5Digit SS:_____

LABORATORY Intellectual Property



LAB 8: Design with Intellectual Property Cores

(10 points)

In this lab, you will be introduced to IP design for FIR filters and FFTs. In many projects time-to-market constrains require the use of predefined block, so called IP block. IPs may be available as HDL source (soft core), parameterized core (firm core), or hard cores (fixed netlist core). FPGA vendor like Altera or Xilinx provide some basic DSP blocks like, IEEE floating-point operations, NCO, FIR, FFT within a basic subscription of their software tool. More complex blocks such as PCI core, DCT, or MPEG and JPEG coder cost extra.

In the **pre-lab**, you will use "pencil-and-paper" to compute the results you expect later in your design implementation. In the **design part**, you will implement a F5 half-band filter using PDA and a 256 point FFT. This lab does not use the DE2 boards.

Lab Objectives

After completing this lab you should be able to

- Configure, generate and simulate a FIR filter use a FIR Compiler core generator
- Configure, generate and simulate a 256 point FFT use a FFT MegaCore Function
- Apply filtering to different noise types

Pre-lab (3 points)

1. A key point when using an IP core is to understand the functionality of the I/O ports and possible design choices. This data can be found in the "User Guide" of the core and are posted on the class webpage. Using the user guide give a brief description of the I/O ports. For control signal specify if input/output, active low/high and synchronous/asynchronous.

Signal	Direction	Brief Description
clk	Input	Positive edge trigger clock for the registers
reset_n		
ast_sink_data		
ast_sink_valid		
ast_source_ready		
ast_sink_error		
ast_source_data		
ast_sink_ready		
ast_source_valid		
ast_source_error		

1

5Digit SS:_____



2. The FFT core used a block-floating-point format for the data, i.e., all integers are assigned ONE power-of-two exponent 2^{-b}. For instance 720 with exponent -3 is represented as: 720=90/2⁻³ For the exponent -3 complete the following table

Integer	BFP representation
	$100 / 2^{-3}$
80	
	$1/2^{-3}$
736	

3. For the 256 point FFT testbench the following input data are used: x(1) = 20; x(2) = 40; x(3) = 60; x(4) = 80, x(5) = 100; x(6) = 120; x(7) = 140; x(8) = 160, and x(9...256) = 0 for all others. Using MatLab determine the first and last two FFT values of **X**=fft(**x**) rounded to the nearest integer:

X(1)	X(2)	X(255)	X(256)

4. From the FFT "User Guide" manual of the core give a brief description of the I/O ports. For control signal specify if input/output, active low/high and synchronous/asynchronous.

Signal	Directio	Brief Description
	n	
clk	Input	Positive edge trigger clock for the registers
reset_n		
inverse		
sink_valid		
sink_sop		
sink_eop		
sink_real		
sink_imag		
sink_error		
source_ready		
sink_ready		
source_error		
source_sop		
source_eop		
source_valid		
source_exp		
source_real		
source_imag		

2

5Digit SS:_____

LABORATORY Intellectual Property



VHDL Design-lab

Follow the directions below to implement a parallel DA filter design for the F5 filter and a 256 point FFT.

A. Getting Started

If you are in B114 or the digital logic lab:

- 1. On the desktop double click on **Engineering folder** and start version 9.1 of **Quartus II**. <u>Do not</u> use Quartus version 10 since it does not has a build-in simulator.
- 2. You should not save anything on the local hard disk. You will have to use a USB flash drive, or your "mapped" home directory to save the files. Create (if not done in a previous lab) a new folder named **DSPwFPGAs** on your mapped network drive.

B. Configure FIR Compiler for F5

- 1. Within your **DSPwFPGAs** folder make a new folder called **DE2_lab8**. Download the file "f5.dat" from the class webpage and put the file in your **DE2_lab8** folder. Create a project f5 and as device select EP2C35F672C6.
- Start the from Tools->MegaWizard Plug-In Manager. From the Installed Plug-Ins select DSP->Filter->FIR Compiler v9.1 and use £5 as output file name. The FIR Compiler v9.1 IP Toolbench should open.
- 3. Select **Step: Parameterize** and the Window similar to Fig. 1 below opens. Click on **Edit Coefficient Set** and a window called **Coefficients Generator Dialog** opens. Then select **Imported Coefficient Set** and Browse to the **File** you downloaded from the course webpage: f5.dat then click Ok.
- 4. Back in the Parameterize Window click on Time Response & Coefficient Values tab and you should see the length-11 impulse response as shown in the Figure 1 below. Use Input Bit Width 16 and the output Full Resolution should be 26. For Coefficients Scaling select None. The Device should be set already to Cyclone II if you have set up the project correctly.
- The FIR Compiler let you choose between different architecture and resource types and provides approximations on the used LEs and multipliers. For Data and Coefficient Storage select Logic Cells. Complete the following table by selecting the Structure and Multiplier Implementation if applicable.

Lastname:	

5Digit SS:___

LABORATORY Intellectual Property



Structure	Multiplier Implementation	Logic Cells	Multipliers
Fully Parallel Filter	N/A		
Variable/Fixed Coefficient:	Logic Cells		
Variable/Fixed Coefficient:	DSP blocks		

9 Parameterize - Fi	IR Compiler					>
Coefficients Specifi	cation - (Low Pass Set [1])				Rate Speci	ification
New Coefficient Set Edit Coefficient Set Remove Coefficient Set					Single Rat	te 💌 Factor 2 💌
Low Pass Set [1]						il clock enable nin
Plot Option Fixed	/Floating Coefficients 💌	🗖 Dark Background			-Input Spec	ification
					Number of	Input Channels 1
256.0				256 ^{5 Col}	-Innut Nun	nber System
					Signed B	inany
					loigheat	
0	·			11	Input Bit W	/idth 16
-25.0				-25		<u> </u>
Coefficients	Original Value	Scaled Value	Eived Point V		-Output Spe	ecification
1	3.0	3.0	3		Full Resolu	ition Bit Width is 26
2	0.0	0.0	0		Based on N	Aethod Actual Coefficients 💌
3	-25.0	-25.0	-25		-Output Ni	mbor System
4	0.0	0.0	0		Output Ni	imper system
5	150.0	150.0	150		Eull Reso	Jution
6	256.0	256.0	256		I un reso	
	150.0	150.0	150			
8	-25.0	-25.0	-25			
10	-20.0	-23.0	-25			
Frequency Resp	onse Time Response & (Coefficient Values				
p						
0						
Coefficients Scaling	g INone					
Architecture Snecifi	cation					
Device Family		Earce Non-Symm	notric Structure	Resource	Litilization	Throughput (Fully Streamin
Device Farmy		I Torce Non-Oynm	ienic onuclare	Logic Cells	600	
\frown				M512	000	- An input data is
Structure	Distributed Arithmetic : I	-ully Serial Filter		M4K	0	processed
\smile				M-RAM	0	every 17 clock periods.
Pipeline Level	1 💌			M9K	0	A now output data is
				M144K	0	denerated
Data Storage	Logic Cells 💌 Mu	Itiplier Implementation	ogic Cells 🛛 🔽	MLAB	0	every 17 clock periods.
	,	1		Multipliers	U	
Coefficient Storage	Logic Cells 💌 🗖 🕻	Coefficients Reload 🔲 Us	e Single Clock			
D Woming: Cooffici	ionto rolond in anoblad an	uuban aaaffisiant ataraga	io ootto o blook r			=
						Cancel Finish

Figure 1: f5.dat FIR compiler parameters.

5Digit SS:_____

LABORATORY Intellectual Property



C. Generate F5 PDA VHDL, Compile and Simulate

After the FIR compiler is configured we now generate the netlist, compile and simulate the filter.

- 1. Select Structure->Distributed Arithmetic: Fully Parallel Filer and press then Finish.
- 2. Now click on **Step 3: Generate** of the **IP Toolbench.** A window open and shows a brief description of the generated files and the I/O ports of the FIR core.
- 3. Perform a Functional Simulation of the design f5.vhd with the impulse response and try to match with the f5 simulation shown in Figure 2. The following input port needs to be specified: clk, reset_n, ast_sink_data, ast_sink_valid, ast_source_ready, ast_sink_error. Remember that reset_n is active low and "sink" are input signals and "source" are output signals! All filter coefficients should be seen in the output signal ast_source_data. Add your e-signature and print this simulation in landscape format.

f 5.vwf												
	Master T	ime Bar:	Pointer:	12.64 (ns In	terval:		Start:		End:		
∖⊧A ⊛€€		Name	0 ps 80.0 ns	160 _, 0 ns	240 _, 0 ns	320 _, 0 ns	400 _, 0 ns	480 _, 0 ns	560 _, 0 ns	640 _, 0 ns	720 _, 0 ns	800.0 ns
		clk reset n										
#	₽ 2	⊞ ast_sink_data		х				0				
	■>19 ≥20	ast_sink_valid ast_source_readu										
Ů.Å z VG	21	■ ast_sink_error	<u></u>				0					
XE XE	۩) 24 ©) 51	■ ast_source_data ast sink ready	K		0			<u> </u>	<u>) X-25 X 0 X15</u>	0 <u>X256X150X_0</u>	<u>X-25X 0 X 3</u>	
	 52 	ast_source_valid										
	™ 253	I ast_source_error	ſ				<u> </u>]
					F	5 core simula	tion by Dr. Uwe	Meyer-Baese				
<u>₽</u> ₽; Ž↓	•	•	ļ									

Figure 2: f5.vhd PDA core simulation.

4. Now run a full compilation. From the **Compilation Report** and the **Classic Timing Analyzer Tool (Processing** menu) for clk determine

Total logic elements	=	-
Embedded Multiplier 9-bit elements	=	
Total memory bits	=	
Registered Performance	=	MHz

Enter these resource and performance data also in your f5.vhd header file.

5. Compare this design with the estimation given by the Parameterize ToolBench under B.5 and compute the error

5Digit SS:_____

LABORATORY	
Intellectual	
Property	



Total logic elements estimation error = ____(LEs)

Estimation error in Percent = ____%

D. Configure FFT MegaCore Function for length 256 FFT

In the second part of the lab we now will use the FFT core generator to build a length-256 FFT.

- 1. Within your folder called DE2_lab8 create a project fft and as device select EP2C35F672C6.
- Start the Tools->MegaWizard Plug-In Manager. From the Installed Plug-Ins select DSP->Transfroms->FFT v9.1 and use fft as output VHDL file name. The FFT Megacore Toolbench should open.
- 3. Select Step 1: Parameterize and a window opens. Click on the Architecture tab and select I/O Data Flow Streaming. Then click on the Implementation Options tab and set Implement Multipliers in: DSP Blocks/Logic Cells. With Structure you can choose how the complex multiplier is implemented: 4 Mults/2 Adders or with 3 Mults/5 Adders. Select the default 4 Mults/2Adders configuration. Now click on the Parameters tab and set FFT parameter: Transform Length=256; Data Precision=16; Twiddle Precision=16. Complete the data for the resource estimation:

LEs	=
M4K RAM Blocks	=
DSP Block 9-bit elements	=
Transform Calculation Cycles	=
Block Throughput Cycles	=

4. The Cylcone C35 device on the DE2 has limited resources, i.e., 35K LEs;105 M4Ks, and 70 DSP 9-bit blocks. Determine with the **Resource Estimation** what the maximum FFT transform length is such that it can be build with C35 resources. Determine also which resource the limiting factor is by modifying the **Transform Length**.

Maximum length FFT with C35 device	e:		_
Resource limit (cycle one) :	LEs	M4Ks	DSP block 9-bit

E. Generate Length 256 FFT VHDL, Compile and Simulate

In the pre-lab we became familiar with the block-floating point format and we developed testbench data for the length-256 FFT we now will use.

1. Select again **Transform Length = 256 points** and 16 bit precision, 4 Mults /2Adders and **Streaming Architecture** in case you had changed it then press **Finish**.

Lastname: ______
5Digit SS:_____



2. Next click on **Step 3: Generate** of the **IP Toolbench.** A window opens and shows a brief description of the generated files and the I/O ports of the FFT MegaCore.

LABORATORY Intellectual

Property

3. Perform a Functional Simulation of the design fft.vhd with the testbench data from the prelab, i.e., x=20:20:160 as in Figure 3. The following input ports need to be specified: clk,reset_n, inverse, sink_valid, sink_sop, sink_eop, sink_real, sink_imag, sink_error, source_ready. Rember that reset_n is active low and sink_sop is needed to be high one clock cycle when the sequence starts and sink_eop is high one clock cycle when the sequence ends. Make sure the input sequence is exact 256 cycles long. Use a clock = 10ns for easy counting clock cycles. The output data start at 5.3 μs and the end is at 7.8 μs.

Tift.vwf						fft.vwf				
Master Time Bar: 40.0 ns + Pointer: 56.85 ns Interval: 16.85 ns Start: End:			Master Time 25.0 ns Point 2.57 us Interv 2.55 us Sta							
A ₩€		Name	5.0 ns 25.0 ns 40. 1	45.0,ns 65.0,ns 85.0,ns 105,0. 0,ns ,↓	ns 125 _i 0ns	⊳ €		Name	2,52 us 2.56 us 2.6 us 2.6 us ,	2.64 us
	■>0	clk .	لمست				▶0	clk		vvv
纳 公。	2	reset_n	'			#		reset_n		
	<u></u> →3	sink_valid				VTT .×	3	sink_valid		
0 1	a 4	sink_sop				_0_1	× • 4	sink_sop		
z Ver	⊡> 5	sink_eop					5	sink_eop		
	→ 6 → 100	Isink_real		20 <u>X 40 X 60 X 80 X100 X120 X140</u>	<u> x160 x 0</u>		6	Image: Sink_real	0	
	■ 23 - 33 40	⊞ sink_imag	└────	0		AL AS	23	I sink_imag		
	40	source readu	<u> </u>	0		と数	40	t sink_error		
XCX	44	sink readv					43	sink ready		
<u>X?</u> X <u>B</u>	@ 45	source_error		0		<u>X?</u> X!	45	I source error	0	
₽₽; ⊉↓	48	source_sop				88. 2.	48	source_sop		
	49	source_eop					- 💿 49	source_eop		
	50	source_valid					50	source_valid		
		Isource_exp	<u> </u>	0				source_exp	0	
	• 08	source_real	└────	0			258	■ source_real		
	• •••• •	Source_imag	<u> </u>	0			OP 75	Image sourceimage		
							•			Þ
(a)							(k)		

Figure 3: FFT input data specification (a) Start Of (data) Package (b) End Of (data) Package.

4. Using a 10 ns clock period the output data are available at 5.3 μs (see active source_sop) and the end is at 7.8 μs (see active source_eop). Verify this data with your test data from the prelab. Exponent of the data should be -3. Add your e-signature and print all 4 simulations in landscape format.

Lastname: L 5Digit SS:	ABORATORY Intellectual Property		WITH FPGAs			
Image: serie seri	Start End 5.445 us Image: Constraint of the start of	ter Time Bar: 40.0 ns Pointe Name Clk reset_n clk reset_n sink_valid sink_cop sink_cop sink_cop sink_cop sink_eror sink_eror source_ready sink_ready f source_ready source_eop source_eop source_eop source_exp source_real s	t. 7.91 us Intervat 7.87 us Start: End: 7.825 us 7.865 us 7.905 us 7.945 us 7.825 us 7.965 us 7.945 us 7.905			
(a)		(b)				

Figure 4: FFT output data specification (a) Start Of (data) Package (b) End Of (data) Package.

5. Now run a full compilation. From the **Compilation Report** and the **Classic Timing Analyzer Tool (Processing** menu) for clk determine

Total logic elements	=	 -
Embedded Multiplier 9-bit elements	=.	
Total memory bits	=	
Registered Performance	=	 MHz

6. Compare this design with the estimation given by the Parameterize ToolBench under D.3 and compute the error

Total logic elements estimation error = ____(LEs)

Logic elements Estimation error in Percent = ____%

F. Deliverables:

- 1. Solve the problems of the pre-lab. (3 points).
- 2. Complete this report, print the Quartus simulations for f5 (see Fig. 2) and 4 simulations for the length-256 FFTs (see Fig. 3 and 4) including e-signature. (7 points).

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