COMP4300
Computer Architecture, Spring 2022
Midterm Examination

1. (10 points) Suppose that a given machine spends 1/3 of its execution time before optimization doing jump instructions. The design team for the manufacturer optimized the calculation of target addresses by jump instructions. If the optimization resulted in the speedup of processing jump instructions by a factor of 4, what was the overall execution time after optimization?

$$\text{New Execution Time} = \left[\left(1-\tfrac{1}{3}\right) + \frac{1/3}{4}\right](\text{Old Execution Time})$$

$$= \left[\tfrac{2}{3} + \tfrac{1}{12}\right](\text{OET})$$

$$= \left(\tfrac{8}{12} + \tfrac{1}{12}\right)\text{OET}$$

$$\boxed{= \tfrac{9}{12}\,\text{OET} = \tfrac{3}{4}\,\text{OET} = 0.75\,\text{OET}}$$

2. (7 points) What is the two's complement 12-bit representation of negative 5?

0000 0000 0101 = 12-bit positive 5
1111 1111 1010 = 12-bit 1's compl neg. 5
1111 1111 1011 = 12-bit 2's compl neg. 5

3. (13 points) The following PDP-8 machine language program loads the contents of address 0x400 into the accumulator, subtracts 5 from it, and stores it back to address 0x400. Use address 0x711 to store the address 0x400. Fill in the blanks in the binary. See the pdp-8 programming card in the files section of Canvas for opcodes. Note that all numbers on the PDP-8 programming card are in *octal,* while numbers in the question and below are either hexadecimal (if prefixed by 0x) or binary (if made up of zeroes and ones). Each blank is one binary digit.

or 111 010 000 000
or 111 110 000 000

Addr	Mnemonic	Binary
0x700	CLA	111 100 000 000
0x701	TAD I 0x711	0011 10 010 001
0x70_	TAD 0x712	001 0 1 001 0010
0x70_	DCA I 0x711	0111 10 010 001
0x70_	HLT	111 100 000 010
0x711	(addr stored here)	010 000 000 000
0x712	(data stored here) -5	111 111 111 011

C

4. (7 points) Is code size of a RISC machine typically larger or smaller than with a CISC machine? Why?

RISC code size is larger b/c RISC instructions are simpler and do less. Often more than one is necessary to accomplish the same work as a CISC instruction.

5. (7 points) What computer did the term "computer architecture" first apply to?

IBM System/360

6. (7 points) Is 80x86 a RISC or CISC design? Why?

CISC: It has register-memory instructions.

7. (8 points) In the following VHDL process, if input A changes at time 20nS and no other inputs change after that time , at what time will all the output signals be *guaranteed* to have assumed their final value, regardless of their initial values?

```
foo: process(A,B,C) is
begin
  if A = '1' then
      X <='1' after 15 ns;
  end if;
  Z <= '1' after 5 ns;
  W <= '0' after 30 ns;
end
```

must take into account possibility that A = '1', but it doesn't affect the answer

20+30=50nS – all times in "after" clauses are relative to current simulation time

8. (5 points) Suppose in the code below that signal Z changes at 20ns and no other signal changes after that time. When will the output signals of process foo have their final values?

```
foo: process(A,B,C) is
begin
  if A = '1' then
      X <='1' after 15 ns;
  end if;
  Z <= '1' after 5 ns;
  W <= '0' after 30 ns;
end
```

20nS – not sensitive to change in Z

9. (7 points) Consider the following computer system, which has a load-store architecture. What is the average CPI?

-ALU operations take 1 cycle and make up 30% of the dynamic instruction count
-Floating point instructions are 4 cycles, and make up 7%
-Branches take 5 cycles and make up 25%
-Loads and stores are 3 cycles and make up the rest of the instructions

$$\text{AVG CPI} = \sum_{i \in \text{instr}} \text{cycles}_i \cdot \text{fraction}_i$$

$$\text{AVG CPI} = 1 \times 0.3 + 4 \times 0.07 + 5 \times 0.25 + 3 \times (1 - 0.3 - 0.07 - 0.25)$$

$$= 0.3 + 0.28 \times 1.25 + 3 \times 0.38$$

$$\boxed{= 2.97 \text{ cycles}}$$

10. (10 points) For the machine of problem 9 (copied below), optimization A reduces cycle time to 0.8 of the original cycle time (cycle time must be the same for all instructions) and speeds up branches by 1 cycle. Optimization B speeds up floating point operations by 1 cycle but lengthens the original cycle time by 5%. Which optimization improves performance more?

Machine from problem 9:

-ALU operations take 1 cycle and make up 30% of the dynamic instruction count
-Floating point instructions are 4 cycles, and make up 7%
-Branches take 5 cycles and make up 25%
-Loads and stores are 3 cycles and make up the rest of the instructions

$CPI_A = 2.97 - 1 \times 0.25 = 2.72$ cycles

$CPI_B = 2.97 - 1 \times 0.07 = 2.91$ cycles

Faster → $\boxed{\text{Speedup}_A} = \dfrac{\cancel{CT} \times \cancel{IC} \times 2.97}{0.8\cancel{CT} \times \cancel{IC} \times 2.72} = \dfrac{2.97}{0.8 \times 2.72} = 1.36$

$\text{Speedup}_B = \dfrac{\cancel{CT} \times \cancel{IC} \times 2.97}{1.05\cancel{CT} \times \cancel{IC} \times 2.91} = 0.97$

11. (10 points) Of the following architectures we have discussed, which one is likely to have the most complicated instruction decode mechanism, and why? PDP-8, RISC-V, or 8080?

8080 b/c variable length instructions, many addressing modes, register-memory instructions
(RISC-V has 4 modes, PDP-8 has 1)

12. (9 points) Suppose a given computer has a power supply that must be cooled by a separate fan. The computer fails if either the power supply or its fan fails. The MTTF of the fan is 1 year. The MTTF to failure of the power supply is 5 years. What is the MTTF of the computer due to the power supply/fan system?

$$prob_{sys} = \frac{1}{MTTF_{ps}} + \frac{1}{MTTF_{fan}}$$

$$= \frac{1}{5\,yr} + \frac{1}{1\,yr}$$

$$= 0.2\,yr^{-1} + 1\,yr^{-1} = 1.2\,yr^{-1}$$

$$MTTF_{sys} = \frac{1}{prob_{sys}} = \frac{1}{1.2\,yr^{-1}} \boxed{= 0.83\,yr = \frac{5}{6}\,yr}$$