**Course: Computer Organization – ENCM 369  
Lab #:** Lab 5  
**Instructor Name:** Norm Bartley **Student Name:** Stephen Ravelo **Lab Section:** B03 **Date submitted:** February 28, 2025

# Exercise A

|  |  |
| --- | --- |
| **Instruction** | **Change** |
| srli t1, t0, 6 | 0000\_1110\_0110\_1011\_0000\_0000\_0000\_0000  shift 6 bits right  0000\_0000\_0011\_1001\_1010\_1100\_0000\_0000  t1 = 0x0039\_ac00 |
| slli t2, t0, 2 | 0000 \_1110\_0110\_1011\_0000\_0000\_0000\_0000  shift 2 bits left  0011\_1001\_1010\_1100\_0000\_0000\_0000\_0000  t2 = 0x39ac\_0000 |
| or t3, t5, t6 | {20 zeroes} \_1100\_0101\_0000  or {20 zeroes} \_1010\_0011\_0000  {20 zeroes} \_1110\_0111\_0000  t3 = 0x0000\_0e70 |
| andi t4, t5, 0x3ff | {20 zeroes} \_1100\_0101\_0000  and {20 zeroes} \_0011\_1111\_1111  {20 zeroes} \_0000\_0101\_0000  t4 = 0x0000\_0050 |
| xor s0, t5, t6 | {20 zeroes} \_1100\_0101\_0000  xor {20 zeroes} \_1010\_0011\_0000  {20 zeroes}\_0110\_0110\_0000  s0 = 0x0000\_0660 |
| xori s1, t5, -32 | {20 zeroes} \_1100\_0101\_0000  xor {20 ones} \_1111\_1110\_0000  {20 ones} \_0011\_1011\_0000  s1 = 0xffff\_f3b0 |

# Exercise B

## Part 1

lw t1, alpha

* auipc x6, 0x0000fc10

The address of the instruction is 0x0040\_0000, so t1 would get a value of

0x0040\_0000 + 0x0fc1\_0000 = 0x1001\_0000

## Part 2

alpha is stored at 0x1001\_0000 + 0x18 = 0x1001\_0018 (24 bytes ahead)

## Part 3

sw t2, beta, t3

* auipc x28, 0x0000fc10

The address of the instruction is 0x0040\_000c, so t1 would get a value of

0x0040\_000c + 0x0fc1\_0000 = 0x1001\_000c

## Part 4

beta is stored at 0x1001\_000c + 0x10 = 0x1001\_001c (16 bytes ahead)

# Exercise C

|  |  |
| --- | --- |
| **Instruction** | **Machine code** |
| beq t6, zero, L2 | B-Type   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | imm12,10:5 | rs2 | rs1 | funct3 | imm4:1,11 | op |   imm12:0 = 0x0040\_00ec - 0x0040\_0084  = 0x68  = 104  = 0\_0000\_0110\_0000   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 0000\_011 | 0\_0000 | 1111\_1 | 000 | 0000\_0 | 110\_0011 |   0x060f\_8063 |
| j L1 | jal x0, L1  J-Type   |  |  |  | | --- | --- | --- | | imm20,10:1,11,19:12 | rd | op |   imm20:0 = 0x0040\_0080 - 0x0040\_00e8  = 0xffff\_ff98  = -104  = 1\_1111\_1111\_1111\_1001\_1000   |  |  |  | | --- | --- | --- | | 1111\_1001\_1001\_1111\_1111 | 0000\_0 | 110\_1111 |   0xf99f\_f06f |

# Exercise D

## Part 1

Estimated time for use\_pointers: 317000 ns

Estimated time for use\_indexes: 320000

## Part 2

### use\_pointers:

### use\_indexes:

Applying complier optimization generates a much larger speedup compared to using pointers over indexes.

# Exercise E

## Part 2

ldr     w10, [x8], *#4*

cmp w10, w0

csel     w0, w10, w0, gt

cmp x8, x9

b.lo     LBB0\_2

## Part 3

ldr w0, x[0]

* 0xb940\_0000

## Part 4

for (i = 0; i < N\_ELEMENT; i++)

* for (*int* i = 0; i < ((*int*) 1e5); i++)

add x8, x8, #1

* 0x91000508