Digital Logic

Read: Sections 3.1 – 3.3 in the textbook.
Handwritten digital lecture notes on the course web page.

Study Questions:

1. On pages 226-227 of the textbook, questions: 3, 5, 8, 9, 11.

2. Build a decoder with three input lines but with only six output lines. If the value of the input corresponds to 6 or 7, then all output lines should be asserted to signal an error.

3. Modify the 1-bit ALU covered in class (supporting AND, OR, +, and -) to implement an XOR (exclusive OR) operation. Draw the entire ALU, adding the fewest possible gates. Use the available op code 11 to select the XOR operation.

4. Build a register file out of (already built) registers, multiplexors, decoders, and logic gates that has the following inputs:
   - read select line
   - two write select lines, selecting two registers that can be written to.
   - two write data lines, for writing data to the two registers selected.
   - two write enable lines, for enabling writing to one or both of the selected registers.

   and an output line for the data that was read from the selected register. Note that this is like the register file covered in class, except that instead of reading from two registers and writing to one register, it supports the writing to two registers and the reading from one register.

5. Can you build a device that, logically, behaves like an or gate from only and and not gates? If so, do so (just for the case where and and or gates have only two inputs). If not, explain why not. Then, answer the same question, except with regards to constructing an and gate from only or and not gates.

6. Consider the register file discussed in class (shown on the first page of the November 11 lecture notes), constructed from D flip-flops.
   a. Suppose that, when the clock line is asserted, the values of the “Read register number 1” and “Write register” inputs are both 4 (i.e. 000...100) and “Write” (i.e. write-enable) is asserted. When the “Read data 1” output stabilizes (while the clock is still asserted), would the value of “Read data 1” necessarily be the same as the value of the “Write data” input? Explain.
   b. If your answer to part b was “yes”, then describe a way of reading the old value of a register at the same time as writing a new value to the same register. If your
answer was “no”, then indicate (precisely) when the value of “Write data” would appear on the “Read data 1” line, assuming that the values of “Read register number 1” and “Write register” are held constant.

7. The last page of the November 9 lecture notes shows D Flip-Flop with a falling edge trigger. Draw the logic for a D Flip-Flop with a rising edge trigger (following the example in the lecture notes, not in the textbook).

8. Build a two-bit multiplier out of gates (i.e. it multiplies two two-bit numbers resulting in a 4-bit number). As usual, it may be easiest to start out with a truth table.

9. True/False Questions
   a. T F A circuit containing only OR and NOT gates must be a combinational circuit.
   b. T F For two's complement numbers, the negative of a number can be found by adding one and then inverting the bits.
   c. T F An disadvantage of two's complement numbers is that, unlike a sign-and-magnitude representation, you cannot tell if a number is negative by looking at only one bit.
   d. T F The number of rows of a truth table depends on the number of inputs, not the number of outputs.
   e. T F The B negate line is connected to the carry-in line of each 1-bit ALU within a 32-bit ALU.
   f. T F After a DRAM cell is read, the value read has to be written back to the cell.
   g. T F Asserting a word line and asserting (to “high” voltage) a bit line writes a 1 to a DRAM cell.
   h. 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).
   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 If a wire carries a logical value of 0, its voltage level will be 0 Volts.
   k. T F In an S-R Latch, if both S and R inputs are asserted, Q and ¬Q will both be 0.
   l. T F A circuit diagram showing a multiplexor with sixteen 32-bit input lines and four select lines is actually referring to an array of 32 multiplexors, each having sixteen one-bit input lines and four select lines.
   m. T F Doubling the number of registers of a register file (but leaving everything else the same) will double the number of input lines to the register file.
   n. T F In a DRAM, all cells have to be periodically refreshed, not just the ones that have been read.
When shifting a two’s complement number to the left, an overflow can only occur if the number is negative, since positive numbers have a zero in the leftmost bit.

To perform the operation \( A - B \), where \( A \) and \( B \) are numbers represented in two’s complement, one can build hardware to perform the following steps: flip the bits of \( A \), add \( B \) with a carry-in of 1, flip the bits of the result, and then add 1.

### Processors (RISC v. CISC, Pipelining)

Read:  Section 2.1 in the textbook.
Handwritten pipelining lecture notes on the course web page.

Study Questions:

1. If a single CISC instruction can achieve the same result as multiple RISC instructions, why isn’t a program compiled for a CISC machine necessarily faster than the same program compiled (into more instructions) for a RISC machine?

2. How does the clock rate factor into the RISC v. CISC discussion?

3. Describe in your own words what a “control hazard” is in a pipeline. Then, give an example of a sequence of simple x86 assembly instructions that presents a control hazard in a pipelined processor.

4. Assume a simple 5-stage pipeline (IF, ID, EXE, DF, W) as we did in class, where, in the ideal world, each stage takes a single cycle. Also, assume there are no cache misses. How many cycles would the following code take to execute if there is no special hardware to improve performance in the presence of hazards?

   Intel
   
<table>
<thead>
<tr>
<th>Instruction</th>
<th>AT&amp;T</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mov eax,[ecx+100]</code></td>
<td><code>mov 100(%ecx),%eax</code></td>
</tr>
<tr>
<td><code>mov ebx,[ecx+104]</code></td>
<td><code>mov 104(%ecx),%ebx</code></td>
</tr>
<tr>
<td><code>add eax,ebx</code></td>
<td><code>add %ebx,%eax</code></td>
</tr>
<tr>
<td><code>mov [ecx+108],eax</code></td>
<td><code>mov %eax,108(%ecx)</code></td>
</tr>
<tr>
<td><code>mov edx,[ecx+108]</code></td>
<td><code>mov 108(%eax),%edx</code></td>
</tr>
<tr>
<td><code>add eax,edx</code></td>
<td><code>add %edx,%eax</code></td>
</tr>
</tbody>
</table>

   Draw a picture of the kind we drew in class, representing the pipelined execution of the above instructions. (Note: Of course, it won’t look exactly like the above picture. Also, be sure to identify the various stages: IF, ID, etc.).

5. Suppose that a processor feeds a new instruction into the pipeline every cycle, even though the previous instruction might be a branch (i.e. a jump or conditional jump). 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. True/False Questions

   a. T F The number of stall cycles due to data hazards occurring in a pipeline during program execution might be reduced by careful instruction scheduling by a compiler (i.e. by rearranging the order of instructions).

   b. T F A control hazard occurs in a pipelined CPU when two instructions try to access memory at the same time.

   c. T F Having every instruction be the same number of bits simplifies pipeline design.

Caches

Read: Section 4.51 in the textbook.
Handwritten cache lecture notes on the web page (ignore the reference to “Chap. 7 in P&H”).

Study Questions:

1. Explain why set-associative caches might generally result in better performance than direct-mapped caches.

2. Write a simple piece of code in C or assembly for which, when executed, you might expect to have a large number of conflict misses in the cache on a machine with a 1MB direct-mapped cache.

3. On a 32-bit processor (i.e. addresses are 32 bits) with a 4MB direct-mapped cache, where each cache entry is one word (i.e. only one word is brought into cache at a time), how many tag bits must be associated with each word in the cache?

4. On a 32-bit processor with a 4MB 4-way set associative cache, where each cache entry is one word, how many tag bits must be associated with each word in the cache?

5. Suppose a program executes N instructions overall. This means that N instructions must be fetched from memory. Suppose also that 30% of the instructions need to fetch data from memory. Suppose that, on your computer, 95% of these memory accesses, for both instructions and data, result in cache hits. Finally, suppose that an instruction that causes no cache misses can be executed in a single cycle, but every cache miss adds an extra 100 cycles to the overall execution time. How many cycles will the program take to execute (note that if there were a 100% cache hit rate, execution would take N cycles).
6. Suppose you were not happy with the performance of the computer being used in question 5 above. After shopping around, you realized you could, for the same amount of money, replace the processor with one of the following:

   • A processor that was twice as fast (i.e. twice the clock speed), but with the same cache size and cache hit rate (95%).
   
   • A process with the same clock speed, but with a larger cache so that the hit rate went up to 98%.

Which would you buy? Explain why (i.e. show quantitatively which would be faster).

7. True/False Questions

   a. T F A four-way set associative cache means that there are four cache entries in each set.
   
   b. T F A 2-way set associative cache requires fewer gates to implement than a direct-mapped cache (think about it).
   
   c. T F Assuming a cache size of 256K words of data and 4-word cache block size, a fully associative cache will be harder to implement than a 64K-way set-associative cache.