QUESTION #1 (15 minutes: 4 points for each, 40 points total)

#1: What decimal number is represented by “11000000111000000000000000000011” in IEEE-754?

(a) \((2^{129} + 2^{128} + 2^{108} + 2^{107})\)

(b) \(-(2^{129} + 2^{128} + 2^{108} + 2^{107})\)

(c) \((2^2 + 2^1 + 2^{-19} + 2^{-20})\)

(d) \(-(2^2 + 2^1 + 2^{-19} + 2^{-20})\)

(e) None of the above
#2: Dynamic code optimizations usually perform relatively simple optimization algorithms. Why? Select all that apply.

(a) It is because performing complex optimizations can make compile-time longer.

(b) It is because performing complex optimizations can make the programs much harder to debug and/or understood by human programmers.

(c) It is because performing complex optimizations can make the execution time longer.

(d) It is because performing complex optimizations will not be effective to older processors.

(e) It is because performing complex optimizations will increase the risk of the hardware bugs.

(f) None of the above.

#3: When the two different integer data types of “the unsigned integers” and “the two’s complement integers” are used in a computer system, which of the followings is (are) correct? Select all that apply.

(a) You can never mix them (you must not copy the contents of a variable in “the unsigned integers” to “the two’s complement integers”) and vice versa.

(b) If you are sure that the possible values of the integers are in specific range, you can copy the contents of a variable in “unsigned integer” to “two’s complement” (and the results are still correct).

(c) If you are sure that the possible values of the integers are in specific range, you can copy the contents of a variable in “two’s complement” to “unsigned integer” (and the results are still correct).

(d) You can always mix them (what the term “mix” means is same as (a)).

(e) None of the above.
#4: What is the relation between “statements” in high-level programming languages and “instructions” in assembly languages? Select the best option that represents their relation in the following options.

(a) one (statement)-to-one (instruction)
(b) one (statement)-to-many (instructions)
(c) many (statements) to-one (instruction)
(d) many (statements)-to-many (instructions)
(e) none of the above

#5: Which of the following formula is best for the one that calculates “the speed-up factor of a super-pipeline processor to super scalar processor”? Select the best option.

**Assume:**
- $n$: the number of the instructions to be executed
- $k$: the depth of a datapath (number of the datapath stages)
- $c$: clock cycle time
- $p$: the number of datapath

(a) $(p \times (n + k - 1) \times c) / (p \times n \times k \times c)$
(b) $(p \times n \times k \times c) / (p \times (n + k - 1) \times c)$
(c) $(n/p \times k \times c) / ((n/p + k - 1) \times c)$
(d) $((n/p + k - 1) \times c) / (n/p \times k \times c)$
(e) None of the above
#6: What is the difference between “li $a0, 1024” and “la $a0, 1024” instructions? Assume that this computer system is a 32-bit system (i.e., all the registers are 32-bit registers and its ALU can deal with up to 32-bit inputs and outputs).

**Note:** This question is NOT a multiple-choice question. Please type your solution below. Long solutions are NOT expected (no more than 30 words).

#7: What is (are) the benefit(s) of using floating-point numbers in computer systems? Select all that apply.

(a) Large (huge) integers can be accurately represented.
(b) Tiny fraction numbers can be accurately represented.
(c) Processors can handle integers quickly.
(d) Processors can handle fraction numbers quickly.
(e) None of the above.

#8: What is the binary bit pattern of -5_{10} in the bias of 127? Assume that the bit pattern consists of 8 bits. **Show your work.**

**Note:** This question is NOT a multi-choice question.

#9: What is the advantage of using “normalized” format in IEEE-754 standard?

**Note:** This question is NOT a multi-choice question.
#10: For the following statements, select the correct statement(s) (select all that apply).

(a) Since the difference between the largest and the second largest positive number is $2^{104}$, the difference between the tenth and the eleventh largest positive number is $2^{94}$.

(b) Since the difference between the tiniest and the second tiniest positive number is $2^{-149}$, the difference between the tenth and the eleventh tiniest positive number is $2^{-139}$.

(c) Since the difference between the tiniest and the second tiniest negative number is $2^{-149}$, the difference between the tenth and the eleventh tiniest negative number is $2^{-139}$.

(d) Since the difference between the smallest and the second smallest negative number is $2^{104}$, the difference between the tenth and the eleventh smallest negative number is also $2^{104}$.

(e) None of the above.
Our favorite program ("Program X") runs in (exactly) 8.8 seconds on processor A, which has a 1.52 GHz clock (its clock cycle rate = 1.52 GHz). A processor designer is trying to build another processor, processor B, which will run this program in (exactly) 6.1 seconds. The designer has determined that processor will achieve the goal (i.e., run the program in 6.1 seconds) by a substantial increase in the clock rate. However, the substantial increase in the clock rate will affect the rest of the processor’s design, causing processor B to require 1.32 times as many clock cycles as processor A to execute Program X.

**Question:** What clock rate should we tell the designer target for processor B to run the program in 6.1 seconds? Select the best from those listed below. **Show the formula you developed for your solution.**

(a) Less than 1.89 GHz  
(b) 1.89 GHz  
(c) 2.09 GHz  
(d) 2.29 GHz  
(e) 2.49 GHz  
(f) 2.69 GHz  
(g) 2.89 GHz  
(h) 3.09 GHz  
(i) 3.29 GHz  
(j) 3.49 GHz  
(k) Over 3.5 GHz  
(l) It’s impossible to achieve

**Note 1:** Showing your formula is required (although completing your calculation is NOT required).

**Note 2:** Showing a correct formula is for 85% of the credit for this question (the remaining 15% is for selecting the correct option above).
QUESTION #3 (10 minutes) – 20 points

Suppose that we have a $k = 20$ (i.e., “a 20-stage”) pipeline processor (not “a super-pipeline” though). If data dependencies should not slow down this pipeline processor more than 17.5% for a program (“program A”), how often data hazards can happen in programs?

Assume the followings:

(a) Input to (read from) a register will happen at the end of the 4th stage in the 20-stage pipeline.
(b) The first three stages in the 20-stage pipeline works without inputting the content of the common register.
(c) Output (write) to a register will happen at the end of the 20th stage in the 20-stage pipeline.
(d) Assume that every instruction needs all the 20 stages (this is for simplifying your work).
(e) Ignore any other overhead (including control hazards).

**Question**: Answer “how often” in the percentage. For example, if data hazards happen five times in 100 instructions, it should be counted as “5%”.

**Note**: You will earn full credit by showing the correct formula (you do not have to finish your calculation for full credit). Partial credit will be given to show your work.
Without data dependency

Execution Time = \((n + k -1) \times T\)

- \(n\) = the number of the instructions executed
- \(k\) = the depth of a pipeline
- \(T\) = the clock-cycle time

With data dependency

Execution Time = \((n + k -1 + a) \times T\)

where \(a = n \times x \times 16\)

\[
\frac{\text{Execution Time} = (n + k -1 + n \times x \times 16) \times T}{\text{Execution Time} = (n + k -1) \times T} \leq 1.175
\]
QUESTION #4 (10 minutes) – 20 points

What is the interval (= “the absolute difference”) between the 33,499,004th and the 33,449,005th tiniest negative numbers in IEEE-754?

Show your work for full credit (at least as much detail to allow Dr. Fujinoki to follow your idea(s)).

**Note 1**: Failing to show how you reach the conclusion will not let you earn the full credit (only 5/20 points if you do not show your work or your intermediate work is not correct.

**Note 2**: Partial credit will be given to any meaningful intermediate work.

**Solution**: $2^{-146}$

<table>
<thead>
<tr>
<th>Sets of $2^{23}$</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
<th>Difference (“interval”)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1st tiniest (= the tiniest)</td>
<td>$2^{23}$th tiniest (= 8,388,608th tiniest)</td>
<td>$2^{-149}$</td>
</tr>
<tr>
<td>2</td>
<td>($2^{23} + 1$)th tiniest</td>
<td>$2 \times 2^{23} = 2^{24}$ = 16,777,216th tiniest</td>
<td>$2^{-148}$</td>
</tr>
<tr>
<td>3</td>
<td>($2^{24} + 1$)th tiniest</td>
<td>$2 \times 2^{24} = 2^{25}$ = 25,165,824th tiniest</td>
<td>$2^{-147}$</td>
</tr>
<tr>
<td>4</td>
<td>($2^{25} + 1$)th tiniest</td>
<td>$2 \times 2^{25} = 2^{26}$ = 33,554,432th tiniest</td>
<td>$2^{-146}$</td>
</tr>
<tr>
<td>5</td>
<td>($2^{26} + 1$)th tiniest</td>
<td>$2 \times 2^{26} = 2^{27}$ = 41,943,040th tiniest</td>
<td>$2^{-145}$</td>
</tr>
</tbody>
</table>