Brian Humphries

David Roster

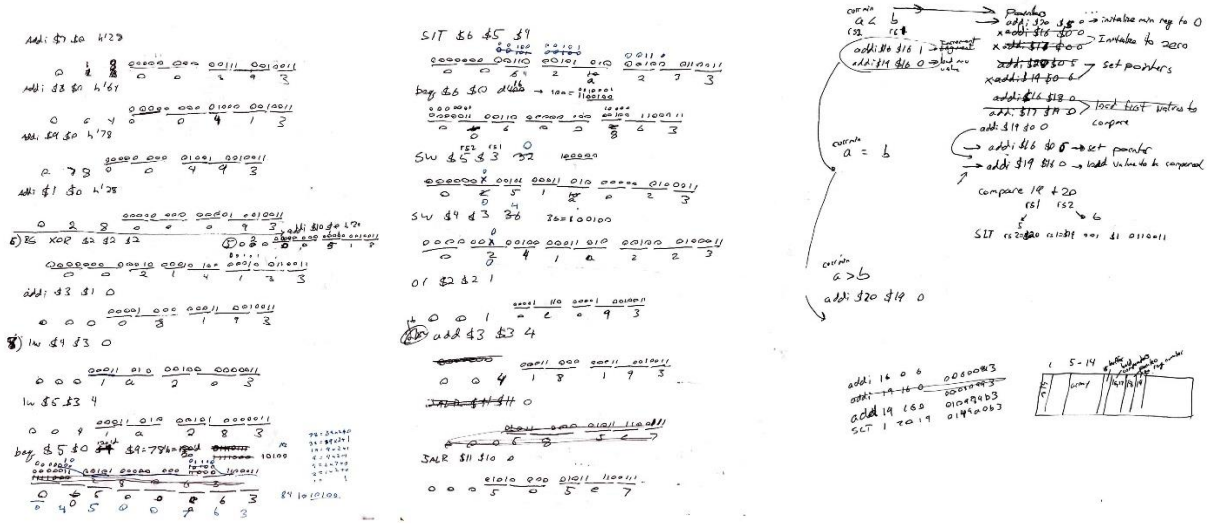
10/25/2018

ECE 156A

Homework 1: Report

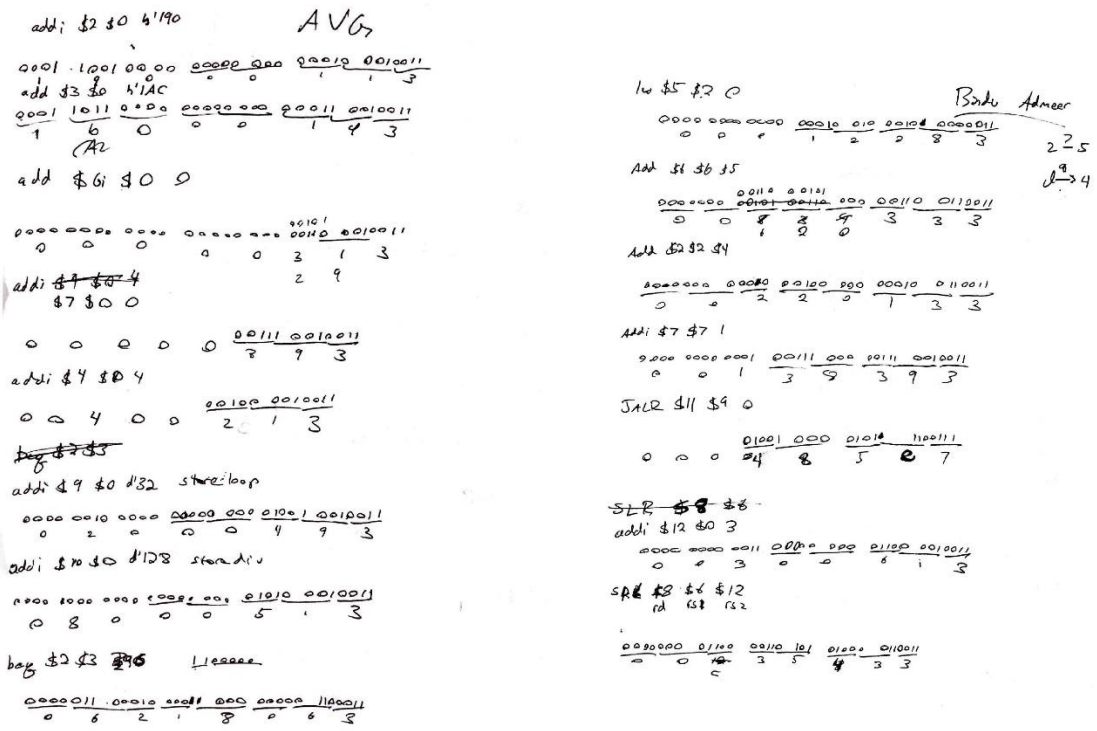
Algorithm Code:

We did all algorithm code by hand by first writing down our pseudo algorithms into Assembly and then translating into binary and finally hex. Pictured below is all algorithms we used. This part of the lab was challenging because we made many translation errors, but our handwritten steps made it easier to debug because we had written evidence of each instruction made and the steps taken to get each answer.

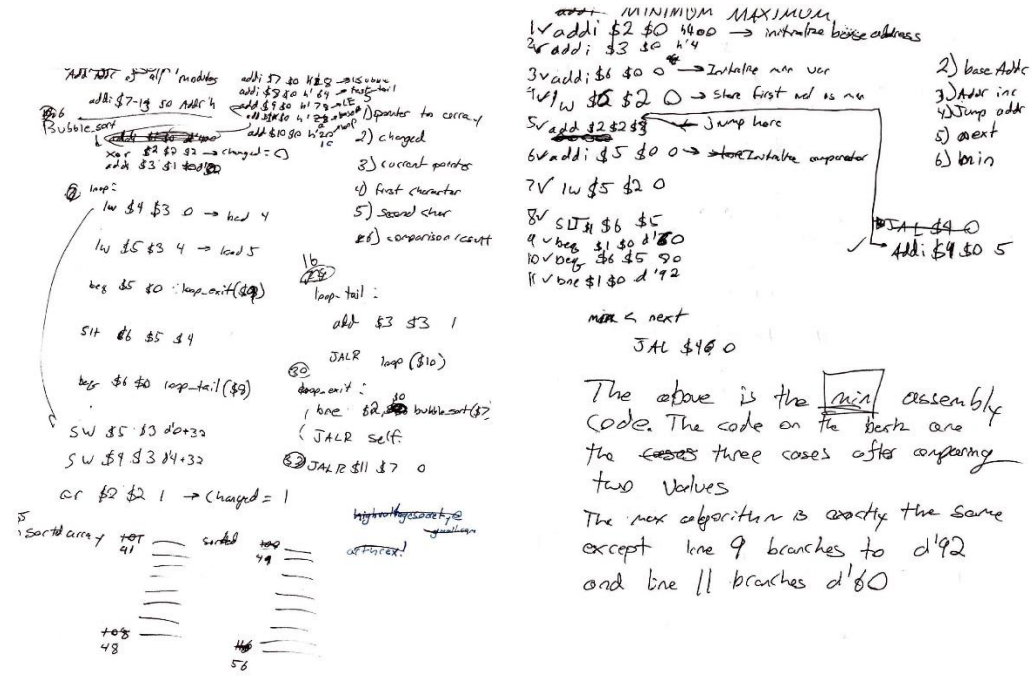


Bubble Sort Assembly Code

AVG Value Assembly Code: Part 1 and Part 2



Minimum/Maximum Assembly Code



Insertion Sort Assembly Code

```
for_init:
    addi $2, $0, 1
    addi $7, $0, 8
    addi $7, $7, -1
    addi $11, $0, 4
    addi $8, $0, 1000
    addi $15, $0, 10
    sw $15, 0($8)
    lw $14, 0($8)
    addi $15, $0, 2834
    sw $15, 4($8)
    addi $15, $0, 2883
    sw $15, 8($8)
    addi $15, $0, 947
    sw $15, 12($8)
    addi $15, $0, 99
    sw $15, 16($8)
    addi $15, $0, 2
    sw $15, 20($8)

    addi $8, $0, 1050
    addi $15, $0, 10
    sw $15, 0($8)
    addi $15, $0, 2834
    sw $15, 4($8)
    addi $15, $0, 2883
    sw $15, 8($8)
    addi $15, $0, 947
    sw $15, 12($8)
    addi $15, $0, 99
    sw $15, 16($8)
    addi $15, $0, 2
    sw $15, 20($8)

for_loop:
    blt $7, $2, end_for
    addi $6, $2, -1
    mul $6, $2, $11
    add $6, $8, $6
    lw $4, 0($6)
```

```
while:
    blt $3, $0, endwhile
    mul $6, $3, $11
    add $6, $8, $6
    lw $5, 0($6)
    blt $5, $4, endwhile
    sw $5, 4($6)
    addi $3, $3, -1
    jal $0, while

endwhile:
    mul $6, $3, $11
    add $6, $8, $6
    sw $4, 4($6)
    addi $2, $2, 1
    jal $0 for_loop


end_for:
    jal $0, end_for
```

32 Bit-Machine Code:

Our process was transcribing our assembly code on paper into machine code. Once our machine code was ready, we inserted it line by line into a text file and debugged it as we went by running simulations. This process allowed us to quickly identify when new errors would pop up and their most likely reason for occurrence. Some issues we did have during this method is by hardcoding data into our text file, deletions and insertions into our file changed our memory mapping and resulted in bugs.

Our machine codes will be uploaded in a Zip file which will be attached along with this report in an email to our designated TA.

Testbench File:

Our Testbench will be uploaded in a Zip file which will be attached along with this report in an email to our designated TA.

Simulation Results:

Our simulation results are displayed below. Any more detailed results of simulations can be given at a moment's notice at the request of the TA.

Bubble sort :

- On every iteration you find a largest number in the array and bubble it out to the last index of the array.

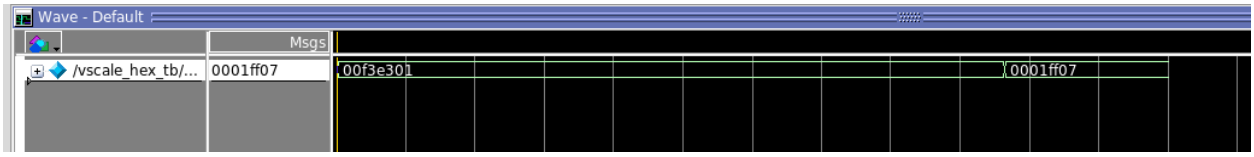
Insertion sort :

- Here you have two regions sorted and unsorted. At Every Iteration you pick up an element from unsorted region and insert at proper location in Sorted region.

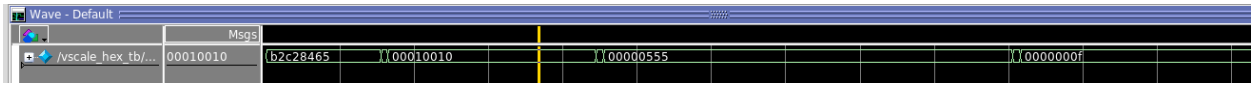
Insertion Sort and Bubble sort both have Worst Case O(N^2). But if the array is mostly sorted Insertion Sort will perform better. Insertion sort is often used with Quick sort: In Quick Sort after some level of recursion (When array is mostly sorted) Insertion sort is used.

AVG WaveForms

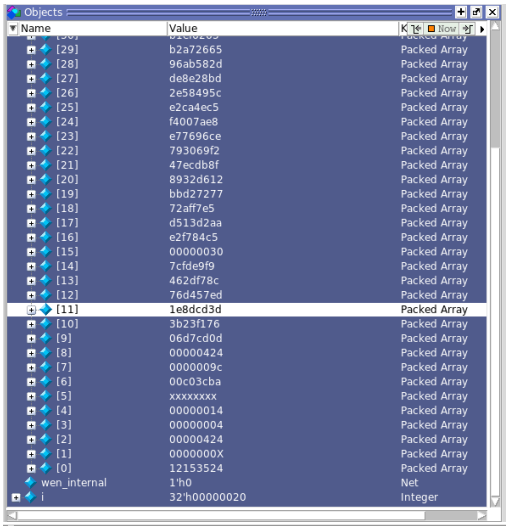
```
 94  00000000
 95  00000000
 96  00000000
 97  00000000
 98  00000000
 99  00000000
100  00000000
101  00000001
102  00000113
103  00000010
104  00000004
105  00000400
106  00008005
107  00007300
108  000f0010
```

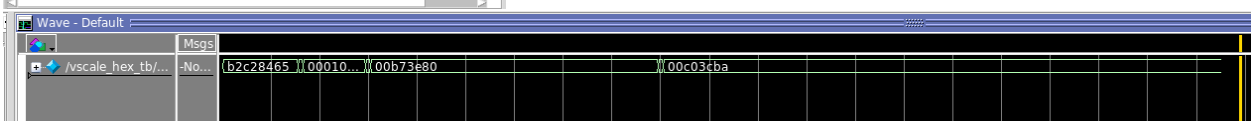


MIN WaveForms



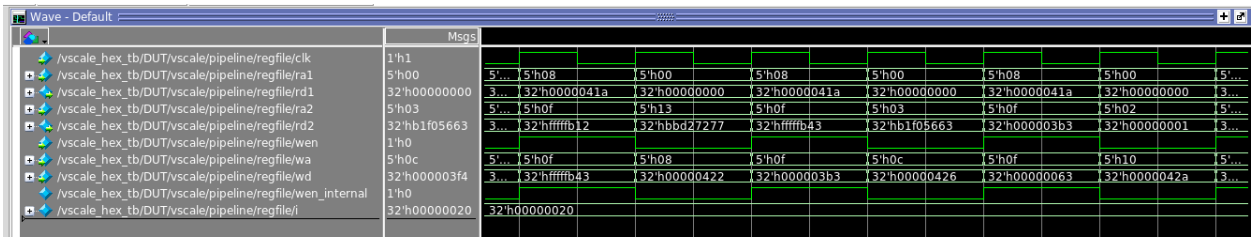
MAX WaveForms





BubbleSort WaveForms

INSERTION SORT Waveform



Feedback:

Lab 01 was challenging and tedious due to my partner and I have little experience using Modelsim. The lab however gave us an excellent understanding how valuable a processor is and just how vital its role is in translating higher level code into lower level code. We understood why the creators of MIPS emphasized its 4 principles that governed the conditions on which MIPS worked.

The lab could be better implemented by having a couple workshops on using MIPS and Modelsim in general. Many of our classmates were struggling to finish the lab on time. If the lab were not extended, it would have resulted in less productive work overall. Many of the underclassmen are taking ECE 156A and ECE 154A concurrently so while Modelsim was also introduced in ECE 154, we have only been working with this software for less than two weeks.

CODE:

19000113

1b000193

00000313

00000393

00400213

02000493

08000513

06218063

00000293

00012283

00530333

00220133

00138393

000485e7

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00300613

00c35433

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000
00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000001

00000113

00000010

00000004

00000400

00008005

00007300

000f0010