Lastname: ___________________ **LABORATORY DFT**

5Digit SS:___________________

LAB DFT: INTRODUCTION TO DISCRETE FOURIER TRANSFORM (10 points)

In this lab you will be introduced to the design for a Discrete Fourier Transform (DFT) using the Goertzel iterative computation. The DFT is described by the following equation:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad k = 0,1,......N-1 \tag{1}$$

$$\text{with } W_N^{kn} = e^{-j2\pi kn/N} \tag{2}$$

The DFT is an important DSP object and can be used not only to compute an approximation of the Fourier Transform but also to build narrow band filters without the need of sophisticated filter design tools. The Goertzel algorithm can be used to implement a single DFT component via a first order IIR filter.

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 an 8 point DFT using the Goertzel algorithm.

Lab Objectives

After completing this lab you should be able to

- Develop a basic Goertzel IIR loop and compute test data
- Create and configure a sub design with I/O ports
- Instantiate a previously developed block
- Design and simulate selected DFT components

Pre-lab (3 points)

The following figure shows the basic building block used to build the Goertzel DFT:

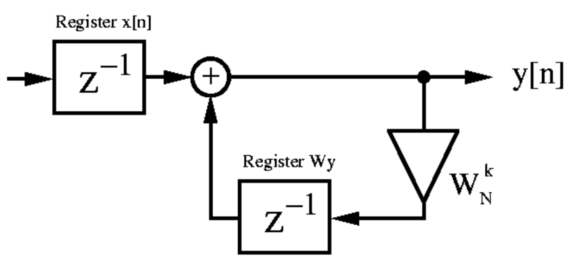


Fig. 1

1. Determine, for N=8, the values $W_N{}^k$ for k=0, 1, 2 and 3. Use no more than 3 fractional digits for $W_N{}^k$. Compute the values for sin and cos, scaled by 128 and quantized to signed integers [-128,128]. Also, compute cos+sin and cos-sin, used for the complex multiplier.

k =	0	1	2	3
$W_N{}^k$ =				
cos				
sin				
cos+sin				
cos-sin				

2. Determine for N=8 and k=0, from Fig.1, the values for Register *Wy* and for $y[n]$. Note that the incoming data $x[n]$ is added with the previous value. Both real and imaginary parts show a triangular input sequence. Use integer values for register *Wy* and for $y[n]$. A short MatLab script is called `goertzel8.m` provided at the class's website that should aid you with these calculations.

Time step	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Register x[n]	16+j16	14+j14	12+j12	10+j10	8+j8	6+j6	4+j4	2+j2
Register Wy	0	16+j16	30+j30					
y[n]	16+j16	30+j30						72+j72

3. Repeat (2) for k=1. Use integer values for final values in registers *Wy* and $y[n]$.

Time step	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Register x[n]	16+j16	14+j14	12+j12	10+j10	8+j8	6+j6	4+j4	2+j2
Register Wy	0							
y[n]	16+j16							

4. Repeat (2) for k=2. Use integer values for registers *Wy* and $y[n]$.

Time step	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Register x[n]	16+j16	14+j14	12+j12	10+j10	8+j8	6+j6	4+j4	2+j2
Register Wy	0	16-j16						
y[n]	16+j16	30-j2						

5. Repeat (2) for k=3. Use integer values for registers *Wy* and $y[n]$.

Time step	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Register x[n]	16+j16	14+j14	12+j12	10+j10	8+j8	6+j6	4+j4	2+j2
Register Wy	0							
y[n]	16+j16							

6. Using MatLab, compute the FFT, X = fft(x), for t = 2:2:16, x = t + j*t and complete the following table using no more than 4 fractional digits:

	X[0]	**X[1]**	**X[2]**	**X[3]**	**X[4]**	**X[5]**	**X[6]**	**X[7]**
Real								
Imag								

Verify the results from 6 with the data computed in 2-5. Do all final values y[7] and X[k] match?

7. Discuss advantages and disadvantages of the Goertzel algorithm, in term of algorithm flexibility, design size, sensitivity to coefficient quantization and latency of the computation.

<u>Advantages (name at least 2)</u>

<u>Disadvantages (name at least 3)</u>

Simulink Design-lab (7 points)

Follow the directions below to implement the 8-point Goertzel DFT circuit.

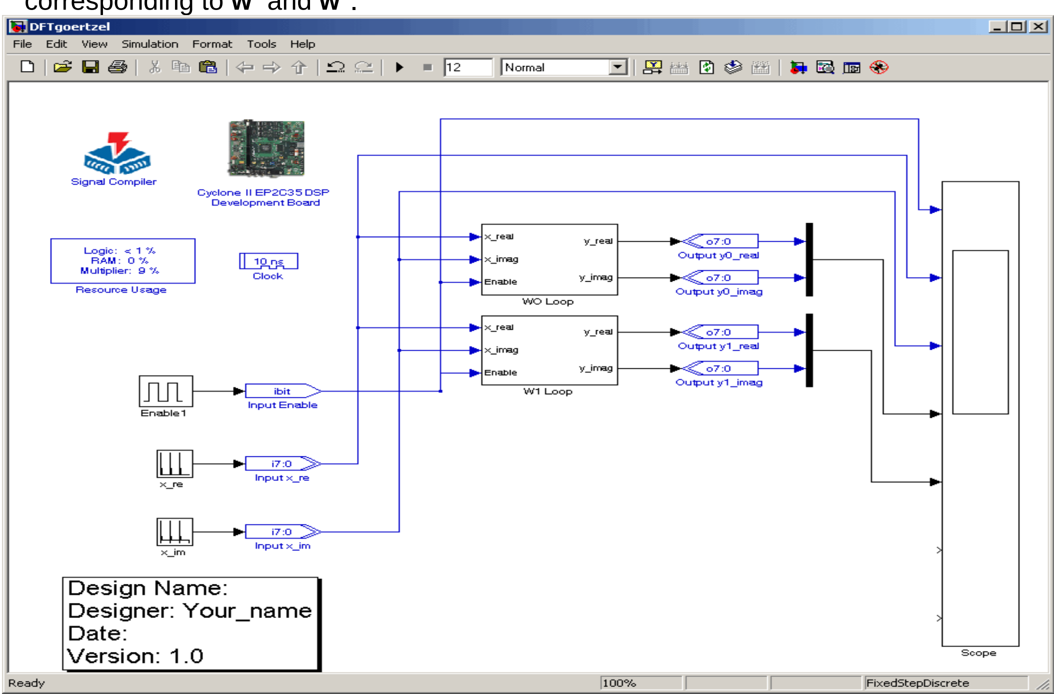
A. Getting Started

If you are in room B114 or the digital logic lab:

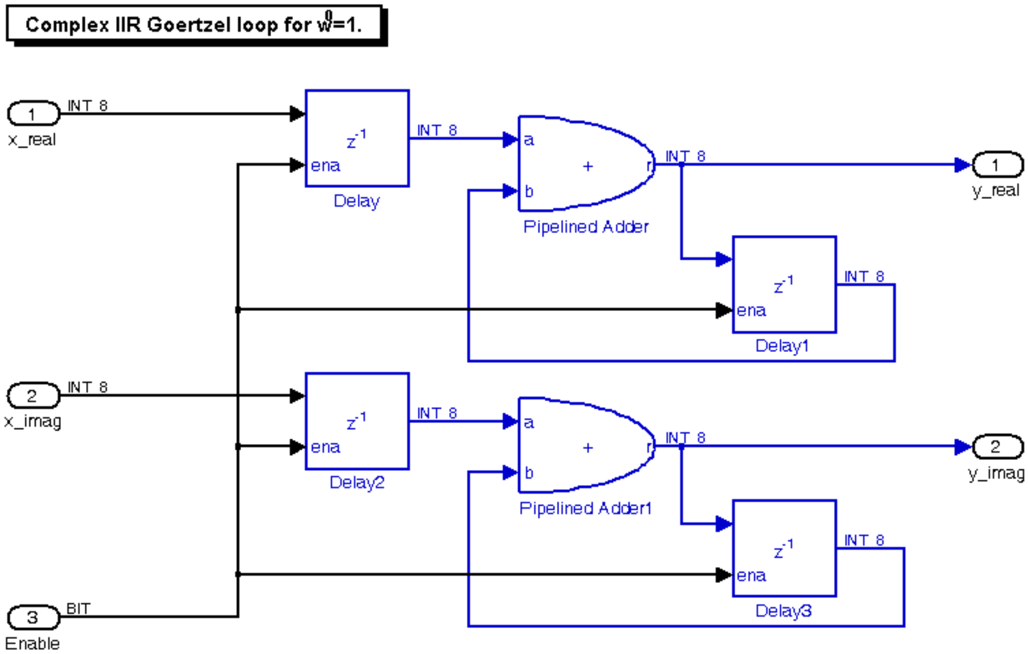
1. On the desktop, double click on the **Engineering Folder**.
2. Double click on the MatLab icon to start MatLab. It will take a few seconds to load.
3. From the top icon list in the **MatLab** window click on the **Simulink** icon to start **Simulink.**
4. Create a New Folder on your mapped network drive and name it **DSPwFPGAs.** Use this folder to save your designs. Never save your files to the local drive, use your network drive or a USB drive instead.

B. Design the W^0 and W^1 Butterflies

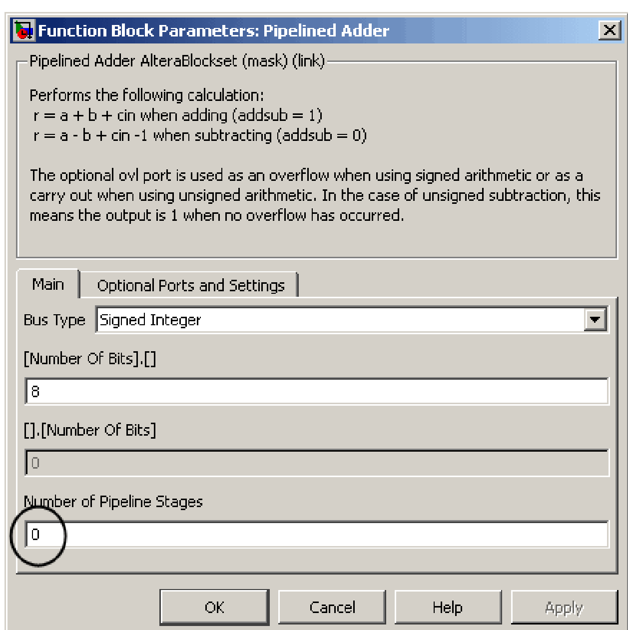
1. Download the file `DFTgoertzel.mdl` from the class webpage and place it in your **DSPwFPGAs** folder.
2. Click on the **Current Directory** selection icon and select as current directory your **DSPwFPGAs** folder.
3. The files in the **DSPwFPGAs** can now be easily accessed with the "open file" button on the **MatLab** toolbar**.** Double click on the `DFTgoertzel.mdl` file. After a moment, you should see the incomplete design shown in the figure below. Note that only the blocks for $\mathbf{w^0}$ and $\mathbf{w^1}$ are present. You will have to **complete** these two blocks, and **create** and **connect** the blocks corresponding to $\mathbf{w^2}$ and $\mathbf{w^3}$.



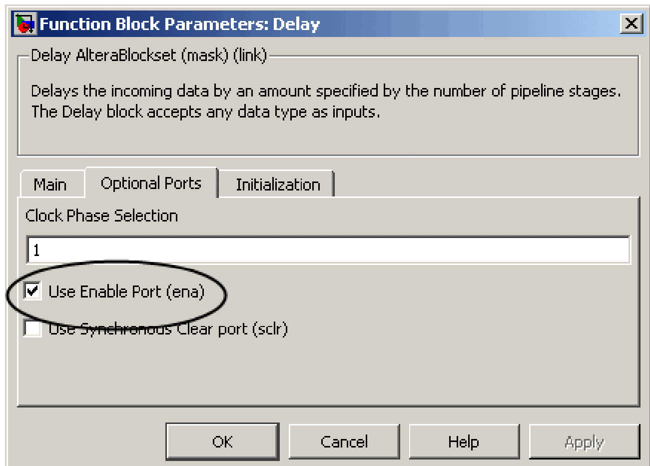
4. Double click on the block corresponding to w^0 and edit it to match the figure shown below.



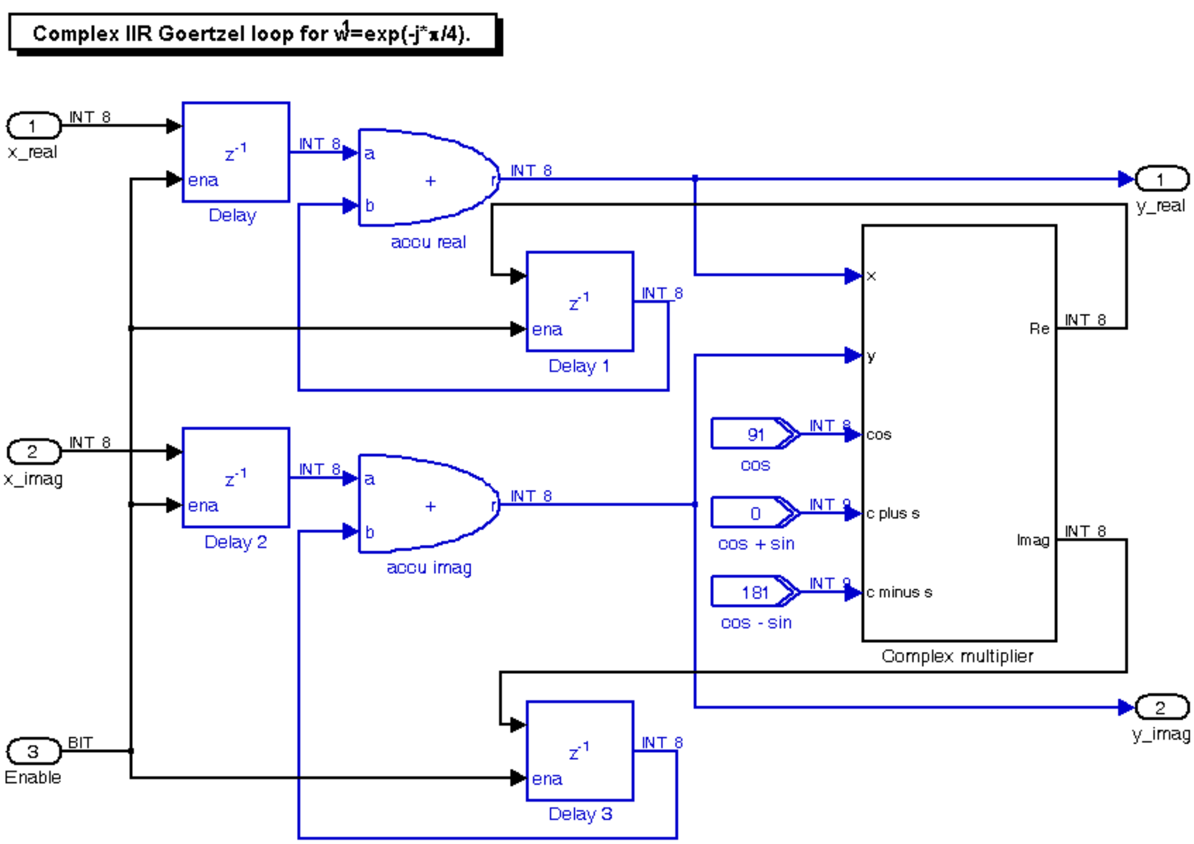
5. You will have to edit the properties of the **Pipelined Adder** and **Delay** blocks to match the design shown. For the case of the **Pipelined Adder**, set the **Bus Type** to **Signed Integer,** the **[Number of bits].[]** to 8, and the **Number of Pipeline Stages** to 0. Note that you could have achieved the same by using the **Parallel Adder Subtractor** block followed by the **AltBus** quantization to 8 bits. Often, there is more than one way to complete a design.



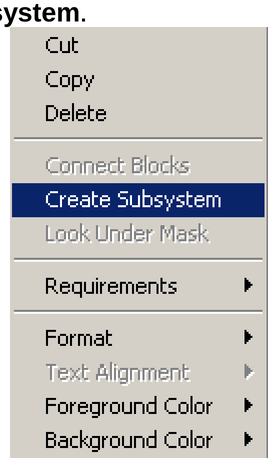
6. To complete the connection of the **Delay** block, you will have to change its **Optional Ports** setting and check the option: **Use Enable Port (ena)**. Note that now the block has a second input called **ena**. Leave the other options of the **Delay** block in the default setting.



7. Modify the block corresponding to $\mathbf{w}^1$ to match the figure shown below. Use the data for cos, cos+sin and cos-sin computed in the Prelab. Hint: you may use the 3*/5+ complex multiplier you designed in lab 4 to complete the block, as detailed by step **8**. <u>Make sure your 3*/5+ design from lab 4 is without pipeline stage in the multipliers</u>!



8. To use the complex multiplier you created in lab 4**,** you will need to first select your design, with the exception of **Input** and **Output** blocks and non Altera blocks. Then, right click on the selection and choose **Create Subsystem**.



9. Simulink will create a block containing the complex multiplier. Copy this block to your $\mathbf{w}^1$ design and wire it according to the figure previously shown. Remember that you can resize the block that you created. Also, you can rename your block by clicking over its current name textbox and typing “Complex multiplier.”

10. If you wish to, you can change the name of the block’s inputs and outputs by double clicking on the block and editing the individual inputs and outputs’ names.

11. Simulate your design and compare the results shown by the scope for $\mathbf{w}^0$ and $\mathbf{w}^1$ with the data you obtained for the Prelab.

C. Completion of the 8-point DFT

Follow the steps below to complete the other two subsystems, $\mathbf{w}^2$ and $\mathbf{w}^3$.

1. Create a backup of your working $\mathbf{w}^0$ and $\mathbf{w}^1$ Simulink model.

2. Based on the two previous sub systems and the data you obtained for the Prelab, add the necessary Altera blocks and wire them to obtain $\mathbf{w}^2$. Note that you do not need a complex multiplier for $\mathbf{w}^2$,i.e., modify a copy of the $\mathbf{w}^0$ block.

3. Add the **Output** and **Mux** corresponding to $\mathbf{w}^2$. Name the outputs **Output y2_Real** and **Output y2_Imag,** and wire them to the multiplexer. Wire the multiplexer to the **scope.** Note that you may simply copy the **Output** and **Mux** from the ones corresponding to $\mathbf{w}^0$ or $\mathbf{w}^1$, rather than add each block from the Simulink library.

4. Wire $\mathbf{w}^2$ by connecting the inputs of the block to the **Input Enable**, **Input x_re** and **Input x_im Input** blocks, and the outputs of the block to the **Output y2_Real** and **Output y2_Imag**.

5. Repeat the same steps for the $\mathbf{w}^3$ sub system, by copying the $\mathbf{w}^1$ sub system, modifying the complex multiplier values, and connecting the input and outputs of the block.

6. Run a simulation of the completed design. Verify that the design produces the correct real and imaginary parts using the **Scope** and the data you obtained for the Prelab.

7. Compile the design using **Signal Compiler** and determine

Lastname: ___________________ **LABORATORY DFT**

5Digit SS:___________________

Logic Cells = _________________

DSP 9x9 =__________________

DSP 18x18 =__________________

Slack = __________________

from the report files or the **Resource Usage** block.

F. Deliverables:

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

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