Please answer question 1 on this paper and put all other answers in the blue book.

1. True/False. Please circle the correct response.
   a) T  F  A four-way set associative cache means that there are four words in a cache block.
   b) T  F  A TLB miss causes a page fault, requiring a page to be brought into memory from disk.
   c) T  F  In the simple MIPS processor we studied, the “sign-extend” serves to convert a 16-bit offset to a 32-bit number to add to the contents of a register.
   d) T  F  Asserting a word line and asserting (to “high” voltage) a bit line writes a 1 to a DRAM cell.
   e) T  F  A control hazard occurs in a pipelined CPU when two instructions try to access memory at the same time.
   f) T  F  In a modern PC using a PCI bus system, the backside bus should operate at a clock speed that is at least as fast as the L2 cache.
   g) T  F  SRAM is “static” in the sense that if the power is turned off, SRAM will continue to store data (e.g. as in flash memory in MP3 players or USB thumb drives).
   h) T  F  In class, we presented a simple analytical model of system performance, given some assumptions about cache miss rates and cache miss penalties, which showed that doubling CPU speed may increase overall system speed by only 13%. Using that same model, system speed would double if both CPU speed and memory speed was doubled.
   i) T  F  In the two’s complement number representation, a negative number with more leading ones is larger (i.e. less negative) than a negative number with fewer leading ones.
   j) T  F  Interleaved memories are used to provide a multiple-word cache block in response to a single memory read request (e.g. due to a load instruction).
2. In addition to “branch on equal”, most processors support “branch on not equal”, “branch on greater than”, “branch on greater than or equal”, etc. For each of the following instructions, describe precisely how you would modify the simple MIPS processor discussed in class in order to support the instruction in the hardware:
   a) branch on greater than or equal
   b) branch on less than
   c) branch on not equal

3. 
   a) What precisely is the data that a TLB (translation lookaside buffer) contains?
   b) Draw the hardware to support a two-way set-associative TLB. Assume a page size of 2Kbytes and be sure to show the number of bits for each line (assuming a word size of 32-bits).
   c) Just as a single element of an ordinary instruction or data cache can be a multi-word cache block, a single TLB element could contain a block of multiple entries of data. What would these multiple entries of data be?

4. 
   a) Below is a table showing, for the various MIPS instructions supported by the simple MIPS processor, the values of the 6-bit Funct field of the instruction, the 2-bit ALUOp field from the main control, and the 3-bit ALU control output.

<table>
<thead>
<tr>
<th>opcode</th>
<th>ALUOp</th>
<th>operation</th>
<th>funct field</th>
<th>ALU action</th>
<th>ALU cntl</th>
</tr>
</thead>
<tbody>
<tr>
<td>LW</td>
<td>00</td>
<td>load word</td>
<td>xxxxxxx</td>
<td>add</td>
<td>010</td>
</tr>
<tr>
<td>SW</td>
<td>00</td>
<td>store word</td>
<td>xxxxxxx</td>
<td>add</td>
<td>010</td>
</tr>
<tr>
<td>BEQ</td>
<td>01</td>
<td>branch equal</td>
<td>xxxxxxx</td>
<td>subtract</td>
<td>110</td>
</tr>
<tr>
<td>R-type</td>
<td>10</td>
<td>add</td>
<td>100000</td>
<td>add</td>
<td>010</td>
</tr>
<tr>
<td>R-type</td>
<td>10</td>
<td>subtract</td>
<td>100010</td>
<td>subtract</td>
<td>110</td>
</tr>
<tr>
<td>R-type</td>
<td>10</td>
<td>AND</td>
<td>100100</td>
<td>and</td>
<td>000</td>
</tr>
<tr>
<td>R-type</td>
<td>10</td>
<td>OR</td>
<td>100101</td>
<td>or</td>
<td>001</td>
</tr>
<tr>
<td>R-type</td>
<td>10</td>
<td>SLT</td>
<td>101010</td>
<td>set on less than</td>
<td>111</td>
</tr>
</tbody>
</table>

Given that there are 8 bits of input, one would expect to build a circuit for the ALU control corresponding to a truth table with 256 entries (i.e. $2^8$). Describe, in your own words, why the actual truth table needed has far few fewer entries.

b) Suppose the ALUOp bits indicate that the current instruction is an R-type instruction but suppose also that the funct field doesn’t have any of the values
listed in the bottom five rows of the above table. It is desirable for the processor to signal an error (“incorrect instruction”). Build a circuit that would, in the case of an R-type instruction, test to see if the funct field was valid.

5. Suppose a computer has the simple 5-stage pipelined MIPS processor that we described in class. Suppose further that, to address data hazards, the pipeline has “forwarding” between stages of the pipeline. That is, suppose the Execute or Memory stage of one instruction in the pipeline can forward data to the Execute stage of another instruction in the pipeline, for use in the next cycle. Assume that there are separate instruction and data caches.

a) How many cycles would the execution of the following code take to complete? Assume each stage, in the absence of hazards, takes one cycle (i.e. there are no cache misses).

```
add $s1,$s2,$s3
sub $s2,$s4,$s5
add $s3,$s1,$s2
lw $s4,34($s3)
sub $s5,$s4,$s2
```

b) Illustrate the progress of these instructions through the pipeline (using the kind of drawing found in my notes or whatever drawing you feel best shows the progress of the instructions).

c) Suppose that the processor feeds a new instruction into the pipeline every instruction, even though the previous instruction might be a branch. If the branch is taken, though (as determined by the Execute stage), the two succeeding instructions currently in the pipeline, i.e. those in the instruction fetch and instruction decode stages, should not be executed. It would be complicated and expensive, though, to try to stall the pipeline and remove those two instructions from the pipeline. Describe, in detail, how you would implement the pipeline to allow the two instructions immediately following the branch to proceed through the pipeline but not violate the meaning of the program being executed.

6. Suppose that your PC, on the typical programs you run, had the following properties:

- L1 cache hit rate (both instructions and data): 90%
- L2 cache hit rate (both instructions and data): 99%
- Percent of all instructions that are memory instructions: 35%
- Penalties:
  - L1 miss but L2 hit: 20 cycles
  - L1 miss and L2 miss: 300 cycles

Suppose also that an L1 cache hit has no penalty and that an instruction can be executed every cycle (if there are no cache misses). How much slower is your PC than a PC with the same processor speed but which never has a cache miss?