ASIC/FPGA Design Spring 2025 Assignment 3

Professor Shabany

Problem 1

Introduction to IP Core

(a) Explain the concept of an IP Core and its role in digital system design.

(b) State the differences between Hard IP Core, Soft IP Core, and Firm IP Core, and provide an example for each.

(c) Discuss the advantages of using IP Cores in various design stages, such as design time, verification, and implementation.

(d) Suppose in your chip design project you want to use a Memory Controller for RAM. Which type of IP Core (Hard or Soft or Firm) would you choose, and why? Discuss considerations such as performance, flexibility, and design requirements.

Problem 2

Direct Digital Synthesizer (DDS)

(a) Using the Xilinx DDS Compiler IP Core, implement a simple DDS that generates a 1 kHz sine wave. The LUT inside the IP Core should be used for initialization. Sampling frequency is 100 kHz.

(b) Suppose you want to change the output frequency by modifying the Frequency Control Word (FCW) in your code. How can this be done?

(c) Explain how a Phase Accumulator can be used in DDS design for generating different frequencies.

(d) Verify the output using a testbench and present the results.

Problem 3

FFT Implementation

1. Implement a 16-point FFT unit using Radix-2 architecture. Explain the processing structure and data flow.
2. Design the system such that input data are complex numbers stored in Fixed-Point format, and the output is the frequency spectrum.
3. You are given two files (I and Q) that contain the real and imaginary parts of the input signal. Store them as complex inputs of the form $I + jQ$, apply FFT, and then save the output to a file. Finally, plot the magnitude spectrum using:

   ```
   plot(abs(FFTI + 1i * FFTQ))
   ```

   Compare the spectrum to the original signal.

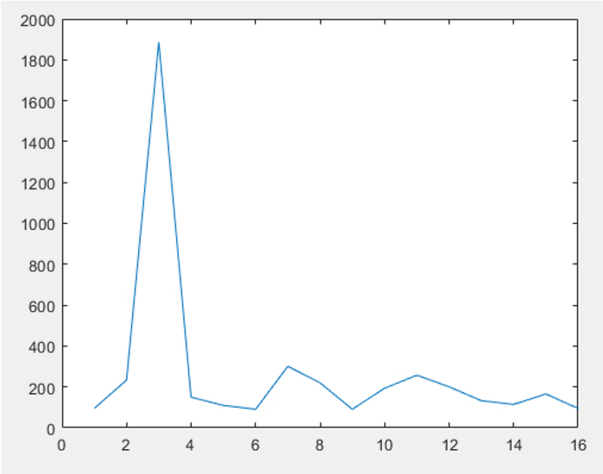

Figure 1: original spectrum

Problem 4

FFT using Xilinx IP Core

(a) By referring to the Xilinx FFT IP Core datasheet, explain how to configure it, its input/output signals, and connection.

(b) Use the output of your DDS module (from Problem 2) as the input to the FFT IP Core. Implement verification, store the data in Fixed-Point format, and check the correctness using a testbench.

Problem 5

Comparison: Manual vs IP Core FFT

(a) Compare the output of the manual FFT (Problem 3) with the FFT IP Core (Problem 4) in terms of accuracy, latency, and FPGA resource usage (e.g., LUTs, RAM, DSP Slices).

(b) Discuss under what conditions manual implementation is preferable over using an IP Core.

(c) What challenges exist in optimizing manual FFT implementations to approach IP Core performance?

(d) In real-world applications (e.g., audio/image processing, communication systems), would you recommend using manual FFT or IP Core FFT? Explain your reasons.