Due Date: April 27, 2004 at the beginning of class

Deliverables: Turn in answers to each part, with each section clearly marked and appropriate work shown on a sheet of paper. A copy of the assignment need not be attached.

Part 1: Floating Point Representation

Encode the following four numbers/expressions into Single Precision IEEE 754 Floating-Point format. Be sure to round as needed. Partition the bits to show each field of the Floating-Point format.

- A.) 15.2
- B.) -726.15
- C.) \((2^{24} + 3)\)
- D.) \(+\infty\)

Part 2: Code on different types of architecture

Using the following function:

\[
A = \frac{(B + C)^3 \cdot (D - C)}{B \cdot C \cdot D}
\]

Code this function for:

- A generic stack architecture
  - MULtiply, DIVide (first operand popped is the denominator), ADDition, SUBtraction are the available arithmetic operations, DUPlicate creates another copy of the top operand and puts it on top of the stack.

- A generic accumulator architecture
  - MULtiply, DIVide, ADDition, SUBtraction are the available arithmetic operations

- A generic register-register architecture (not MIPS).
  - MULtiply, DIVide, ADDition, SUBtraction are the available arithmetic operations
Part 3: MIPS Architecture

[0x0040A030]  add $1, $0, $0  // int x
[0x0040A034]  lw $2, 80($10)  // int a used in your C code
[0x0040A038]  loop: slti $3, $1, 25
[0x0040A03C]  beq $3, $0, end
[0x0040A040]  addi $4, $2, 45
[0x0040A044]  add $5, $4, $4
[0x0040A048]  lw $6, 0($11)  // int b used in your C code
[0x0040A04C]  sub $7, $5, $6
[0x0040A050]  sw $7, 0($11)
[0x0040A054]  addi $11, $11, 4
[0x0040A058]  addi $1, $1, 1
[0x0040A05C]  j loop
[0x0040A060]  end:  addi $8, $2, 100
[0x0040A064]  sw $8, 80($10)

Use this code from a portion of a program above, to answer the questions below.

- Encode the instruction at address [0x0040A05C] into a standard binary MIPS format. Identify the format and addressing mode(s) used by this instruction.

- Encode the instruction at address [0x0040A048] into a standard binary MIPS format. Identify the format and addressing mode(s) used by this instruction.

- Encode the instruction at address [0x0040A03C] into a standard binary MIPS format. Identify the format and addressing mode(s) used by this instruction.

- Encode the instruction at address [0x0040A04C] into a standard binary MIPS format. Identify the format and addressing mode(s) used by this instruction.

- Write a C equivalent for the code above.