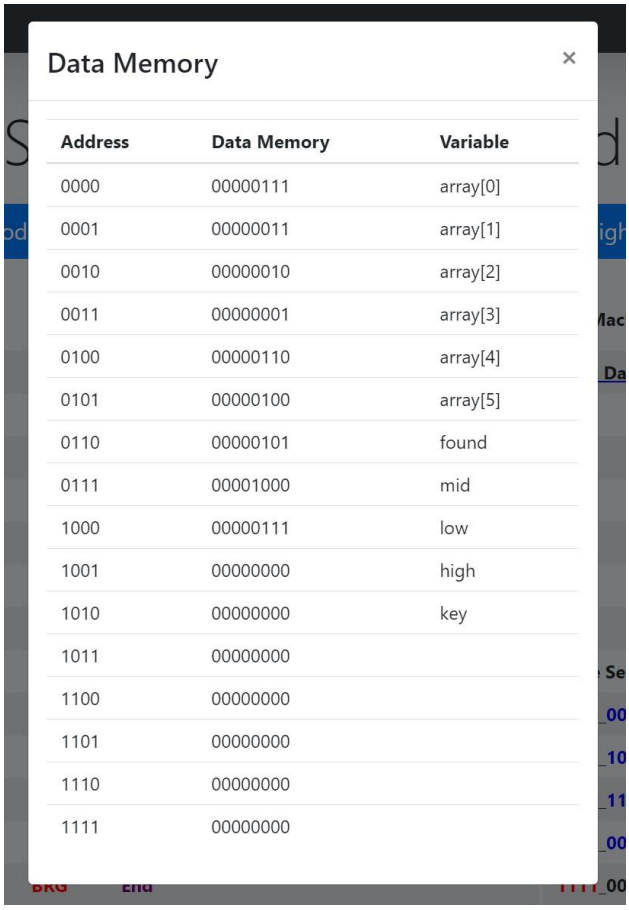
Senior Design Bi-Weekly Status Report; Fall 2020

Team Name: sdmay21-38
Team Members: Colby McKinley, Aiman Priester, Eric Marcanio, Brady Kolosik, Byrce Snell, Jacob Betsworth
Report Period: March 1st-March 15th

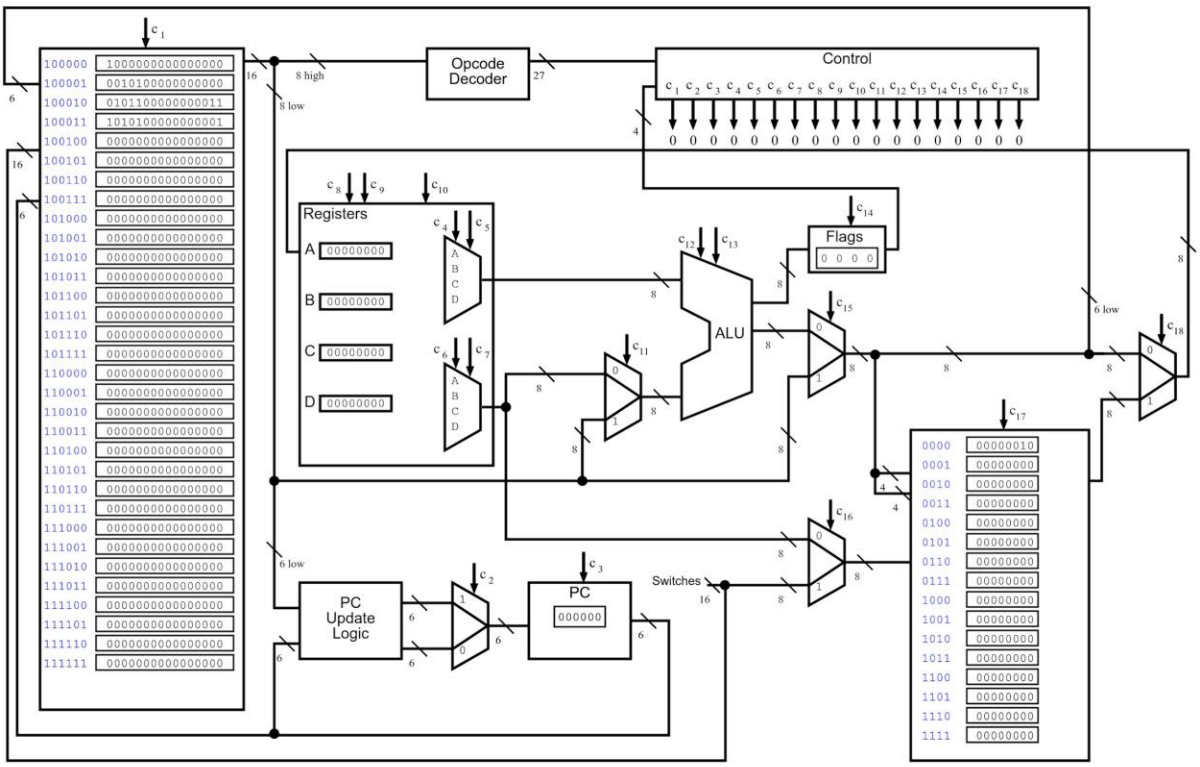
Past week accomplishments

- Eric - Assembler
  - Created a data memory table that pops up when a program is ran ○ Below is the output of the table from “Bubble sort”



- Colby - Simulator
  - Added more bus mux info
  - Made font changes in register file
  - Meet with Bryce and Brady to understand CPU

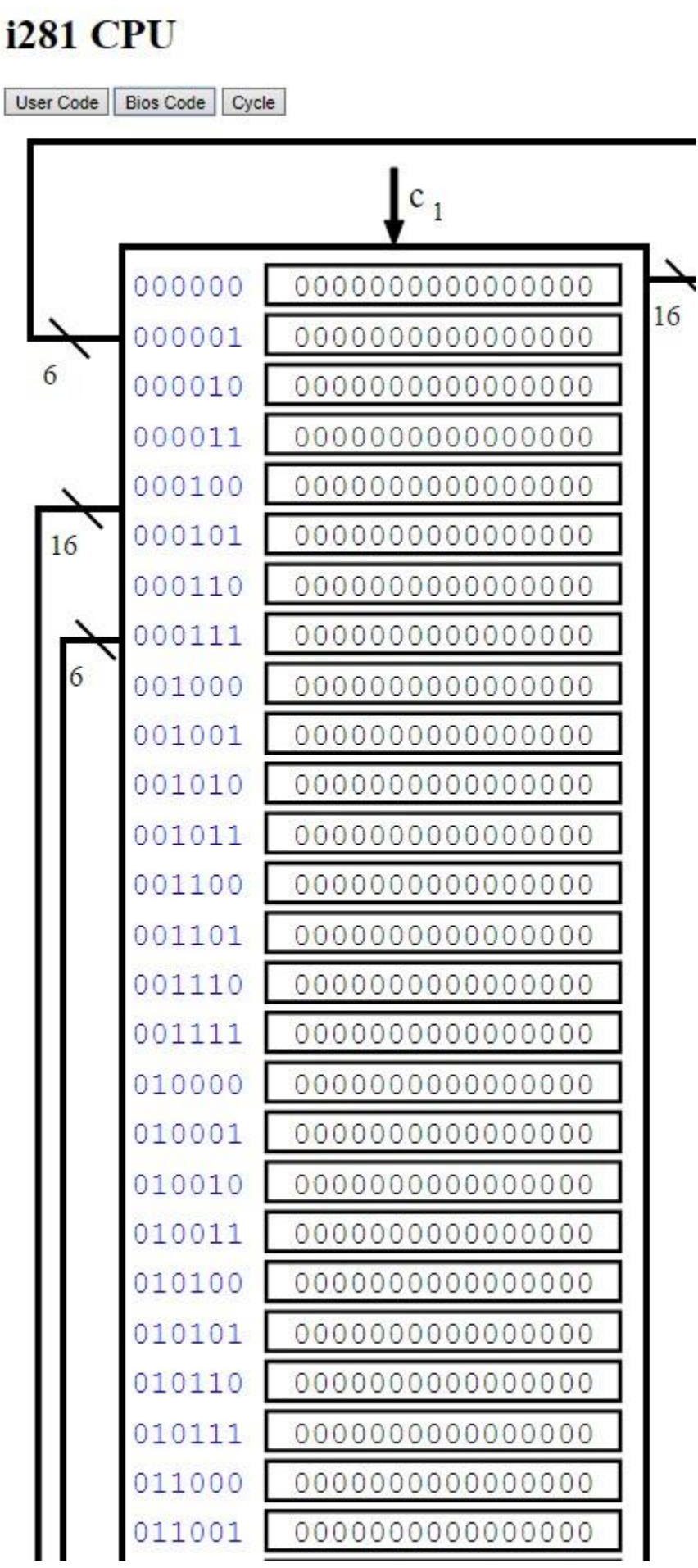
i281 CPU

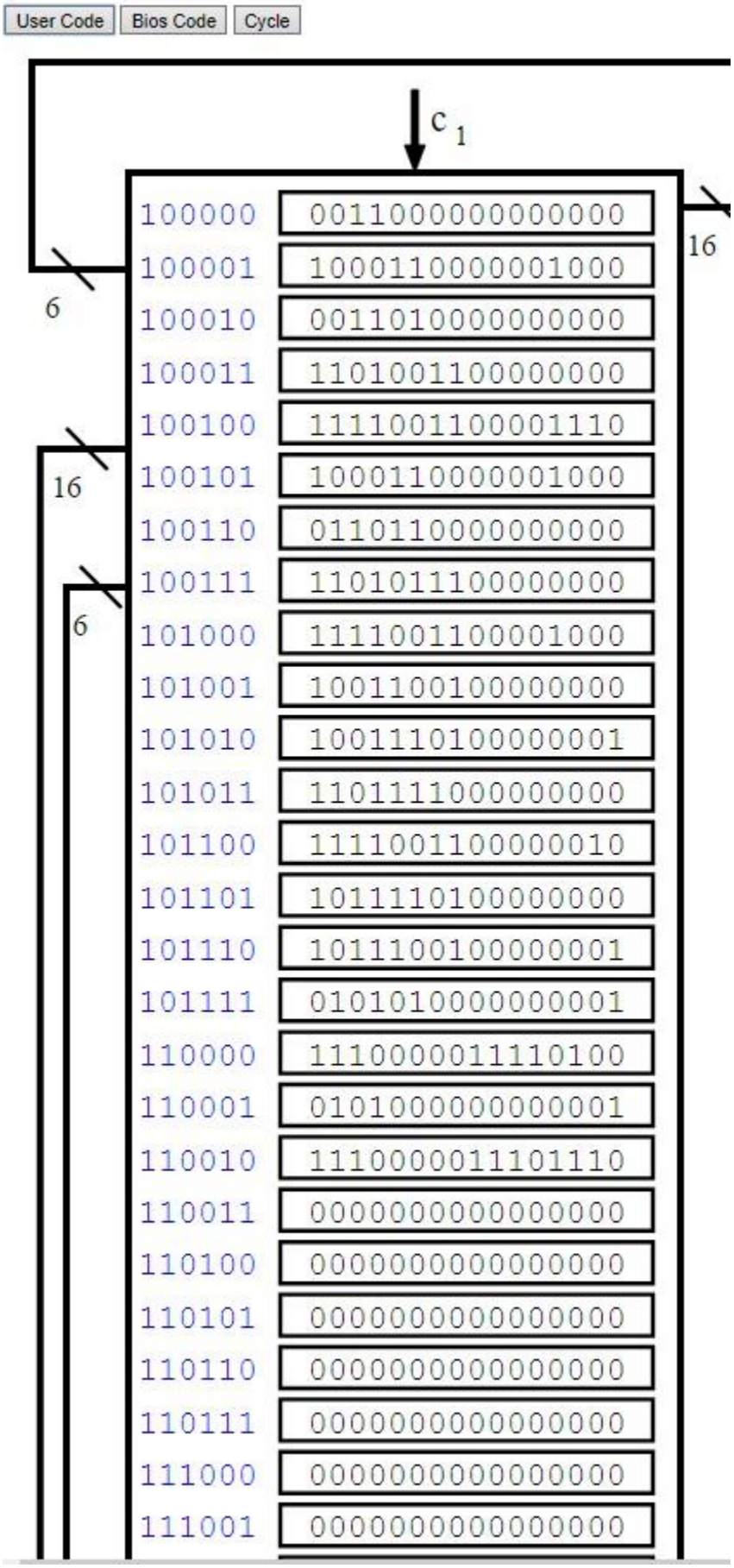


- Aiman - Verilog
  - Made final touches with regards to Verilog testing
  - Assembled and tested more sorting algorithms that was successfully ran on the FPGA
  - Communicated with the team on their questions regarding the hardware development side of the project to be incorporated into the final piece.
- Brady - Simulator
  - Various visual updates
    - Mux select lines
    - Fixing up wire segments and lining them up with the mux select lines
    - Moving intersection dots accordingly
    - Lining up lots of wires to be centered with certain components ○

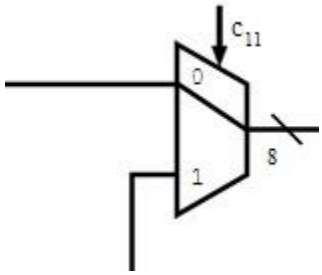
  Began interfacing assembler and CPU with the visualization.
    - Instruction memory and data memory are both populated with values from the CPU simulation, which in turn gets its data from objects stored in the browser cache provided by the assembler, above image shows instructions and data memory
    - PC value is updated by taking the new value from the simulation after every cycle ran, printed in 6 bit binary regardless of the number(pad function required as js will remove all leading 0's).

- Ability to switch below lower 32 instructions in bios, and upper 32 instructions in user space(for total accuracy with the hardware). Currently 32 lower instructions are hardcoded to noops but will be updated as it is implemented.

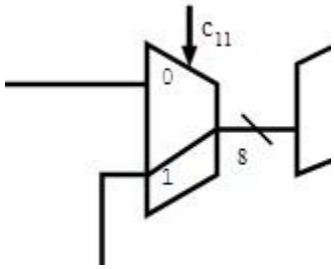

i281 CPU
User Code
Bios Code
Cycle
c 1
16
6
16
6
000000 0000000000000000
000001 0000000000000000
000010 0000000000000000
000011 0000000000000000
000100 0000000000000000
000101 0000000000000000
000110 0000000000000000
000111 0000000000000000
001000 0000000000000000
001001 0000000000000000
001010 0000000000000000
001011 0000000000000000
001100 0000000000000000
001101 0000000000000000
001110 0000000000000000
001111 0000000000000000
010000 0000000000000000
010001 0000000000000000
010010 0000000000000000
010011 0000000000000000
010100 0000000000000000
010101 0000000000000000
010110 0000000000000000
010111 0000000000000000
011000 0000000000000000
011001 0000000000000000


User Code
Bios Code
Cycle
c 1
100000 0011000000000000
100001 1000110000001000
100010 0011010000000000
100011 1101001100000000
100100 1111001100001110
100101 1000110000001000
100110 0110110000000000
100111 1101011100000000
101000 1111001100001000
101001 1001100100000000
101010 1001110100000001
101011 1101111000000000
101100 1111001100000010
101101 1011110100000000
101110 1011100100000001
101111 0101010000000001
110000 1110000011110100
110001 0101000000000001
110010 1110000011101110
110011 0000000000000000
110100 0000000000000000
110101 0000000000000000
110110 0000000000000000
110111 0000000000000000
111000 0000000000000000
111001 0000000000000000
16
6
16
6

- Fully implemented function to change which wire is selected within the mux objects(see below).



Before selecting



After selecting

```
function test(){

  cpu.singleCycle();
  mux0.select(1);
  document.getElementById("pc_val_id").innerHTML = pad((cpu.pc.currentPC).toString(2), 6);
}
```

Relevant code snippet to show what function call looks like for updating PC and selecting a mux

```
select(val) {
    this.selected = val
    var t =  document.getElementById(this.mux_id+"_true_wire");
    var f = document.getElementById(this.mux_id+"_false_wire");

    if(val==0){
    t.style.visibility = "hidden";
    f.style.visibility = "visible"
    }

    else{
        t.style.visibility = "visible";
        f.style.visibility = "hidden";
    }
}
```

Code to select an input

- Bryce - CPU Simulator
    - Created a small demo of how the simulator would interface with the assembler.
        - This demonstrated that our design for connecting the assembler and simulator was valid and could work.
    - Worked with the visualizer team to connect the assembler and simulator to the visualizer.
        - This allowed us to show the iMem and dMem from the given assembly program on the visualizer.
        - This was a major milestone of our project. This initial setup demonstrated that the entire project was coming together and would likely work.
        - My role has shifted from design and implementation to guiding the visualizer team on connections and documentation of my software.
    - Worked across teams to begin integrating our entire software stack.
- Jacob - Board design
    - Made final updates to seven segment displays that have been given the final okay
    - Added tooltips to switches and buttons to give details of functionality ○
    Added a speed slider that will be used for simulation speed
    - Made push button function to control them with keys u, i, o, p, and k.

Pending Issues

Assembler - Eric

- Store f - Syntax highlighting. The syntax highlighting works but when it is turned off it does not return to the correct state. I will need to repopulate the entire table of user code from the original file.

Aiman

- Block files naming should be changed due to clients request. Block3 -> _2to1mux Colby
- No major issues. GUI needs minor corrections to placement of some wires and bus width info.

Individual Contributions

Team Member	Contribution	Weekly Hours	Total Hours

Eric Marcanio	Able to view the internals of the data memory in a new pop up table	**6**	**31**
Colby McKinley	Made UI updates	**6**	**33**
Aiman Priester	Further testing Extensive debugging	**7**	**27**
Brady Kolosik	Synthesis of the CPU and visualizer, various visual changes and functional changes to pre existing modules as well	**9**	**29.5**
Bryce Snell	Connected the assembler and visualizer to the simulator.	**7**	**25**
Jacob Betsworth	Made significant UI changes	**8**	**25**

Plans for Upcoming Reporting Period

Eric - Fix bugs in the syntax highlighting and also change the numbering in the data table to show the spot in memory for each variable rather than the line number.

Jacob - Have a meeting with Bryce to take on the UI and CPU integration, have switch 17 and 16 properly defined for game function for sw16 and reg view/mem view for sw17

Aiman - Client has new tasks which involves renaming and splitting of the files from BDF to strictly verilog.

Bryce - Continue having meetings with product owners to ensure a smooth integration period as the project enters the final stages.

Colby -

- Make corrections to the wires as requested by Dr. Stoytchev.
- Make corrections bus width info as requested by Dr. Stoytchev.
- Shrink and recenter PC Value box
- Move Register file registers to the right

Brady-

- Continue to interface with the CPU, write control signals to proper locations
- Potentially animate a mux based on a control signal
- Reimplement instruction highlighting
- Move buttons to switch between lower and upper 32 instructions into the imem module
    - Also add a checkbox to start in user space(make it preferred default as well) ●

Continue to refactor and clean up existing code as the design is finalized