**Lab 1B Report**

a) I have attached the original ISA, and the edited ISA.

Changes:

Branch Lengths

- I had to adjust a couple of these because I forgot to account for the

automatic PC increment

Branch Conditions

- I had to change a couple conditions because the original conditions were

not sufficient for the programs

Add instruction

- The original Add instruction used a 16-bit register. This one now utilizes

a carry flag and an 8-bit register.

b)

i) The simulator runs each instruction as they are supposed to work. For this reason, the

simulator will be a helpful tool to check my design in later designs.

ii) The simulator first checks, via if statements, the op-code (2 bits), then checks the next

3 bits, and then checks the final 3 bits. After the final check, it runs the correct

instruction.

iii) I did not run into any bugs, outside of the ISA flaws which I have now changed.

iv) The simulator will be updated every time there needs to be a change to the ISA.

However, I do not expect the ISA to change.

c)

i) I implemented the Assembler, so the code is available in asm.txt. The code comments

are listed below.

ii) I have commented on log.txt below using the clock cycle number.

Code Comments

Note: This program runs both the Square and Width programs.

1. Load R1 with the value at mem. loc. 0 (keep count of number of additions)
2. Load R2 with the value at mem. loc. 0 (add to R3 every loop)
3. Load R3 with the value at mem. loc. 0 (the low byte)
4. Add R2 to R3
5. Check if carry flag is set, add 1 to R4 (the high byte)
6. Decrement R1 by one
7. Check if R1 is one. If one, do not branch. Else, branch back to line 4
8. Store R4 to mem. loc. 1
9. Store R3 to mem. loc. 2
10. Set R1 to 0
11. Set R2 to 0
12. Set R3 to 0
13. Set R4 to 0
14. Load R1 with 31
15. Increment R1 by one
16. Load R2 with mem. loc. [R1]
17. Store width of R2 in R3
18. If R3 is greater than R4, do not branch, else branch to line 23
19. Store R3 to mem. loc. 3 (These next few lines are only reached if a wider number is found)
20. Store R2 to mem. loc. 4
21. Store R1 to mem. loc. 5
22. Load R4 with R3
23. If R1 = 63 or if R4 = 8, branch to line 25, else do not branch
24. Branch to line 15
25. Halt program.

**Log Comments**

1. R1 is loaded with the number to be squared at mem. loc. 0 (4)
2. R2 is loaded with the number to be squared at mem. loc. 0 (4)
3. R3 is loaded with the number to be squared at mem. loc. 0 (4)
4. R2 is added to R3 (8)
5. No carry, R4 remains unchanged.
6. R1 decremented (3)
7. R1 is not equal to 1, branch to add.
8. R2 is added to R3 (12)
9. No carry, R4 remains unchanged.
10. R1 decremented (2)
11. R1 is not equal to 1, branch to add.
12. R2 is added to R3 (16)
13. No carry, R4 remains unchanged.
14. R1 decremented (1)
15. R1 is equal to 1, no branch
16. R4 stored to mem. loc. 1 (0)
17. R3 stored to mem. loc. 2 (16) [This is correct, the square of 4 is 16]
18. R1 is loaded with 0
19. R2 is loaded with 0
20. R3 is loaded with 0
21. R4 is loaded with 0
22. R1 is loaded with 31
23. R1 incremented (32)
24. R2 is loaded with mem. loc. [R1 -> 32] (00100100)
25. R3 is loaded with the width of R2 (4) [Which is correct]
26. R3 is greater than R4, do not branch
27. Store R3 at mem. loc. 3 (4) [width]
28. Store R2 at mem. loc. 4 (00100100) [number]
29. Store R1 at mem. loc. 5 (32) [location]
30. Load R4 with R3 (4)
31. R1 is not equal to 63 and R4 is not 8, do not branch
32. Branch to inc
33. R1 incremented (33)
34. R2 is loaded with mem. loc. [R1 -> 33] (01000000)
35. R3 is loaded with the width of R2 (1) [Which is correct]
36. R4 is greater than R3, branch to beq
37. R1 is not equal to 63 and R4 is not 8, do not branch
38. Branch to inc
39. R1 incremented (34)
40. R2 is loaded with mem. loc. [R1 -> 34] (10000001)
41. R3 is loaded with the width of R2 (8) [Which is correct]
42. R3 is greater than R4, do not branch
43. Store R3 at mem. loc. 3 (8) [width]
44. Store R2 at mem. loc. 4 (10000001) [number]
45. Store R1 at mem. loc. 5 (34) [location]
46. Load R4 with R3 (8)
47. R4 is 8, branch to halt
48. Program halted