COMP.CE.320 High-level Synthesis

Circular Buffer Exercise

This week, we implement two different FIR filters to demonstrate how coding style affects performance when arrays are mapped to memory (instead of registers).

Task 1

Create a new Catapult project for a shift register based FIR filter. The source code and testbench are given ready made for you with the exercise files. First, examine the pre-given code files. Then, run the Catapult flow to RTL stage for the code with the following options:

1. In the "Libraries" phase, use technology to Vivado Xilinx Artix-7 FPGA (part xc7a50tcsg324-1). Also, enable the Xilinx RAM models in the Compatible Libraries window at this stage.
2. Use 50 MHz clock.
3. In the "Architecture" phase, map the internal shift register memory array to Xilinx_RAMS.BLOCK_DPRAM_RBW_DUAL (dual-port RAM, read-before-write). I/O should be mapped to wires, as default for them is.

Explore and select unrolling and pipelining options that give the best possible throughput without modifying the code.

Question 1: What was the best achieved throughput? What were the pipelining and unrolling options for it?

Task 2

Implement a circular buffer based version of the FIR filter. Page 161 of the HLS Blue Book shows how to implement the circular buffer. Verify with the given test bench that your circular buffer based filter gives the same results as the shift register based filter did (you will need to modify the test bench to invoke your new function). Using the same options as in Task 1, try to obtain an architecture with the best possible throughput.

Question 2: What was the best achieved throughput with the circular buffer based shift register? What were the pipelining and unrolling options now?

Question 3: Explain as best you can, why the circular buffer based FIR filter can achieve better throughput than the shift register based version. Also explain, what is the major difference in the schedule of the two different architectures. You can draw diagrams to demonstrate the differences.

Grading:

- Grade 1: The circular buffer based shift register has been implemented and the answers to the questions make sense.
- Grade 2: As above, and the analysis is deeper and it is easy to see from the provided text and diagrams that the differences between the two implementations have been understood.

To return:

Return your source code files for the circular buffer based shift register in a zip file. The questions should be answered in a separate PDF file that can include text and figures. Put the PDF in the same zip file.