

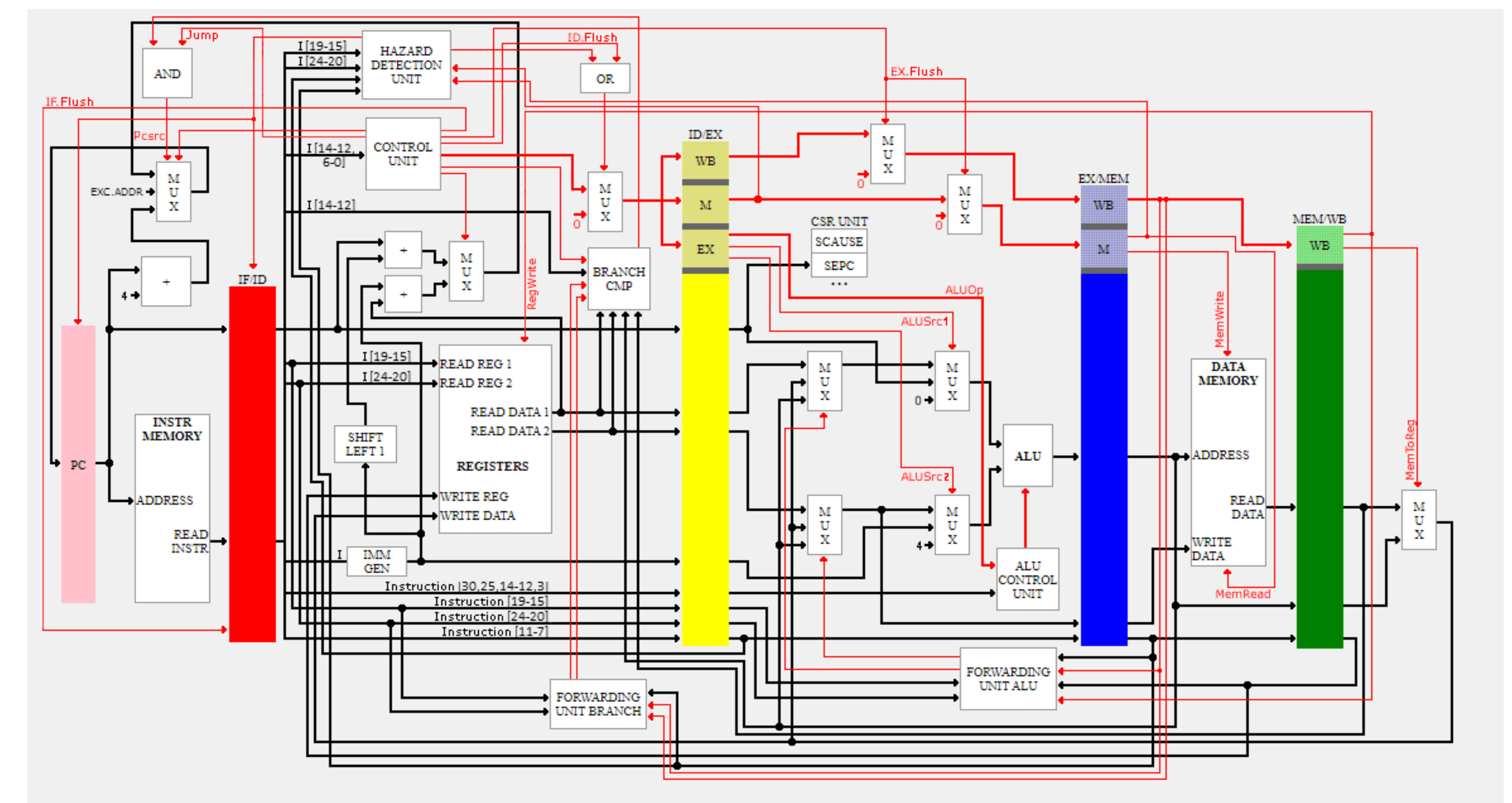


## RV32 Core (Team 11)

Andrew MacGillivray (CS) Jarrod Grothusen (CS) Aditi Darade (CS) Daniel Ginsberg (CS)

### Description & Purpose

- Implemented a core for a 32-bit RISC-V CPU
- We wanted to gain familiarity with Verilog and computer architectures / CPUs, because specialized hardware is a growing domain.
- Specialized hardware is increasingly necessary for the computational costs of things like AI and machine learning
- We implemented the core but were unable to run it on the metal.

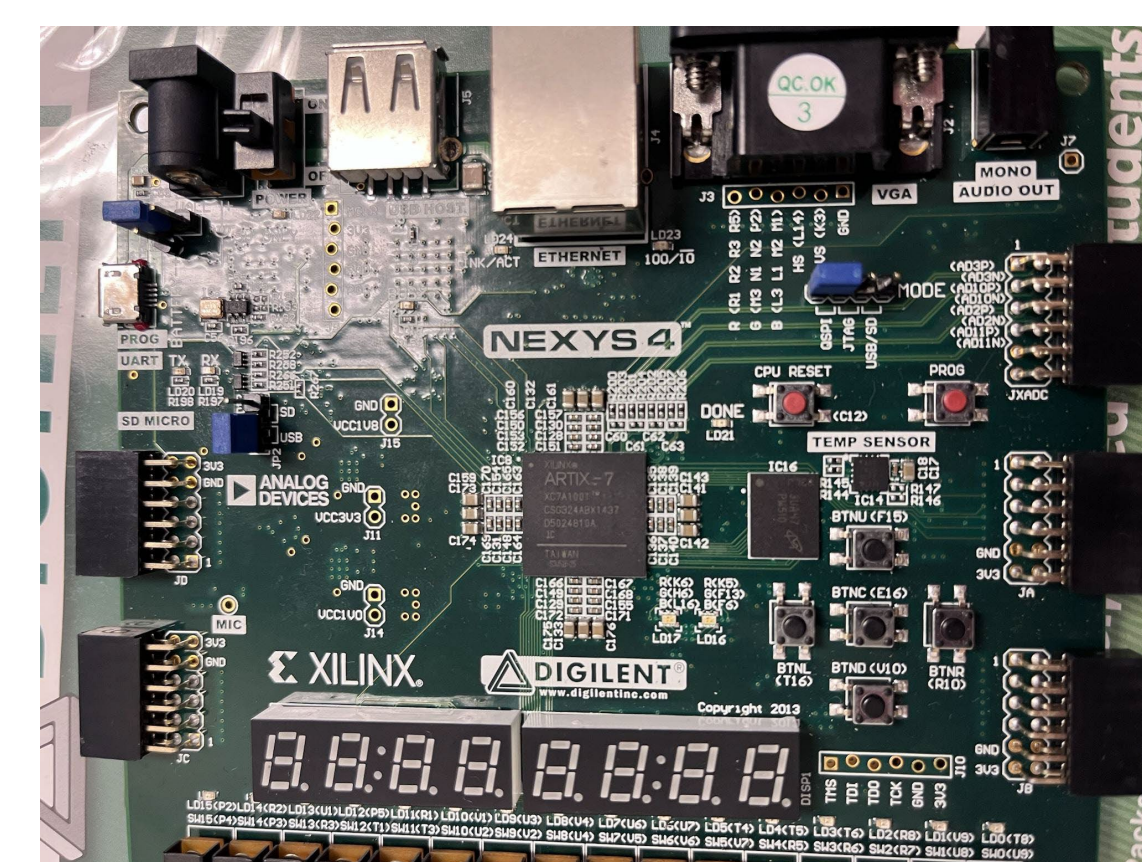
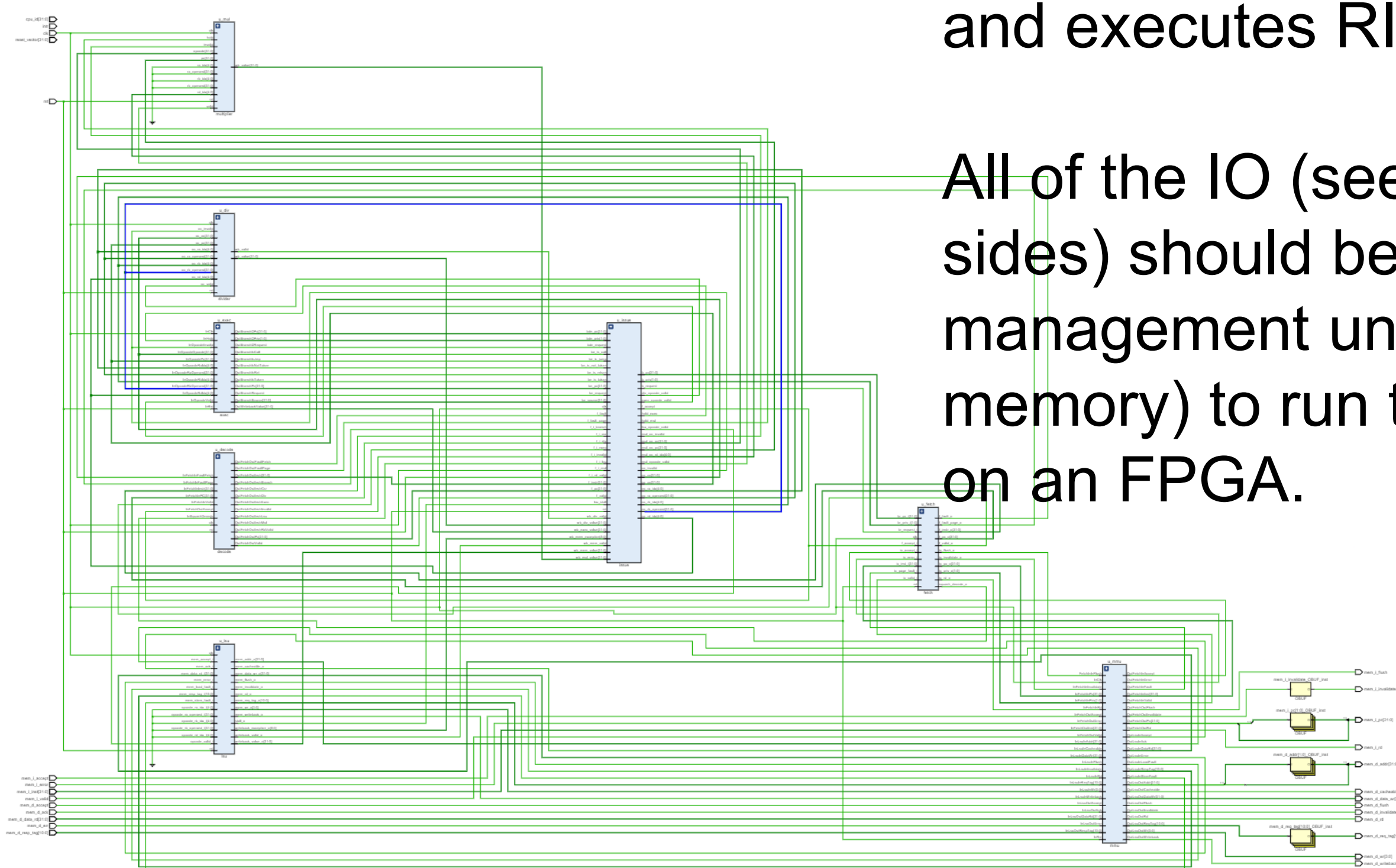


5-Stage Pipeline: illustrates how instructions are read into the system, decoded, and performed, with results written back into data memory. Our core followed this general behavior.

### Design

Our schematic, a core that decodes and executes RISC-V instructions.

All of the IO (seen on the left and right sides) should be wired to a memory management unit (with associated memory) to run the CPU on the metal on an FPGA.



### Ethical & Intellectual Property Issues

#### Ethical:

- **Security:** Open source ISAs like RISC-V create different security threat models compared to propriety ones like ARM. Our goal was to learn about computer architectures, and security was not specifically considered. This could pose a risk to users.

#### Intellectual Property:

- **Reverse Engineering:** We heavily referenced and reverse engineered open-source designs by at least one entity. We respected their licenses and credited them in our documentation.
- **Proprietary Tools:** We used the Xilinx toolset (specifically Vivado) to create our project, which may be a hindrance to those who want to reference it that do not have access.