RTL test plan

This documentation walks through brief examples of how each command in the RTL well work.

**R-Type (ALU Commands)**:

**Initialize:** $m = 10 (0b1010), $t0 = 5 (0b101), PC = 0

**Instructions**: Add $m, $t0 (0000 0000 000 100 XX)

And $m, $t0 (0000 0001 000 100 XX)

Add $m, $t0

**Fetch:** IR = 0000 0000 000 100 XX

PC = 0 + 1

**Decode:** AiA = 10

AiB = 5

**Execute:** ALUout = 10 + 5 = 15

**Mem/Reg:** Reg[0] = 15

And $m, $t0

**Fetch:** IR = 0000 0001 000 100 XX

PC = 1 + 1

**Decode**: AiA = 15 (0b1111)

AiB = 5 (0b0101)

**Execute**: ALUout = 1111 && 0101 = 0101

**Mem/Reg**: Reg[0] = 5

**I-type (ALU commands)**:

**Initialize**: $m = 0, PC = 10

**Instructions**: Addi 15

Ori 32

Addi 15

**Fetch**: IR = 0001 000000001111

PC = 10 + 1 = 11

**Decode**: AiA = 0b0000000000000000

AiB = 0b0000000000001111

**Execute**: ALUout = 15 + 0 = 15

**Mem/Reg**: reg [0] = 15

Ori 32

**Fetch**: IR = 1001 000000100000

PC = 11 + 1 = 12

**Decode**: AiA = 0b0000000000001111

AiB = 0b0000000000100000

**Execute:** ALUout = 15 || 32 = 0000000000101111 = 47

**Mem/Reg:** reg [0] = 47

**Load Immediate**

**Initialize**: $m = 0, PC = 10

**Instructions**: Li 10

Li12

**Fetch**: IR = 1000 000000001010

PC = 10 + 1 = 11

**Decode**: Reg[0] = 10

**Fetch**: IR = 1000 000000001100

PC = 11 + 1 = 12

**Decode**: Reg[0] = 12

**Jump**

**Initialize**: PC = 0x014a

**Instructions**: j 10

J 100

j 10

**Fetch**: IR = 0100 000000001010

PC = 0x014a + 1 = 0x014b

**Decode**: PC = 0x0 concatenate 0x00a = 0x000a

j 100

**Fetch**: IR = 1000 000000001100

PC = 0x000a + 1 = 0x000b

**Decode**: PC = 0x0 concatenate 0x064 = 0x0064

**Branch**

**Initialize**: $m = 0, $t0 = 0, $t1 = 10 PC = 10

**Instructions**: beq $t0, 15

beq $t1, 32

beq $t0, 15

**Fetch**: IR = 0010 000000001111

PC = 10 + 1 = 11

**Decode**: AiA = 0b0000000000000000

AiB = 0b0000000000000000

**Execute**: ALUout = 0 - 0 = 0

PC = 11 + 15 = 26

beq $t1, 32

**Fetch**: IR = 0010 000000100000

PC = 26 + 1 = 27

**Decode**: AiA = 0b0000000000000000

AiB = 0b0000000000001010

**Execute:** ALUout = 0 – 10 = -10

PC = 27