Homework #3
Due: Sunday, May 9th at 11:59pm

Your submission should be comprised of one item: a .pdf file containing answers to the questions. **Legible**, handwritten solutions are acceptable for undergraduate students (472). The handwritten solutions must be electronically scanned and submitted as a PDF file. Graduate students (572) must compose their solutions in MS Word, LaTeX, or some other word processor and submit the results as a PDF file.

1. (16 pts) Derive the even-parity Hamming code that corresponds to the following data bits. Be sure to show your work and illustrate how each parity bit was determined.

   `0b10111000101110`

2. (24 pts) Suppose that you receive a Hamming code with the following contents. (containing data bits and parity bits):

   `0b111011010010101`

   For all parts of this question, assume that a maximum of one bit was corrupted.

   a. How many bits of data (not including parity bits) were originally encoded?
   b. Assuming that the transmitted Hamming code was using even parity, is it possible to determine the original data bits that were encoded? If so, derive the original data bits (show your work). If not, explain why.
   c. Now assume that the transmitted Hamming code was using odd parity. Is it possible to determine the original data bits that were encoded? If so, derive the original data bits (show your work). If not, explain why.

3. (8 pts) Imagine that you are considering two potential implementations of a virtual memory system (case “a” and case “b”).
   a. Assuming a memory page of 16K and a 32 bit virtual address space, calculate the amount of memory (in bytes) that is needed to store the full page table. Assume that each PTE requires 5 bytes. Show all calculations.
   b. If the virtual address space is expanded to 48 bits (and all other properties remain identical), how many bytes of memory are now required to store the full page table?

4. (12 pts) Consider Figure 4.10 in the textbook (the image is identical for the 5th edition and 6th edition).
   a. Determine the control signals which will implement the following MIPS instruction:

      ```
      addi  $15, $7, -36
      ```

      Provide the required binary values of all control lines during this clock cycle (the figure uses **blue ink** to notate the relevant signals). Assume that control lines are "active high". If a particular binary value is not relevant, you may indicate this by placing an "x" in the corresponding bit position.

   b. Similar to part A, determine the binary value of all control signals needed to implement the following MIPS instruction:

      ```
      lw  $22, 76($10)
      ```

5. (25 pts) In this question we will be working with a CPU that exhibits the following latencies for each stage of the datapath:

IF	ID	EX	MEM	WB
250ps	270ps	190ps	330ps	225ps

For our particular program, the percentage of instructions (in each category) are as follows:

ALU/Logic	Load	Store	Jump/Branch
52%	18%	17%	13%

   a. What will be the clock period for a pipelined implementation?
   b. What minimum clock period is expected for a non-pipelined design?
   c. Suppose we want to improve the performance of our pipeline. We've been informed that it's possible to split a single stage of the pipeline into three stages, but there's a caveat: each of the three newly created stages will exhibit 40% of the original latency (not 33.3%). Which stage would be the best to split, and what latency would be exhibited by the newly pipelined CPU? I.e. how much time would this new CPU implementation require to fully execute an instruction?
   d. Ignoring the effects of stalls and hazards, what is the utilization of the data memory? Express your answer as the percentage of clock cycles when the data memory is actively being used.
   e. Ignoring the effects of stalls and hazards, what is the utilization of the write port on the register bank? Note: I'm asking specifically asking about the write port (controlled by the RegWrite signal in Figure 4.10). Express your answer as the percentage of clock cycles when the register write circuitry is being used to update a register.