ECEN302 : Integrated Digital Electronics
Assignment 3 Submission

Daniel Eisen : 300447549

October 13, 2020

1. ***Describe three key features about FPGAs that make them suitable for hardware acceleration applications.***

   The user/developers ability to reconfigure and tune the device to the specific requirements of a task.

   The ability for that configuration to make use of large scale parallel processing to accelerate lots of independent operations.

   High IO bandwidth capacities allow to the fast input, processing and output of data.

2. ***Discuss the relative advantages and disadvantages of an FPGA solution compared to an ASIC solution.***

   FPGA's ability for reconfiguration leads to quick design-build-test cycles, the ability to do "firmware updates" after deployment and even for use in the ASIC development cycle in design and functional prototyping/testing.

   ASIC however are faster with leaner design, and less logical overhead than an FPGA as they are custom manufactured for the deployment task. Also for this this reason, ASICs often use less power than an FPGA.

   FPGA: smaller volume is cheaper as development/testing cycle cost is lower. ASIC: higher volume is cheaper, as once is mass manufacture the individual device cost is smaller for bulk orders.

3. ***Discuss the key advantage of using HLS in FPGA code development.***

   The use of HLS as a the primary language/framework for writing FPGA is to drastically increase development time for design/testing and eliminating what would normally be the FPGA engineer bottleneck from a products development.

4. ***Describe how a test bench can be developed for testing HLS C code and the resulting RTL.***

   Second main file that calls some function (the is being tested), passing it set inputs, and compares the output of the function call to a set of 'golden' outputs either in a file or an array etc.

   Note this doesn't include timing analysis, which is done later in the design process.

5. ***Describe a situation where significant processing speed gains can be made and how you go about implementing the improvement.***

   Unrolling a large sequential looping operation. Such as performing a kernel based image processing algorithm in (or close to) real-time and doing these independent operations in parallel in significantly less clock cycles.

6. ***Explain, using the terms "control, data path, scheduling and binding", how C code is converted to RTL.***

   The process of HLS is basically the construction and mapping of a state-machine from the the C code that is to to implemented on device.

   The first stage is the Control Extraction, where the C/C++ code is analysed and the internal loops are correlated to the FSM's states to define the behaviours of the hardware. After this the operations within each of those loops are at unified into the known control states to form the data path and dataflow behaviour.

   Next scheduling maps the operations in the dataflow into the clock cycles. The output/result of the scheduling I dependant on user defined constraints, i.e. clock speed etc and can result is more/less operation per clock cycle for example. Once the operations have been scheduled, binding maps these to specific IP cores and configures them in accordance to the scheduling, i.e. how many multiplications per cycle, recourse sharing etc.