UNIVERSITY OF CALIFORNIA IRVINE

EECS 298: System-on-Chip Design (Fall 2023)

Lab 1: Introduction to High-Level Synthesis

NOTE: This is an individual lab. Please complete it by yourself.

Due: November 10, 2023

In this lab, you will use FPGA high-level synthesis (HLS) tools to generate an hardware accelerator for vector addition (Part I) and matrix multiplication (Part II), targeting the **KR 260** platform. The PL frequency target is **100 MHz**. For each design, generally, there are two steps involved:

- *Step 1*: use Vitis HLS to compile vector addition code (in C++) into an AMD Xilinx IP block;
- *Step 2*: use Vivado tool to integrate your `pl_vecadd` with other IP blocks (mainly the ZYNQ processing system, or PS), and generate the bitstream configuration file (*.bit) that can be used to program the SoC FPGA platform.

In this lab, you don't need to actually program the KR 260 platform (we will do that in the next lab). You only need to use design tools to generate bitstream configuration files. Please note that this lab is for every student, please complete it individually. We will use version 2023.1 of the Vitis HLS and Vivado. They are available on UCI remote labs and EECS instructional servers. We covered their usage in our lectures.

Submission: Submit your solutions/report to Canvas, in PDF format.

PART I: Vector Addition Example *(30 pts)*

Take the C++ code (pl_vecadd.cpp) in the provided vector addition example (pl_vecadd_example.zip), and generate the bitstream configuration file for KR 260. Report the total latency and resource utilizations from Vitis HLS synthesis summary and Vivado post-implementation summary.

PART II: Matrix Multiplication *(70 pts)*

In this Part II, you will build an accelerator for matrix multiplication (`matmul`) and optimize it in HLS. Please watch this instruction video (and slides) first. Then, create two versions of matmul accelerators: (a) a plain version without any HLS pragmas; (b) an optimized version that uses `ARRAY_PARTITION` and `PIPELINE` (or `UNROLL`) pragmas. Run Vitis HLS and Vivado for both versions, and **report** the end-to-end latency, resource utilizations from Vitis HLS and Vivado for both versions. Show your code in your submitted PDF file. Assume matrix sizes: M = K = N = 16, i.e., input matrices A, B and output matrix AB are all 16×16 squares. Assume matrix elements are all `int32` integers.

References

- Lecture recordings on usage of Vitis HLS and Vivado on remote lab and EECS instructional servers. (recording 2)
- AMD UG1399: Vitis HLS User Guide version 2023.1 (PDF)
- Parallel Programming for FPGAs, Chapter 7. [PDF]