ISA Tradeoffs

1. What advantages do RISC ISAs offer over CISC ISAs? Why were CISC ISAs popular for so long?

2. What is the maximum physically addressable amount of memory for the LC3b ISA?

3. a) Give one advantage and one disadvantage of using a variable length instruction set (as opposed to fixed length instructions).

b) What is an advantage to having a large number of architectural registers? A disadvantage?

4. There are four types of operand models; 1) Memory only, 2) Stack, 3) Accumulator, and 4) Registers model. List at least two metrics to categorize pros and cons of each model and write the models in order from good to bad.

5. You have a 16 bit instruction. In this ISA there are 22 opcodes, and 8 registers. If an instruction involves a destination register and two source registers, how many ALU opcodes can you include?

In the same ISA, how many bits can you use for an offset in a memory instruction with one destination register and one source register?
**Performance**

1. a) Does it make sense to compare two processors based on their frequency? Why or why not?

b) Briefly explain the difference between latency and throughput.

c) Suppose the following access times and hit rates.

- L1 cache: 5ns / 95%
- L2 cache: 30ns / 98%
- Main Memory: 1000ns / 99%
- Disk: 20ms / 100%

What is the average memory access time?

d) For large $N$, why is it likely that the following loop will perform poorly?

   *Give specifics on cache size. Ask what performance will be?*

   ```
   for (i = 0; i < N; i++) {
       for (j = 0; j < N; j++) {
           A[j * N +i] *= 2;
       }
   }
   ```

   Rewrite it to improve performance.

2. We are comparing the performance of two different systems: S1 and S2. System S1 costs $10,000 and system S2 costs $15,000. There is a program which takes 10 seconds on S1 and 5 seconds on S2. Answer the following questions.

   a) Which machine is more cost effective for running the program? How much?
b) Consider the following additional measurements with the conditions given in the question.

<table>
<thead>
<tr>
<th>Program</th>
<th>Instruction executed on S1</th>
<th>Instruction executed on S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given</td>
<td>$20 \times 10^6$</td>
<td>$16 \times 10^6$</td>
</tr>
</tbody>
</table>

Find the instruction execution rate (instruction per second) for each machine when running the program.

c) If the clock rate of system S1 is 20MHz and the clock rate of system S2 is 30MHz, find the clock cycles per instruction (CPI) for the program on both systems using the data given in the question and part b).

3. Consider two different implementations, M1 and M2, of the same instruction set. There are four classes of instructions in the instruction set. M1 has a clock rate of 50MHz. The average number of cycles for each instruction on M1 is as follows:

<table>
<thead>
<tr>
<th>Instruction class</th>
<th>CPI for this instruction class</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
</tr>
</tbody>
</table>

M2 has a clock rate of 75MHz. The average number of cycles for each instruction class on M2 is as follows:

<table>
<thead>
<tr>
<th>Instruction class</th>
<th>CPI for this instruction class</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>4</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
</tr>
</tbody>
</table>
a) Consider the program below composed of four instructions mentioned above.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program [instructions]</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Compute average performance, CPI, to run the given program on each machine M1 and M2?

b) If the number of instructions executed in a certain program is divided equally among the above classes of instructions, how much faster is M2 than M1?

c) Assuming the instruction distribution from part b), at what clock rate would M1 have the same performance as the 75MHz version of M2?

**Cache/Memory**

1. In what situations might direct mapped caches be a better choice than caches with high associativity? When might high associativity be preferred?

2. What effects do block size have on cache performance?

3. An unaligned memory access is a memory access that crosses the boundary of the word size supported by the memory hierarchy. In the LC3b, all accesses are aligned: a single load can fetch bytes 0 and 1 or bytes 2 and 3, but there is no single instruction to fetch both bytes 1 and 2.

Out of your love for computer architecture, you decide to modify the LDR instruction in a pipelined LC3b processor to support unaligned accesses.

(i) What changes must be made to the implementation of the instruction? Assume that the memory hierarchy cannot be changed to support unaligned accesses, and avoid decreasing performance for aligned loads.
(ii) The memory hierarchy of your processor contains the following:

- 256 B 2-way set associative L1 cache with 128-bit line size; 30ns access time
- 1KB 8-way set associative L2 cache with 256-bit line size; 60 ns access time (256 bit memory bandwidth; 600ns access time).

Give an address that, when loaded, will cause a worst-case unaligned load in this system? How long does it take (consider only the delay from memory)?

* For simplicity, assume that the access times are cumulative (e.g. an L1 miss that can be resolved by the L2 takes a total of 30+60 = 90 ns).

4. A computer system has a 64-bit virtual address space, a 48-bit physical address space, and 8-KB pages.

   a) Why do we need virtual memory?

   b) Why do we need a page table? Is page table a cache? Why or why not?

   c) How large (in bits) is a page table entry in this system.

   d) Rounding your answer to part c) up to the nearest byte if necessary, how much space would be required to hold the page table for this system?

   e) What is an issue with physically indexed, physically tagged caches?

5. Consider a system with a 24-bit address space and a cache. The cache has 256-byte cache lines and the capacity is 128KB.

   Give the number of comparators required, and specify the fields for tag, index, and offset if the cache is...

   a) Direct mapped
b) 8-way set associative

c) Fully associative

6. Define the two types of locality that caches help to exploit.

7. There is 64KB of L1 cache with 8-way set associativity with 128 bytes per cache line. Each address in the physical memory is composed of 32 bits and byte addressable. How many bits and which bits of the memory would represent the followings?

a) Offset bits

b) Index bits

c) Tag bits

8. Give one advantage and one disadvantage of each policy in each category:

i. Associativity

   a) Direct Mapped
      Advantage:

      Disadvantage:

   b) Fully Associative
      Advantage:

      Disadvantage:

   c) Set Associative
      Advantage:

      Disadvantage:
ii. Replacement policy

a) Least Recently Used (LRU)
   Advantage:
   Disadvantage:

b) Not Most Recently Used (not MRU)
   Advantage:
   Disadvantage:

c) Random
   Advantage:
   Disadvantage:

9. Show the final contents of the following caches with memory accesses of addresses: 5, 10, 3, 14, 12, 8, 11, 3, 15, 21 for part a) and b). Assume a Least Recently Used replacement policy and one word per cache line (i.e. no offset bits)

a) An 8 line direct mapped cache

```
  0  1  2  3  4  5  6  7
--- --- --- --- --- --- ---
```

b) A 2 way set-associative cache with 4 lines per way

```
  Set 0
  0  1  2  3
  Set 1
  4  5  6  7
  Set 2
  8  9 10 11
  Set 3
 12 13 14 15
```

c) How many non-compulsory misses in the direct mapped in part a)? the set associative in part b)?

10. Consider a computer with following features:

- The CPU sends references to the cache at the rate of $10^7$ words per second
- The bus, between cache and memory, can support $10^7$ words per second, read or writes (bandwidth=$10^7$)
- 25% of the above references are writes; 75% are reads
- 90% of all memory accesses (reads and writes) are found in the cache (hit rate=0.9)
- The cache block size is 2 words and the whole block is read on any miss
- The bus reads or writes a single word on a write hit
- Assume at any one time, 30% of the lines in the cache have been modified; 30% of the lines in the cache should be replaced on miss
- The cache uses write allocate on a write miss

You are considering adding a peripheral to the bus, and you want to know how much of the bus bandwidth is already used. Calculate percentage of the bus bandwidth used on the average in the two cases below. The percentage is called the traffic ratio.

a) The cache is write through policy

b) The cache is write back policy

11. A system with a 16 bit address space has a direct-mapped unified I/D-cache consisting of 16 lines of 2 bytes each. The cache has a write-back, write-allocate policy.

a) Compute the total bits of storage (including valid, dirty, and tag bits) needed to implement the cache.

b) Suppose we run the following program on the system:

```
0     LDR R1, R0, COUNT
2 LOOP:   ADD R2, R1, 10
4     STR R2, R0, DATA1
6     LDR R3, R0, CONST
8     ADD R4, R3, R1
10    STR R4, R0, DATA2
```
12  ADD R1, R1, -1
14  LDI R5, R0, POINT
16  JSRR R5
18  BRp LOOP
20  HALT:
22  COUNT: #2
24  CONST: #42
26  DATA1: ??
28  DATA2: ??
...
40  POINT: SUB
...
56  SUB: ADD R4, R4, R2
58  AND R4, R4, R3
60  RET

Assuming that execution ends immediately before fetching the instruction at HALT, what is the hit ratio for the cache?

c.) If we separate the I and D caches into separate direct-mapped caches with 8 lines of 2 bytes each, what is the hit ratio for each cache?

**MP related questions**

1. We are to implement a new instruction SWAP which swaps the value of two given source registers. The specifications of this instruction are as follows.

Instruction: **SWAP** SR1 SR2


Operation: { R7 = SR1;
SR1 = SR2;
SR2 = R7; }  

a) How would you implement this instruction in the state diagram?
b) What kind of changes do you have to make in the datapath to implement this instruction?

2. Consider the following program, executing on your multicycle MP2.1 LC3-b datapath.

There is a single level of cache between the datapath and the memory. Assume that there is no penalty for L1 hits and a 10-cycle penalty for each memory request. Thus, a clean L1 miss incurs a 10-cycle penalty (what about a dirty L1 miss?)

The cache is write-allocate and write-through.

The cache is 2-way set associative; each way has 2 4-word lines (total size = 32B). It uses a true LRU replacement policy.

Recall the following from MP1:
- IFETCH cost is 3 cycles + (cost of memory)
- All instructions go through DECODE (1 state)
- Thus, $F_D = 4$ cycles + (cost of memory).
- ADD requires $F_D + 1$ additional cycle
- Taken branches require $F_D + 2$ additional cycles; not taken branches cost $F_D + 1$ cycles
- LDR requires $F_D + 3 +$ (cost of memory)
- LDR requires $F_D + 3 +$ (cost of memory)

The following code executes on the processor:

```
0: LDR R4, R0, COUNTER
2: TOP: LDR R2, R0, MEM_OFF
4: ADD R2, R2, 15
6: STR R2, R0, MEM_OFF
8: ADD R0, R0, 2
10: ADD R4, R4, -1
12: BRp TOP
14: DONE: BRnzp DONE
16: COUNTER: #16
18: MEM_OFF: #0
20:   #1
22:   #2
24:   #3 ; etc
```

Please read through all the questions before attempting to answer them.
a) How many cycles does this code take to run until we reach the instruction at address 14 (0xe)? (E.g., until just before IF1 of the BRnzp instruction).

b) How many clean cache misses are there? How many dirty misses?

c) How many compulsory misses are there? Conflict misses? Capacity misses?

d) Would this code speed up by making the cache write-back instead of write-through? Why, or why not?

e) Would using a write-no-allocate policy on the cache change the performance of the program? Why, or why not?

f) Assume both the total cache size and line size must remain constant. What would happen to the performance of the program if the cache were made:

   i) Fully associative?

   ii) Direct mapped?

   iii) Fully associative, if the line size changes?

   iv) Direct mapped, if the line size changes?

   v) Fully associative, if the total cache size changes?

   vi) Direct mapped, if the total cache size changes?

   vii) Fully associative and line size changes, and the total cache size changes?

   viii) Direct mapped and line size changes, and the total cache size changes?

   ix) Fully associative, if the line size changes, if the total cache size changes?

   x) Direct mapped, if the line size changes, if the total cache size changes?

   xi) Fully associative and line size changes, if the total cache size changes?

   xii) Direct mapped and line size changes, if the total cache size changes?

   xiii) Fully associative, if the line size changes, if the total cache size changes?

   xiv) Direct mapped, if the line size changes, if the total cache size changes?

   xv) Fully associative and line size changes, if the total cache size changes?

   xvi) Direct mapped and line size changes, if the total cache size changes?

   xvi) Fully associative, if the line size changes, if the total cache size changes?

   xix) Direct mapped, if the line size changes, if the total cache size changes?

   xx) Fully associative and line size changes, if the total cache size changes?

   xxi) Direct mapped and line size changes, if the total cache size changes?

   xxx) Fully associative, if the line size changes, if the total cache size changes?

   xxxi) Direct mapped, if the line size changes, if the total cache size changes?

   xxxii) Fully associative and line size changes, if the total cache size changes?

   xxxiii) Direct mapped and line size changes, if the total cache size changes?

   xxxiv) Fully associative, if the line size changes, if the total cache size changes?

   xxxv) Direct mapped, if the line size changes, if the total cache size changes?

   xxxvi) Fully associative and line size changes, if the total cache size changes?

   xxxvii) Direct mapped and line size changes, if the total cache size changes?

   xxxviii) Fully associative, if the line size changes, if the total cache size changes?

   xxxix) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxi) Fully associative and line size changes, if the total cache size changes?

   xxxxii) Direct mapped and line size changes, if the total cache size changes?

   xxxxiii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxiv) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxv) Fully associative and line size changes, if the total cache size changes?

   xxxxvi) Direct mapped and line size changes, if the total cache size changes?

   xxxxvii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxviii) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxix) Fully associative and line size changes, if the total cache size changes?

   xxxxi) Direct mapped and line size changes, if the total cache size changes?

   xxxxii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxiii) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxiv) Fully associative and line size changes, if the total cache size changes?

   xxxxv) Direct mapped and line size changes, if the total cache size changes?

   xxxxvi) Fully associative, if the line size changes, if the total cache size changes?

   xxxxvii) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxviii) Fully associative and line size changes, if the total cache size changes?

   xxxxix) Direct mapped and line size changes, if the total cache size changes?

   xxxxi) Fully associative, if the line size changes, if the total cache size changes?

   xxxxii) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxiii) Fully associative and line size changes, if the total cache size changes?

   xxxxiv) Direct mapped and line size changes, if the total cache size changes?

   xxxxv) Fully associative, if the line size changes, if the total cache size changes?

   xxxxvi) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxvii) Fully associative and line size changes, if the total cache size changes?

   xxxxviii) Direct mapped and line size changes, if the total cache size changes?

   xxxxix) Fully associative, if the line size changes, if the total cache size changes?

   xxxxi) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxii) Fully associative and line size changes, if the total cache size changes?

   xxxxiii) Direct mapped and line size changes, if the total cache size changes?

   xxxxiv) Fully associative, if the line size changes, if the total cache size changes?

   xxxxv) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxvi) Fully associative and line size changes, if the total cache size changes?

   xxxxvii) Direct mapped and line size changes, if the total cache size changes?

   xxxxviii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxix) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxi) Fully associative and line size changes, if the total cache size changes?

   xxxxii) Direct mapped and line size changes, if the total cache size changes?

   xxxxiii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxiv) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxv) Fully associative and line size changes, if the total cache size changes?

   xxxxvi) Direct mapped and line size changes, if the total cache size changes?

   xxxxvii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxviii) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxix) Fully associative and line size changes, if the total cache size changes?

   xxxxi) Direct mapped and line size changes, if the total cache size changes?

   xxxxii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxiii) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxiv) Fully associative, if the line size changes, if the total cache size changes?

   xxxxv) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxvi) Fully associative and line size changes, if the total cache size changes?

   xxxxvii) Direct mapped and line size changes, if the total cache size changes?

   xxxxviii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxix) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxi) Fully associative and line size changes, if the total cache size changes?

   xxxxii) Direct mapped and line size changes, if the total cache size changes?

   xxxxiii) Fully associative, if the line size changes, if the total cache size changes?

   xxxxiv) Direct mapped, if the line size changes, if the total cache size changes?

   xxxxv) Fully associative and line size changes, if the total cache size changes?

   xxxxvi) Direct mapped and line size changes, if the total cache size changes?

   xxxxvii) Fully associative, if the line size changes, if the total cache size changes?