**EE 310 – Lab 6 Report**

**NAU, 10 April 2020**

**By Alenn Wright**

**Problem Description**

In this lab, we are going to run a program on a simplified version of MIPSzy. The simplified MIPSzy design is created from the Verilog code given in section 6.7 of our book. It has the following properties:

 It can execute only lw, sw, addi, add instructions,

 Its data memory space is limited to 1024 32-bit words, from addresses 4096 to 8188 (recall that addresses are incremented by 4, i.e. 4096, 4100, 4104, …),

 It can store programs of up to 1024 lines long (more than enough for our 10-20 line programs),

 It can NOT implement any branches, jumps, comparisons, and hence NO loops or subroutines are possible at this time.

|  |
| --- |
| first program    second program |

Figure 1. Expected behavior of the circuit

**Solution Plan**

In order to solve the problem explained above, I will for the first program save the opposite ends of the lists in registers and swap the values. The second program will have 1 register hold the sum and another to hold the current data position, then save the new sum in the data position.

|  |
| --- |
|  |

Figure 2. State diagram for the proposed solution

**Implementation and Test Plan**

I have implemented the solution plan explained above, by *the code is exactly what I set out to do however, I found that the emulator does not accept ajusted data positions so I had to save the data address in a the address register then save the new values.*

|  |
| --- |
| //first program  addi $t6, $zero, 5000  lw $t0, 0($t6)  addi $t6, $t6, 28  lw $t1, 0($t6)  sw $t0, 0($t6)  addi $t6, $t6, -28  sw $t1, 0($t6)  addi $t6, $t6, 4  lw $t0, 0($t6)  addi $t6, $t6, 20  lw $t1, 0($t6)  sw $t0, 0($t6)  addi $t6, $t6, -20  sw $t1, 0($t6)  addi $t6, $t6, 4  lw $t0, 0($t6)  addi $t6, $t6, 12  lw $t1, 0($t6)  sw $t0, 0($t6)  addi $t6, $t6, -12  sw $t1, 0($t6)  addi $t6, $t6, 4  lw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  sw $t0, 0($t6)  addi $t6, $t6, -4  sw $t1, 0($t6)  MIPSzy\_0.IM.memory[0] = 'b00100000000011100001001110001000;  MIPSzy\_0.IM.memory[1] = 'b10001101110010000000000000000000;  MIPSzy\_0.IM.memory[2] = 'b00100001110011100000000000011100;  MIPSzy\_0.IM.memory[3] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[4] = 'b10101101110010000000000000000000;  MIPSzy\_0.IM.memory[5] = 'b00100001110011101111111111100100;  MIPSzy\_0.IM.memory[6] = 'b10101101110010010000000000000000;  MIPSzy\_0.IM.memory[7] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[8] = 'b10001101110010000000000000000000;  MIPSzy\_0.IM.memory[9] = 'b00100001110011100000000000010100;  MIPSzy\_0.IM.memory[10] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[11] = 'b10101101110010000000000000000000;  MIPSzy\_0.IM.memory[12] = 'b00100001110011101111111111101100;  MIPSzy\_0.IM.memory[13] = 'b10101101110010010000000000000000;  MIPSzy\_0.IM.memory[14] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[15] = 'b10001101110010000000000000000000;  MIPSzy\_0.IM.memory[16] = 'b00100001110011100000000000001100;  MIPSzy\_0.IM.memory[17] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[18] = 'b10101101110010000000000000000000;  MIPSzy\_0.IM.memory[19] = 'b00100001110011101111111111110100;  MIPSzy\_0.IM.memory[20] = 'b10101101110010010000000000000000;  MIPSzy\_0.IM.memory[21] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[22] = 'b10001101110010000000000000000000;  MIPSzy\_0.IM.memory[23] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[24] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[25] = 'b10101101110010000000000000000000;  MIPSzy\_0.IM.memory[26] = 'b00100001110011101111111111111100;  MIPSzy\_0.IM.memory[27] = 'b10101101110010010000000000000000;  //second prog  addi $t6, $zero, 5000  lw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  add $t0, $t0, $t1  sw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  add $t0, $t0, $t1  sw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  add $t0, $t0, $t1  sw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  add $t0, $t0, $t1  sw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  add $t0, $t0, $t1  sw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  add $t0, $t0, $t1  sw $t0, 0($t6)  addi $t6, $t6, 4  lw $t1, 0($t6)  add $t0, $t0, $t1  sw $t0, 0($t6)  MIPSzy\_0.IM.memory[0] = 'b00100000000011100001001110001000;  MIPSzy\_0.IM.memory[1] = 'b10001101110010000000000000000000;    MIPSzy\_0.IM.memory[2] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[3] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[4] = 'b00000001000010010100000000100000;  MIPSzy\_0.IM.memory[5] = 'b10101101110010000000000000000000;    MIPSzy\_0.IM.memory[6] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[7] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[8] = 'b00000001000010010100000000100000;  MIPSzy\_0.IM.memory[9] = 'b10101101110010000000000000000000;    MIPSzy\_0.IM.memory[10] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[11] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[12] = 'b00000001000010010100000000100000;  MIPSzy\_0.IM.memory[13] = 'b10101101110010000000000000000000;    MIPSzy\_0.IM.memory[14] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[15] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[16] = 'b00000001000010010100000000100000;  MIPSzy\_0.IM.memory[17] = 'b10101101110010000000000000000000;    MIPSzy\_0.IM.memory[18] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[19] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[20] = 'b00000001000010010100000000100000;  MIPSzy\_0.IM.memory[21] = 'b10101101110010000000000000000000;    MIPSzy\_0.IM.memory[22] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[23] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[24] = 'b00000001000010010100000000100000;  MIPSzy\_0.IM.memory[25] = 'b10101101110010000000000000000000;    MIPSzy\_0.IM.memory[26] = 'b00100001110011100000000000000100;  MIPSzy\_0.IM.memory[27] = 'b10001101110010010000000000000000;  MIPSzy\_0.IM.memory[28] = 'b00000001000010010100000000100000;  MIPSzy\_0.IM.memory[29] = 'b10101101110010000000000000000000; |

Figure 3. Verilog code for the proposed solution

|  |
| --- |
|  |

Figure 5. Lab pictures of the running solution