Getting Started with System Generator for DSP

Lab 4 – System Control

XILINX®

System Control

Introduction

In this lab you will be creating a simple state machine using the **MCode** block to detect a sequence of binary values **1011**. The FSM needs to be able to detect multiple transmissions as well, such as **10111011**.

Objectives

After completing this lab, you will be able to:

- Create a finite state machine using the Mcode block in System Generator

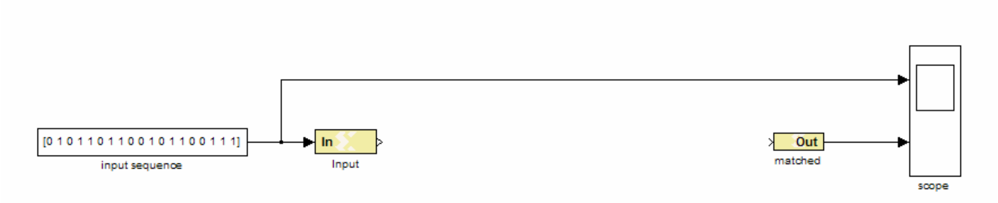
Lab Setup

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

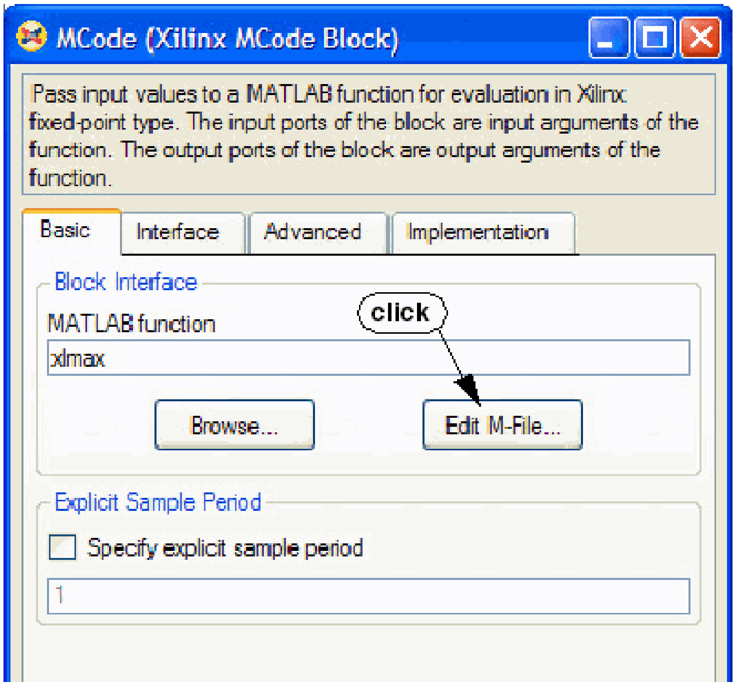
❶ Launch the MATLAB program and change the working directory to:
…sysgen/examples/getting_started_training/lab4

❷ Open up the file **lab4.mdl**. You will see the following uncompleted diagram.



❸ Add an **MCode** block from the Xilinx Blockset "Index" library. Do not wire up the block yet - first you will edit the MATLAB function to create the correct ports and function name.

❹ Double-click on the **MCode** block and as shown below, click on the **Edit M-file…** option.



MCode (Xilinx MCode Block)

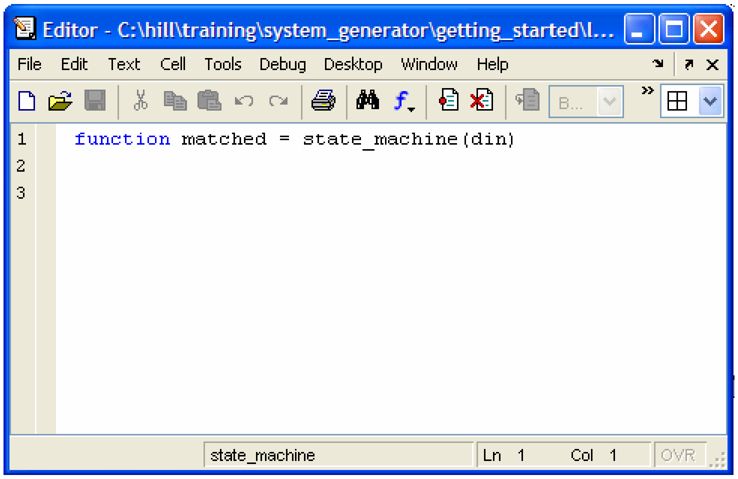
Pass input values to a MATLAB function for evaluation in Xilinx fixed-point type. The input ports of the block are input arguments of the function. The output ports of the block are output arguments of the function.

MATLAB function: xlmax

□ Specify explicit sample period

1

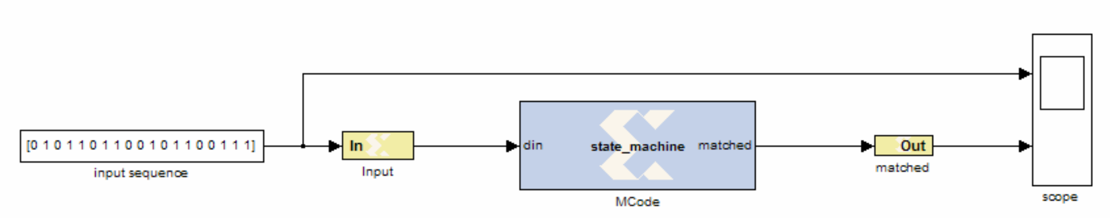
❺ Edit the default MATLAB function to include the function name "state_machine" and the input **din** and output **matched**. The sample M-code can now be deleted.


Editor - C:\hill\training\system_generator\getting_started\l...
File Edit Text Cell Tools Debug Desktop Window Help
state_machine
Ln 1
Col 1
OVR

```
function matched = state_machine(din)
```

❻ Once the edits have been made, save the MATLAB file to the lab4 folder and use the **Browse** button to make sure that the **MCode** block is referencing the local MATLAB file.

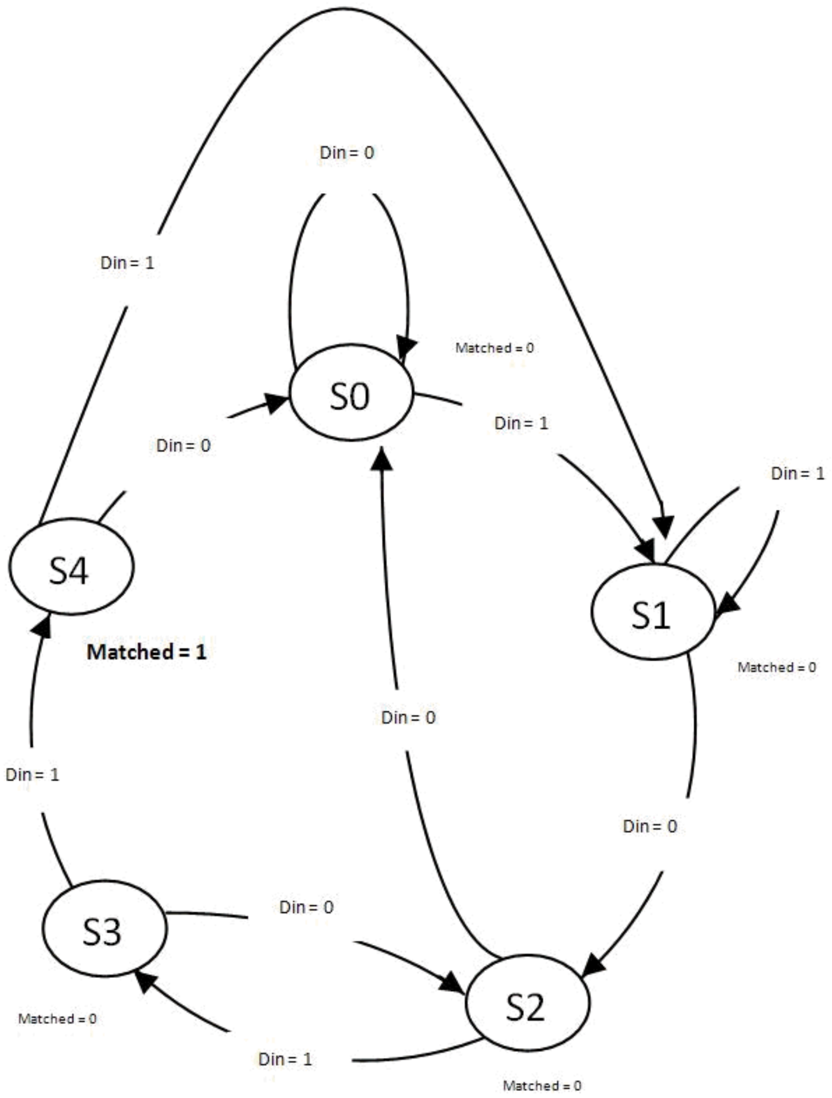
❼ Click **OK** on the MCode Properties form. You will see the **MCode** block assume the new ports and function name. Now connect the **MCode** block to the diagram as shown below



You are now ready to start coding the state machine. The bubble diagram for this state machine is shown below. This FSM will have 5 states and will be capable of detecting two sequences in succession.

❽ Edit the MATLAB and define the state variable using the Xilinx **xl_state** data type as shown below. This requires that the variable be declared as a persistent variable. The **xl_state** function requires two arguments, the initial condition and a fixed-point declaration. Since you need to go up to 4, you will need 3 bits.

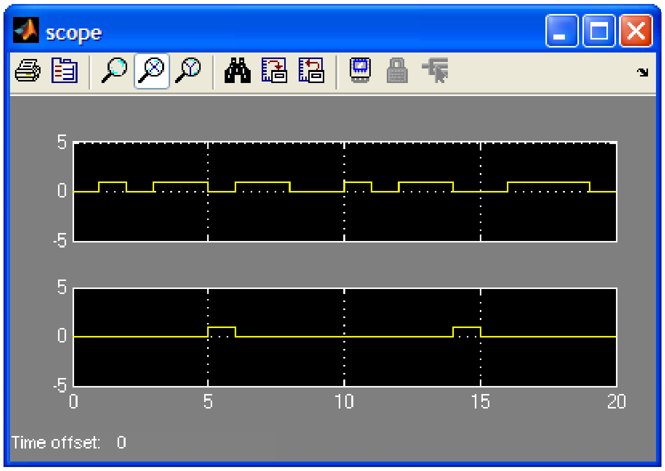
```
persistent state, state = xl_state(0,{xlUnsigned, 3, 0});
```



❾ Use a `switch - case` statement to define the FSM states shown. A small sample is provided below to get you started. Note that you will need an `otherwise` statement as your last case.

```
switch state
    case 0
        if din == 1
            state = 1;
        else
            state = 0;
        end
        matched = 0;
```

⓾ Save the MATLAB file and run the simulation. The waveform should look like the following. You should notice two detections of the sequence.



Solution

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

...<*sysgen_tree*>**/examples/getting_started_training/lab4/solution**