Fall
2018

# TEST CODE 1 – Has all of the instructions defined by the project requirement (V1).

ADVANCED HARDWARE DESIGN (TEAM 5)

TENZING RABGYAL

| PC | INSTRUCTION (imm values shown as decimal. Convert this to binary for opcode) | OPCODE | COMMENTS |
|---|---|---|---|
| **0** | Addi r0,r1,9 | 00000100000000010000000000001001 | R1 now has 0x9 |
| **4** | Addi r0,r2,2 | 00000100000000100000000000000010 | R2 now has 0x2 |
| **8** | Subi r1,r1,2 | 00001000001000010000000000000010 | R1 now has 0x7 |
| **12** | Addi r1,r2,1 | 00000100001000100000000000000001 | R2 now has 0x8 |
| **16** | Or r1,r2,r3 | 00000000001000100001100000000111 | R3 now has 0xF |
| **20** | And r2,r3,r4 | 00000000010000110010000000000101 | R4 now has 0x8 |
| **24** | Andi r3,r5,11 | 00001100011001010000000000001011 | R5 now has 0xB |
| **28** | Ori r3,r6,11 | 00010000011001100000000000001011 | R6 now has 0xF |
| **32** | Sub r2,r1,r1 | 00000000010000010000100000000011 | R1 now has 0x1 |
| **36** | Nor r1,r2,r7 | 00000000001000100011100000001001 | R7 now has 0xFFFFFFF6 |
| **40** | Shl r1,r8,2 | 00010100001010000000000000000010 | R8 now has 0x4 |
| **44** | Sw r1,r8,4 | 00100000001010000000000000000100 | Location 5 now has the 0x4 |
| **48** | Lw r1,r9,4 | 00011100001010010000000000000100 | R9 now has 0x4 |
| **52** | Add r9,r9,r9 | 00000001001010010100100000000001 | First Time: R9 now has 0x8<br>Second Time: R9 now has 0x10 |
| **56** | Beq r9,r2,-8 | 00101001001000101111111111111110 | First Time: Since R9 == R2. Branch to prev ins.<br>Second Time: R9 now has 0x10. Go to next ins. |
| **60** | Addi r0,r10,3 | 00000100000010100000000000000011 | R10 now has 0x3 |
| **64** | Addi r0,r11,1 | 00000100000010110000000000000001 | R11 now has 0x1 |
| **68** | Addi r11,r11,1 | 00000101011010110000000000000001 | First Time: R11 now has 0x2<br>Second Time: R11 now has 0x3 |
| **72** | Bne r10,r11,-8 | 00101101010010111111111111111110 | First Time: Since R10 != R11. Branch to prev ins.<br>Second Time: R10 == R11. Go to next ins. |
| **76** | Addi r0,r12,1 | 00000100000011000000000000000001 | R12 now has 0x1 |
| **80** | Addi r12,r12,1 | 00000101100011000000000000000001 | First Time: R12 now has 0x2.<br>Second Time: R12 now has 0x3. |
| **84** | Blt r12,r10,-8 | 00100101100010101111111111111110 | First Time: Since R12 < R10. Branch to prev ins.<br>Second Time: R12 == R10. Go to next ins. |
| **88** | Jmp 96 | 00110000000000000000000000011000 | Jumps to 96 (PC Value). Will skip the next instruction. |
| **92** | Addi r0,r14,255 | 00000100000011100000000011111111 | *THIS INSTRUCTION WILL NOT BE EXECUTED* |
| **96** | Addi r0,r15,170 | 00000100000011110000000010101010 | R15 now has 0xAA |
| **100** | Hal | 11111100000000000000000000000000 | Program end. |

NOTE: All registers and dmem locations have 0x0 value prior to this program's execution.

NOTE: For jump and branch instructions, divide the decimal value by 4 before converting it to binary. This is done to negate the fact that MIPS concatenates "00" at the end of these values.

REGISTER AND DMEM VALUES AFTER THIS PROGRAM FINISHES

| REGISTER/DMEM | VALUE (in hex) |
|---|---|
| r1 | 0x00000001 |
| r2 | 0x00000008 |
| r3 | 0x0000000F |
| r4 | 0x00000008 |
| r5 | 0x0000000B |
| r6 | 0x0000000F |
| r7 | 0xFFFFFFF6 |
| r8 | 0x00000004 |
| r9 | 0x00000010 |
| dmem location = 5 | 0x00000004 |
| r10 | 0x00000003 |
| r11 | 0x00000003 |
| r12 | 0x00000003 |
| r15 | 0x000000AA |

NOTE: Registers and dmem locations that are not shown on the table above have 0x0 value.

ASSEMBLY CODE

```
addi r0,r1,0000000000001001
addi r0,r2,0000000000000010
subi r1,r1,0000000000000010
addi r1,r2,0000000000000001
or1 r1,r2,r3
and r2,r3,r4
andi r3,r5,0000000000001011
ori r3,r6,0000000000001011
sub r2,r1,r1
nor r1,r2,r7
shl r1,r8,0000000000000010
sw1 r1,r8,0000000000000100
lw1 r1,r9,0000000000000100
add r9,r9,r9
beq r9,r2,1111111111111110
addi r0,r10,0000000000000011
addi r0,r11,0000000000000001
addi r11,r11,0000000000000001
bne r10,r11,1111111111111110
addi r0,r12,0000000000000001
addi r12,r12,0000000000000001
blt r12,r10,1111111111111110
jmp 00000000000000000000011000
addi r0,r14,0000000011111111
addi r0,r15,0000000010101010
hal
```

OPCODE

"00000100","00000001","00000000","00001001",
"00000100","00000010","00000000","00000010",
"00001000","00100001","00000000","00000010",
"00000100","00100010","00000000","00000001",
"00000000","00100010","00011000","00000111",
"00000000","01000011","00100000","00000101",
"00001100","01100101","00000000","00001011",
"00010000","01100110","00000000","00001011",
"00000000","01000001","00001000","00000011",
"00000000","00100010","00111000","00001001",
"00010100","00101000","00000000","00000010",
"00100000","00101000","00000000","00000100",
"00011100","00101001","00000000","00000100",
"00000001","00101001","01001000","00000001",
"00101001","00100010","11111111","11111110",
"00000100","00001010","00000000","00000011",
"00000100","00001011","00000000","00000001",
"00000101","01101011","00000000","00000001",
"00101101","01001011","11111111","11111110",
"00000100","00001100","00000000","00000001",
"00000101","10001100","00000000","00000001",
"00100101","10001010","11111111","11111110",
"00110000","00000000","00000000","00011000",
"00000100","00001110","00000000","11111111",
"00000100","00001111","00000000","10101010",
"11111100","00000000","00000000","00000000"