1. Short answer questions (10 points)
Answer the following questions in no more than 30 words.

a) What is a virtual cache? (2 points)
   Solution: A cache that is accessed with virtual addresses. (A more detailed answer could be a cache that is indexed and tagged with virtual addresses.

b) What is the benefit of a variable instruction format? (2 points)
   Solution: A smaller amount of memory is required to store the program and a smaller number of bytes need to be fetched for execution.

c) What is a TLB? (2 points)
   Solution: Translation Look Aside Buffer, a cache that stores the frequently used page table entries for accelerating the virtual address translation process.

d) What was the low temperature on the day of the second lecture? (2 points)
   Solution: -2F

e) What type of information should be stored in a cache tag store? (2 points)
   Solution: Tags, valid bits, (approximate) LRU bits, dirty bits, etc.
2. Performance (10 points)
Consider two different processors: with clock frequency of 20MHz for processor A and 30MHz for processor B. We want to run some programs for performance analysis. Assume that there is no memory delay; i.e. magic memory.

a. You are going to run a program with 100 instructions on both single-cycle processors with clock frequency specified above. What is the latency for each processor? Also, calculate the speedup of processor B over processor A. Note: Speedup should be greater than 1 if the execution time of processor B is shorter. (3 points)

Solution:
Processor A: 100 instructions * 1/20MHz = 100 instructions * 50ns/instruction = 5000ns
Processor B: 100 instructions * 1/30MHz = 100 instructions * 33ns/instructions = 3300ns
Speedup = 5000ns/3300ns = 1.5x

b. With two new multi-cycle processors A’ and B’, we ran 100 instructions, and both processors took the same time to complete them. Processor A’ clocks at 2GHz and performs 3 CPI while processor B’ performs 15 CPI. What is the clock frequency of processor B’? Note that CPI is identical for all instructions. (3 points)

Solution: Latency of Processor A’ = Latency of Processor B’
100 instructions * 3 CPI * 1/(2GHz) = 100 instructions * 15 CPI * 1/x
3/(2 GHz) = 15/x
x = 5*2 GHz = 10GHz

c. There is a new program with 100 instructions as summarized in the table below. The rightmost two columns indicate the number of cycles each type of instruction takes on each multi-cycle processor in (b). Which processor would run faster and by how much? Show your calculations. (4 points)

<table>
<thead>
<tr>
<th>Instruction Type</th>
<th>Percentage of instructions in program</th>
<th>Cycles taken on processor A’</th>
<th>Cycles taken on processor B’</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-type</td>
<td>30%</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>Load</td>
<td>40%</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>Store</td>
<td>20%</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>Beq</td>
<td>10%</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>

Solution: Processor A’: (2*30 + 4*40 + 3*20 + 3*10)/2GHz = 155ns
Processor B’: (13*30 + 16*40 + 15*20 + 12*10)/10GHz = 145ns
Speedup: B over A is 155ns/145ns = 1.069x
Processor B’ is faster than Processor A’ by 1.069x
3. Machine Problem (12 points)
You recently joined the LEG group as an engineer and have been tasked with adding the SWAP instruction to the MP1 LC-3b datapath. The instruction swaps the contents of registers REGA and REGB. The format of the instruction is given below:

| 1000 | REGA   | 000000 | REGB |

You have been given the following design constraints by your manager:
- The ALU and registers (e.g., IR, RegFile, etc.) are black boxes and cannot be modified.
- Minimize the number of cycles after the DECODE state used by the instruction with respect to the previous constraint.

a) Identify the changes that need to be made to the datapath to implement this instruction. (6 points)

b) Fill in the state machine to control the new datapath. For each state, show all necessary control signals (state actions) and transition conditions. (6 points)
Because the register file cannot be modified, we need at least two states after DECODE to write to registers REGA and REGB.

To swap the contents in two cycles, first store the contents of REGA into REGB while simultaneously storing the content of REGB into a temporary register (here MDR is used to avoid adding an additional register). In the following state, write the stored value into REGA.
4. Cache (16 points)
Consider a cache configuration:
- The address space is 16 bits;
- The total size of the cache is 1KB (Data array only);
- The line size (block size) is 64 bits;
- The associativity is 4-way set-associative;
- The replacement policy is LRU;
- The memory is byte addressable;

a) What’s the length of the tag, set(index), offset parts when we break down the memory address? (3 points)
Solution: Tag: 8 bits, Set: 5 bits, Offset: 3 bits

b) The total size of the cache is 1KB (Data array only). However, there is some overhead storage. For example, the tag array and so on. Calculate the overhead storage if the write policy is “write through”. Do another calculation for the overhead storage if the write policy is “write back”. Don’t include the overhead storage for LRU unit. Express the overhead storage in bytes. (4 points)
Solution: There is no “dirty bit” for write through cache.
- Write through: (tag+valid)*associativity*sets=(8+1)*4*32=1152 bits =144B
- Write back: (tag+valid+dirty)*associativity*sets=(8+1+1)*4*32=1280 bits =160B

c) How many additional bits do you need to implement the true LRU policy? How many additional bits do you need if you want to implement the Pseudo LRU policy? (4 points)
Solution: LRU: 2*4*32=256 bits (naive implementation, might be optimized to 5*32)
- Pseudo LRU: 3*32=96 bits

d) Consider the following read pattern:
(0x0000, 0x0100, 0x0200, 0x0300, 0x1000) for 100 times.
What is the hit rate? (2 points)
Solution: All the five address are in the same set. But the associativity is 4. The LRU replacement policy will give a hit rate of 0%.

e) If the cache is changed to a direct mapped cache (keep other configurations the same), what is the hit rate for the read pattern in part d)? (3 points)
Solution:
1 and 5 are in the same set. They will be in conflict and don’t have any cache hit.
2, 3, 4 are in different sets. So they will only encounter a compulsory miss. All the other accesses are cache hit. So the hit rate is (99*3)/100*5=59.4%
5. ISA Problem (12 points)

You’ve been tasked to implement a new instruction MAX DR, SR which compares the DR and SR values and puts the max value into DR and sets the condition code based on the value in DR. If DR is already the max, then no registers change.

a) You realize you can implement this instruction with a hardware or software (assembler) modification. Identify a positive and negative characteristic of each approach. (2 points)

Solution: HW
Positive – potentially a higher performance solution
Negative – need to deal with bit encoding challenges of new instruction, need to modify HW data path and control unit design, not applicable to existing hardware

SW/Assembler
Positive – no change to HW, can be applied to existing hardware
Negative – potentially a lower performance solution, may need to reserve registers to hold intermediate results between instructions in a sequence that implements a new mnemonic.

b) Give a 16-bit instruction format that would require minimum rewiring of the datapath (unmodified given above). Describe the hardware modifications required to implement the instruction. (4 points)

Solution: [ Opcode ][ DR ][ 000000 ][ SR ]
Add a comparator in the ALU which outputs the greater value.

c) Write the assembly sequence for MAX R0, R1. R0 should only change if it is less than R1. All other registers should remain the same. (6 points)

Solution: MAX:
NOT R0, R0
ADD R0, R0, 1
ADD R0, R0, R1
BRp MOV
Restore values:
NOT R1, R1
ADD R0, R0, R1
ADD R0, R0, 1
NOT R1, R1
NOT R0, R0
ADD R0, R0, 1
BR 1
MOV:
ADD R0, R1, 0
6. Cache/VM Interaction (20 points)
A small memory system has the following attributes.

- 10-bit virtual addresses
- 8-bit physical addresses
- Fully associative TLB with 2 entries
  - Random replacement policy
- 64B pages
- Single level page table
- 2-way set associative cache
  - Single level
  - 32B capacity
  - Block size of 4B
  - LRU replacement
  - Virtually indexed, physically tagged
  - Write-back policy
  - No write-allocation
  - Byte-addressable

a) Draw a flow chart showing, for all cases of TLB miss, TLB hit, page fault, page hit, cache hit, cache miss, read, write, dirty bit set or not set. Show where the dirty bit is set and cleared. Show where the TLB, cache, and page table are updated. Show which things happen in parallel, and where two concurrent courses of events must both finish before progress can continue. Show where cache entries are invalidated. (6 points)
b) Suppose your boss wants you to increase the cache size eightfold without changing the associativity. What were you able to do in parallel before that you can no longer do, and why not? (4 points)

Solution: The index field now overlaps with the page number field in the address, meaning that the virtual index is different from the physical index, and you can’t just use the virtual index to draw out the set in parallel with the TLB lookup.

c) What can be done to fix the problem in (b) without violating your boss’s new constraints, or changing block size, page size, or the total number of bits in the virtual or physical addresses? (4 points)

Solution: Draw two sets out of the cache in parallel with the TLB lookup. When TLB lookup is complete, check tags in parallel with separate comparators.