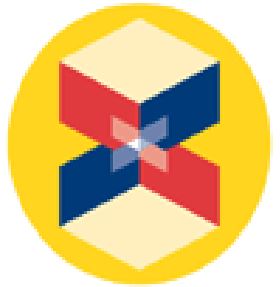
Digitronix Nepal
Innovation from Engineers...

FPGA Design with Matlab & Simulink

Course Prepared by
Digitronix Nepal
www.digitronixnepal.com
Email: digitronixnepali@gmail.com



XILINX
SYSTEM
GENERATOR™
For DSP

Sections and Lecturers of the Course:

Section 1. Installing Matlab/Simulink and Xilinx VIVADO/ISE Design Suit

Lecture 1 : Downloading and Intalling Matlab/Simulink and Xilinx ISE

Lecture 2 : Version Compatibility for Matlab/Simulink and ISE/VIVADO

Section 2. Introduction to Matlab/Simulink and HDL Coder/System Generator

Lecture 1 : Matlab and Simulink Introduction

Lecture 2 : HDL Coder and System Generator Introduction

Lecture 3 : Program Xilinx Zynq SoC Devices with Embedded Coder and HDL Coder

Lecture 4 : Program Xilinx FPGAs Using HDL Coder with Xilinx System Generator

Section 3. Basic Project with System Generator and FPGA

Lecture 1 : Setting up FPGA and JTAG configuration (Adding your Board) to Sys Gen.

Lecture 2 : Dump in to Spartan 3E FPGA

Lecture 3 : Dump in to Zynq FPGA (Zedboard/Zybo)

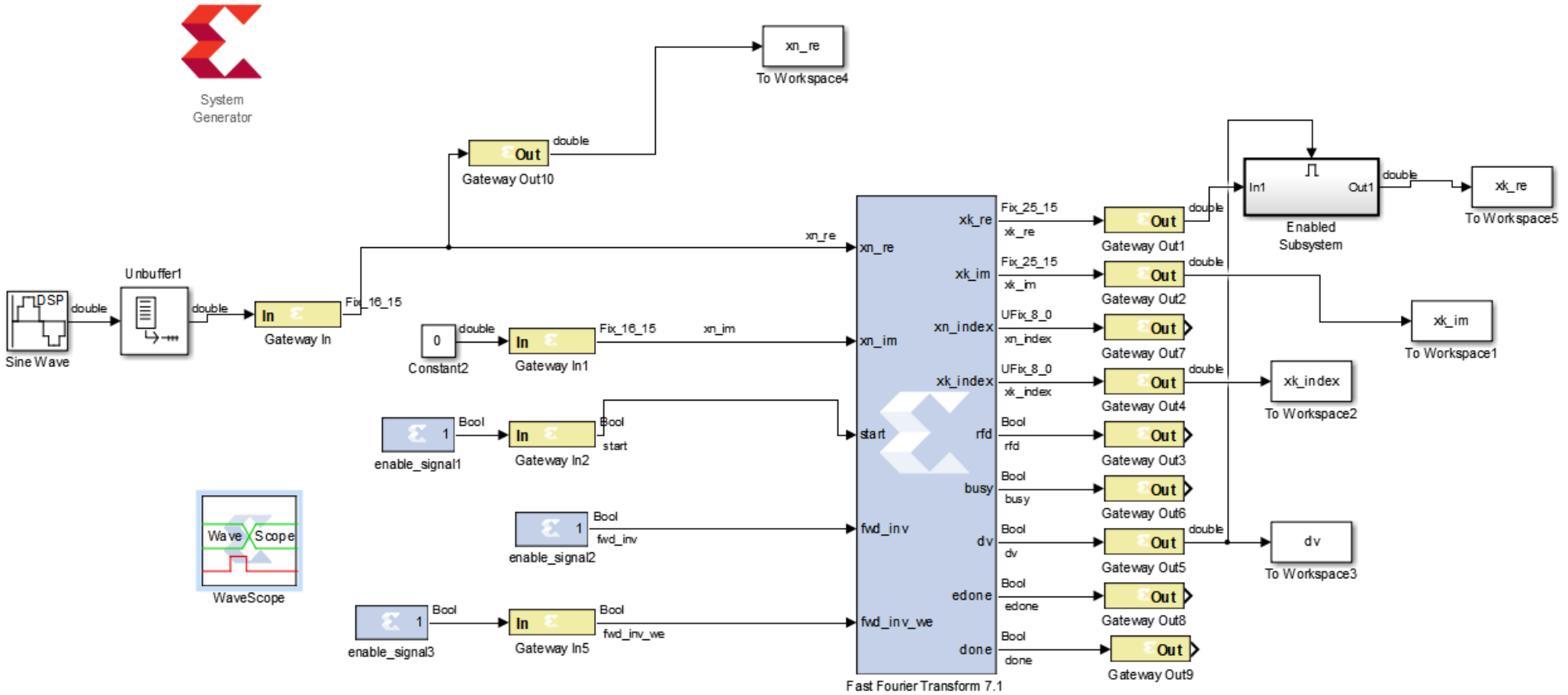
Section 3. Basic Project with System Generator and FPGA

- Lab 31: Basic System Generator Design for FFT
- Lab 32: Creating JTAG Configuration for FPGA Board in System Generator

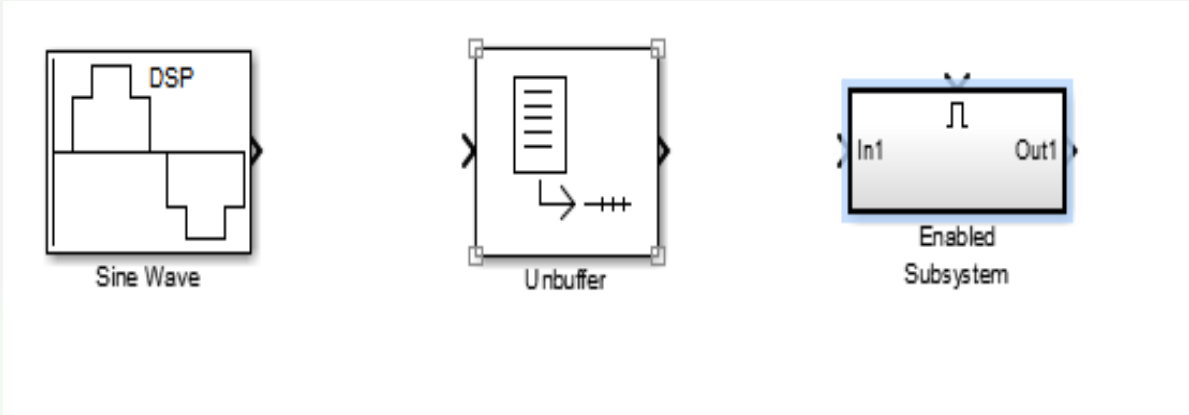
Lab 31: Basic System Generator Design for FFT

Setting up Blocks

- First setup the blocks as shown in the picture.



Simulink Blocks



DSP sine wave: DSP sine wave block is used to generate a digital sinusoidal signal.

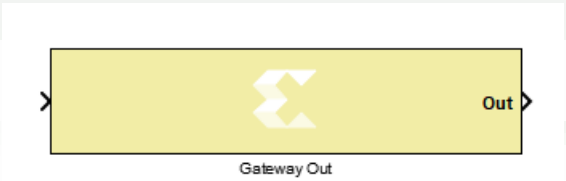
Unbuffer: Unbuffer block converts the frame output from DSP sinewave into a single dimensional scalar output.

Enabled Subsystem: Enabled subsystem will only output values when enable port is high. This is used after the system generator FFT block to get the valid outputs only.

System Generator Blocks

- Gateway In
  - Gateway in block is used to provide input from Simulink blocks to the FPGA blocks

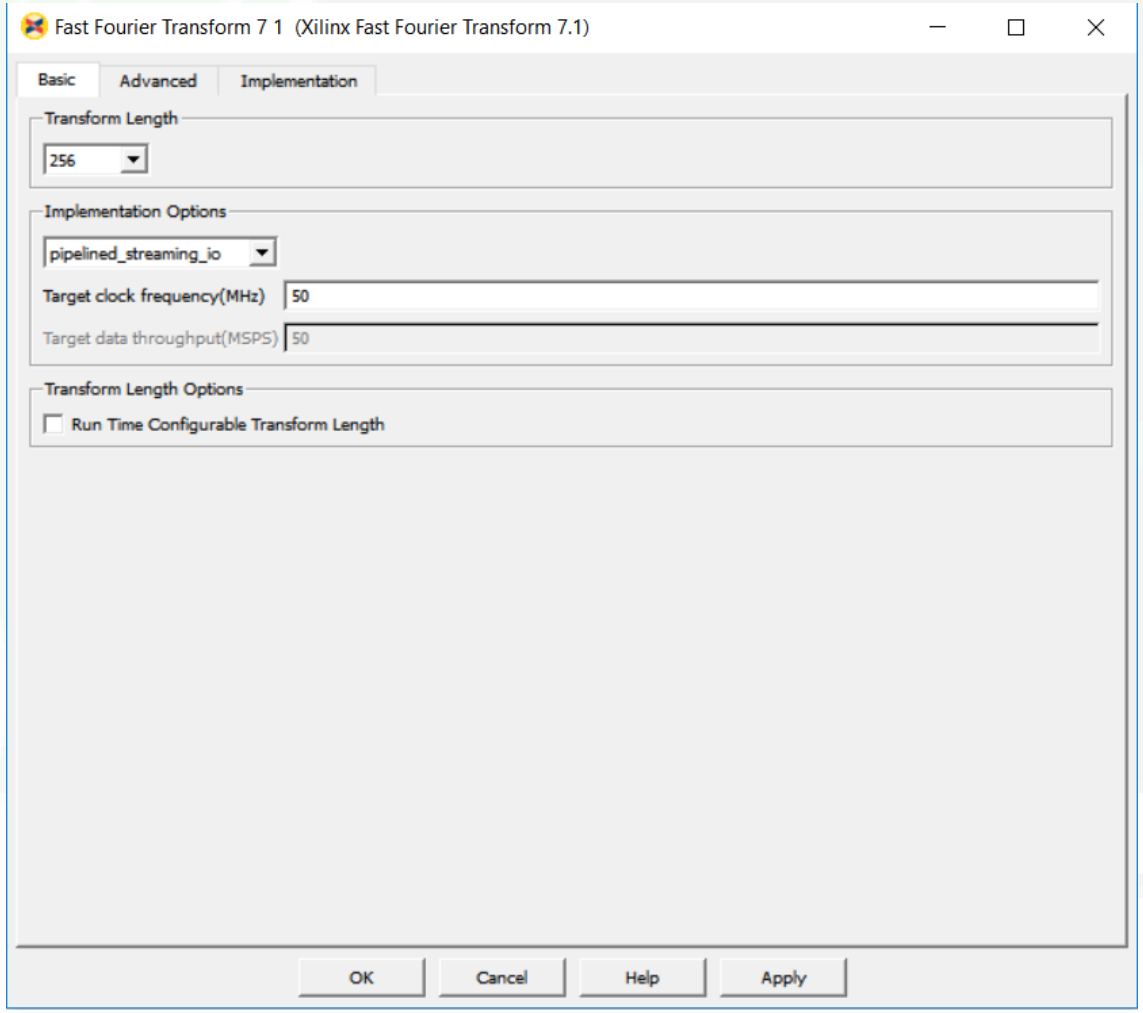
- Gateway Out
  - Gateway out is used to output data or signal from system generator blocks back to Simulink environment.

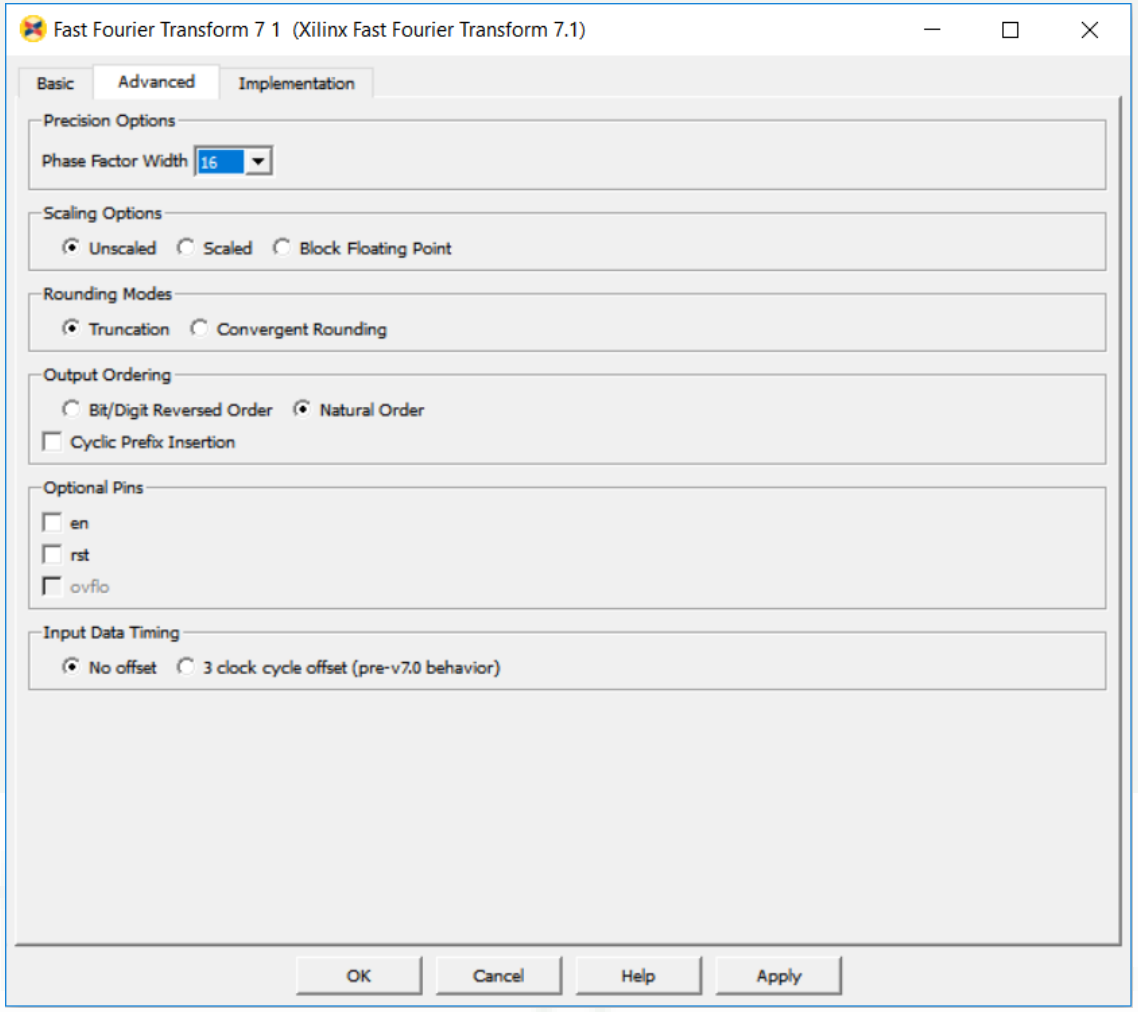


FFT v7.1 core

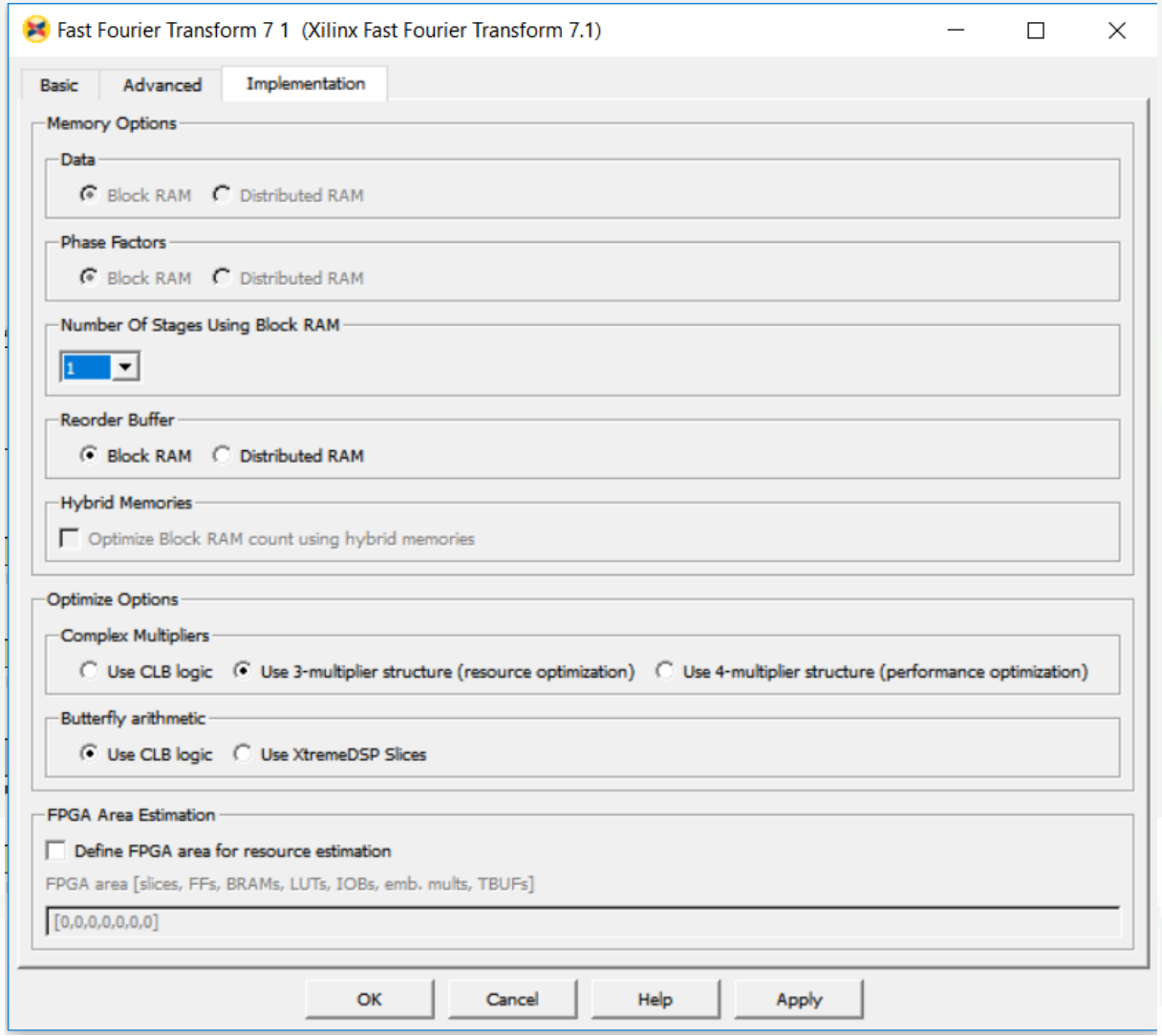
- FFT v7.1 is a DSP block in System Generator and is core part of our design.
- It can process both FFT and IFFT algorithm.
- FFT v7.1 works in burst or pipelined mode and supports Radix-2 and Radix-4 algorithm.
- It can use either block ram or distributed ram resource in FPGA according to our need.
- FFT is used to process signal in frequency domain and it is also used in digital communication for different modulation scheme.

FFT v7.1 Settings

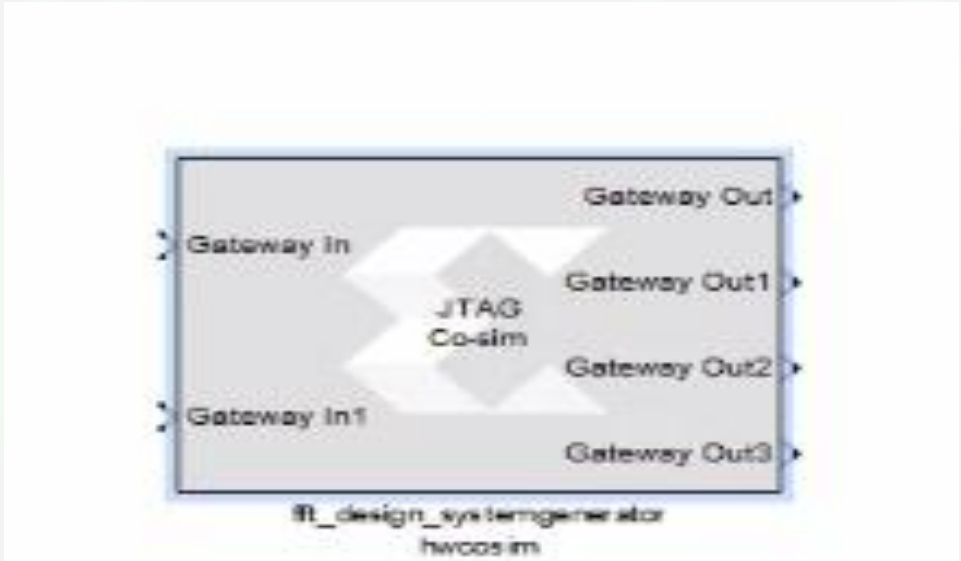


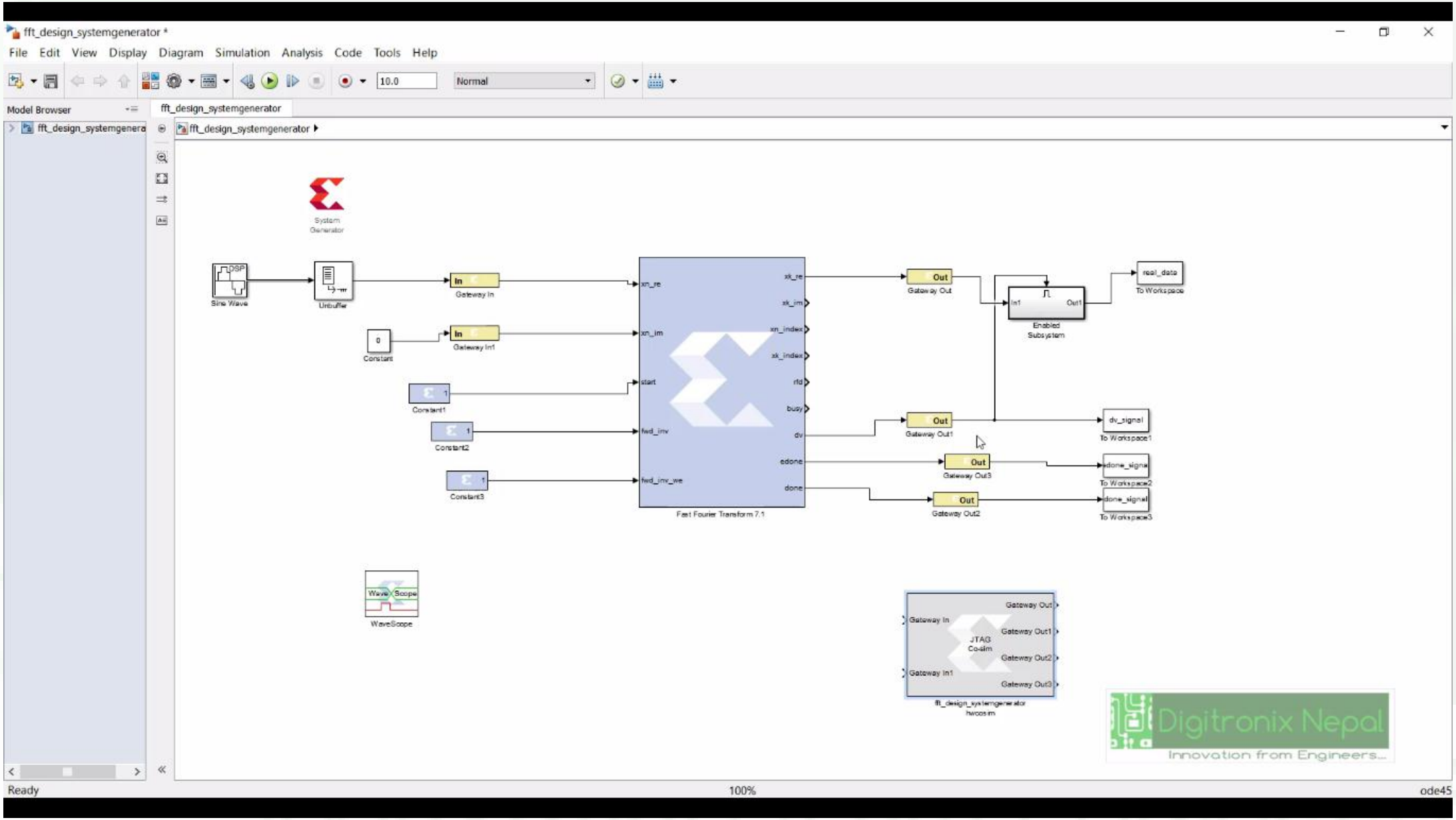


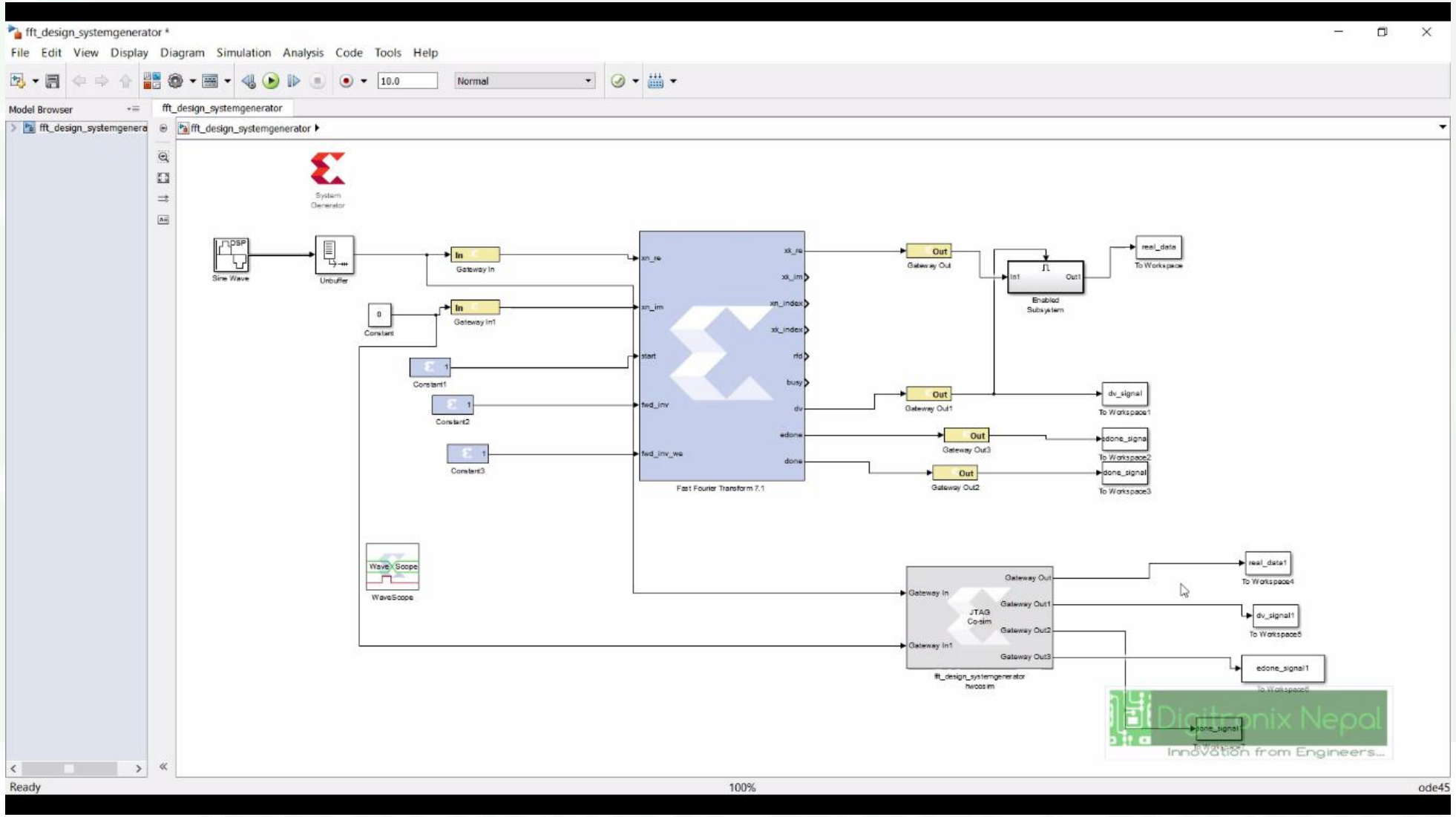
FFT v7.1 Settings



Lab 32: Creating JTAG Configuration for FPGA Board in System Generator




fft_design_systemgenerator *
File Edit View Display Diagram Simulation Analysis Code Tools Help
10.0
Normal
Model Browser
fft_design_systemgenerator
fft_design_systemgenerator
System Generator
Sine Wave
Unbuffer
Gateway In
Gateway In1
Constant
Constant1
Constant2
Constant3
xn_re
xn_im
start
fwd_inv
fwd_inv_we
xk_re
xk_im
xn_index
xk_index
rfd
busy
dv
edone
done
Fast Fourier Transform 7.1
Gateway Out
Gateway Out1
Gateway Out3
Gateway Out2
Enabled Subsystem
real_data
To Workspace
dv_signal
To Workspace1
edone_signal
To Workspace2
done_signal
To Workspace3
WaveScope
Gateway In
Gateway In1
JTAG Co-sim
Gateway Out
Gateway Out1
Gateway Out2
Gateway Out3
fft_design_systemgenerator hwcosim
Digitronix Nepal
Innovation from Engineers...
Ready
100%
ode45

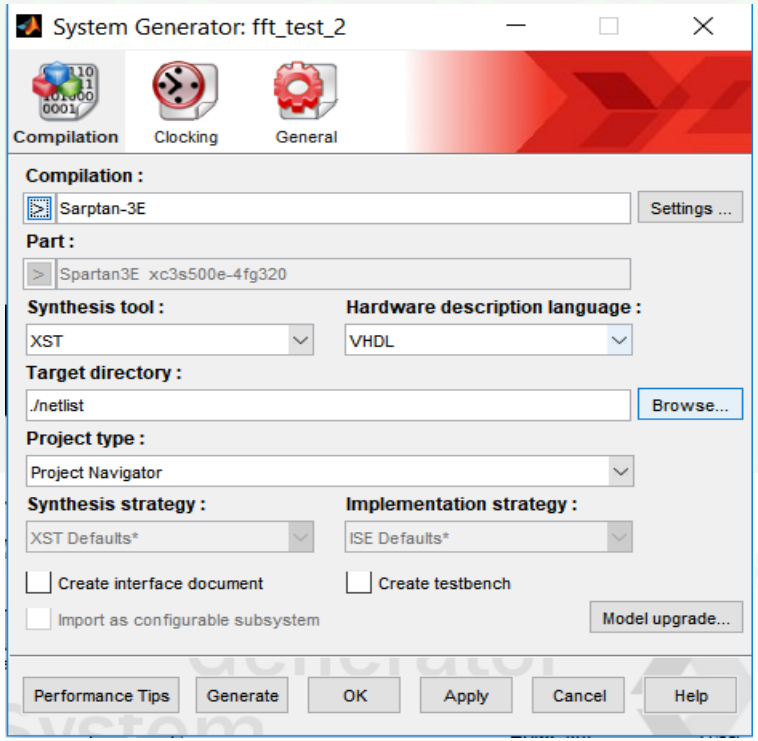

fft_design_systemgenerator *
File Edit View Display Diagram Simulation Analysis Code Tools Help
Model Browser
fft_design_systemgenerator
Sine Wave
Unbuffer
Gateway In
Gateway In1
Constant
Constant1
Constant2
Constant3
Fast Fourier Transform 7.1
Gateway Out
Gateway Out1
Gateway Out2
Gateway Out3
Enabled Subsystem
To Workspace
To Workspace1
To Workspace2
To Workspace3
WaveScope
JTAG Co-sim
To Workspace4
To Workspace5
Digitronix Nepal
Innovation from Engineers...
Ready
100%
ode45

Hardware Co-simulation(HW-cosim)

- For hardware cosimulation we create a JTAG block that contains all the system generator portion.
- The JTAG block processes the our design in physical FPGA device.
- JTAG block can be referred as the physical device itself.
- JTAG configurations are device specific.
- To generate JTAG block we goto system generator token.
- Click Hardware Cosimulation in
- Choose the specific board for hardware cosimulation
- And then click on generate which we create a JTAG block of our design.

JTAG Configuration

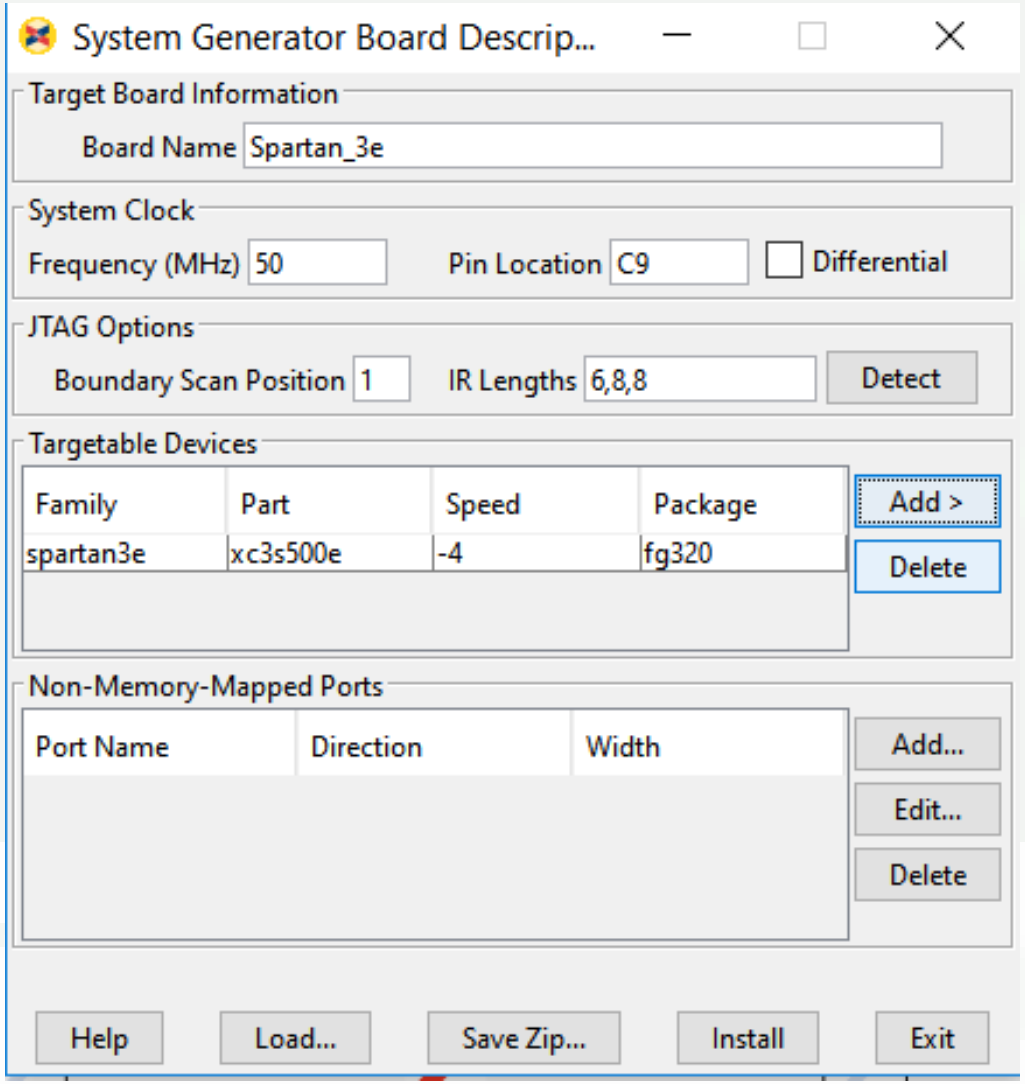
- For hardware co-simulation we have different pre-defined FPGA board configuration in System Generator Token.
- We can also add custom board configuration for different Xilinx FPGA boards



Custom JTAG Configuration

- Goto System generator token
- Select from compilation list and choose Hardware Co-simulation
- In hardware cosimulation choose New compilation target
- A new dialog box will appear where we have to set parameters of our target board.
- We have to input frequency of target FPGA and pin location of its system clock
- We also have to choose FPGA model and packaging configuration.

Custom JTAG Configuration for Spartan 3E Starter Kit



Lets go to

System Generator

Thank You!