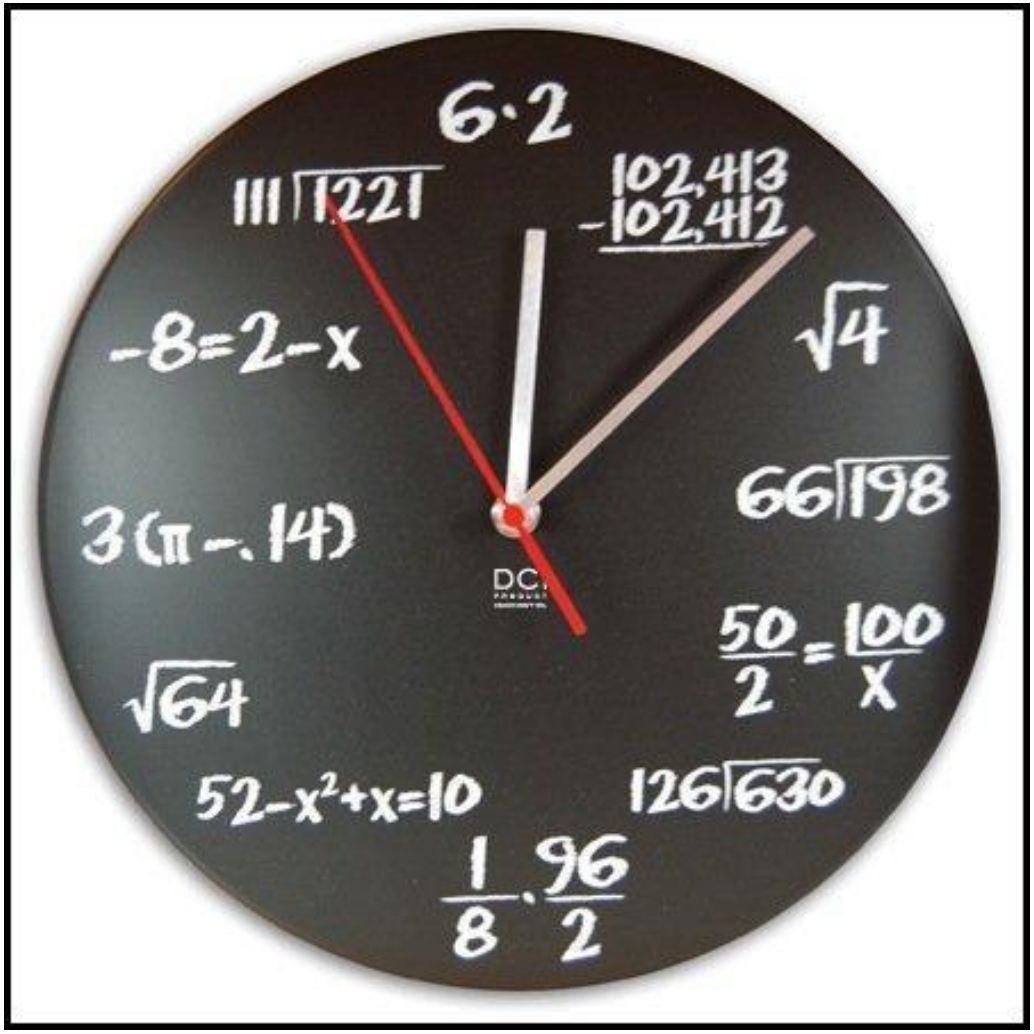
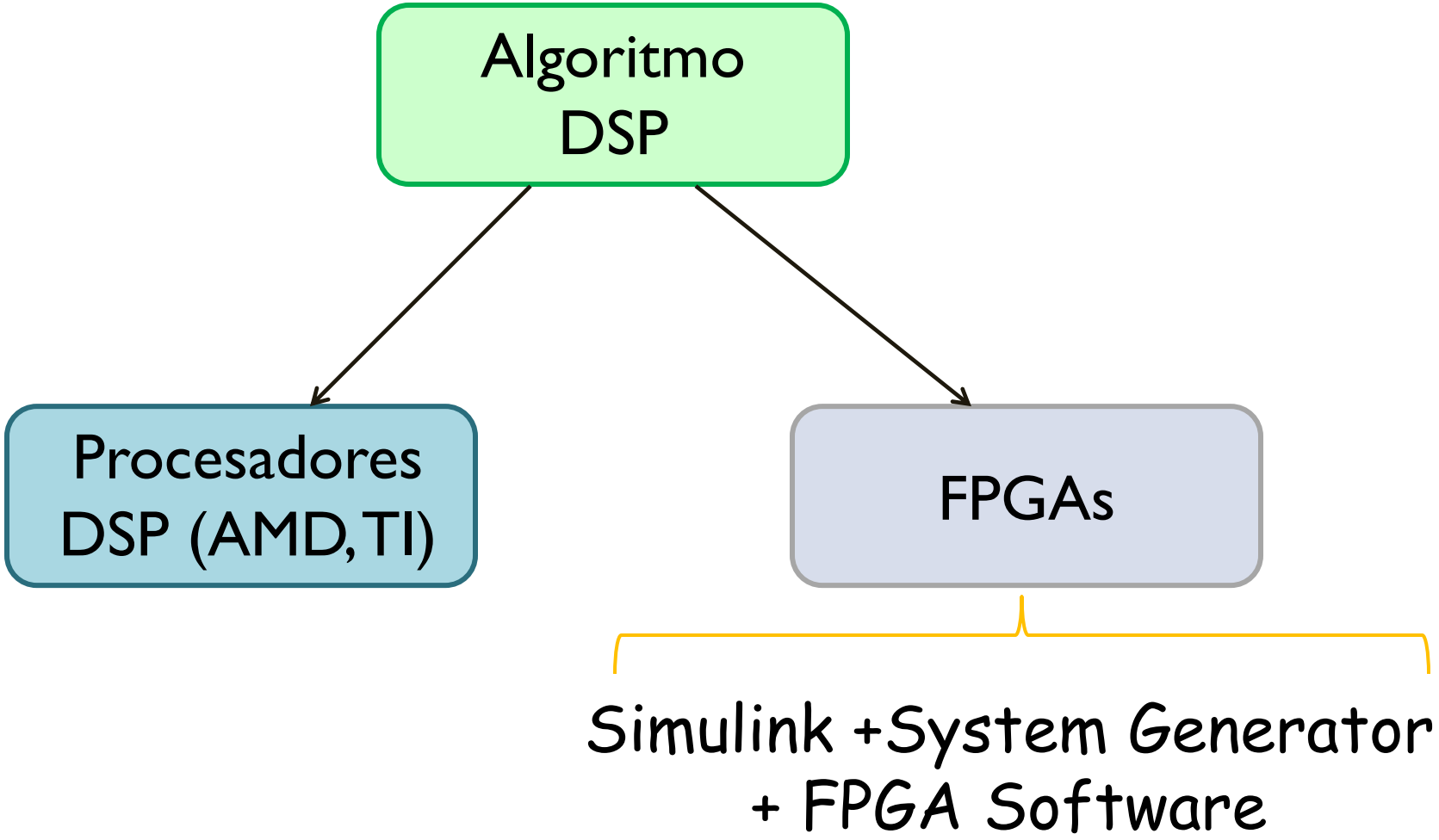
# INTRODUCCIÓN A SIMULINK/SYSTEM GENERATOR (XILINX FPGA)

C7 Technology

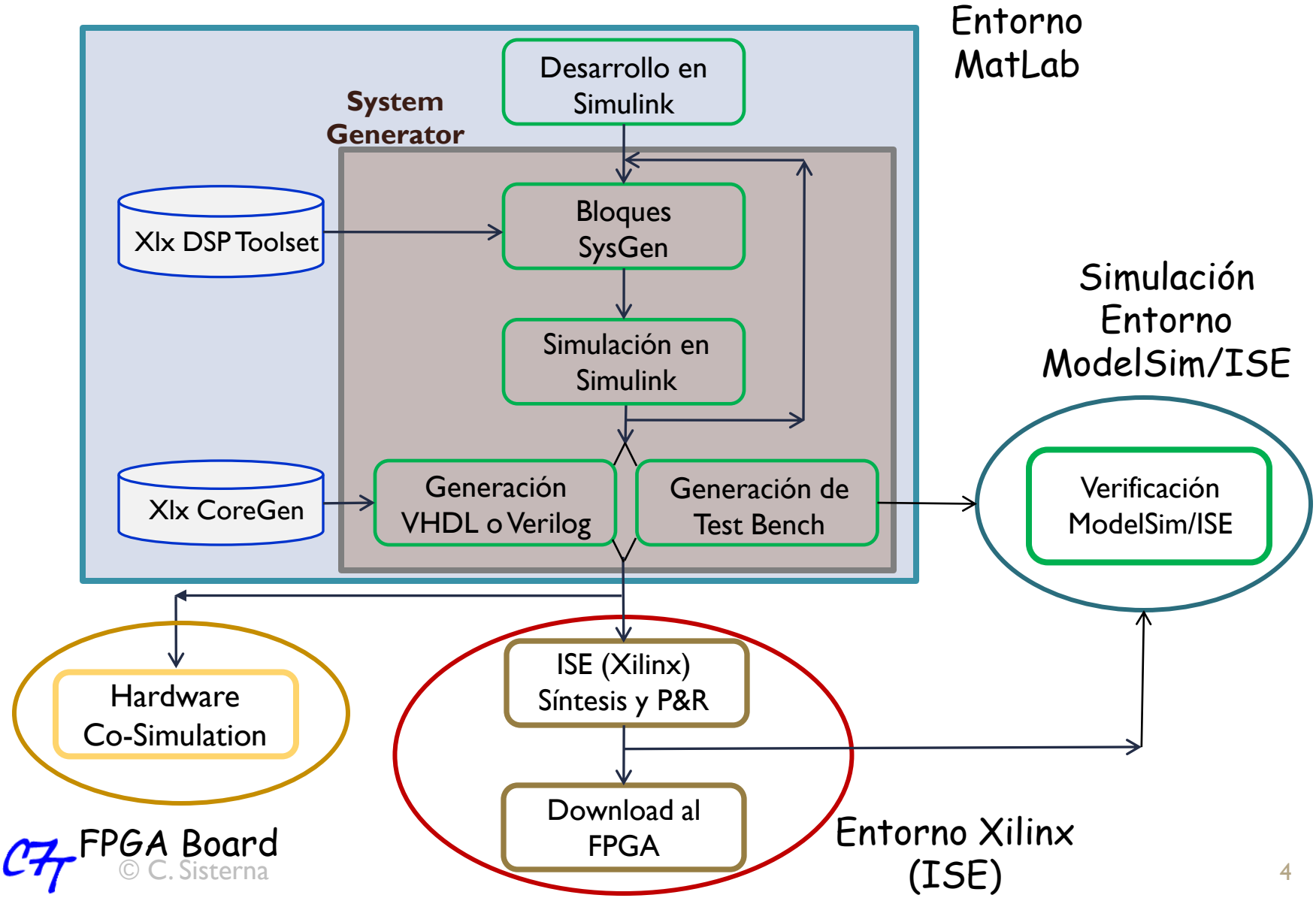
# Algoritmos DSP



# Implementación Algoritmos DSP



# Flujo de Diseño

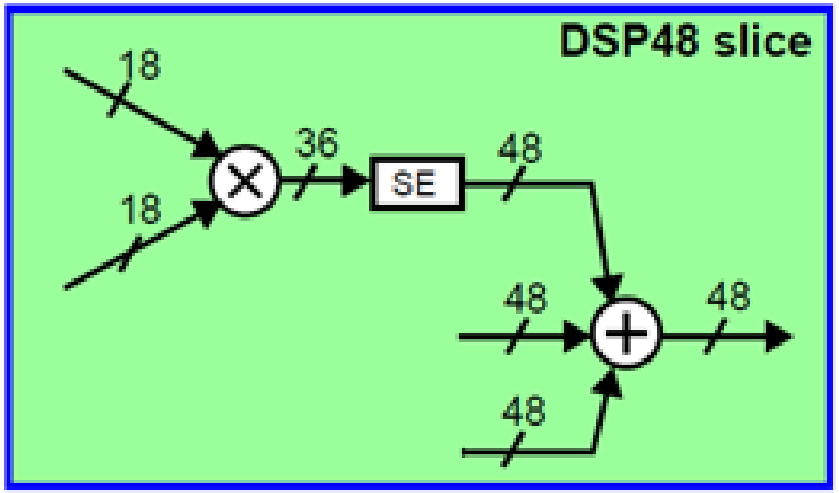


# DSP Blocks

Xilinx FPGAs

# Xilinx FPGA – Bloque DSP48

- Desde un punto de vista simple el DSP48 puede multiplicar dos números de 18 bits (18x18) y acumular el resultado en un ACC de 48 bits. Todo expresado en complemento a dos.

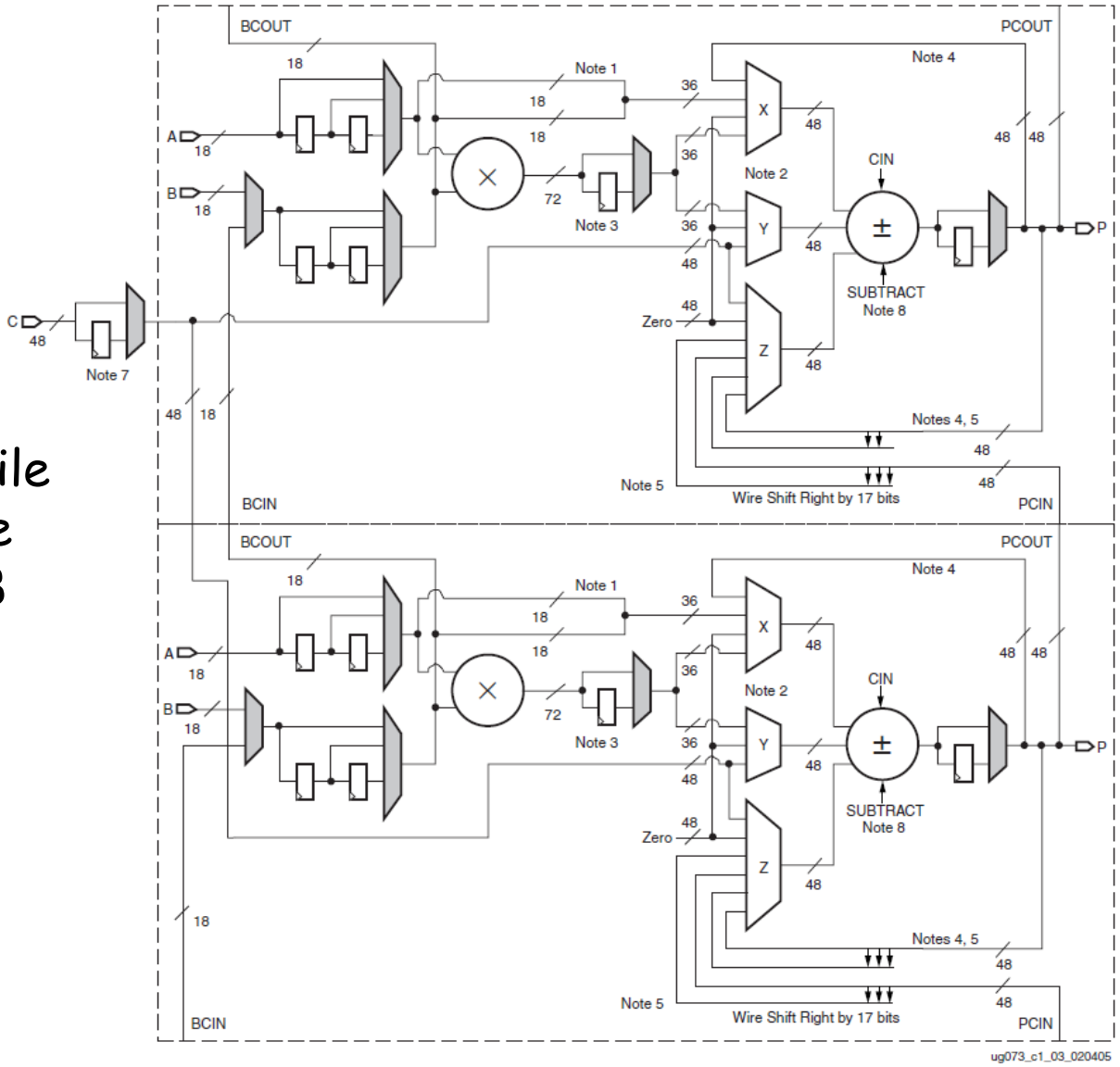



**SE** - Sign Extension

Nota: Dos DSP48 slices componen lo que se llama un Xtreme DSP Tile

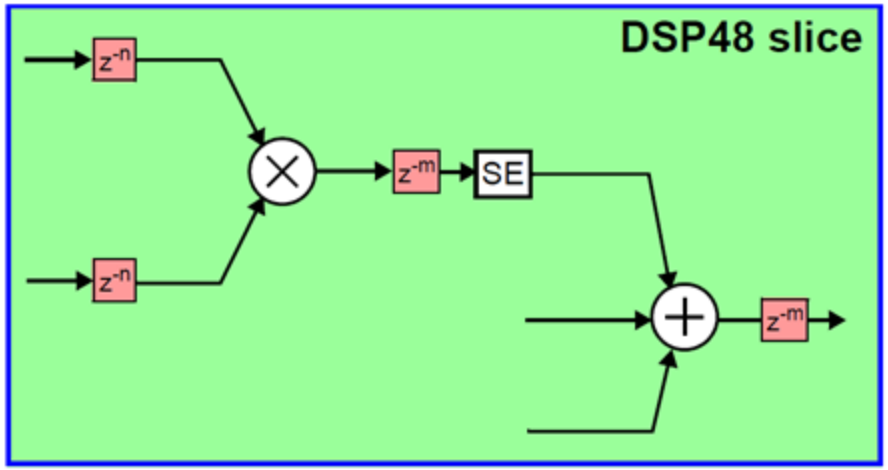
# Xilinx DSP48 Tile

Un DSP48 Tile consiste de dos DSP48 Slices



# Bloque DSP48 - Registros

Existen diferentes opciones del uso de los registros disponibles en el DSP48



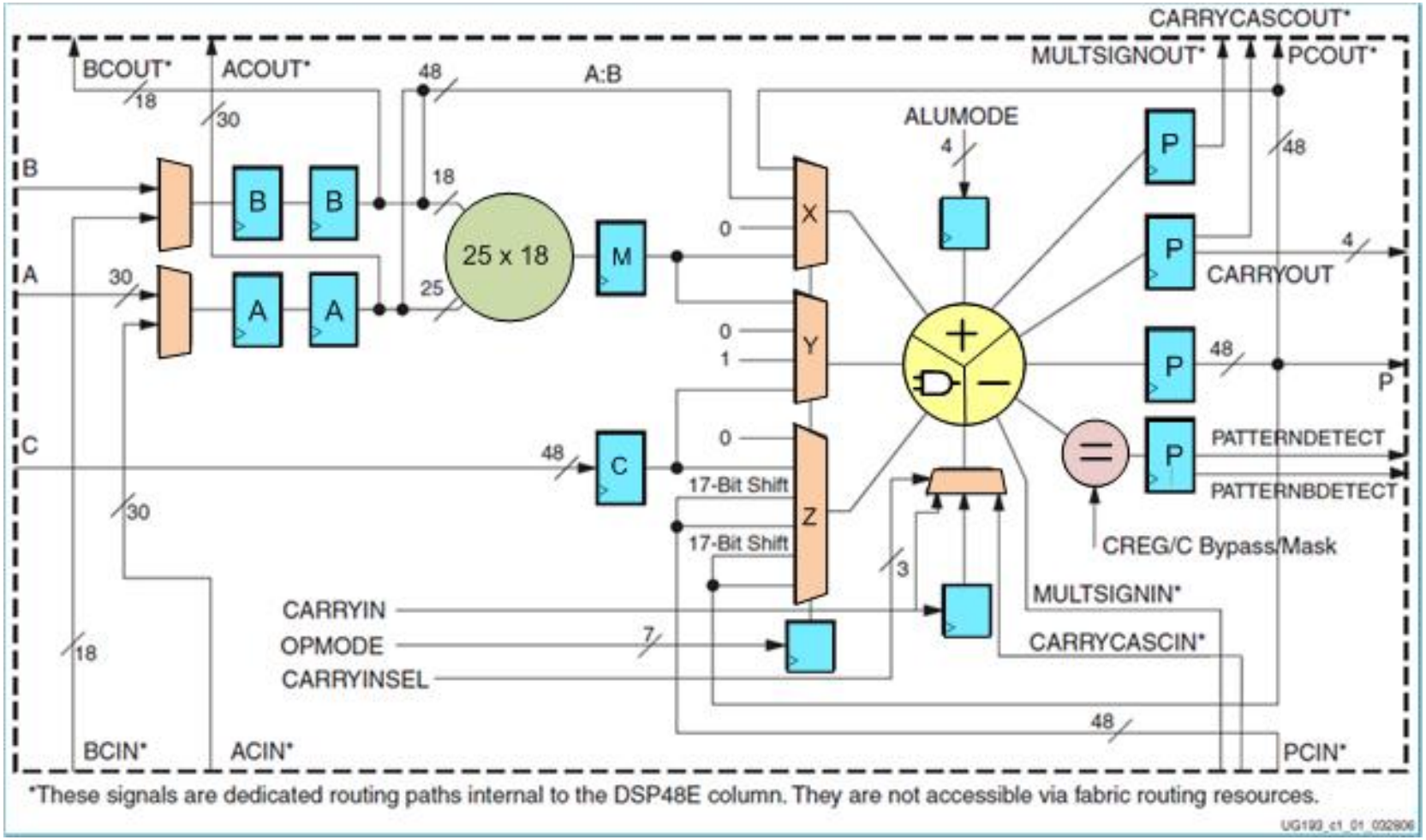

Los registros $z^{-n}$ pueden tener n = 0, 1 o 2 para las entradas del multiplicador.

Los registros $z^{-m}$ pueden tener m = 0 o 1.

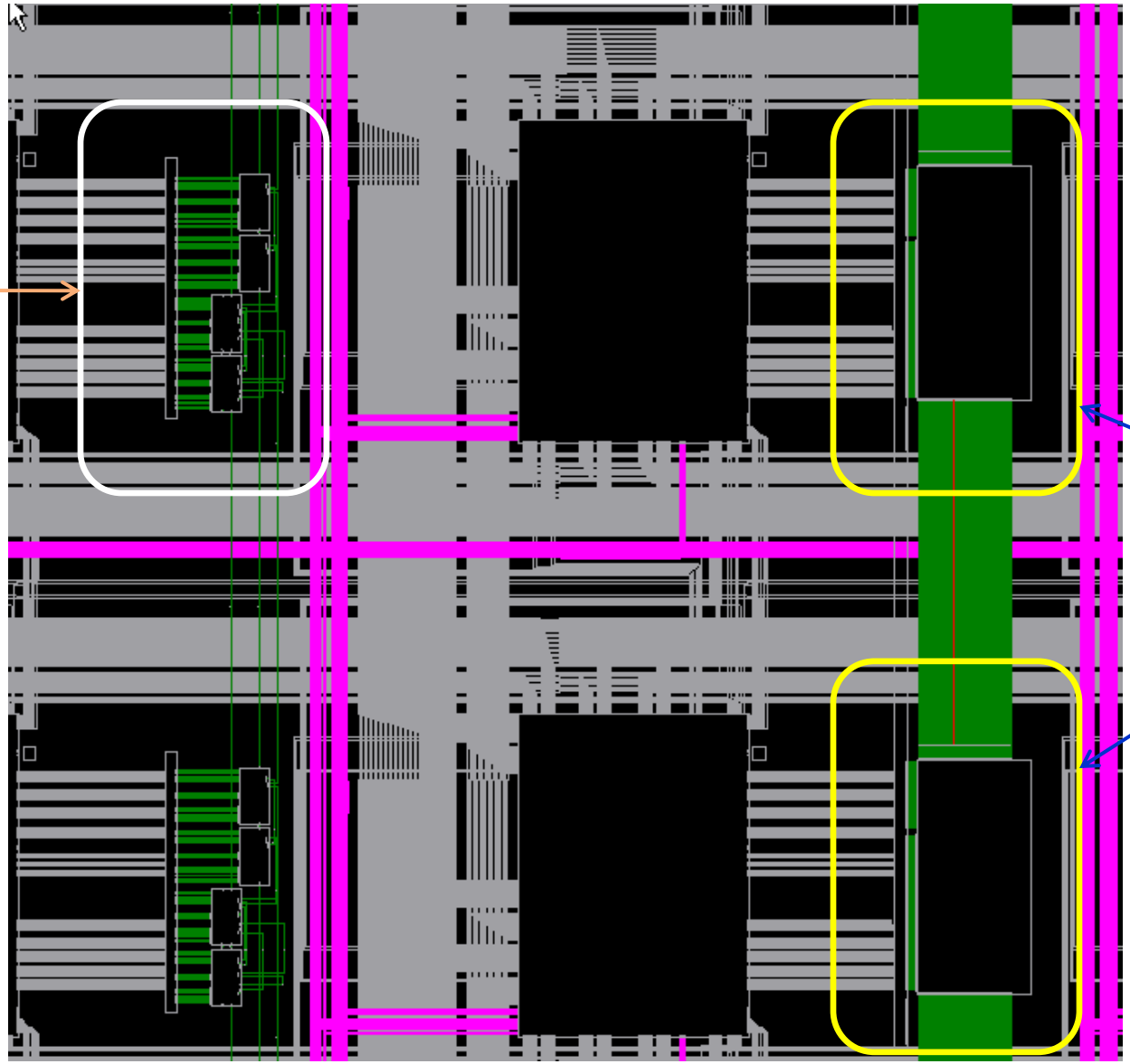
# Xilinx FPGA – Bloque DSP48

- El bloque DSP48 es parte del los bloques prediseñados *ASMBLs* (Application Specific Modular Blocks).
- El bloque DSP48 casi no usa el ruteo del FPGA, solo entrada y salidas. Ello implica: bajo consumo de potencia, muy alta frecuencia de trabajo y una implementación en silicio muy eficiente.
- Bloques DSP48E pueden fácilmente conectarse con sus bloques DSP48 vecinos. Pudiendo formar una cascada de bloques DSP48.

# DSP48E – Virtex 5/6



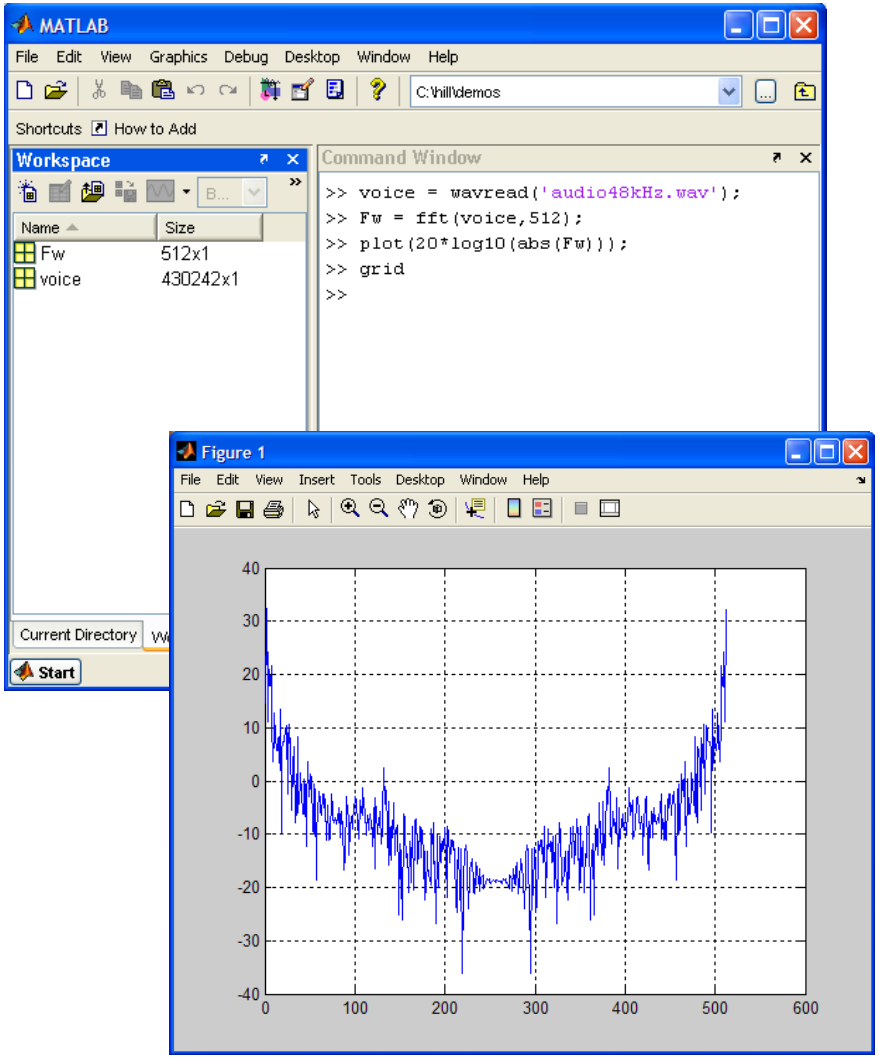
# DSP48 – FPGA Vista Interna I



# Introducción a MatLab – Simulink

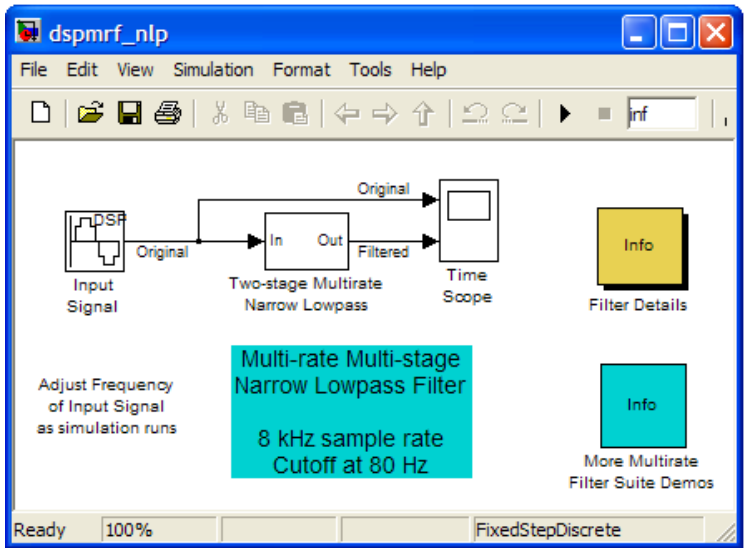

# Matlab

- Matlab provee un entorno de computación técnica que facilita la exploración de soluciones matemáticas a problemas de sistemas:
  - Extensas librerías de funciones matemáticas, procesamiento de señales, comunicaciones, etc.
  - Visualización: gran variedad de funciones para el ploteo y visualización de datos.



# Simulink

- Simulink provee un entorno de diseño gráfico para el desarrollo y simulación de sistemas dinámicos:
  - Totalmente integrado a Matlab
  - Editor gráfico
  - Simulador disparado por evento
  - Extensa librería de funciones parametrizadas
    - Bloques: matemática, fuentes, destinos, comunicaciones, DSP, Sistemas de Control, etc.



# Simulink

- Visual Data Flow.
- Alternativa al uso de lenguajes de programación (‘C’).
- Posibilita visualizar la naturaleza dinámica de un sistema.
- **Puede modelar concurrencia en un sistema:**
  - Ejecuta secciones de un sistema en paralelo.
  - Similar a la concurrencia en HDLs

# Fuentes de Señales en Simulink

- Algunos de las fuentes de señales mas usadas en diseños DSP son:
  - White-Noise (band limited)
  - Constante
  - Rampa
  - Onda Senoidal
  - Función Escalón
  - Generador de Pulsos
  - Contador/Contador Libre
  - Datos binarios de un archivo
  - Datos proveniente del espacio MatLab

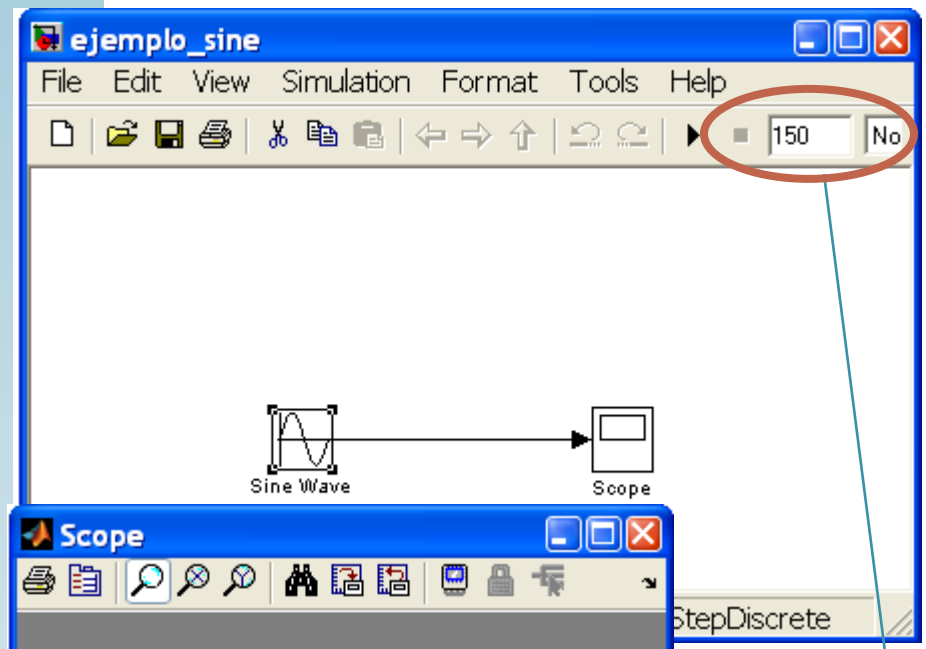
# Periodo de Muestreo

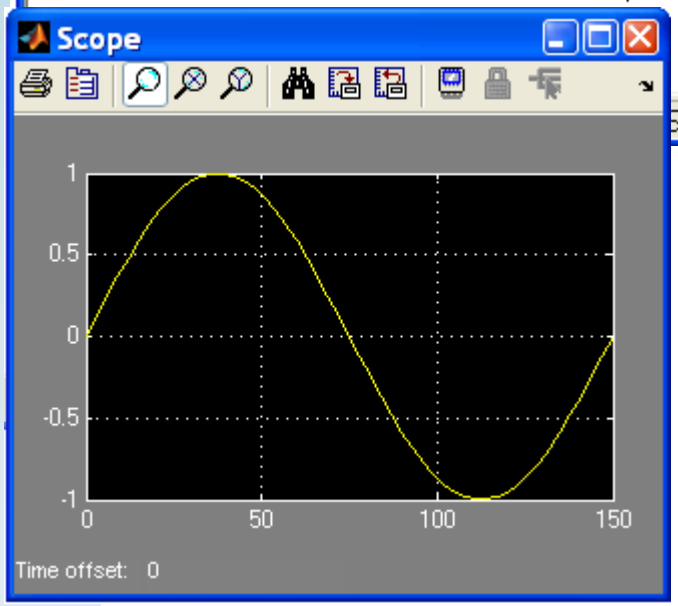
- Cada señal usada en un diseño DSP en Simulink debe ser *muestreada* cada cierto tiempo llamado *periodo de muestreo*.
- Si bien el periodo de muestro puede ser *arbitrario*, los diferentes bloques disponibles en Simulink usan un periodo de muestreo.
- Un periodo de muestreo de 1/1000 (0.001) indica que la función del bloque se ejecutará cada 0.001 segundo y producirá la respectiva salida.
- Recordar el Teorema de Nyquist:
  - **Fm >= $2f_{max}$**

# Periodo de Muestreo (cont.)

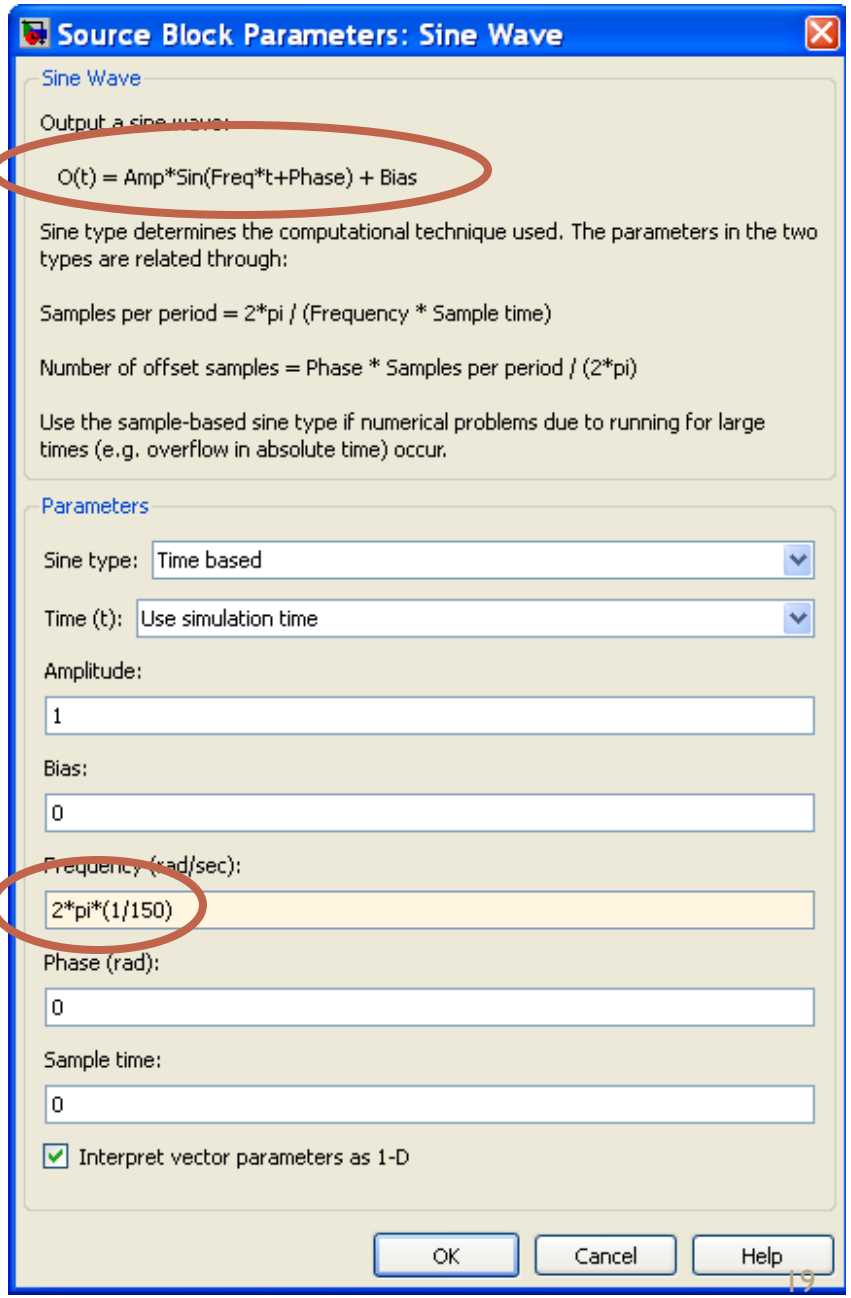
- El periodo de muestreo de un bloque tiene relación directa con la frecuencia del reloj del hardware que implementará el bloque
- El periodo de muestreo debe ser fijado sí o sí en:
  - Bloque denominado 'Gateway In'
  - Bloque sin entradas
- En los bloques que no es especificado, el periodo de muestreo es derivado del periodo de muestreo de las entradas al mismo.

# Ejemplo Simulink I

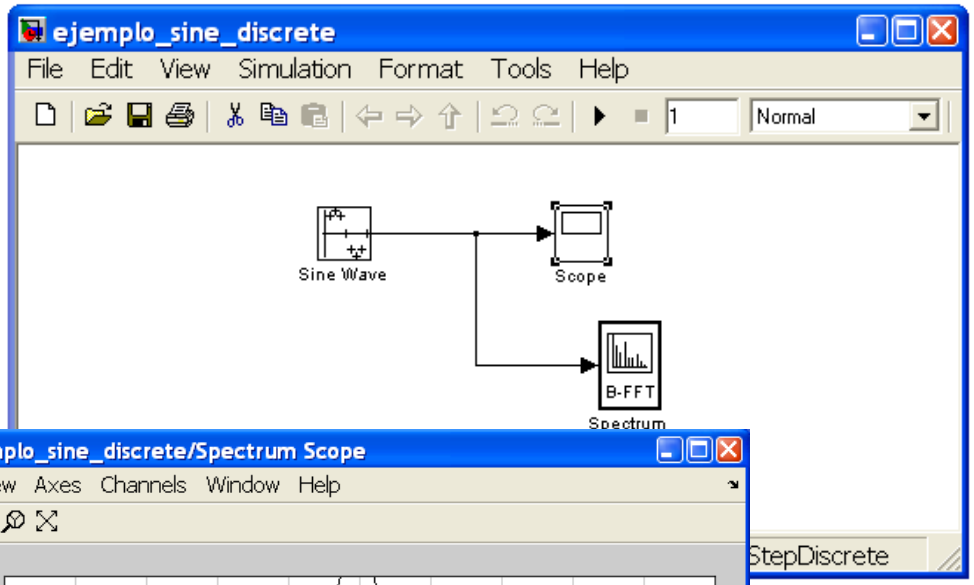


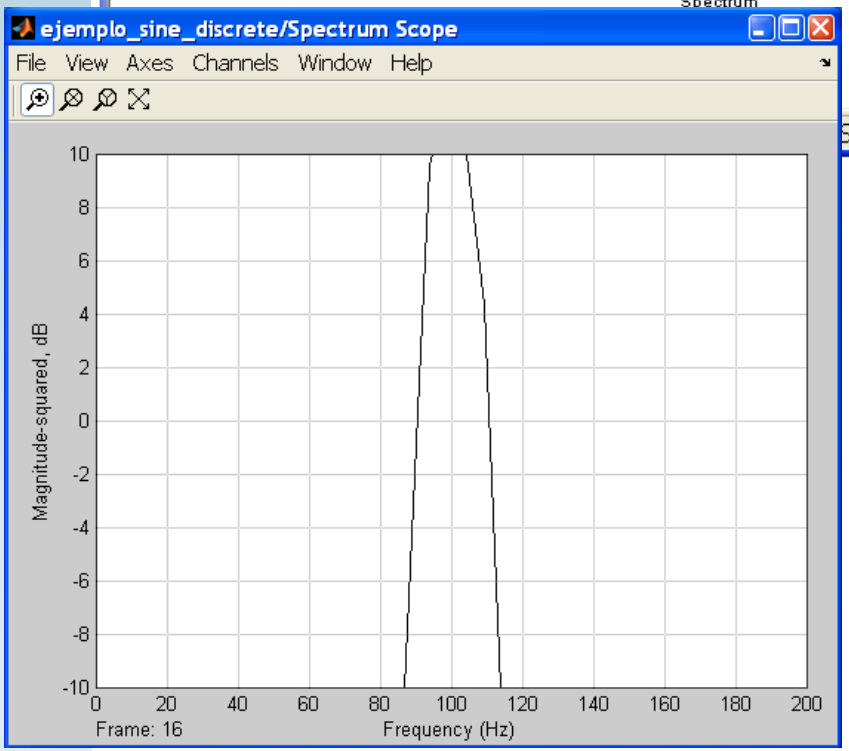



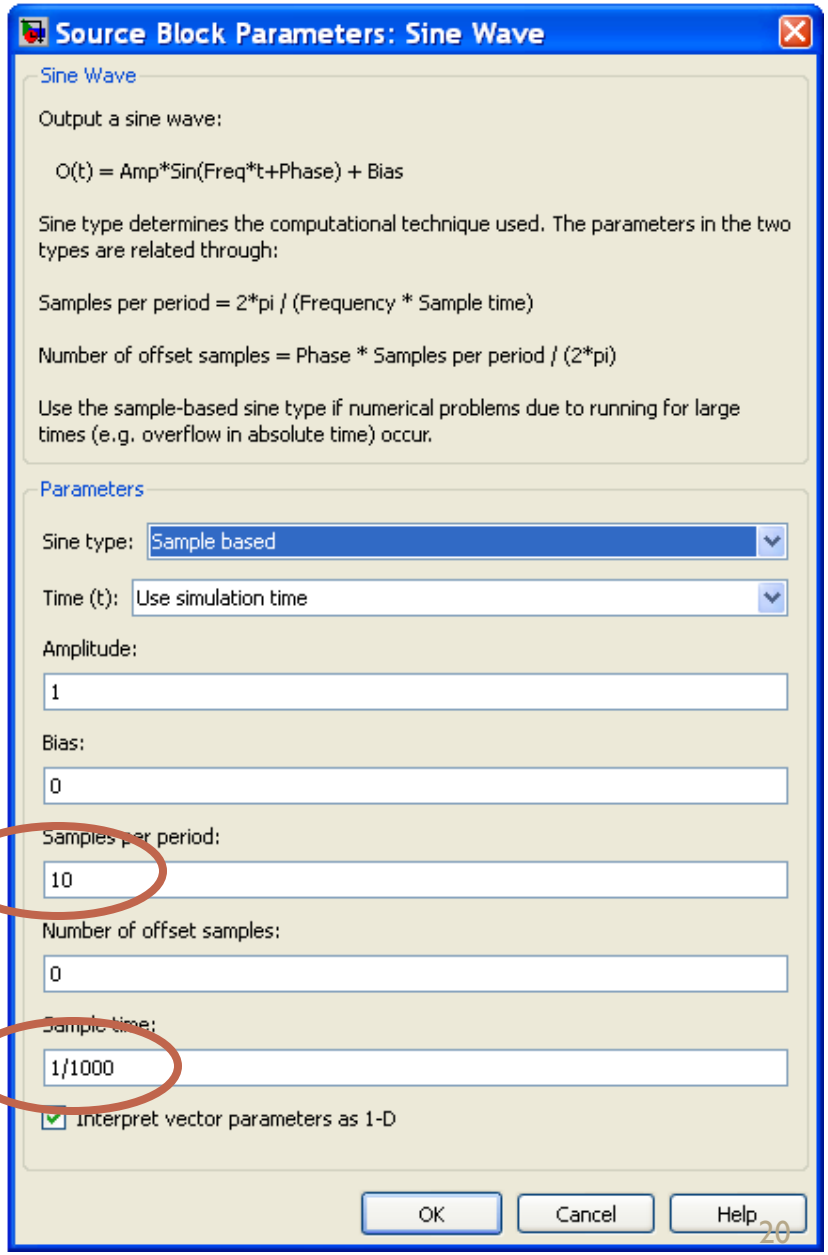

1Hz



# Ejemplo Simulink 2




100Hz

# SYSTEM GENERATOR

# System Generator

- Flujo de diseño integrado a Simulink, generando directamente el archivo de configuración del FPGA (.bit)
- Integra:
  - MatLab, Simulink
  - HDL Síntesis
  - Librerías de DSP
  - Herramientas de implementación del FPGA
  - Simulación en doble precisión y punto fijo
- Abstracción Aritmética
  - Punto-fijo arbitrario, incluyendo cuantización y overflow

# System Generator

- Es un 'Toolbox' plug-in al entorno Simulink.
- Provee:
  - Bloques de distintas funciones implementables en FPGA
  - Generación de archivos VHDL
  - Simulación
  - IP Cores (48, hasta ahora)
  - Test benches

# Generación de Archivos FPGA

- Generación del código VHDL y Verilog para:
  - Virtex-7, Virtex-6, Virtex-5, Virtex-4, Spartan-7, Spartan-6, Spartan-3E
- Mapeo y expansión del hardware respectivo a cada bloque
- Preservación de la jerarquía del modelo una vez generados los códigos VHDL y Verilog
- Utilización de IP cores de CoreGenerator
- Generación automática de:
  - Un proyecto ISE
  - Archivo de restricciones (constraint file *.ucf)
  - Archivo de simulación Test Bench y script *.do (ModelSim)

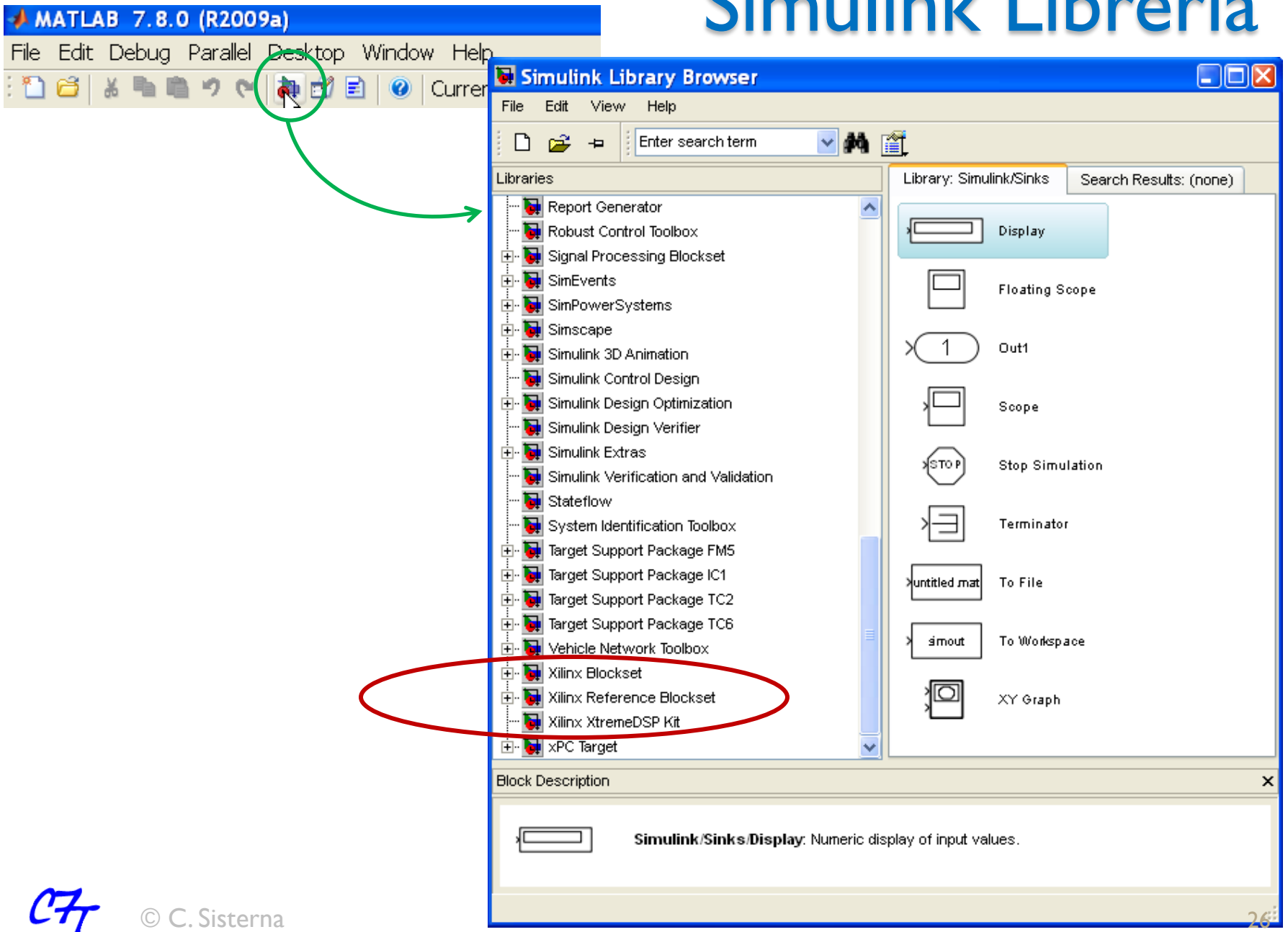
# SysGen – Archivos Generados

Archivos de Diseño
- .vhd o .v
- .edn o .ngc
- .xcf

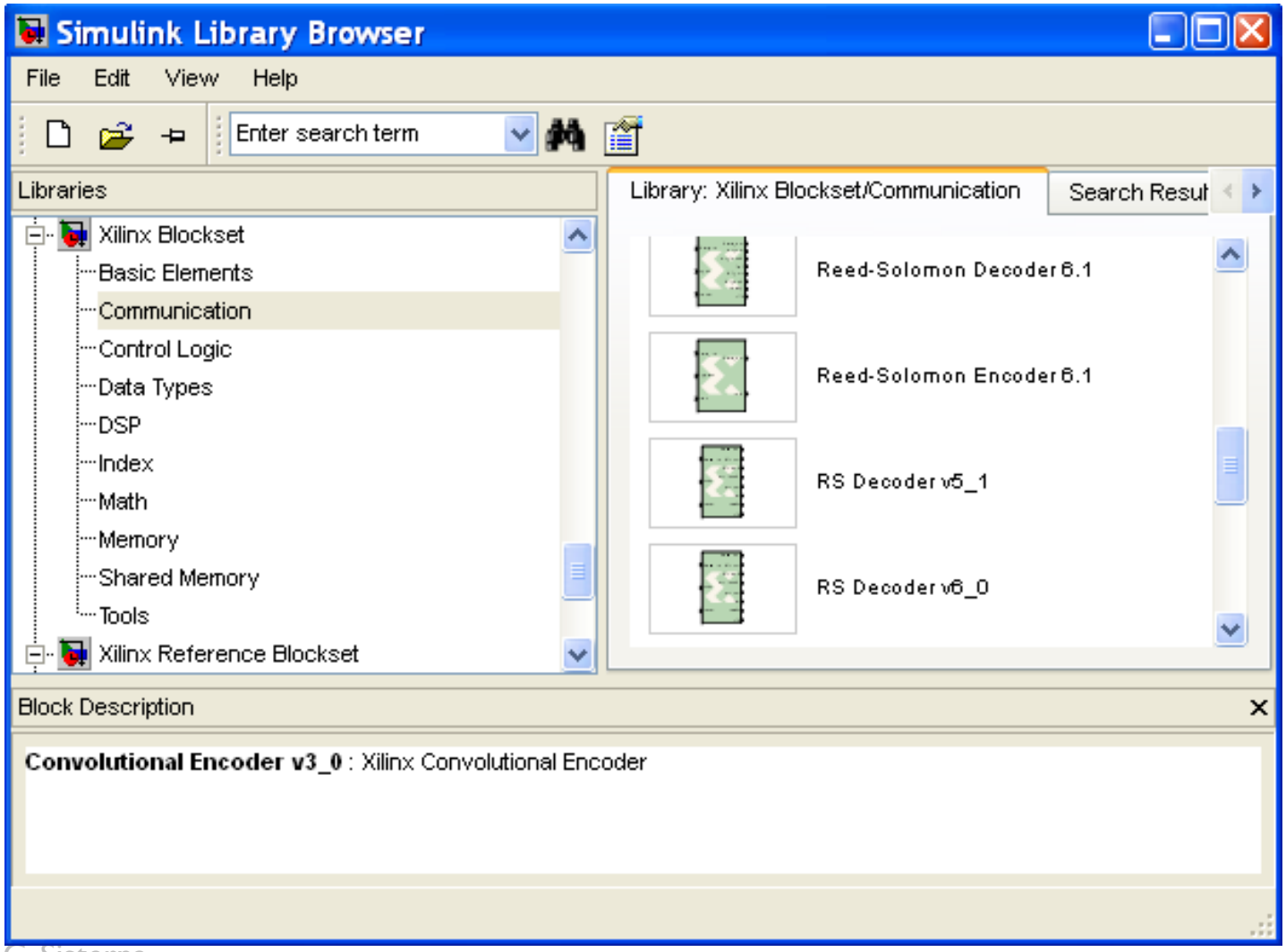
Archivos de Proyecto
- .ise
- .tcl

Archivos de Simulación
- .do
- .dat
- _tb.vhd o _tb.v

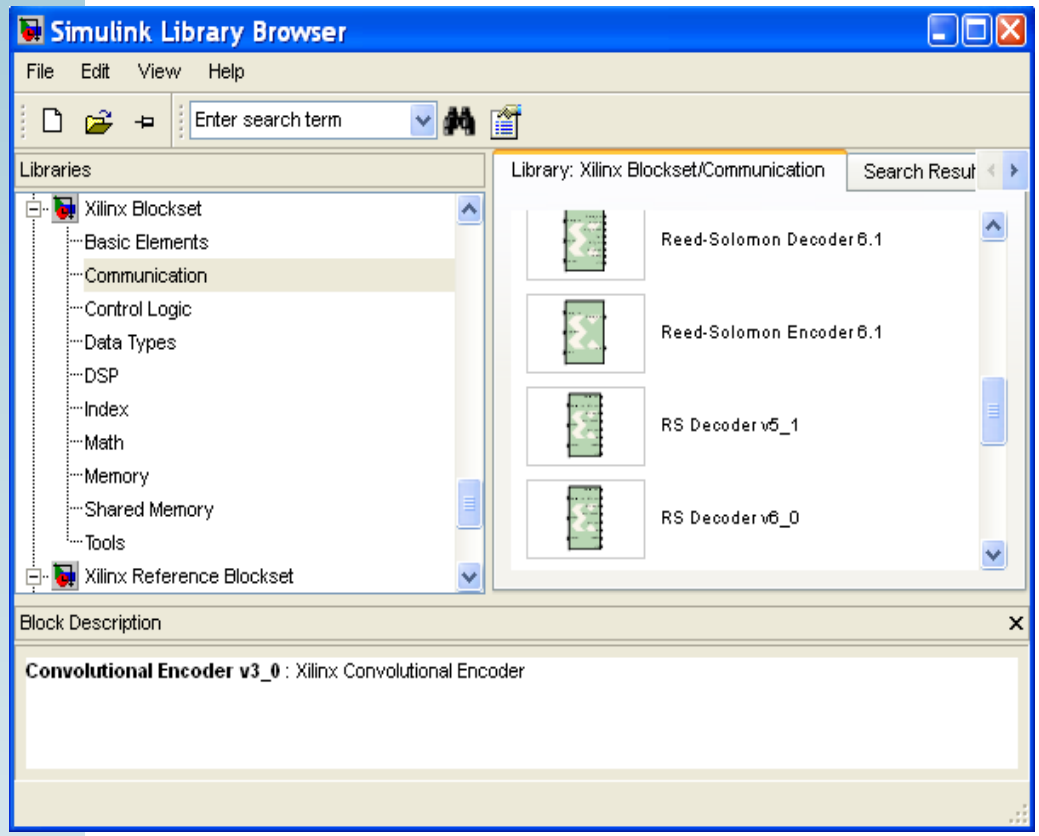
# Simulink Librería



# Librería de Xilinx

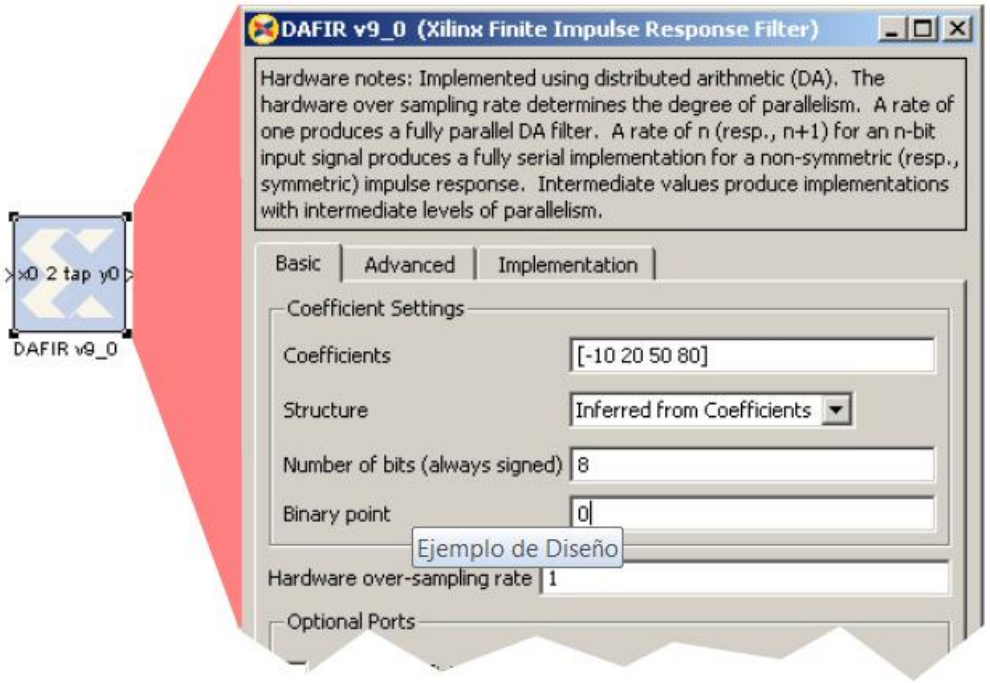


# Librería de Xilinx

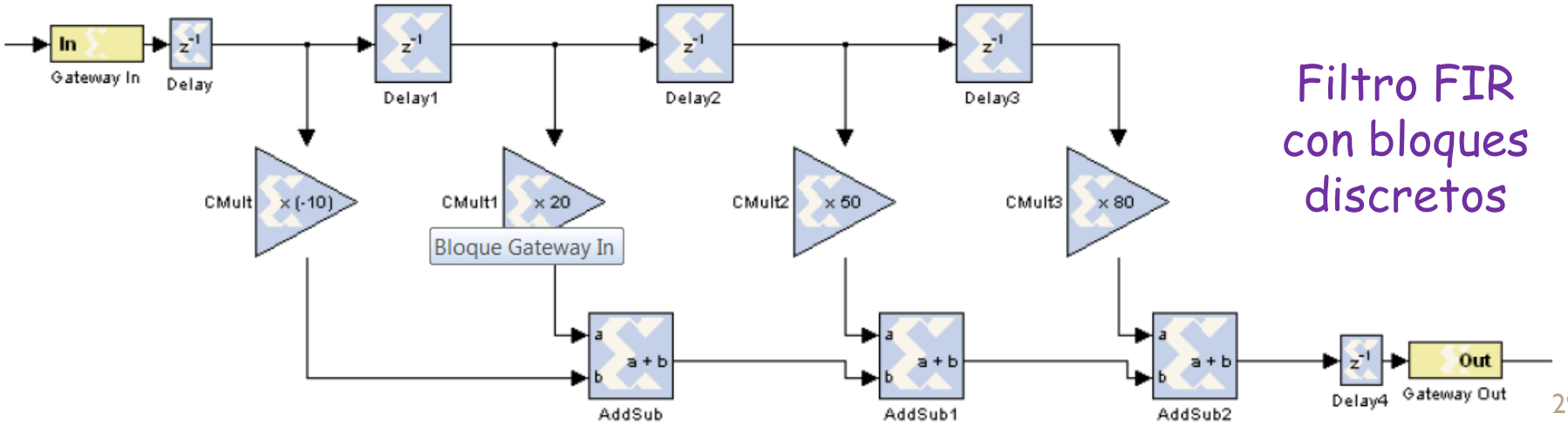



- Elementos básicos: contadores, retardos, lógica binaria, etc.
- Comunicaciones: ECC bloques, Viterbi, Reed Solomon, etc.
- Tipo de Datos: convertir, rotar, concatenar, etc.
- DSP: IIR, FIR, FFT, etc.
- Math: multiplicar, acumular, invertir, contadores, etc.
- Memoria, dual port RAM, single port RAM, ROM, FIFO.
- Tools: ModelSim, Estimador de Recursos, SysGen Token

# Xilinx Librería



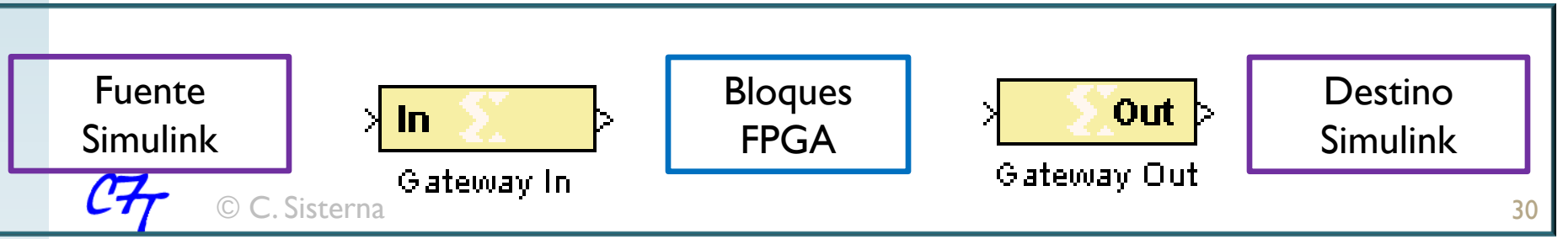

Bloque de Filtro FIR disponible en la librería



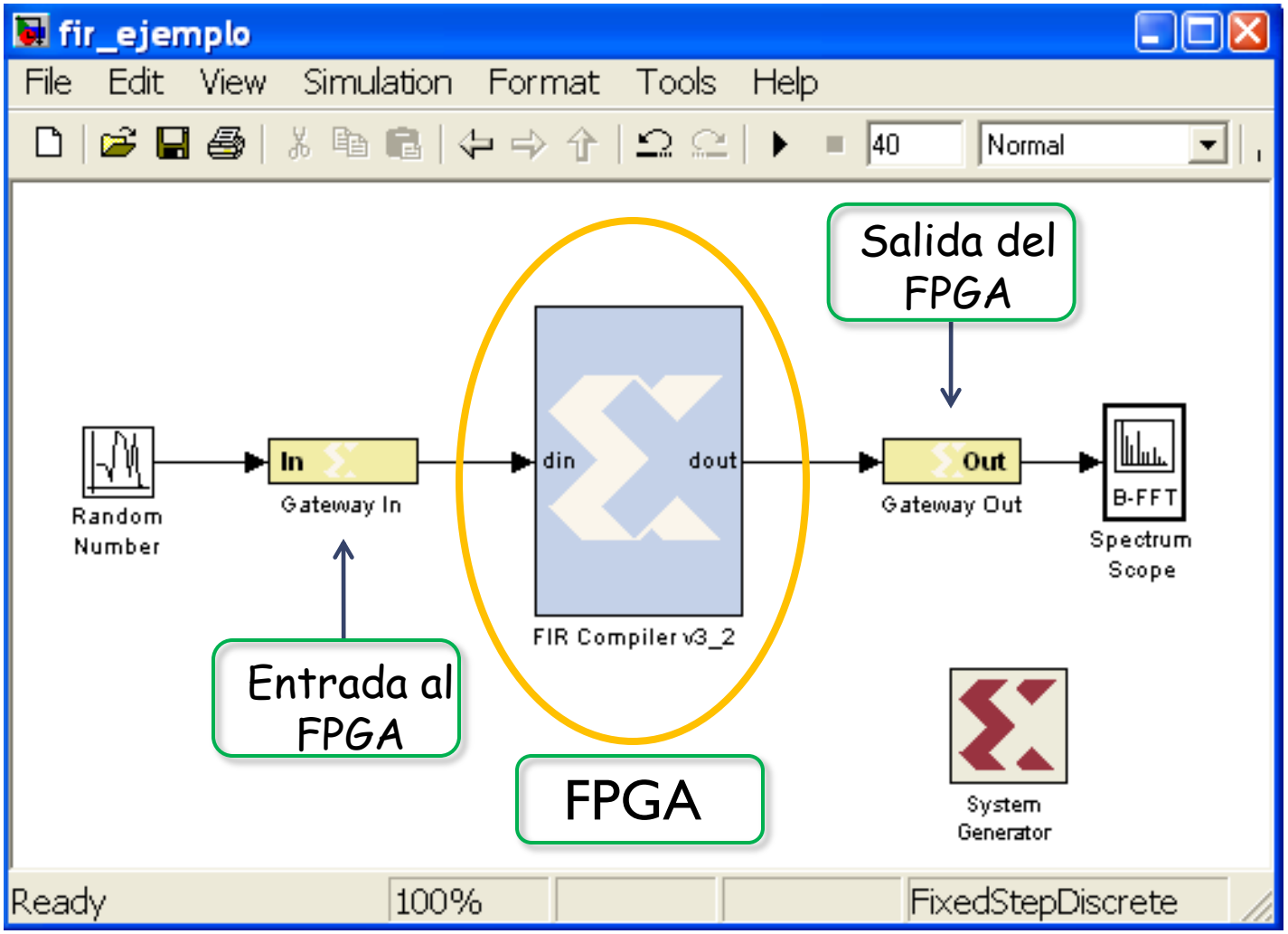

Filtro FIR con bloques discretos

# Interface Simulink-FPGA

- Simulink usa 'double' para representar números en simulación. A double es un número en punto flotante en complemento a dos de 64 bits.
- TODOS los bloques de Xilinx usan representación en n-bits en punto fijo (complemento a dos es opcional).
- Por ello se usa un bloque para la conversión de datos desde Simulink a un bloque de Xilinx, llamado *Gateway In*. Y otro bloque para la conversión desde un bloque de Xilinx a Simulink, llamado *Gateway Out*.



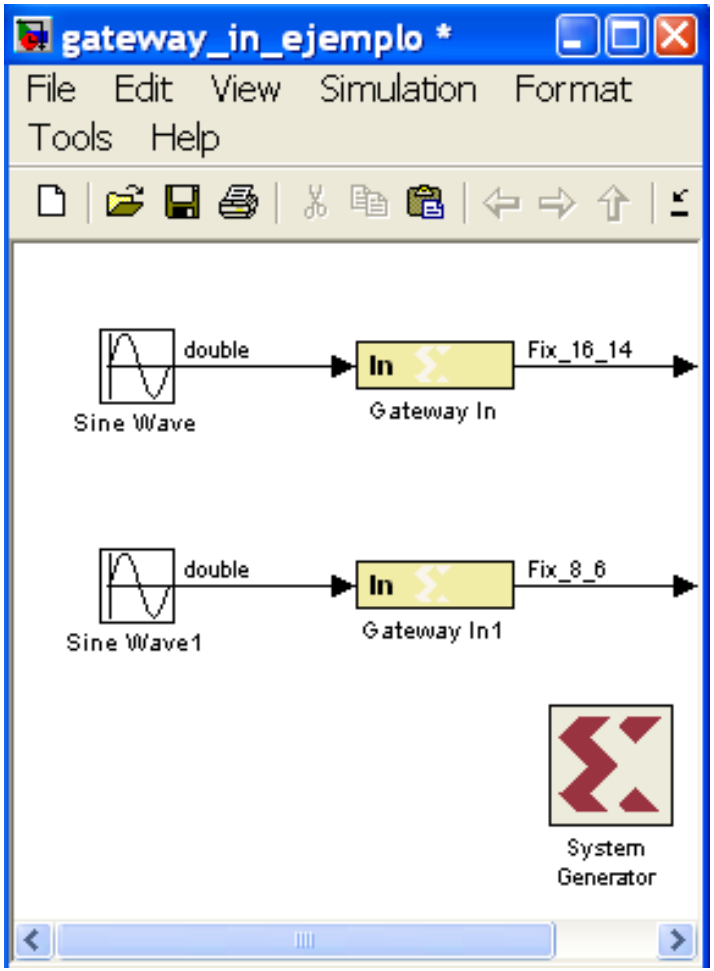
# Ejemplo de Diseño

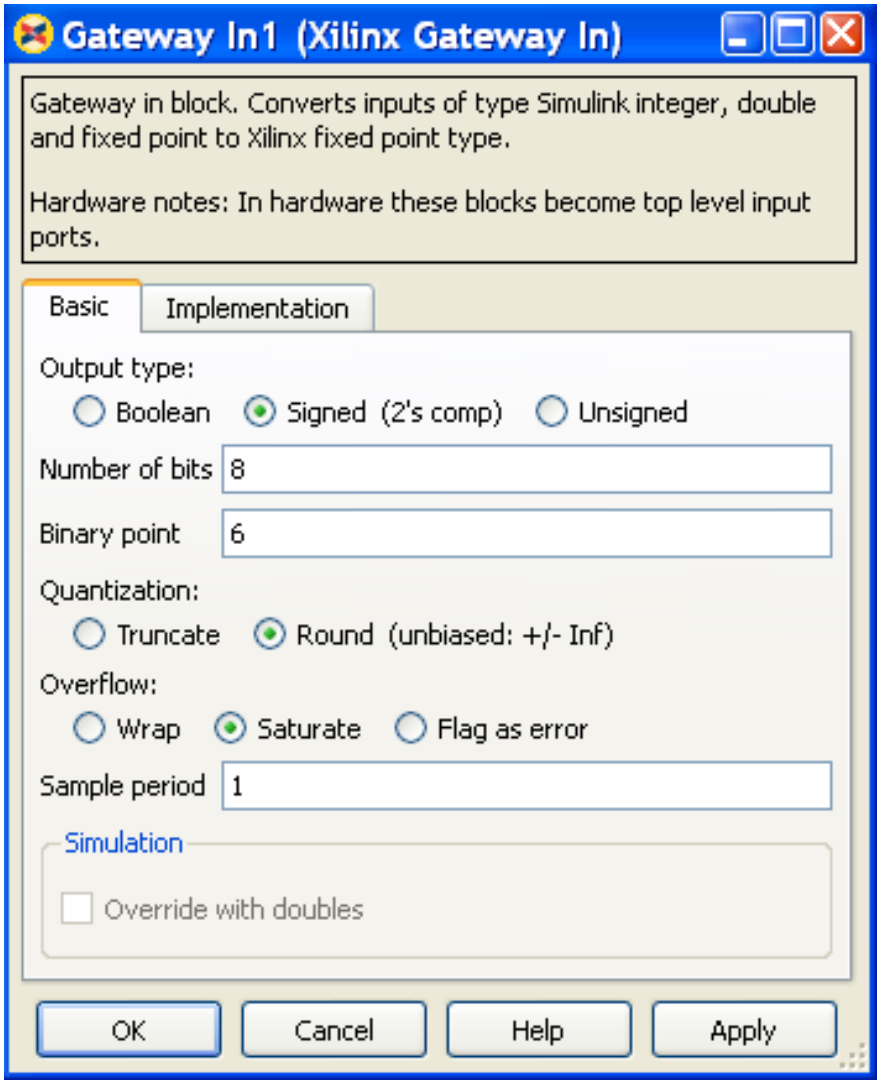


# Bloque Gateway In

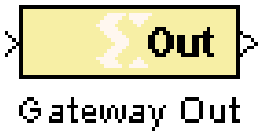

- Controla la conversión desde un número representado en double, entero  o punto fijo a un número booleano de N-bits, que puede ser con signo (complemento a dos) o sin signo, con o sin punto fijo.
- Presenta la opción de manejar los extra bits durante la conversión (ovreflow).
- Este bloque define cuales van a ser las entradas del diseño codificado en HDL por SysGen.
- Define los estímulos en caso de que la opción de 'Create Testbench' haya sido seleccionada.
- Nombra los correspondientes puertos de entrada de la entidad (entity) generada por System Generator.

# Bloque Gateway In



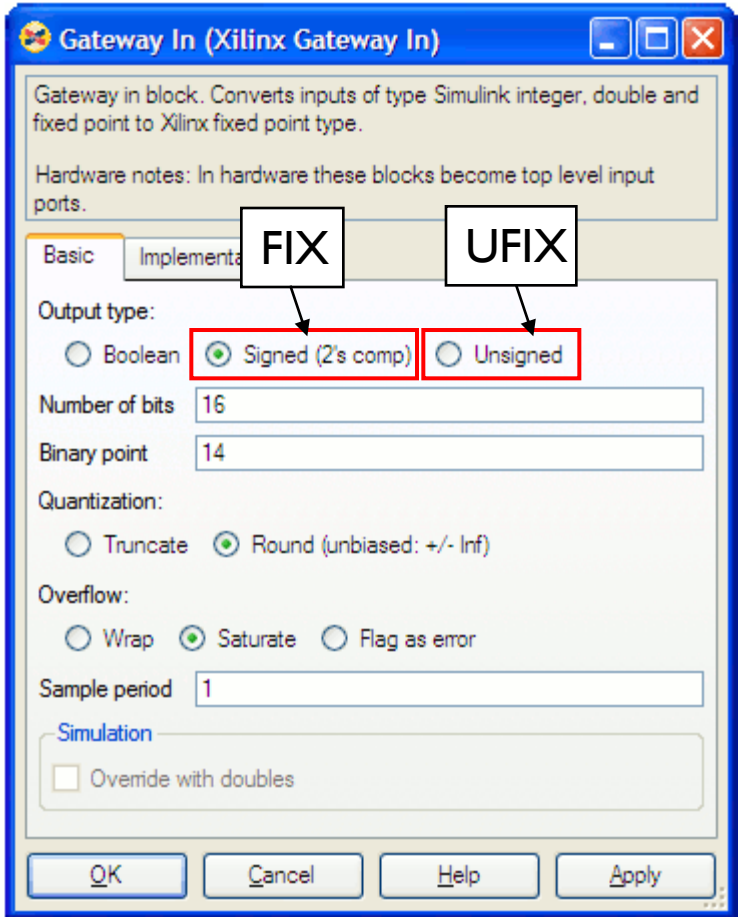



# Bloque Gateway Out



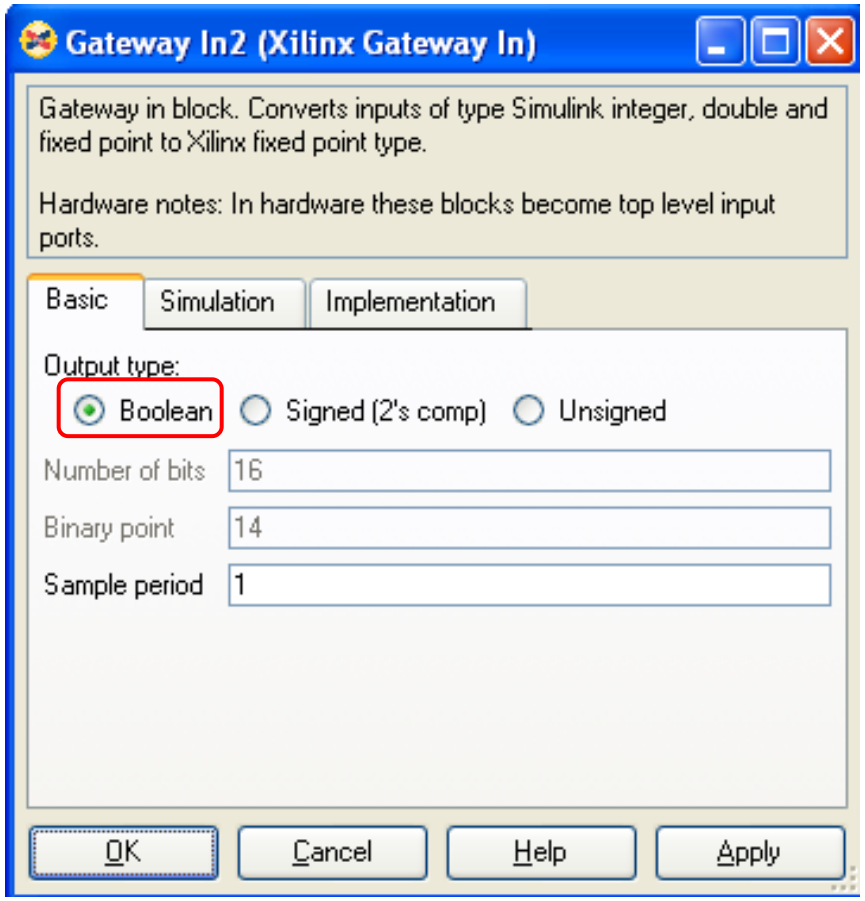

- Convierte los datos de punto fijo del sistema en el FPGA a punto flotante Simulink.
- Define cuales van a ser los puertos de salida del sistema generado por System Generator.
- Nombra los correspondientes puertos de salida de la entidad (entity) generada por System Generator.

# SysGen – Tipo de Datos



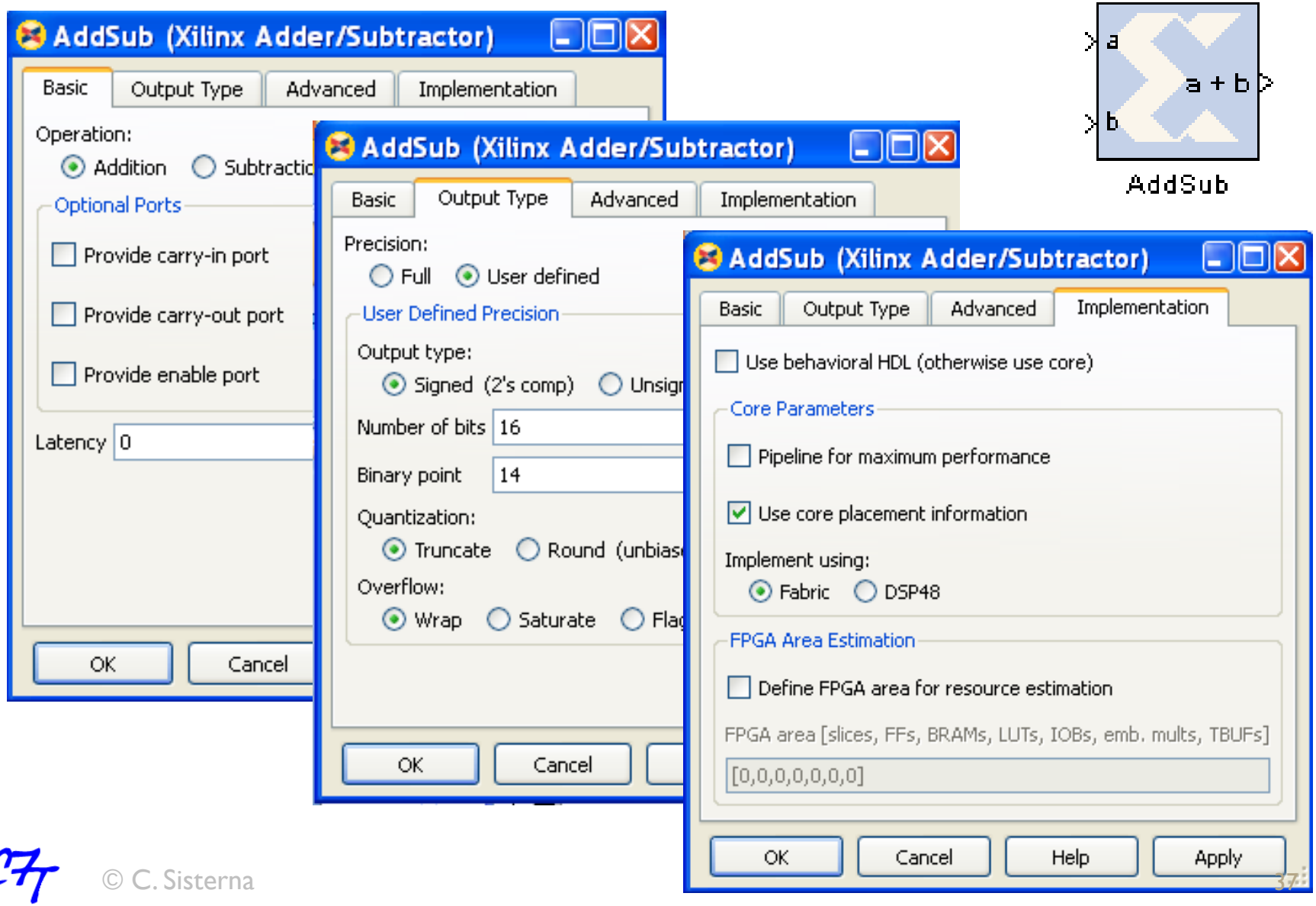

- **FIX**: tipo de dato fijo, es un número en punto fijo representado en complemento a dos.
- **UFIX**: es un número en punto fijo representado sin signo.

# SysGen – Tipo de Datos

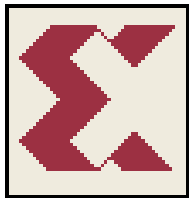



- **Boolean**: pueda tomar valores ‘1’ o ‘0’. Se usa para controlar señales como LOAD, CS, RESET, etc.

# SysGen – Parámetros de un Bloque

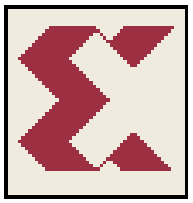


# Bloque System Generator

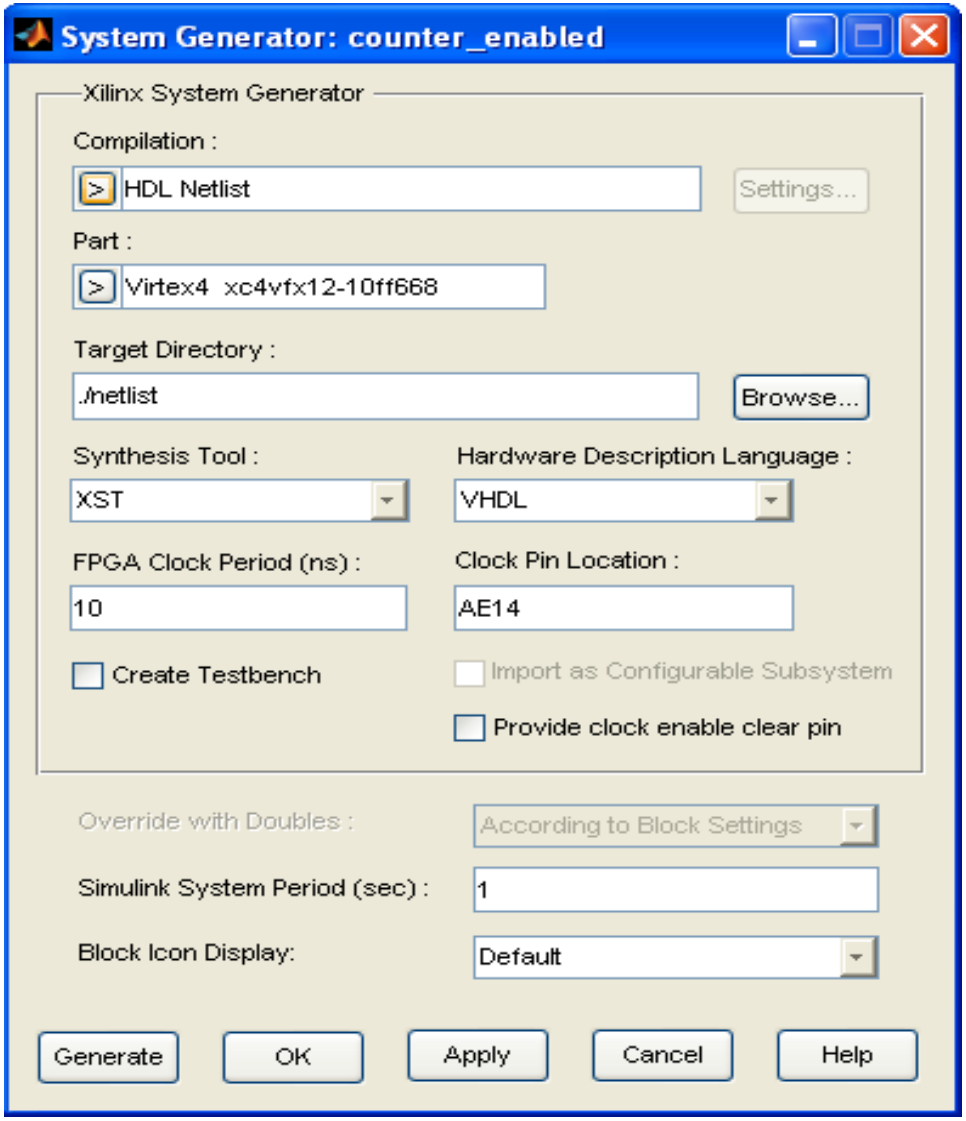



- **Cada** diagrama usando bloques de Xilinx in Simulink requiere que el bloque System Generator sea colocado en el diagrama.
- El bloque **no** se conecta a nada. Pero controla el proceso de generación del código HDL y de la implementación del sistema.
- Parámetros del bloque SysGen determinan FPGA a usar, código HDL, frecuencia objetivo, directorio de trabajo, etc.

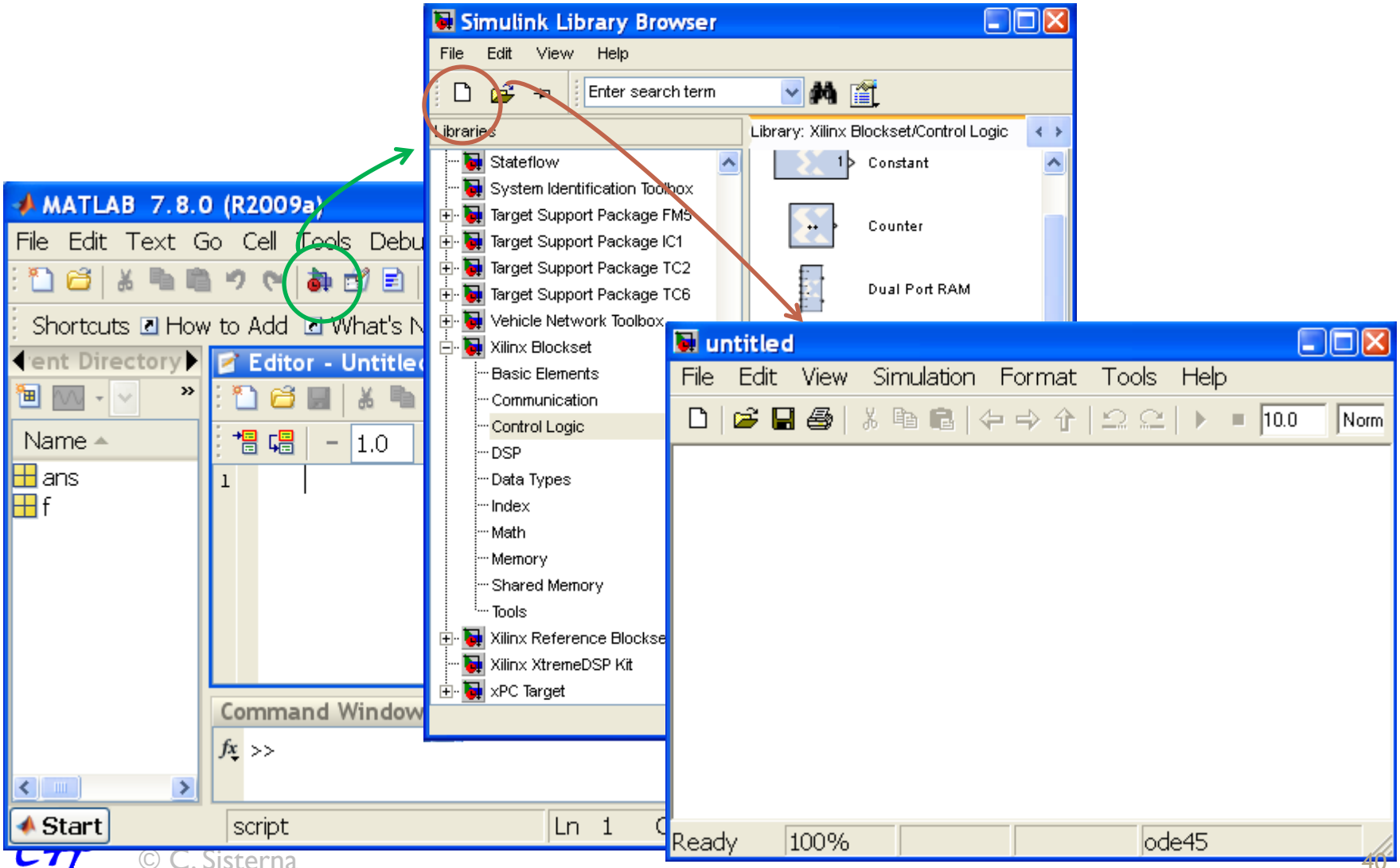
# Bloque System Generator



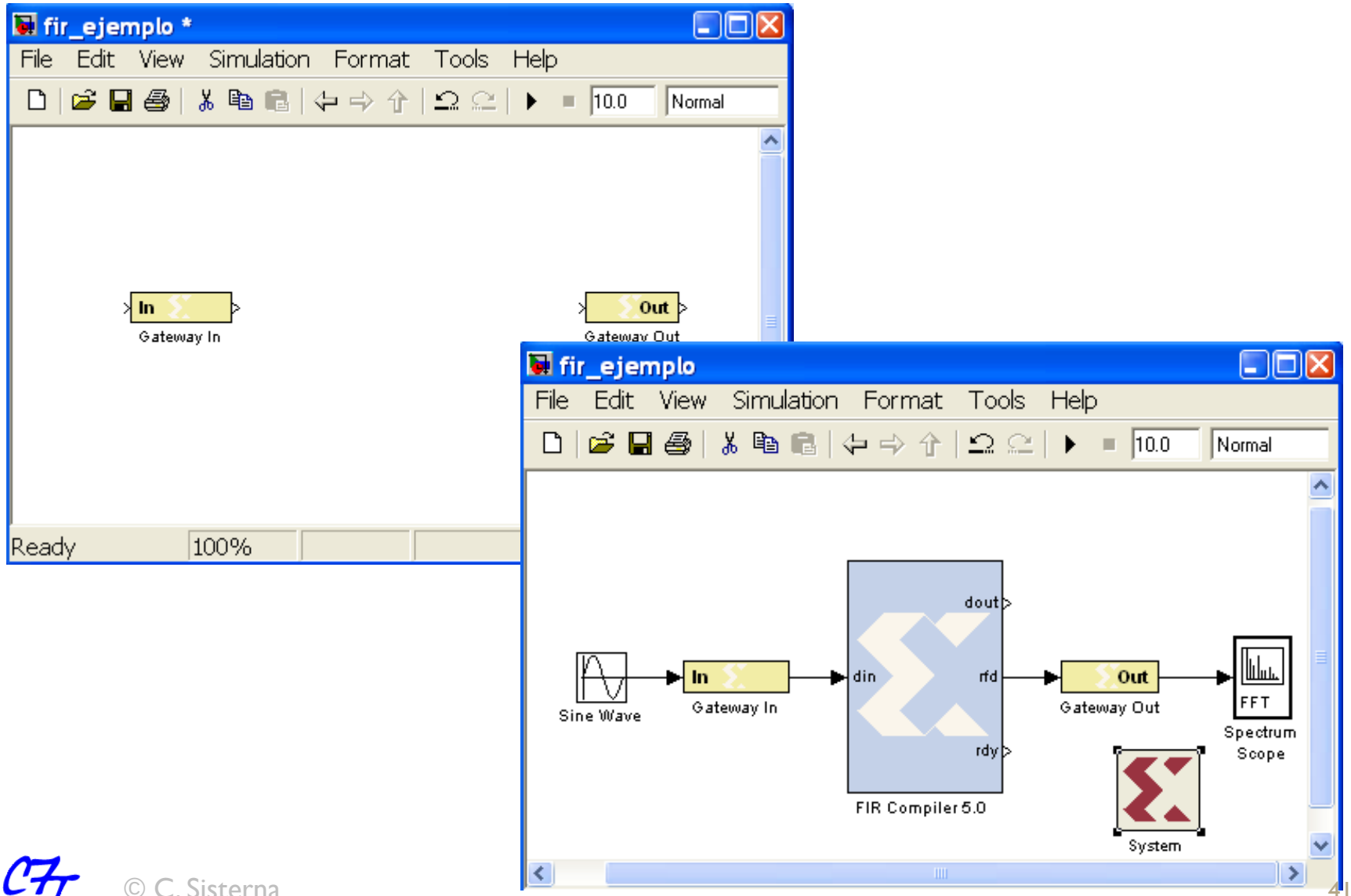
System Generator



# Creando un Diseño en Simulink

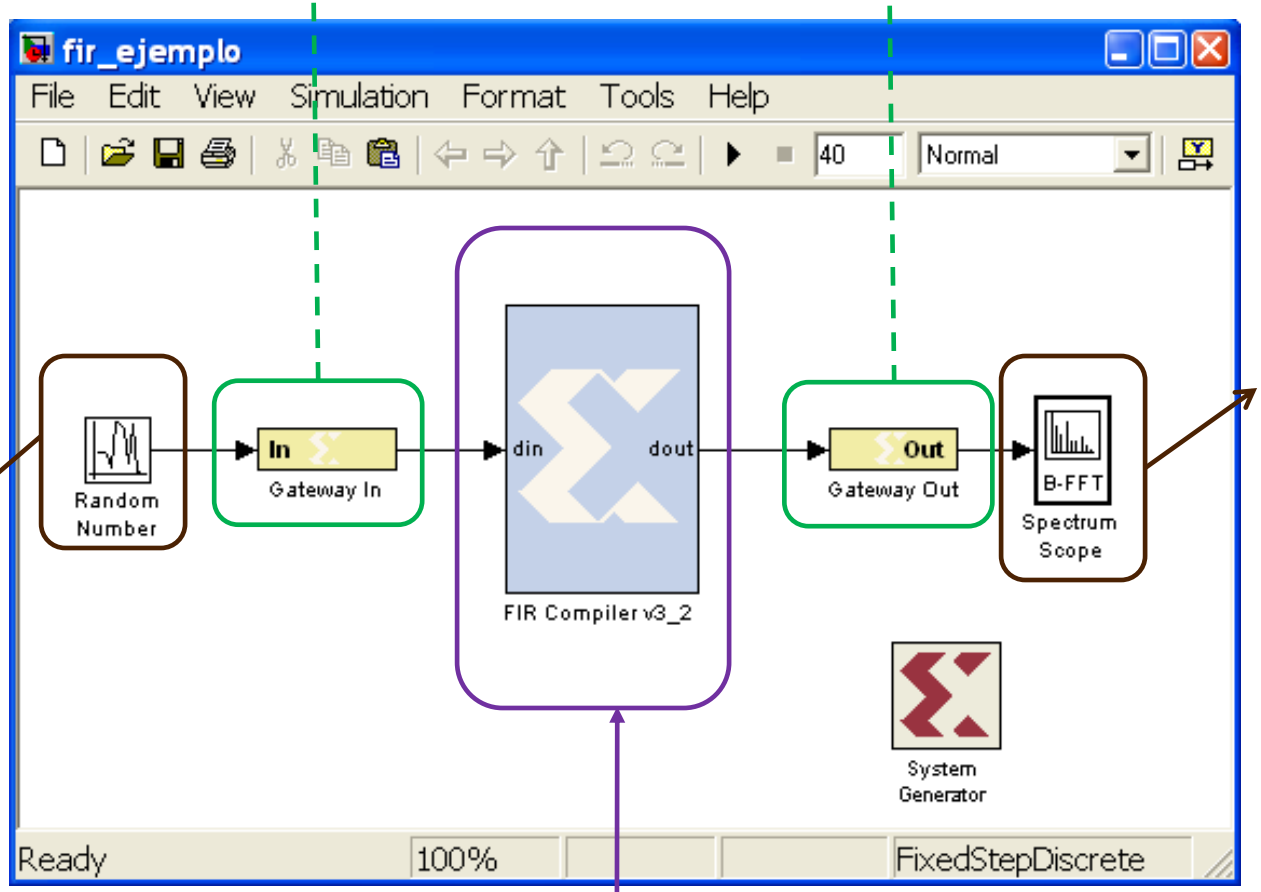


# Usando Xilinx Blockset



# Ejemplo de Diseño

Gateway In/OUT: interface entre SysGen y Simulink

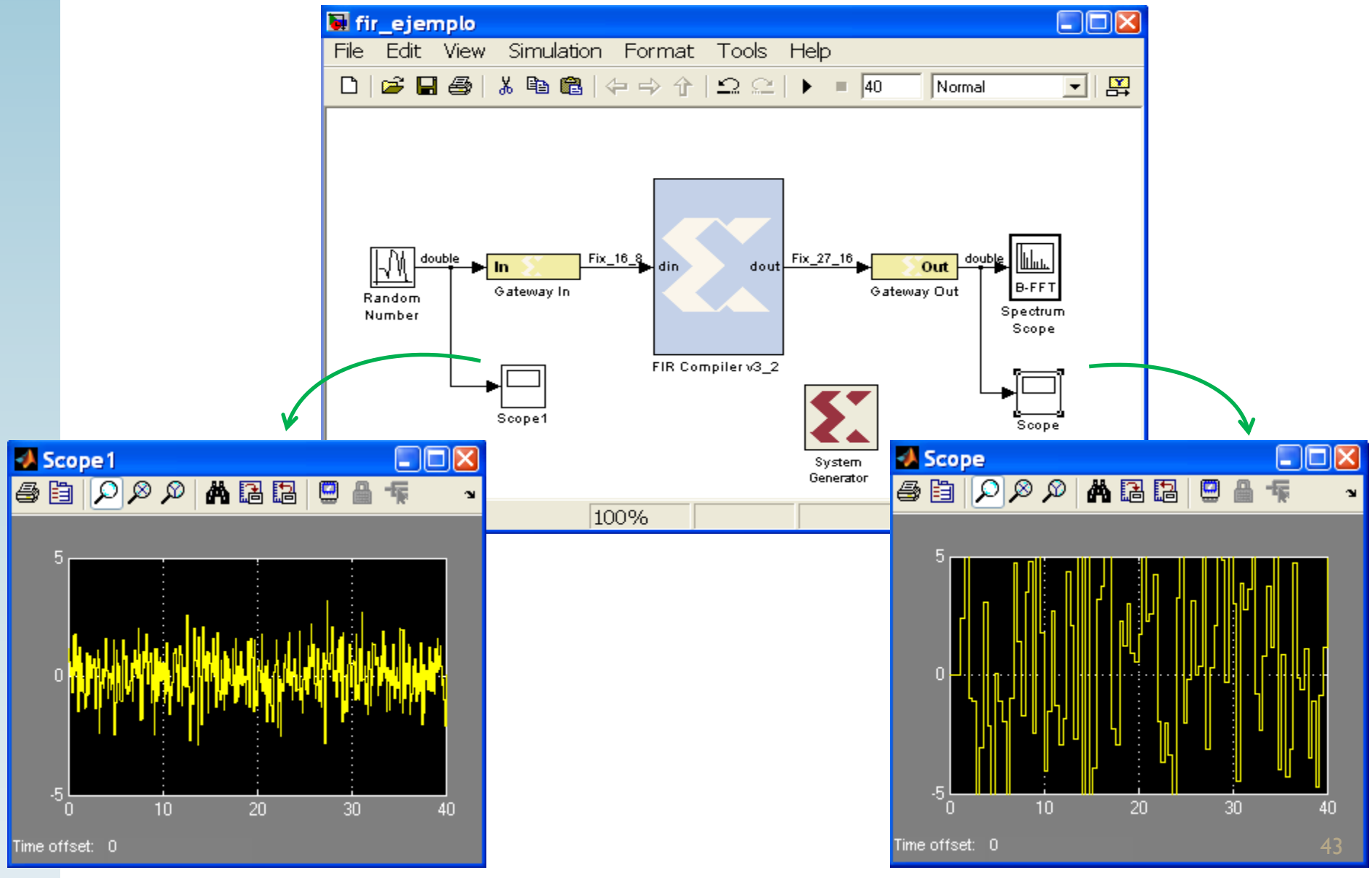



Simulink Graficos de Senal

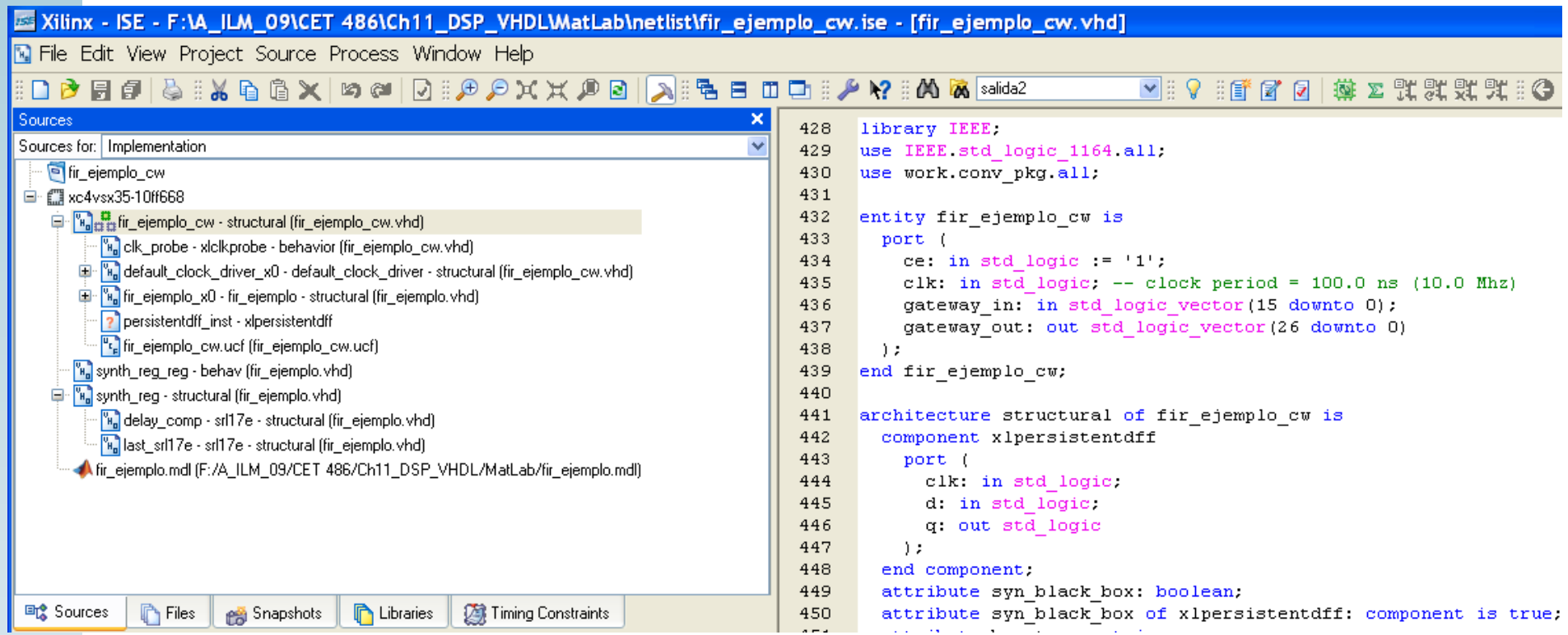
Simulink Fuente de Senal

Lógica a implementar en el FPGA

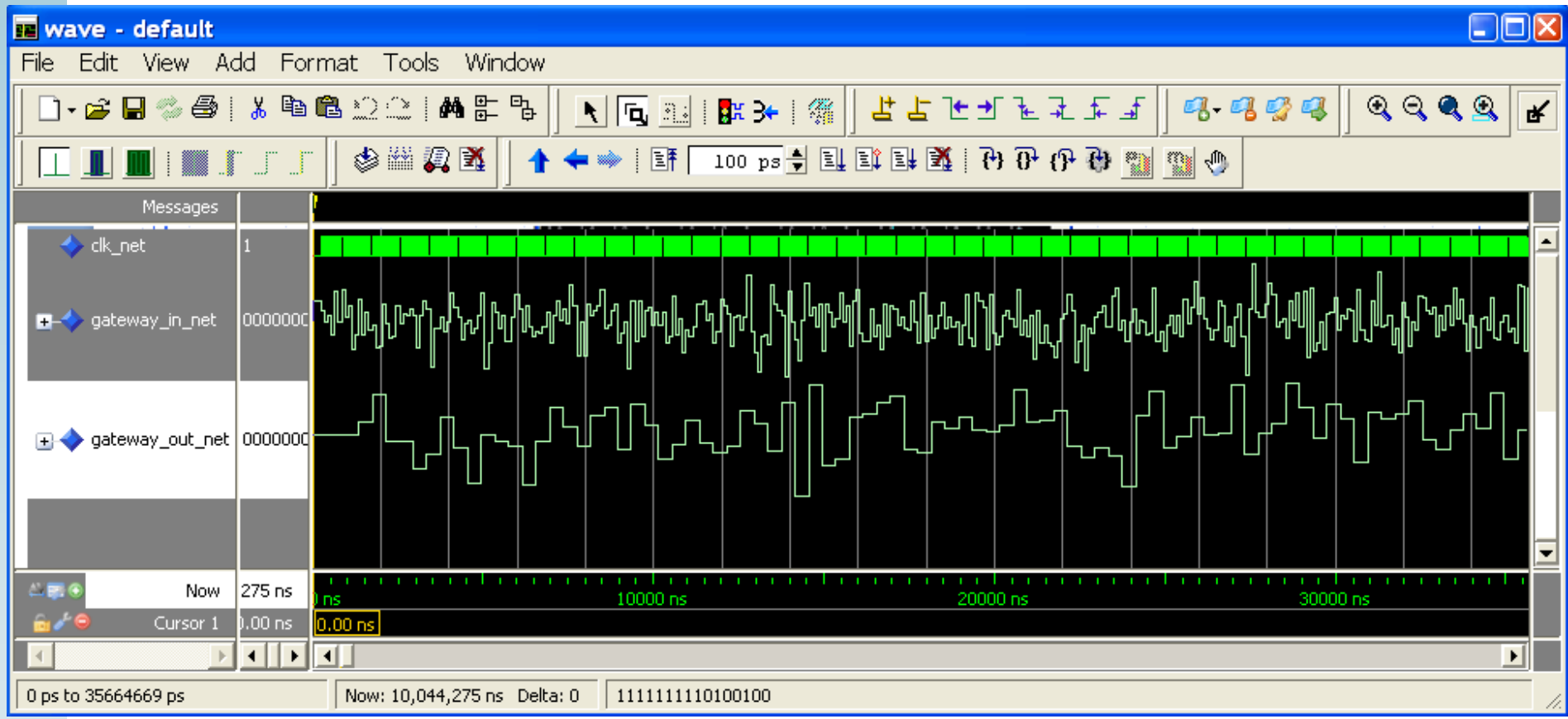
# Simulación en Simulink



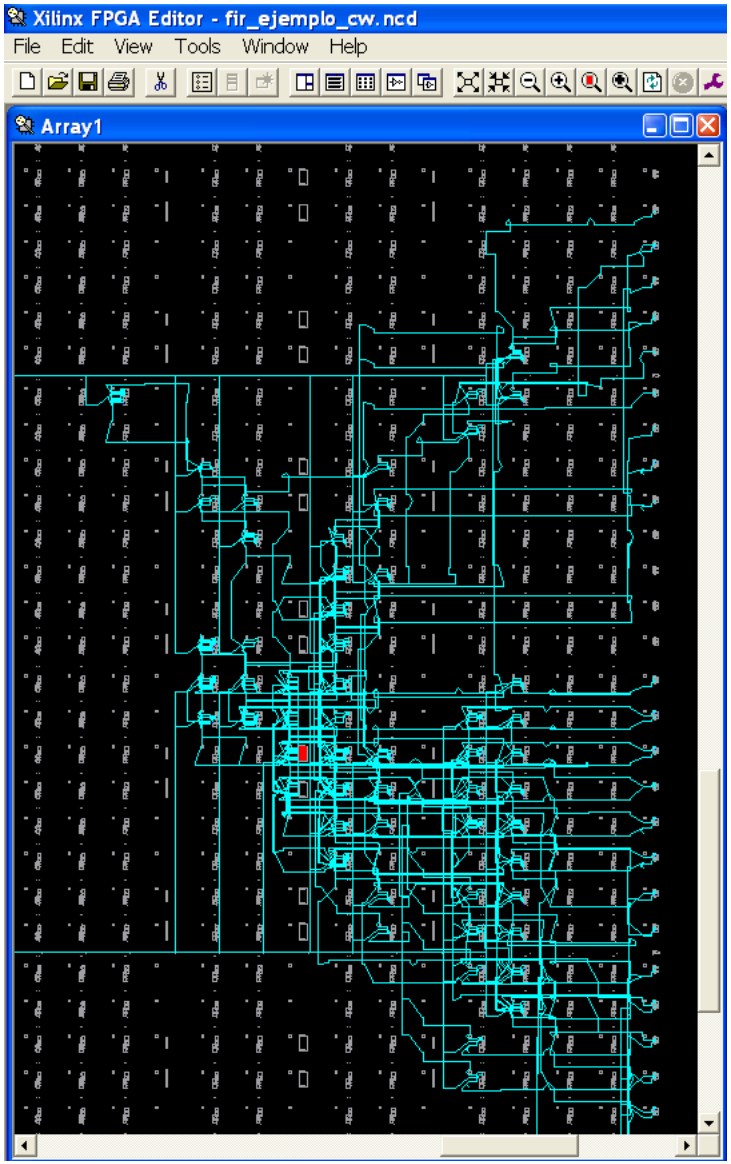
# Proyecto FPGA – Xilinx ISE



# Simulacion en Xilinx



# Implementación del Diseño



# Muchas Gracias !

## C7 Technology

http://www.c7t-hdl.com

http://hdl-fpga.blogspot.com.ar/