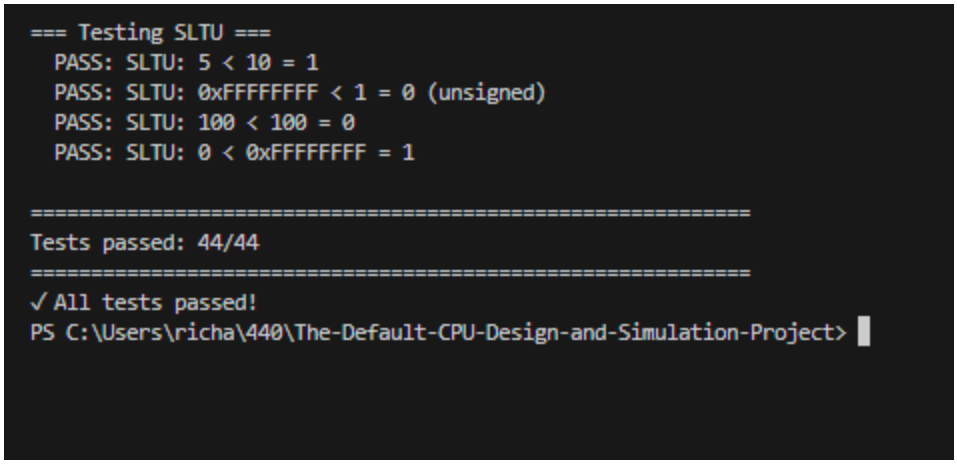
Alu.py test

```
============================================================
RISC-V ALU Unit Tests
============================================================

=== Testing ADD ===
  PASS: ADD: 5 + 10 = 15
  PASS: ADD: 100 + 0 = 100
  PASS: ADD: 0xFFFFFFFF + 1 = 0 (overflow)
  PASS: ADD: Max positive + 1

=== Testing SUB ===
  PASS: SUB: 10 - 5 = 5
  PASS: SUB: 0 - 1 = 0xFFFFFFFF (underflow)
  PASS: SUB: 42 - 42 = 0
  PASS: SUB: 100 - 50 = 50

=== Testing AND ===
  PASS: AND: 0xFF & 0x0F = 0x0F
  PASS: AND: anything & 0 = 0
  PASS: AND: x & x = x
  PASS: AND: Non-overlapping bits = 0

=== Testing OR ===
  PASS: OR: 0xF0 | 0x0F = 0xFF
  PASS: OR: x | 0 = x
  PASS: OR: x | 0xFFFFFFFF = 0xFFFFFFFF
  PASS: OR: 0 | 0 = 0

=== Testing XOR ===
  PASS: XOR: 0xFF ^ 0x0F = 0xF0
  PASS: XOR: x ^ x = 0
  PASS: XOR: x ^ 0 = x
  PASS: XOR: Bit inversion

=== Testing SLL ===
  PASS: SLL: 1 << 4 = 16
  PASS: SLL: x << 0 = x
  PASS: SLL: 1 << 31 = 0x80000000
  PASS: SLL: Shift with overflow
  PASS: SLL: Shift amount wraps (36 & 0x1F = 4)

=== Testing SRL ===
  PASS: SRL: 16 >> 4 = 1
  PASS: SRL: x >> 0 = x
  PASS: SRL: Logical shift (no sign extension)
  PASS: SRL: 0xFFFFFFFF >> 31 = 1

=== Testing SRA ===
  PASS: SRA: 16 >> 4 = 1 (positive)
  PASS: SRA: Sign extension for negative
  PASS: SRA: -1 >> 4 = -1
  PASS: SRA: x >> 0 = x
  PASS: SRA: Positive number (no sign ext)

=== Testing SLT ===
  PASS: SLT: 5 < 10 = 1
  PASS: SLT: 10 < 5 = 0
  PASS: SLT: 7 < 7 = 0
  PASS: SLT: -1 < 1 = 1 (signed)
  PASS: SLT: -2 < -1 = 1
  PASS: SLT: 1 < -1 = 0 (signed)
```


=== Testing SLTU ===
PASS: SLTU: 5 < 10 = 1
PASS: SLTU: 0xFFFFFFFF < 1 = 0 (unsigned)
PASS: SLTU: 100 < 100 = 0
PASS: SLTU: 0 < 0xFFFFFFFF = 1
Tests passed: 44/44
✓ All tests passed!
PS C:\Users\richa\440\The-Default-CPU-Design-and-Simulation-Project>

Test_cpu.py

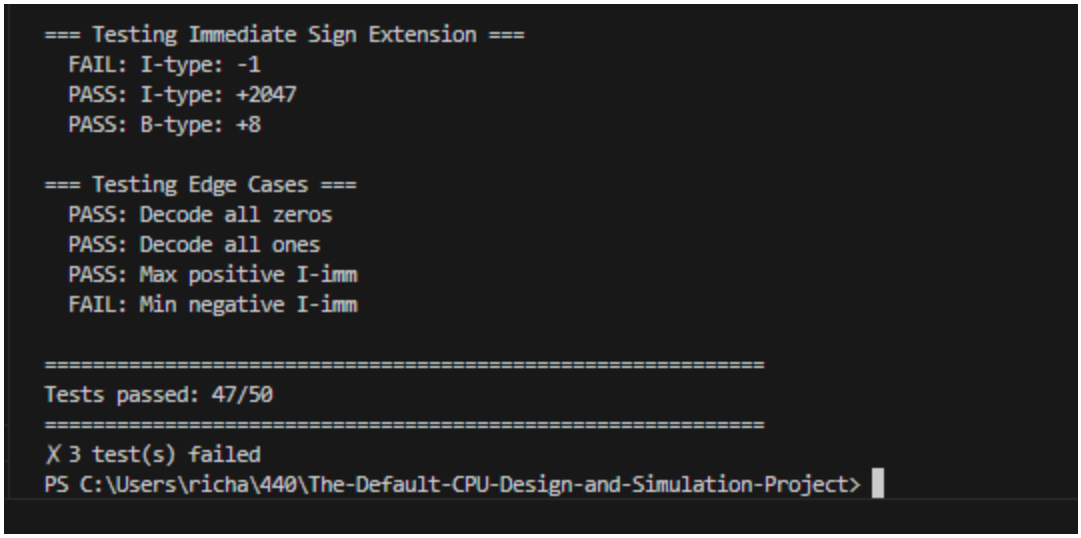
```
============================================================
CPU TESTS
============================================================

=== Test 1: Basic Arithmetic ===
Loading: test_arith.hex
Loaded 5 instructions from test_arith.hex
Loaded 5 instructions

Starting execution...
PC = 0x00000000

Halt detected at cycle 4

Finished after 4 cycles

============================================================
FINAL STATE
============================================================
Cycles: 4
Final PC: 0x00000010

=== Register File ===
x 0-x 3: 0x00000000 0x00000005 0x0000000A 0x0000000F
x 4-x 7: 0x00000005 0x00000000 0x00000000 0x00000000
x 8-x11: 0x00000000 0x00000000 0x00000000 0x00000000
x12-x15: 0x00000000 0x00000000 0x00000000 0x00000000
x16-x19: 0x00000000 0x00000000 0x00000000 0x00000000
x20-x23: 0x00000000 0x00000000 0x00000000 0x00000000
x24-x27: 0x00000000 0x00000000 0x00000000 0x00000000
x28-x31: 0x00000000 0x00000000 0x00000000 0x00000000

Memory (non-zero):
  [0x00000000] = 0x00500093 (5243027)
  [0x00000004] = 0x00A00113 (10486035)
  [0x00000008] = 0x002081B3 (2130355)
  [0x0000000C] = 0x40110233 (1074856499)
  [0x00000010] = 0x0000006F (111)
============================================================
x1 = 5 (should be 5)
x2 = 10 (should be 10)
x3 = 15 (should be 15)
x4 = 5 (should be 5)
PASS

=== Test 2: Memory Operations ===
Loading: test_mem.hex
Loaded 5 instructions from test_mem.hex
Loaded 5 instructions

Starting execution...
PC = 0x00000000

Halt detected at cycle 4

Finished after 4 cycles
```

```
============================================================
FINAL STATE
============================================================
Cycles: 4
Final PC: 0x00000010

=== Register File ===
x 0-x 3: 0x00000000 0x00000014 0x00010000 0x00000014
x 4-x 7: 0x00000000 0x00000000 0x00000000 0x00000000
x 8-x11: 0x00000000 0x00000000 0x00000000 0x00000000
x12-x15: 0x00000000 0x00000000 0x00000000 0x00000000
x16-x19: 0x00000000 0x00000000 0x00000000 0x00000000
x20-x23: 0x00000000 0x00000000 0x00000000 0x00000000
x24-x27: 0x00000000 0x00000000 0x00000000 0x00000000
x28-x31: 0x00000000 0x00000000 0x00000000 0x00000000

Memory (non-zero):
  [0x00000000] = 0x01400093 (20971667)
  [0x00000004] = 0x00010137 (65847)
  [0x00000008] = 0x00112023 (1122339)
  [0x0000000C] = 0x00012183 (74115)
  [0x00000010] = 0x0000006F (111)
  [0x00010000] = 0x00000014 (20)
============================================================
x1 = 20 (should be 20)
Memory[0x10000] = 20 (should be 20)
x3 = 20 (should be 20)
PASS

=== Test 3: Branches ===
Loading: test_branch.hex
Loaded 7 instructions from test_branch.hex
Loaded 7 instructions

Starting execution...
PC = 0x00000000

Halt detected at cycle 5

Finished after 5 cycles

============================================================
FINAL STATE
============================================================
Cycles: 5
Final PC: 0x00000018

=== Register File ===
x 0-x 3: 0x00000000 0x00000005 0x00000005 0x00000002
x 4-x 7: 0x00000000 0x00000000 0x00000000 0x00000000
x 8-x11: 0x00000000 0x00000000 0x00000000 0x00000000
x12-x15: 0x00000000 0x00000000 0x00000000 0x00000000
x16-x19: 0x00000000 0x00000000 0x00000000 0x00000000
x20-x23: 0x00000000 0x00000000 0x00000000 0x00000000
x24-x27: 0x00000000 0x00000000 0x00000000 0x00000000
x28-x31: 0x00000000 0x00000000 0x00000000 0x00000000
```

```
Memory (non-zero):
  [0x00000000] = 0x00500093 (5243027)
  [0x00000004] = 0x00500113 (5243155)
  [0x00000008] = 0x00000193 (403)
  [0x0000000C] = 0x00208463 (2131043)
  [0x00000010] = 0x00100193 (1048979)
  [0x00000014] = 0x00200193 (2097555)
  [0x00000018] = 0x0000006F (111)
============================================================
x3 = 2 (should be 2, not 1)
PASS - branch worked!

=== Test 4: Full Program (test_base.hex) ===
Loading: test_base.hex
Loaded 11 instructions from test_base.hex
Loaded 11 instructions

Starting execution...
PC = 0x00000000

Halt detected at cycle 9

Finished after 9 cycles

============================================================
FINAL STATE
============================================================
Cycles: 9
Final PC: 0x00000028

=== Register File ===
x 0-x 3: 0x00000000 0x00000005 0x0000000A 0x0000000F
x 4-x 7: 0x0000000F 0x00010000 0x00000002 0x00000000
x 8-x11: 0x00000000 0x00000000 0x00000000 0x00000000
x12-x15: 0x00000000 0x00000000 0x00000000 0x00000000
x16-x19: 0x00000000 0x00000000 0x00000000 0x00000000
x20-x23: 0x00000000 0x00000000 0x00000000 0x00000000
x24-x27: 0x00000000 0x00000000 0x00000000 0x00000000
x28-x31: 0x00000000 0x00000000 0x00000000 0x00000000

Memory (non-zero):
  [0x00000000] = 0x00500093 (5243027)
  [0x00000004] = 0x00A00113 (10486035)
  [0x00000008] = 0x002081B3 (2130355)
  [0x0000000C] = 0x40110233 (1074856499)
  [0x00000010] = 0x00010287 (66231)
  [0x00000014] = 0x0032A023 (3317795)
  [0x00000018] = 0x0002A203 (172547)
  [0x0000001C] = 0x00418463 (4293731)
  [0x00000020] = 0x00100313 (1049363)
  [0x00000024] = 0x00200313 (2097939)
  [0x00000028] = 0x0000006F (111)
  [0x00010000] = 0x0000000F (15)
============================================================
x1 = 5 (expected 5)
x2 = 10 (expected 10)
x3 = 15 (expected 15)
x4 = 15 (expected 15)
x5 = 0x00010000 (expected 0x00010000)
x6 = 2 (expected 2)
mem[0x10000] = 15 (expected 15)
PASS - Everything correct!

============================================================
Results: 4/4 passed
============================================================
```

Test_decoder.py

```
PS C:\Users\richa\440\The-Default-CPU-Design-and-Simulation-Project> python3 test_decoder.py
============================================================
Instruction Decoder Tests
============================================================

=== Testing R-type ===
  PASS: Type is R
  PASS: rd = 3
  PASS: rs1 = 1
  PASS: rs2 = 2
  PASS: funct3 = 0
  PASS: funct7 = 0x00 (ADD)
  PASS: Name is ADD
  PASS: funct7 = 0x20 (SUB)
  PASS: Name is SUB

=== Testing I-type ===
  PASS: Type is I
  PASS: rd = 1
  PASS: rs1 = 0
  PASS: imm = 5
  PASS: Name is ADDI
  FAIL: Negative immediate = -1
  PASS: Load opcode
  PASS: Name is LW

=== Testing S-type ===
  PASS: Type is S
  PASS: rs1 (base) = 5
  PASS: rs2 (source) = 3
  PASS: imm (offset) = 0
  PASS: Name is SW

=== Testing B-type ===
  PASS: Type is B
  PASS: rs1 = 3
  PASS: rs2 = 4
  PASS: Branch offset = 8
  PASS: Name is BEQ

=== Testing U-type ===
  PASS: Type is U
  PASS: rd = 5
  PASS: Upper immediate = 0x00010000
  PASS: Name is LUI

=== Testing J-type ===
  PASS: Type is J
  PASS: rd = 0
  PASS: Jump offset = 0
  PASS: Name is JAL

=== Testing All Instruction Names ===
  PASS: 0x002081B3 -> ADD
  PASS: 0x40110233 -> SUB
  PASS: 0x00500093 -> ADDI
  PASS: 0x0032A023 -> SW
  PASS: 0x0002A203 -> LW
  PASS: 0x00418463 -> BEQ
  PASS: 0x000102B7 -> LUI
  PASS: 0x0000006F -> JAL
```




Test_integration.py

```
PS C:\Users\richa\440\The-Default-CPU-Design-and-Simulation-Project> python3 test_integration.py
============================================================
Integration Tests - Component Interaction
============================================================

=== Testing ALU + Register Integration ===
x1 = 5
x2 = 10
x3 = x1 + x2 = 15 (expected 15)
PASS: ALU + Register integration working!

=== Testing Register + Memory Integration ===
Stored 0xDEADBEEF to memory at 0x00010000
Loaded back to x5 = 0xDEADBEEF
PASS: Register + Memory integration working!

=== Testing Decoder Integration ===
Instruction: 0x02A00093
Decoded: ADDI
rd=1, rs1=0, imm=42
Result: x1 = 42 (expected 42)
PASS: Decoder integration working!

=== Testing Full Pipeline Simulation ===
ADDI: x1 = 5
ADDI: x2 = 10
ADD: x3 = 15
SW: mem[0x00000000] = 15
LW: x4 = 15
PASS: Full pipeline simulation working!

=== Testing Hex File Loader Integration ===
Created test file: test_integration.hex
Loaded 3 instructions from test_integration.hex
✓ Successfully loaded 3 instructions
First instruction: 0x00500093

=== Testing Branch Simulation ===
x1 = 10, x2 = 10
BEQ should be taken: True
PASS: Branch condition correct!

=== Testing Memory Alignment ===
Wrote 0x12345678 to 0x1000
Read from 0x1000: 0x12345678
Read from 0x1002: 0x12345678 (should be same, auto-aligned)
PASS: Memory alignment working!

=== Testing Register x0 Hardwired to Zero ===
Tried to write 0xFFFFFFFF to x0
Read back: 0x00000000 (should be 0x00000000)
PASS: x0 hardwired to zero!

============================================================
Results: 8/8 test suites passed
============================================================
✓ All integration tests passed!
PS C:\Users\richa\440\The-Default-CPU-Design-and-Simulation-Project>
```

CPU.py

```
PS C:\Users\richa\440\The-Default-CPU-Design-and-Simulation-Project> python3 cpu.py test_base.hex
Loading: test_base.hex
Loaded 11 instructions from test_base.hex
Loaded 11 instructions

Starting execution...
PC = 0x00000000

[0] PC=0x00000000 | 00500093 | ADDI
[1] PC=0x00000004 | 00A00113 | ADDI
[2] PC=0x00000008 | 002081B3 | ADD
[3] PC=0x0000000C | 40110233 | SUB
[4] PC=0x00000010 | 00010237 | LUI
[5] PC=0x00000014 | 0032A023 | SW
[6] PC=0x00000018 | 0002A203 | LW
[7] PC=0x0000001C | 00418463 | BEQ
[8] PC=0x00000024 | 00200313 | ADDI
Halt detected at cycle 9

Finished after 9 cycles

============================================================
FINAL STATE
============================================================
Cycles: 9
Final PC: 0x00000028

=== Register File ===
x 0-x 3: 0x00000000 0x00000005 0x0000000A 0x0000000F
x 4-x 7: 0x0000000F 0x00010000 0x00000002 0x00000000
x 8-x11: 0x00000000 0x00000000 0x00000000 0x00000000
x12-x15: 0x00000000 0x00000000 0x00000000 0x00000000
x16-x19: 0x00000000 0x00000000 0x00000000 0x00000000
x20-x23: 0x00000000 0x00000000 0x00000000 0x00000000
x24-x27: 0x00000000 0x00000000 0x00000000 0x00000000
x28-x31: 0x00000000 0x00000000 0x00000000 0x00000000

Memory (non-zero):
  [0x00000000] = 0x00500093 (5243027)
  [0x00000004] = 0x00A00113 (10486035)
  [0x00000008] = 0x002081B3 (2130355)
  [0x0000000C] = 0x40110233 (1074856499)
  [0x00000010] = 0x00010237 (66231)
  [0x00000014] = 0x0032A023 (3317795)
  [0x00000018] = 0x0002A203 (172547)
  [0x0000001C] = 0x00418463 (4293731)
  [0x00000020] = 0x00100313 (1049363)
  [0x00000024] = 0x00200313 (2097939)
  [0x00000028] = 0x0000006F (111)
  [0x00010000] = 0x0000000F (15)
============================================================
PS C:\Users\richa\440\The-Default-CPU-Design-and-Simulation-Project> git status
```