

HW 4 REPORT

Registers before execution:

```
$ zero 00000000000000000000000000000000
$ at   00000000000000000000000001110001 = 113
$ v0   0000000000000000000000011000011110 = 1566
$ v1   0000000000000000000000011000011110 = 1566
$ a0   00000000000000000000000000000000
$ a1   00000000000000000000000000000000
$ a2   00000000000000000000000000000000
$ a3   00000000000000000000000000000000
$ t0   00000000000000000000000000000000
$ t1   00000000000000000000000000000000
$ t2   000000000000000000000000011000111 = 199
$ t3   00000000000000000000000000000000
$ t4   11111111111111111111111011101000 = -280
$ t5   00000000000000000000000000000000
$ t6   11111111111111111111111011101000 = -280
$ t7   00000000000000000000000000000000
$ s0   000000000000000000000001110000100 = 900
$ s1   00000000000000000000000000000000
$ s2   000000000000000000000000000000111
$ s3   00000000000000000000000000000000
$ s4   11010100000010010100000111100011
$ s5   00000000000000000000000000000000
$ s6   00000000000000000111111110001110
$ s7   00000000000000000000000000000000
$ t8   00100000100101001000000001110001
$ t9   00000000000000000000000000000000
$ k0   10001001010100111111001010111100
$ k1   00000000000000000000000000000000
$ gp   00000000000000000111111110001110
$ sp   00000000000000000000000000000000
$ fp   00000000000000000000000000000000
$ ra   00000000000000000000000000000000
```

Registers before execution:

They dont have any value on them. They all are X.

Instructions:

```

000010000000000000000000000000000001 --> j 0x0000009          jumps to line
1 101011000000000100000000000000001000 --> sw v0 0x0008(zero)
2 101011000000000100000000000000001100 --> sw v0 0x000c(zero)
3 1010110000000001000000000000000010000 --> sw v0 0x0010(zero) } Dummy sw instructions show
4 0011010000100100000000000000100011 --> ori a0 at 0x0023      jumps and branch instructions work
5 000000111110000000000000000000001000 --> jr ra
6 10101100000000010000000000000010100 --> sw v0 0x0014(zero)    jumps to ra = line 10
7 10101100000000010000000000000011000 --> sw v0 0x0018(zero)
8 10101100000000010000000000000011100 --> sw v0 0x001c(zero)
9 000011000000000000000000000000000100 --> jal 0x0000004         jumps to line 4, ra = 9 + 1 = 10
10 0001000001000011000000000000000011 --> beq v0 v1 0x0003      v0 = v1
11 1010110000000001000000000000100000 --> sw v0 0x0020(zero)    jumps to line 10 + 1 + 3 = 14
12 1010110000000001000000000000100100 --> sw v0 0x0024(zero)
13 1010110000000001000000000000101000 --> sw v0 0x0028(zero)
14 00111100000001101111111111111111 --> --> lui a2 0xffff      at != v0, does not jump
15 00010000001000101111111111111111 --> beq at v0 0xffff
16 0001010000100010000000000000000011 --> bne at v0 0x0003      at != v0
17 1010110000000001000000000000101100 --> sw v0 0x002c(zero)    jumps to line 16 + 1 + 3 = 20
18 1010110000000001000000000000110000 --> sw v0 0x0030(zero)
19 1010110000000001000000000000110100 --> sw v0 0x0034(zero)
20 00111100000001110101010101010101 --> lui a3 0x5555
21 00010100010000111111111111111111 --> bne v0 v1 0xffff      v0 = v1, does not jump
22 00001000000000000000000000000011111 --> j 0x0000001f          jumps to line 32
23 1010110000000001000000000000111000 --> sw v0 0x0038(zero)
24 1010110000000001000000000000111100 --> sw v0 0x003c(zero)
25 1010110000000001000000000000100000 --> sw v0 0x0040(zero)
26 0011010000100101000000000000000000 --> ori a1 at 0x0000
27 0000001111100000000000000000001000 --> jr ra                  jumps to ra = line 33
28 1010110000000001000000000000100100 --> sw v0 0x0044(zero)
29 1010110000000001000000000000100100 --> sw v0 0x0048(zero)
30 10101100000000010000000000001001100 --> sw v0 0x004c(zero)
31 00001100000000000000000000000011010 --> jal 0x0000001a        jumps to line 26, ra = 32 + 1 = 33
32 00000001010000010101100000100000 --> add t3 t2 at
33 00000001100000010110100000100000 --> add t5 t4 at
34 00000001110000010111100000100010 --> sub t7 t6 at
35 00000010000000011000100000100010 --> sub s1 s0 at
36 00000010010000011001100000100110 --> xor s3 s2 at
37 0000001010000001101100000100110 --> xor s5 s4 at
38 00000010110000011011100000100100 --> and s7 s6 at
39 00000011000000011100100000100100 --> and s9 s8 at
40 00000011000000011100100000100100 --> or k1 k0 at
41 00000011100000011110100000100101 --> or sp gp at
42 10101100010000011111100111100010 --> sw at 0xf9e2(v0)
43 101011000000000110000000000000100 --> sw v1 0x0004(zero)
44 10001100010010001111100111100010 --> lw t0 0xf9e2(v0)
45 100011000000100100000000000000100 --> lw t1 0x0004(zero)
46

```

Register Output:

```

$ zero 00000000000000000000000000000000
$ at   00000000000000000000000000001110001
$ v0   00000000000000000000000000001100011110
$ v1   00000000000000000000000000001100011110
$ a0   00000000000000000000000000001110011 --> ori a0 at 0x0023
$ a1   00000000000000000000000000001110001 --> ori a1 at 0x0000
$ a2   111111111111111110000000000000000 --> lui a2 0xffff
$ a3   010101010101010100000000000000000 --> lui a3 0x5555
$ t0   00000000000000000000000000001110001 --> lw t0 0xf9e2(v0)
$ t1   00000000000000000000000000001100011110 --> lw t1 0x0004(zero)
$ t2   0000000000000000000000000000100111000 --> addn t3 t2 at --> 199 + 113 = 312
$ t3   00000000000000000000000000000000011 --> t2 > 0
$ t4   1111111111111111111111111101011001 --> addn t5 t4 at --> -280 + 113 = -167
$ t5   00000000000000000000000000000000010 --> t4 < 0
$ t6   11111111111111111111111111001110111 --> subn t7 t6 at --> -280 - 113 = -413
$ t7   00000000000000000000000000000000010 --> t6 < 0
$ s0   0000000000000000000000000000110010011 --> subn s1 s0 at --> 900 - 113 = 767
$ s1   00000000000000000000000000000000011 --> s0 > 0
$ s2   00000000000000000000000000001110110 --> xorn s3 s2 at
$ s3   00000000000000000000000000000000011
$ s4   11010100000010010100000110010010 --> xorn s5 s4 at
$ s5   00000000000000000000000000000000010
$ s6   00000000000000000000000000000000000 --> andn s7 s6 at
$ s7   00000000000000000000000000000000001
$ t8   00000000000000000000000000001110001 --> andn s9 s8 at
$ t9   00000000000000000000000000000000011
$ k0   10001001010100111111001011111101 --> orn k1 k0 at
$ k1   00000000000000000000000000000000010
$ gp   0000000000000000000000000000111111111111111 --> orn sp gp at
$ sp   00000000000000000000000000000000011
$ fp   00000000000000000000000000000000000
$ ra   000000000000000000000000000010000000

```

Memory output:

```

00000000000000000000000000001110001 --> sw at 0xf9e2(v0)
00000000000000000000000000001100011110 --> sw v1 0x0004(zero)
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

0xf9 = -1566, v0 = 1566, adress = 0
 adress = 4 + 0 = 4th byte = 1st word

The rest is all x since I didn't executed any sw instructions. The sw instructions I wrote as dummy instructions didn't work which shows that jump and branch instructions work too.