Paper review of

FPGA Programming for the Masses

**A1.**

GPP (General Purpose Processor): It is a processor that can execute any algorithm by running a set of instructions. CPU is a general purpose processor. To improve performance, GPPs have on-chip caches and on-chip DMA.

ASIC (Application Specific Integrated Circuit): These are integrated circuits that are designed to run a dedicated application. A hardware description language (HDL) is used to program an ASIC. Since these chips are custom manufactured for special applications, a functional change will need designing a complete new circuit from scratch.

FPGA (Field Programmable Gate Array): These are integrated circuits that are designed to be programmed according to the user’s application. HDLs are used to program a FPGA. They contain combinational logic (Look-Up Tables) and sequential logic (FFs) blocks which are configured to function based on the application. Apart from these programmable logic blocks, there are other discrete components like Block RAMs, DSPs, processor cores and communication cores in a FPGA.

Similarities/differences:

In GPPs the computational speed is lower compared to ASIC and FPGA. The cost of building a GPP is also higher.

ASICs are quick, run at frequencies in GHz and are faster than FPGAs. Since it is dedicated to an application, unlike FPGAs a new ASIC has to be manufactured for other applications which is costly. Hardware in not wasted since every part of the ASIC is built to serve an application.

FPGAs are relatively larger than ASIC. They typically run at frequencies in MHz which is slower than ASIC. The reconfigurable logic is something that saves on cost compared to ASICs. Since only some part of FPGA will be used during an application, there is hardware that is idle.

In a FPGA, since it is possible to configure multiple logic blocks to perform multiple functions, parallelism comes into effect. If the volume of data to be processed is high, FPGA can be configured to perform the same function multiple time using multiple parallel execution. Unlike ASIC or CPU where implementing multiple parallel processes is not always possible, FPGAs can outperform even if it runs at a relatively lower frequency.

**2A.**

Yes I think the comparison is fair. Since the levels of abstraction in a HDL are not in as much detail as a high level software programming level language, the efficiency of the code decreases. Efficiency falls because of the size of the code.

**3A.**

A floating point operation can be implemented on hardware by using integer multipliers and shifters. When we have a floating point value, it is possible to reverse engineer and compute value of the multiplicand and number of right shifts needed. Those computed values can be used to perform the function. The result can be converted back to a floating point value. So yes it is possible to implement floating point multiplication and addition on hardware. A Floating-point Unit (FPU) is a co-processor which computes results of floating point operations. The hardware implemented in a FPU does arithmetic operations and bit shifts to perform floating point operations.

**4A.**

HDL-like languages:

Languages like VHDL and Verilog have been in practice. SystemVerilog