Our ISA leaves room for more instructions in almost every addressing mode. Further arithmetic operations could be added. As an example, reverse SUB could be a useful choice. In this form, the destination register would be the one being subtracted from the source register. The general width of the modules is 16 bits. A whole range of arithmetic instructions could be introduced that take one register, and perform operations taking the 2 bytes as operands. The current multiplication block functions for unsigned inputs. Implementing MLS is crucial for a well-rounded general-purpose CPU. Booth's algorithm helps in reducing the operation's latency.

Latency could be improved upon in every segment, particularly in the ALU and the decoder blocks. If that is no longer possible there is a more drastic solution. The introduction of a new state for the decoding segment of each the instruction. This would turn every instruction 1 cycle longer. However, with pipelining all in all would be a significant improvement to the CPU. To achieve this, major modifications must be done to the state machine and in the top-level design. The decoder block needs a clocked output, the instruction register must maintain the original operation for more than one cycle.

Our design methods sacrificed a smaller chip area and lower power consumption in favour of higher processing power. This is not always desirable. Control operations could be added to allow the user to switch between different operation modes. Instructions would be grouped in terms of their importance and power consumption. Some of the less necessary hardware than could be turned off while carefully maintaining the Turing-completeness of the CPU.

Currently, the device has 2 inputs, a clock and reset with no outputs. The processor can only communicate with its internal registers and the RAM. Without an input and output stream, the CPU has many limitations. It must be extended with appropriate hardware. In addition to these, new commands would be included in ISA to choose these new ports. Loaded then into an FPGA, it could take many functionalities. Eventually with the right compiler written higher-level coding can be encoded into this ISA. The CPU could function similarly to microprocessors.