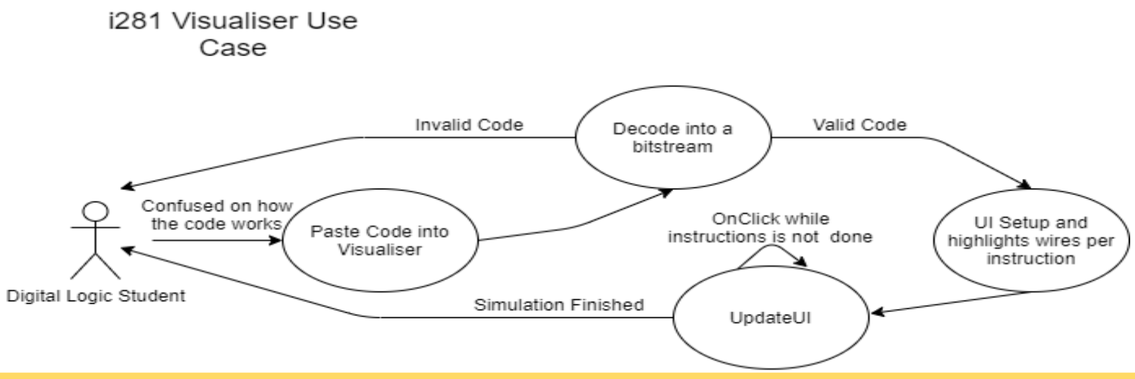
SDMAY-21-38: Visual Debugger for the i281 CPU

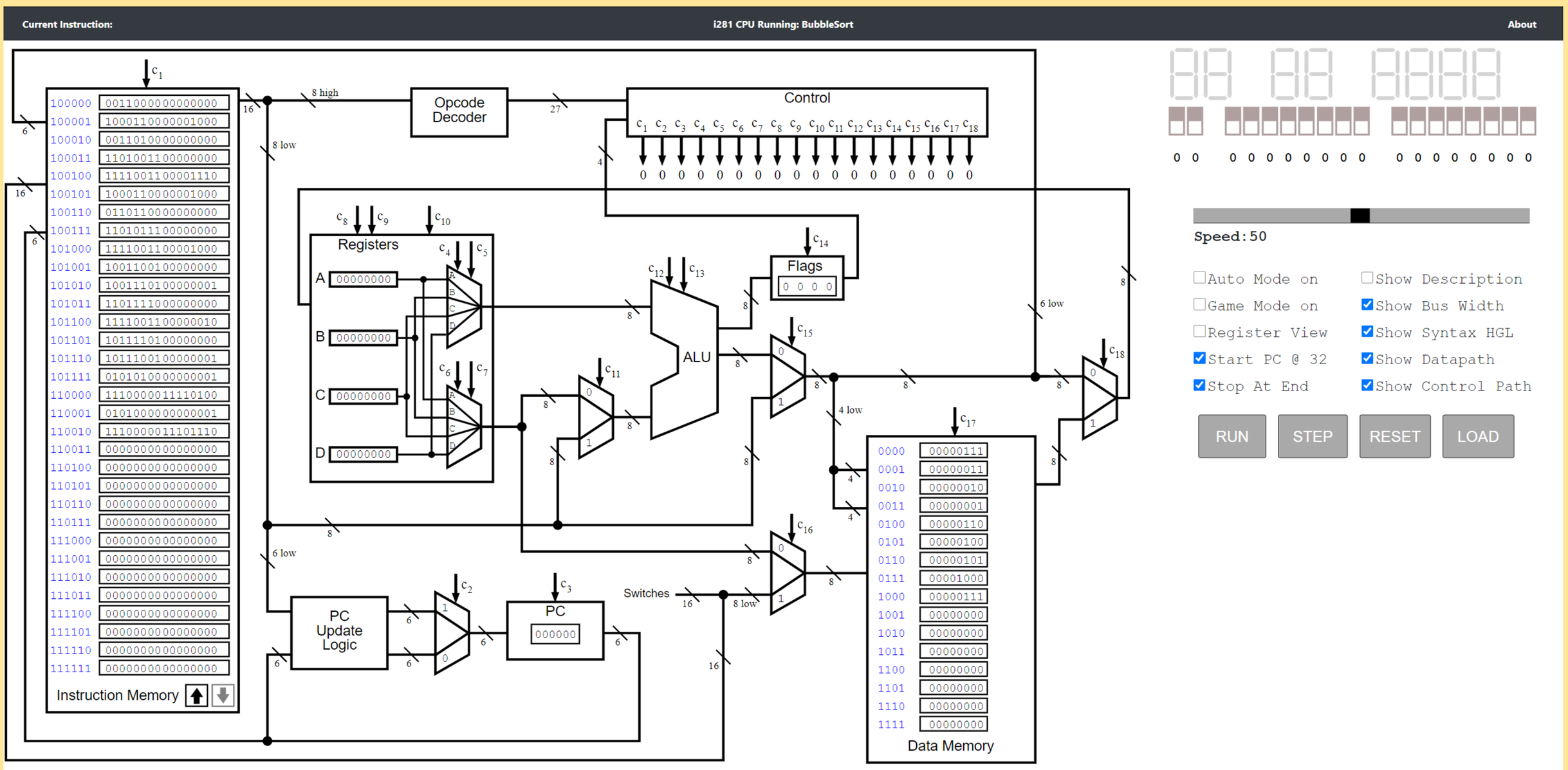
Motivation

- Aid in teaching CprE 281: Digital Logic, primarily to Sophomores.
- To showcase how assembly code is converted into binary strings.
- To show bit propagation from assembled code to different components of the i281 CPU
- Ease the transition to CprE 381, where students will be designing a MIPS Processor



Project Distribution

- Split into five sub-groups:
  - Assembler
  - Main Visualizer
  - I/O Graphical User Interface
  - JS Backend Implementation
  - Quartus BDF to Verilog Conversion
- Constraints:
  - Only JavaScript and Verilog
  - No JS Frameworks
  - All common browser support
  - No use of SystemVerilog
  - Hostable on client's class website without any additional dependencies



Design Requirements

- Functional / Non-Functional Requirements
  - Assembler must be able to parse and pass assembled code to the GUI
  - Visualizer must highlight current executing instruction
  - I/O GUI must be able to select different options when running the simulation
  - JS must be able to run fast enough to be able to visualize i281s version of Pong
  - Components in Verilog Conversion must be done at the gate level / no libraries
- Standards used:
  - ISO/IEC/IEEE 23026:2015
  - IEEE 1008-1987

Testing Strategy

- Each sub-group tested their code extensively through unit tests.
- Integration testing was done through programs given to us by the client, which works perfectly on the FPGA
- The same program was given to the Verilog team to ensure their designs had the same functionality as the i281 that was designed in BDFs.
- Iterations of the visualizer were periodically sent to the client for his own in-depth tests and design critique

Team Members: Aiman Priester, Eric Marcanio, Colby McKinley, Bryce Snell, Brady Kolosik, Jacob Betsworth
Advisor: Dr Alexander Stoytchev