HW2

- In RISCV, only load and store instructions access memory locations
- These instructions must follow a 'format' to access memory
- Assume a 32-bit machine in all problems unless asked to assume otherwise

Problem 1

Assume address in memory of 'A[0]', 'B[0]' and 'C[0]' are stored in Registers x27, x30, x31. Assume values of variables f, g, h, i, and j are assigned to registers x5, x6, x7, x28, x29 respectively

Write down RISC V Instruction(s) to

1. Load Register x5 with content of A[10]
2. Store contents of Register x5 into A[17]
3. add 2 operands: one in x5 - a register, the other in in Register x6. Assume result of operation to be stored in register x7
4. copy contents at one memory location to another: C[g] = A[i+j+31]
5. implement in RISC V these line of code in C:
    - f = g - A[B[9]]
    - f = g - A[C[8] + B[4]]
    - A[i] = B[2i+1], C[i] = B[2i]
    - A[i] = 4B[i-1] + 4C[i+1]
    - f = g - A[C[4] + B[12]]

Problem 2

Assume the following register contents:

x5 = 0x00000000AAAAAAAA, x6 = 0x1234567812345678

1. For the register values shown above, what is the value of x7 for the following sequence of instructions?

   srli x7, x5, 16

   addi x7, x7, -128

   srai x7, x7, 2

   and x7, x7, x6

2. For the register values shown above, what is the value of x7 for the following sequence of instructions?

   slli x7, x6, 4

3. For the register values shown above, what is the value of x7 for the following sequence of instructions?

   srli x7, x5, 3

andi x7, x7, 0xFEF

Problem 3

For each RISC-V instruction below, identify the instruction format and show, wherever applicable, the value of the opcode (op), source register (rs1), source register (rs2), destination register (rd), immediate (imm), func3, func7 fields. Also provide the 8 hex char (or 32 bit) instruction for each of the instructions below

add x5, x6, x7

addi x8, x5, 512

ld x3, 128(x27)

sd x3, 256(x28)

beq x5, x6 ELSE #ELSE is the label of an instruction 16 bytes larger

#than the current content of PC

add x3, x0, x0

auipc x3, FFEFA

jal x3 ELSE

Problem 4

1. For the following C statement, write a minimal sequence of RISC-V assembly instructions that performs the identical operation. Assume x5 = A, and x11 is the base address of C.

   A = C[0] << 16;

2. Find the shortest sequence of RISC-V instructions that extracts bits 12 down to 7 from register x3 and uses the value of this field to replace bits 28 down to 23 in register x4 without changing the other bits of registers x3 or x4. (Be sure to test your code using x3 = 0 and x4 = 0xffffffffffffffff. Doing so may reveal a common oversight.)

3. Provide a minimal set of RISC-V instructions that may be used to implement the following pseudoinstruction:

   not x5, x6 // bit-wise invert

   [Hint: note that there is no 'not' instruction in RISCV. However, an XOR immediate instruction could be used]

Problem 5

Suppose the program counter (PC) is set to $0x60000000_{\text{hex}}$.

1. What range of addresses can be reached using the RISC-V *jump-and-link* (jal) instruction? (In other words, what is the set of possible values for the PC after the jump instruction executes?)
2. What range of addresses can be reached using the RISC-V *branch if equal* (beq) instruction? (In other words, what is the set of possible values for the PC after the branch instruction

executes?)

Problem 6

Assume that the register x6 is initialized to the value 10. What is the final value in register x5 assuming the x5 is initially zero?

```
LOOP:   beq x6, x0, DONE
        addi x6, x6, -1
        addi x5, x5, 2
        jal x0, LOOP
DONE:
```

1. For the loop above, write the equivalent C code. Assume that the registers x5 and x6 are integers acc and i, respectively.
2. For the loop written in RISC-V assembly above, assume that the register x6 is initialized to the value N. How many RISC-V instructions are executed?
3. For the loop written in RISC-V assembly above, replace the instruction "beq x6, x0, DONE" with the instruction "blt x6, x0, DONE" and write the equivalent C code.

Problem 7

1. Translate the following C code to RISC-V assembly code. Use a minimum number of instructions. Assume that the values of a, b, i, and j are in registers x5, x6, x7, and x29, respectively. Also, assume that register x10 holds the base address of the array D.

```
for(i=0; i < a; i++)
    for(j=0; j < b; j++)
        D[4 * j]= i + j;
```

2. How many RISC-V instructions does it take to implement the C code from 7a. above? If the variables a and b are initialized to 10 and 1 and all elements of D are initially 0, what is the total number of RISC-V instructions executed to complete the loop?

Problem 8

Consider the following code:

```
lb x6, 0(x7)
sd x6, 8(x7)
```

Assume that the register x7 contains the address **0×10000000** and the data at address is 0×1122334455667788.

1. What value is stored in 0×10000007 on a big-endian machine?
2. What value is stored in 0×10000007 on a little-endian machine?

Problem 9

Write the RISC-V assembly code that creates the 64-bit constant $0x1234567812345678_{\text{hex}}$ and stores that value to register x10.

Problem 10

Assume that x5 holds the value 128_{10}.

1. For the instruction add x30, x5, x6, what is the range(s) of values for x6 that would result in overflow?
2. For the instruction sub x30, x5, x6, what is the range(s) of values for x6 that would result in overflow?
3. For the instruction sub x30, x6, x5, what is the range(s) of values for x6 that would result in overflow?