Computer Architecture Midterm Exam

---

2. Arithmetic for Computer (25%)

2.1 (5%)

What is the binary value of -375 (using 2-s complement).

A: ____________________

2.2 (5%)

What is the binary bit pattern representation of 7.640625 (using double precision).

A: ____________________

2.3 (15%) Floating arithmetic

Calculate 2.78125 + 0.4375 by hand, assume 1 guard, 1 round bit and 1 sticky bit, and round to the nearest even. Show all the steps, write your answers in single precision floating point format, show all the necessary steps.

---

3. Instruction (35%)

3.1 (5%)

Translate the following RISC-V assembly code into the binary and the hex: `jal x1, 200`

A: ____________________

3.2 (15%)

The following code fragment processes two arrays, A and B, and produces an outcome value stored in register `t0`. Assume that each array consists of 2500 words with values ranging from 0 to 2499. The base addresses of A and B are stored in registers `a0` and `a1`, respectively, and their sizes (2500) are stored in registers `a2` and `a3`. **Translate this code fragment into C code.** Specifically, what value will be returned in `t0`?

```
      slli a2, a2, 2
      slli a3, a3, 2
      add  t0, zero, zero
      add  t1, zero, zero
outer:
      add  t4, a0, t1
      lw   t4, 0(t4)
      add  t2, zero, zero
inner:
      add  t3, a1, t2
      lw   t3, 0(t3)
      bne  t3, t4, skip
      addi t0, t0, 1
skip:
      addi t2, t2, 4
      bne  t2, a3, inner
      addi t1, t1, 4
      bne  t1, a2, outer
```

(1) Translation:

(2) The value of t0 = ____________

3.3 (15%)

Implement the following C code in RISC-V. Partial assembly code is provided. Complete the remaining code in the boxes. The modulo operation can be implemented in two ways: by combining multiplication and division instructions, or by directly using the modulo instruction `rem`:

`rem rd, rs1, rs2` (comment: rd=rs1%rs2)

For readability, use register names like `zero`, `ra`, `sp`, `t0`, `a0`, etc., instead of `x0`, `x1`, `x2`, `x5`, `x10`, etc.

C Code:

```c
int gcd(int a, int b) {
    if (b == 0) return a;
    return gcd(b, a % b);
}

int main() {
    int num1 = gcd(10, 5);
    return 0;
}
```

RISC-V Assembly Code:

```
MAIN:
      addi sp, sp, -16      # stack space
      sw   ra, 12(sp)       # return address
      addi a0, zero, 10     # parameter 1 for gcd
      addi a1, zero, 5      # parameter 2 for gcd
      jal  ra, GCD          # call GCD
      add  t0, a0, zero     # save GCD result
      addi a0, zero, 0      # return value of MAIN
      lw   ra, 12(sp)       # restore return address
      addi sp, sp, 16       # deallocate stack space
      jalr zero, 0(ra)      # return

GCD:
      #
      # (Your code here)
      #
      # if b!=0, go to ELSE
      #
      #
      #

ELSE:
```

```
        #
        # (Your code here)
        #
        #
        #
```

---

4. Single cycle processor (20%)

(Note: The following questions refer to a diagram of a single-cycle processor datapath.)

4.1 (10%)

Suppose you are adding "jalr" instruction to a single cycle processor shown above, modify the figure and add any necessary datapaths and control signals.

4.2 (5%)

Complete the table below. You can add columns if necessary.

INSTRUCTION	ALUSRC	MEMTOREG	REGWRITE	MEMREAD	MEMWRITE	BRANCH	ALUOP1	ALUOP0
R-format	0	0	1	0	0	0	1	0
lw	1	1	1	1	0	0	0	0
sw	1	X	0	0	1	0	0	0
beq	0	X	0	0	0	1	0	1
jalr								

4.3 (5%)

Consider the initial single-cycle datapath (before adding `jalr` instruction). Can we use other existing control signal(s) to implement the `MemtoReg` control signal? Please explain the reason.