Getting Started with System Generator for DSP

Lab 6 – Using Memories

XILINX®

Using Memories

Introduction

In this lab you will learn how to use a Xilinx ROM block to implement a LUT-based operation such as an arcsin using block or distributed RAM. This provides an efficient implementation for trig and math functions with inputs that can be quantized to 10 bits or less.

Objectives

After completing this lab, you will be able to:

- Use a Xilinx ROM block to implement a trig or math function such as arcsin

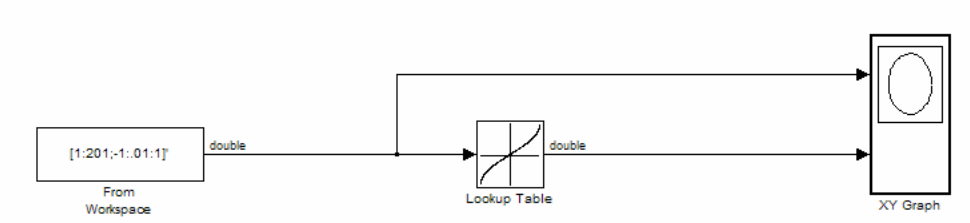
Lab Setup

Please check the System Generator for DSP release notes to insure that the proper versions of ISE Design Suite and MATLAB are installed on your machine. Failure to have the proper tool versions installed may result in unexpected behavior.

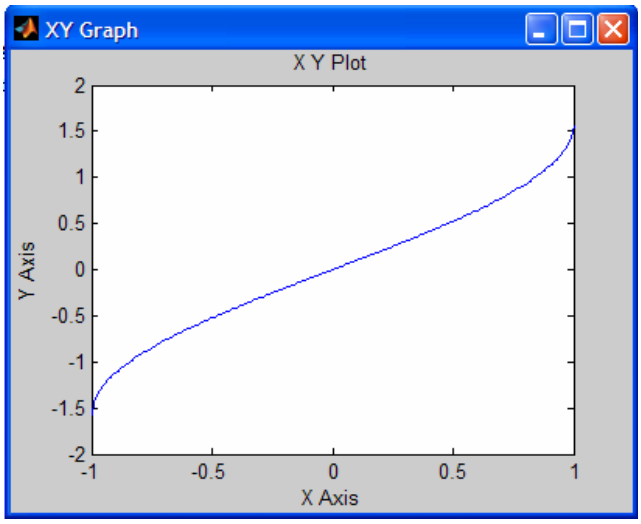
Procedure

❶ Launch the MATLAB program and change the working directory to:
…sysgen/examples/getting_started_training/lab6

❷ As shown below, open the Simulink executable spec named **lab6.mdl**. Double-click on the **Lookup Table** block to see how the arcsine function has been defined.



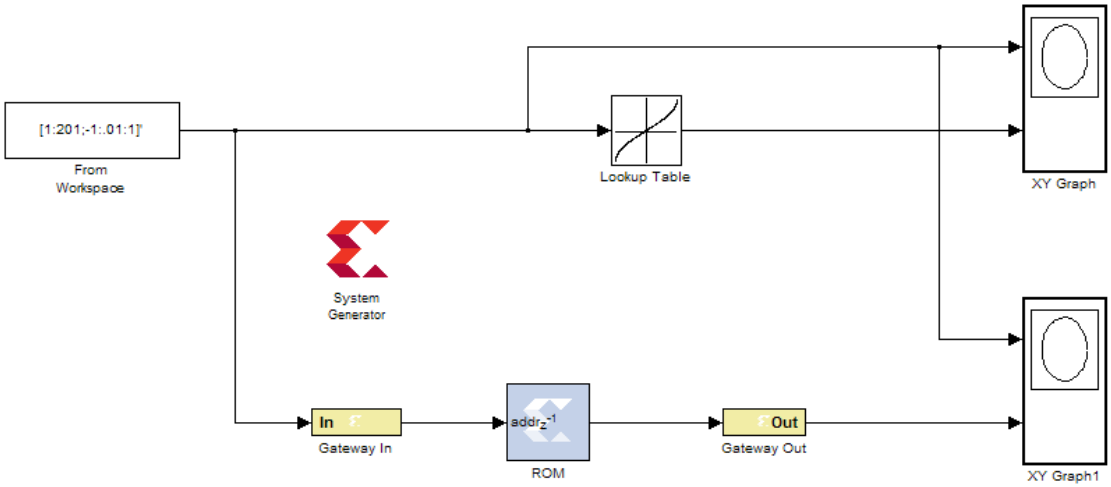
❸ Simulate the counter for 201 simulation cycles and observe the results. The design is using the Simulink **X Y Graph** to plot the output data as a function of the input data.



Note: This plot is a Simulink representation of the MATLAB arcsine example that is displayed when you type **doc asin** from the MATLAB command line prompt.

```
x = -1:.01:1;
plot(x,asin(x)), grid on
```

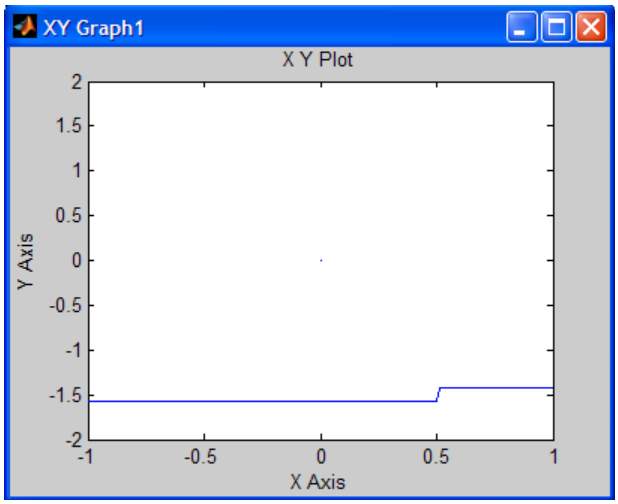
❹ Add a Xilinx **Gateway In**, **Gateway Out**, **System Generator** token and **ROM** block as shown in the figure below.



❺ Double-click on the **ROM** block and set the initialization vector equal to **asin([-1:.01:1])** and the depth to **256**. This is the same initialization as the Simulink **Lookup Table** block. The Xilinx memory blocks are going to require depths to fall on power of 2 boundaries. The MATLAB statement used to initialize the ROM is only going to set 201 locations. The other locations will be uninitialized.

❻ Simulate the design for 201 clock cycles. You are going to get an error indicating an incorrect quantization at the input of the **ROM** block. You left the quantization of the **Gateway In** block to the default value of **fixed [16 14]**. To address a Xilinx memory, the quantization must be **ufix** with no fractional bits. Since you have a 256 element address space, you are going to need an input quantization of **ufix [8 0]**. Change the gateway and re-simulate.

View the waveform in XY Graph1. You will notice that it is incorrect.

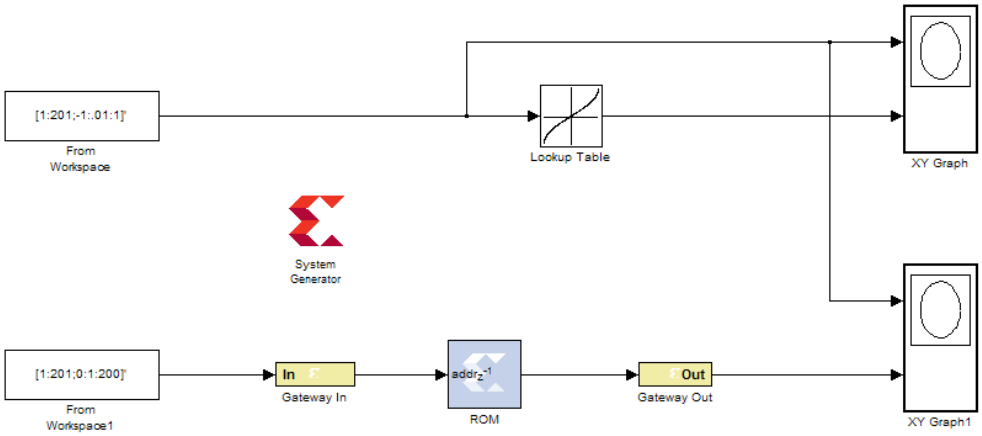


The reason the waveform does not match is that the quantization of the **Gateway In** is truncating the fractional bits and the sign bit. It is not as simple as just re-specifying the quantization. The arcsine value for the input "-1" is stored at address location zero and the

arcsine value for 1 is stored in location 201. To match the behavior of the MATLAB LUT block, which can accept negative and fractional numbers as inputs, you need to convert the input data to an appropriate RAM address.

❼ Add a second **From Workspace** block to the diagram and configure the block to generate outputs from 0 to 201. This is the simplest approach. The "data" field should be specified as follows:

```
[1:201;0:200]'
```



❽ Re-simulate. You should see correct results.

❾ Set the compilation target in the **System Generator** token to **Bitstream**, run System Generator, then record the results.

Registers _____
Block RAMs _____
Slices _____

❿ Double-click on the **ROM** block and set the **Memory Type** field to **Distributed**. Run System Generator and record the results.

Registers _____
Block RAMs _____
Slices _____

Solution

The complete solution to this lab is in the following location:

…sysgen/examples/getting_started_training/lab6/solution