~~R~~ESERVED 1.—2.—3.—4.—5. −10

#8

Tuesday, October 30, 2018 starting at 5:45 p.m. H-433

NAME: 90 ID:

** IMPORTANT NOTE ** : To receive credit for any question, the answers must be copied into the answer spaces provided. However, you may use any white space anywhere in the exam for your scratch work. Answers left as scratch will not be graded.

Also, please do not detach any sheets from this booklet.

Special Rules

1. Only pocket calculators and writing materials (pen or pencil) are allowed. You may not borrow someone else's calculator. Bring a calculator.
2. No books, notes, scratch paper, or other electronics, etc., are allowed.
3. Examination booklet must be returned.
4. Coats, book bags, backpacks, etc., must be stowed against the wall.
5. Cell phones must be powered down (i.e., turned off) and stored with your belongings against the wall.
6. Students must present valid ID. Normally, this is Concordia ID. In unusual cases, a drivers license or passport will be accepted.

1. [20 marks] Digital Logic.

a) 'X' is a ternary connective. 'Xpqr' is true iff 'p' and 'r' are false and 'q' is true. 'F' and 'T' have their usual meanings.

Using {'X', 'F', 'T'}, synthesize: ~p |= =| X p T p

Using {'X', '~', 'T'}, synthesize: p /\ q |= =| X~F p ~q

b) 'p <--> q' is true iff 'p' and 'q' have the same truth value. 'Z' is a ternary connective. We have: Zpqr |= =| p /\ (q <--> r). 'F' and 'T' have their usual meanings.

Using {'Z', 'F', 'T'}, synthesize: ~p |= =| Z T p F

Using {'Z', 'F', 'T'}, synthesize: p /\ q |= =| Z p q T

2. [20 marks] Amdahl's Law.

a) On a uniprocessor, somewhat parallel portion A of program P consumes 33 s, while perfectly parallel portion B consumes 113 s. On a parallel computer, portion A speeds up 4x, while portion B speeds up by the number of processors. Given 1,000 processors, what is the speedup on program P? Two decimal places of accuracy are requested.

su = 17.46 times

b) On a uniprocessor, somewhat parallel portion A of program P consumes 33 s, while perfectly parallel portion B consumes 't' s. On a parallel computer, portion A speeds up 4x, while portion B speeds up by the number of processors. How large must 't' be so that using 1,000 processors gives a speedup of at least 125 times? Two decimal places of accuracy are requested.

t = ~~[illegible]~~ 1,140.86 s

3. [20 marks] Pipeline Information Flow.

```
+------------------------------------------------------------------------+ P
|                                                                        | r
|    <I-cache>              <Register file>               <D-cache>      | o
|                                                                        | c
|                           <Control circuitry>                          | e
|                  +-+                +-+            +-+           +-+   | s
|    <f-box>      | |    <d-box>     | |  <x-box>   | |  <m-box>  | |  <w-box>  s
|                  +-+                +-+            +-+           +-+   | o
|                  f/d                d/x            x/m           m/w   | r
+------------------------------------------------------------------------+
```

```
                     1  2  3  4  5  6  7  8  9  0  1  2
l.d    f2,0(r2)      f  d  xi m  w
mul.d  f4,f2,f0         f  d  s  x1 x2 x3 n  w
l.d    f6,0(r3)            f  s  d  xi m  w
add.d  f8,f4,f6                  f  d  s  x1 x2 x3 n  w
```

a) The x1-box in cycle 5 has two operands. One is "made" in cycle 4 by the m -box and the other is "made" in cycle 3 by the d -box.

b) The x1-box in cycle 8 has two operands. One is "made" in cycle 7 by the x3 -box and the other is "made" in cycle 7 by the m -box.

c) The w-box in cycle 9 has two operands. One is "made" in cycle 3 by the d -box and the other is "made" in cycle 7 by the x3 -box.

d) The m-box in cycle 7 has one operand. It is "made" in cycle 6 by he xi -box.

e) The xi-box in cycle 6 has two operands. One is "made" in cycle 5 by the d -box and the other is "made" in cycle 5 by the d -box.

Hex table:

0	0000	4	0100	8	1000	c	1100
1	0001	5	0101	9	1001	d	1101
2	0010	6	0110	a	1010	e	1110
3	0011	7	0111	b	1011	f	1111

Hex flips:

0 - f 4 - b
1 - e 5 - a
2 - d 6 - 9
3 - c 7 - 8

Hex powers:

1, 16, 256, 4096

Hex naturals:

a	b	c	d	e	f
10	11	12	13	14	15

4. [20 marks] Instructions with an Immediate.

a) Assume 32-bit registers, instructions, and memory addresses, and 16-bit immediates. Array 'a' has 128 floating-point elements. Instruction 'l.d f4,31750(r2)' loads the first element of 'a', viz., a[0]. Find offset 'x' such that 'l.d f6,x(r2)' loads the last element of 'a', viz., a[127].

(decimal integer) x = 32,766 ✓

'x' is then expanded to 32-bits before being added to 'r2'. So now,

(hexadecimal integer) x = 0 0 0 0 7 F F E ✓

b) In a VLIW computer, the compiler routinely packs sets of three 32-bit instructions into _instruction packets_. These executable packets are issued by the computer as indivisible units. Thus, 'PC' only points to packets, and packets are the only legal branch targets. Consider 'bne r1,r2,loop', where 'loop' has the value -1,573. What is the hexadecimal representation of the 32-bit integer that will be added to 'PC'?

ans: F F F F B 6 4 4 ✓

5. [20 marks] Fractional-Number Formats.

An application-specific computer has its own "mock" floating-point system. It is exactly like normal floating point except i) there is no sign bit, ii) the exponent is a 4-bit natural number, and iii) the fractional field is 16 bits long. There is no exponent bias, normalization is strictly enforced, and no values are singled out for special representation.

a) Write the infinite binary expansion of 1 2/9. Indicate any repeating sequence by placing a bar over it, (e.g., 1/3 = 0.3, bar over the 3).

pass ans: $(1.00111\overline{0})_2$ ←

b) Write the hexadecimal representation of the best mock floating-point approximation to 1 2/9. Do not round.

ans: 1 3 C F 3 ←

c) What is the largest number representable in mock floating point? For this question only, extend the fractional field to 20 bits.

ans: 1111.1111 1111 1111 1111 1111 ←

d) Let 'x' be the hexadecimal rational 13,739/(16^6). 'x' has been reduced to lowest terms and is clearly less than one. Our lemma tells us that six hexits suffice to represent the numerator. Write the finite hexadecimal expansion of 'x'.

ans: 0. 0 0 3 5 A B ✓

1/16 1/256 1/4096