CMPT 295: Instruction Set Design

1. **Instruction Set Architecture**
   An instruction set architecture for a CPU consists of 30 instructions. External storage consists of a $2^{16} \times 16$ memory unit, M. Internal storage is provided by a $2^3 \times 16$ register file, R.

   (a) If R[0] = 0x0800, R[1] = 0x0001, and M[0800] = 0x0001, and M[0000] = 0x0800, complete the syntax and semantics for each "ADD" instruction below that increments the contents of register R[4] by 1 by using the address mode specified. (Immediate mode is provided as an example):

	MODE	SYNTAX	SEMANTICS
i.	immediate mode	ADD R4, 1	R[4] ← R[4] + 1
ii.	direct mode	ADD R4,	R[4] ← R[4] +
iii.	register mode	ADD R4,	R[4] ← R[4] +
iv.	base+displacement mode	ADD R4,	R[4] ← R[4] +
v.	register indirect mode	ADD R4,	R[4] ← R[4] +

   (b) Propose physical instruction formats for the different modes of this instruction using as few different formats as possible. Beside each format, indicate which modes of ADD would use it. Indicate the position of all fields and the number of words of memory required for instructions employing each format.

2. **Instruction Set Architecture**

A proposed instruction set architecture consists of 60 instructions. The CPU will incorporate a register file of 8 registers and employ a 2-operand machine model. The CPU for the instruction set will be interfaced with a $2^{20} \times 8$ memory unit.

(a) Provide the syntax (a mnemonic and operands) for a "register mode add" instruction and a "register mode subtract" instruction, and express the semantics of each as register transfer statements.

(b) Provide a physical format for the "add" instruction you defined in (a) indicating how many words of memory would be needed to store the instruction in memory.

(c) Provide the syntax for a "direct mode load" instruction and a "direct mode store" instruction and express the semantics of each as register transfer statements.

(d) Provide a physical format for the "load" instruction you defined in (c) indicating how many words of memory would be needed to store the instruction in memory.

(e) Using just the assembly language instructions you have defined above, write a short program segment that will evaluate the assignment statement: `SUM = 2 * (A + B)`.

(f) Determine the number of memory accesses required to fetch and execute your program segment.

3. **Instruction Set Design**

An instruction set of a CPU consists of 50 instructions. The CPU is interfaced to a $2^{15} \times 12$ external random access memory, labelled "M", and possesses an $2^3 \times 12$ internal memory, labelled "R".

(a) What is the minimum number of bits required to represent the opcode of any instruction, assuming all instructions have opcodes of the same length?

(b) An ADD instruction is included in the instruction set. Its syntax is "ADD A, B".

i. If operands A and B specify addresses to external memory, how much space must be allocated in the machine instruction for each operand?

ii. If operands A and B specify addresses to internal memory, how much space must be allocated in the machine instruction for each operand?

iii. If operands A and B each specify a signed integer between -31 and $+31$, how much space must be allocated for each operand?

(c) Provide a physical instruction format for the ADD instruction of the previous question if the semantics of the instruction "ADD A, B" is "R[A] $\leftarrow$ R[A] + M[B]" .

(d) Suppose R[2] = 5 and M[3] = 7. If the opcode for ADD is 0x19:

i. Using the ADD instruction, provide an assembly language instruction for adding 5 and 7. You may express the operands that are to be provided in the instruction as actual numeric values.

ii. Give the corresponding machine instruction for adding 5 and 7.

4. **Instruction Set Architectures**

(a) In each of the following questions, provide an example (both the syntax and the semantics) of an instruction that satisfies the description provided.

i. An add instruction for a 0-operand machine

ii. A data transfer instruction that uses register indirect mode

iii. A branch instruction that uses base+displacement mode

(b) For each of the following versions of an AND instruction, provide a **physical** format for the instruction that minimizes instruction length and the number of memory accesses required to fetch and execute the instruction. There are 24 instructions, a $2^{16} \times 8$ external memory, **M**, and an 8×8 register file internal memory, **R**. **A** is another register in the CPU.

	SYNTAX	SEMANTICS
i.	AND Ri	$A \leftarrow A \wedge R[i]$
ii.	AND Ri, X	$R1 \leftarrow R[i] \wedge M[X]$
iii.	AND Ri, Rj	$R[i] \leftarrow R[i] \wedge R[j]$
iv.	AND Ri, X	$A \leftarrow A \wedge M[X + R[i]]$
v.	AND Ri, X	$R[i] \leftarrow R[i] \wedge M[X + PC]$

5. **Instruction Set Architecture**

Suppose you are given the following instructions, taken from an instruction set of 30 instructions:

SYNTAX	SEMANTICS (i = 0,1,...,7)	— MODE —
CLR Ri	R[i[$\leftarrow$ 0	
ADD Ri, A	R[i] $\leftarrow$ R[i] + M[A]	
SUB Ri, A	R[i] $\leftarrow$ R[i] + M[A]	
MPY Ri, Rj	R[i] & R[j] $\leftarrow$ R[i] $\times$ R[j]	
STO Ri, A	M[A] $\leftarrow$ R[i]	

(a) Under the column heading "MODE" indicate the addressing mode used by each instruction. Do not guess! Wrong answers will be subtracted from right answers

(b) Using the instructions given, write an assembly language program that evaluates the expression:

$$z = (p + q) * (p - q)$$

(c) What specifically determines the effect of the von Neumann bottleneck during the fetch phase of an instruction execution?

(d) What specifically determines the effect of the von Neumann bottleneck during the execute phase of an instruction execution?

(e) For the instructions given (from a set of 30 instructions) propose physical instruction formats, given $M = 2^{16} \times 8$ external memory and $R = 2^3 \times 8$ internal memory.

(f) For your program in (b) determine the number of fetch accesses and execute accesses and total number of memory accesses that will occur during the execution of the program.