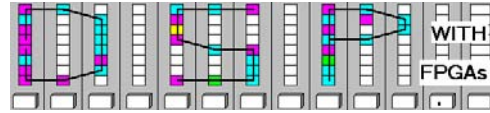


Lastname: _____

LABORATORY
Intro. to M-File



5Digit SS: _____

LAB M-FILE: INTRODUCTION TO MATLAB M-FILE SCRIPTS
(10 points)

In this lab you will be introduced to the MatLab M-file coding. MatLab is a powerful DSP interpreter, that allows you quick and efficient to develop LUT and test bench data for your FPGA design. We will in this lab extend our function generator from Lab 1 with additional test functions and we will also write M-file scripts to test a complex multiplier design that needs 3 real multiplications and 5 add operations.

In the **pre-lab** you will compute with “pencil-and-paper” the results you later expect in your design implementation. In the **design part** you will design a function generator for 4 different functions and a complex multiplier.

Lab Objectives

After completing this lab you should be able to

- write simple M-file scripts to define tables and to use predefined functions.
- use the MatLab help, demo and function library
- design and simulate complex multiplier systems using Simulink

Pre-lab (3 points)

Note that MatLab can be accessed from any computer in the college using the X-Win32 program on dingo.

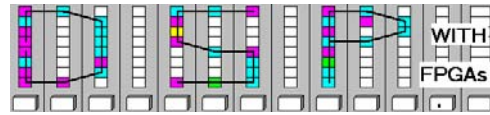
1. MatLab works like a very powerful pocket calculator and has a wide selection of predefined functions. You can easily define data, and process, manipulate, or plot these data. Using the MatLab prompt determine the operation of the following MatLab instructions:

MatLab instruction	Short description
<code>a=zeros(3,5);</code>	
<code>help fir1</code>	
<code>lookfor convolution</code>	
<code>xpsound</code>	
<code>x=0:127;</code>	
<code>y=sin(2*pi*x/128);</code>	
<code>v=100*rand(1,128);</code>	
<code>w=floor(x/16);</code>	
<code>f=fft(y);</code>	
<code>plot(abs(f))</code>	
<code>whos</code>	

Note: If you are not sure about the function just try it out in MatLab!

Lastname: _____

**LABORATORY
Intro. to M-File**



5Digit SS: _____

2. A complex multiplication of the type $(x+jy)(c+js)=xc-sy+j(xs+yc)=Re+jIm$ is a frequent used object of in the DFT computation discussed in later labs. The complex multiplication in the direct form requires 4 real multiplications and 2 add operations. From an implementations standpoint the 4 multipliers realized as array multipliers are the most resource intensive objects and algorithms that use 3 multipliers and 5 adds are usually more resource effective. Such a algorithms works as follows:

Compute first: $a_1=c+s, a_2=c-s, \text{ and } a_3=x-y$
then compute $m_1=c*a_3, m_2=y*a_2, \text{ and } m_3=x*a_1,$
and finally: $r=m_2+m_1 \text{ and } i=m_3-m_1$

as can be seen it uses $3+2=5$ add and 3 multiply operations. Develop a short MatLab script that computes the output for 8 bit data and coefficient quantization by completing the following M-Files:

```
% m file to check the complex multiplier with 3*5+
x=60; y=40;
w=exp(j*pi/5);
c=round(real(w)*128);
s=round(imag(w)*128);
%% add the equation for a1-a3 and m1-m3, re and im here

%% display the results
check=(x+j*y)*w;
str= sprintf('result = %5d+j%5d => check: %5.3f+j%5.3f',...
    floor(re/128), floor(im/128), real(check), imag(check));
disp(str);
```

Store the file under the name `cmtest.m` on your DSP4FPGAs directory. To call the file, just type its name (without the `.m` extension) in the MatLab prompt. Make sure using `dir *.m` and `cd` instructions that your MatLab prompt points to the right directory where you have placed your M-file. Now use the script and produce some test data:

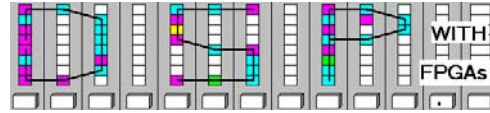
$x+jy$	$w=\exp(j\varphi)=(c+js)$	$(x+jy)*w$	$c*128$	$s*128$	$(c+s)*128$	$(c-s)128$
60+j40	$\varphi=\pi/5$					
70+j50	$\varphi=\pi/9$					
60+j40	$\varphi=-\pi/9$					
70-j50	$\varphi=-\pi/5$					

Make sure your name and SS is on all pages you turn in!

Lastname: _____

5Digit SS: _____

LABORATORY
Intro. to M-File



Simulink Design-lab (7 points)

Follow the directions below to implement the 4 function generator and the complex multiplier.

A. Getting Started

If you are in B114 or the digital logic lab:

1. On the desktop double click on **Engineering folder**.
2. Double click on **MatLab** icon  to start **MatLab**.
3. From the top icon list in the **MatLab** window click on the **Simulink** icon  to start **Simulink**.
4. You should not save anything on the local hard disk. You will have to use a Zip, a floppy disc, or your "mapped" home directory to save the files. Create a New Folder named **DSPwFPGAs** on your mapped network drive.

B. Completion of the Complex multiplier Design

1. Complete you design as shown in Fig. 1.

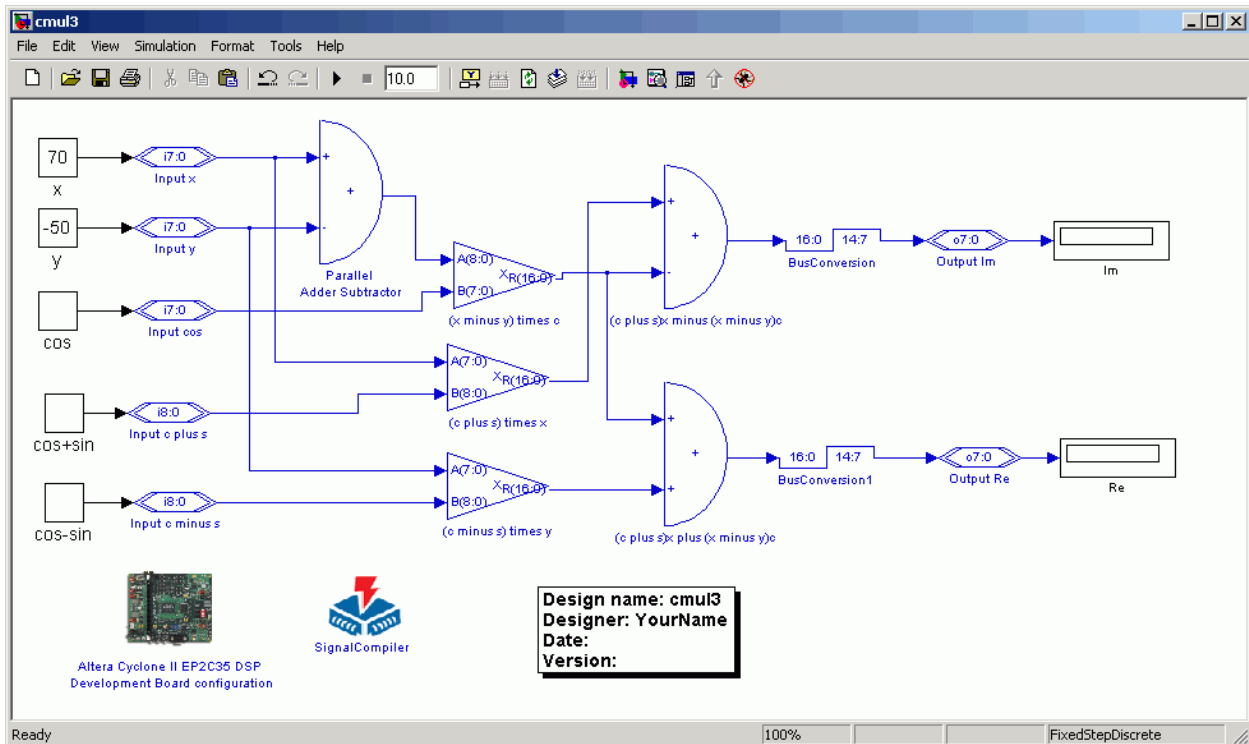
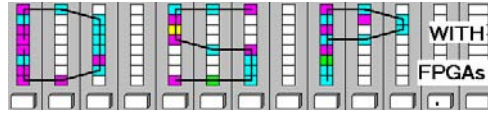


Fig. 1: Complex multiplier using three real multiplications.

Lastname: _____

LABORATORY
Intro. to M-File



5Digit SS: _____

- Verify the design via Simulink simulation for the 4 different test data computed in the prelab and complete the following table:


$x+jy$	$w=\exp(j\varphi)=(c+js)$	Re	Im
$60+j40$	$\varphi=\pi/5$		
$70+j50$	$\varphi=\pi/9$		
$60-j40$	$\varphi=-\pi/9$		
$70-j50$	$\varphi=-\pi/5$		

- Compile the design using **Signal Compiler** and determine

LEs = _____
Embedded 9-bit Multiplier = _____

from the report files.

C. Design the Function generator systems

- Download the file `funcgen.m` from the class webpage and put the file in the **DSPwFPGAs** folder.
- Click on the “Current Directory” selection icon  and select as current directory the **DSPwFPGAs** folder.
- The files in the **DSPwFPGAs** folder are now visible in the upper left **MatLab** window. Double click on the `funcgen.m` file and you should see after a moment the incomplete M-file:

```
clear;
%% m file to compute the 4 functions in the ML lab
x=0:2^7-1;           %% all have length 128
y0= zeros(1,128);  %% first is the sine function
y1= zeros(1,128);  %% second is the multi step function
y2= zeros(1,128);  %% uniform random numbers
y3= zeros(1,128);  %% symmetric triangular

%%%% Plot the results and store in LUT
plot(x,y0,'k-',x,y1,'k--',x,y2,'k.',x,y3,'k-.');
legend('sine','multi step','random','triangular')
xlabel('Sample');ylabel('Amplitude');title('4 Functions generator')
axis([0 127 0 1.1*2^14]);
print -djpeg funcgen.jpg

LUT=[y0, y1, y2, y3];
```

- Complete your M-file so that the required 4 functions, y_0, \dots, y_4 , are plotted as shown in Fig. 2. Note that the Y range is $[0, 2^{14}-1]$ unsigned.

Lastname: _____

5Digit SS: _____

LABORATORY Intro. to M-File

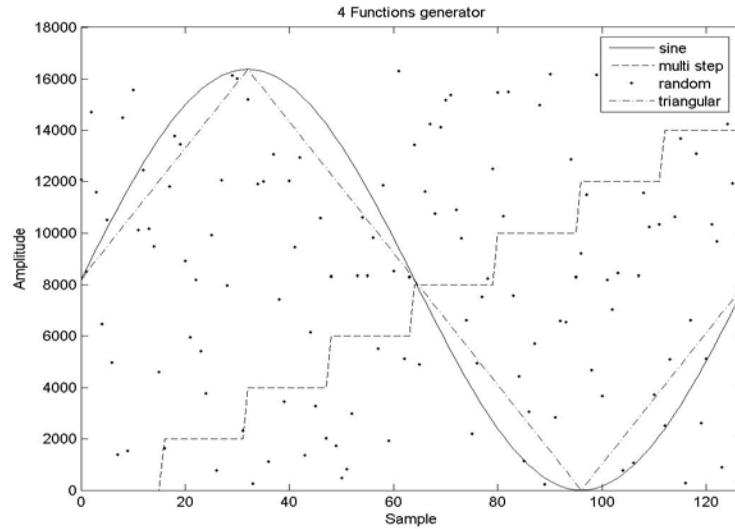
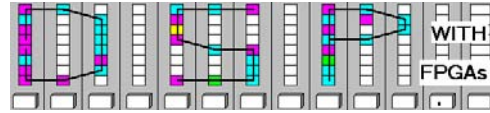


Fig. 2: Plot of 4 different functions.

5. Now add in your lab 1 design file lab1.mdl the **DIP switch**, **LED 6**, **LED 7**, **BusConcatenation** block, **ExtractBit**, **BusBuild**, **Repeating Sequence Stair** block and a **LUT** of 2 more input bits to your design as shown in the following Fig. 3. Store the file under the name **funcgenerator.mdl**.

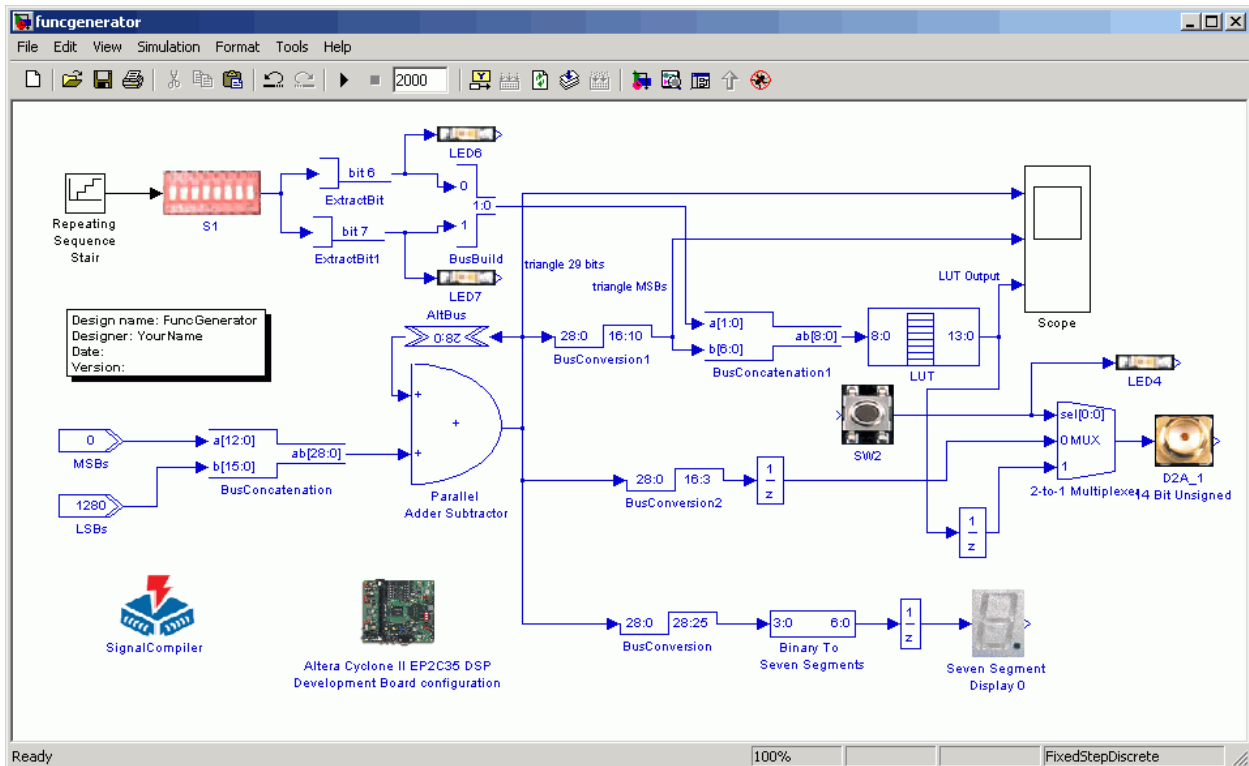
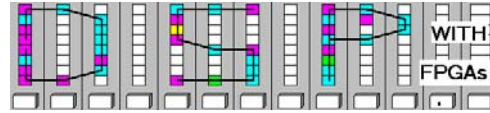


Fig. 3: Lab 1 extended with the **DIP switch** for function selection and a **Repeating Sequence Stair**.

Lastname: _____

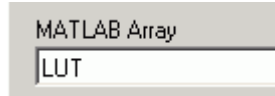
LABORATORY
Intro. to M-File



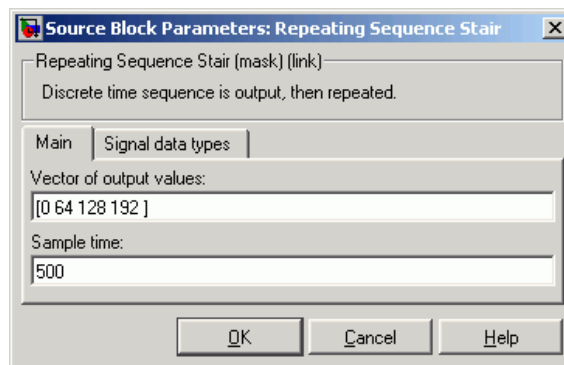
5Digit SS: _____

D. Completion of the Function Generator design

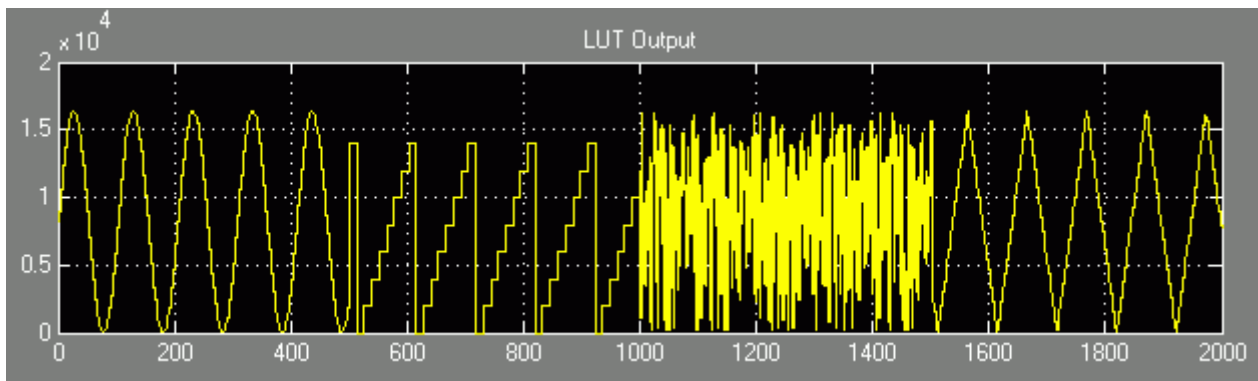
4. Double click the **LUT** block and enter under MATLAB array your new variable LUT:



5. Complete your design as shown in Fig. 3. The **Repeating Sequence Stair** block can be found in the **Simulink->Sources** subdirectory. Double click the **Repeating Sequence Stair** block and enter the data shown below:



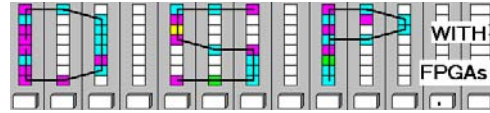
6. Verify the design via Simulink simulation. Change the Simulation stop time to 2000 and the LSBs value to 1280 so that all four set of functions are viewed. (Print out the simulation waveform for LUT scope.) The simulation should look as follows:



7. Next double-click the **SignalCompiler** block. Note That you can only compile the design on a PC that has a board, i.e. has a DSP builder license enabled.
8. Check the box for **Re-run update diagram to solve workspace parameters**, and then click **Analyze**.
9. The **SignalCompiler** block should pop up. Make sure the Reset option is set to Active Low. Click on the **Execute steps 1, 2 and 3** button. (Note: This may take a couple of minutes.)

Lastname: _____

LABORATORY
Intro. to M-File



5Digit SS: _____

10. Watch the output of the D2A output with an oscilloscope or the 7-segment LEDs in case you do not have an oscilloscope connected to the board. Change the bits 6 and 7 of the **DIP switch** so that you can see all 4 functions on the oscilloscope.
11. Show the lab TA your working FPGA board design.
12. Using the Report File determine the number of logic elements and the maximum frequency of the device:

LCs = _____

Total memory bits = _____

MHz = _____

from the report files.

E. Deliverables:

1. Solve the problems of the pre-lab. (3 points).
2. Print the 2 MDF files and the Simulink simulation (7 points).

Make sure your name and SS is on all pages you turn in!