

## Homework 11

1. **A.** Add Hamming parity bits to the value 204, assuming it is stored in unsigned format using 8 bits. (1100011010100)
- B.** Describe the data-recovery process and the exact value recovered if bit 3 is somehow flipped. Do the same if bit 9 is flipped, and if both 3 and 9 are flipped simultaneously. (*recovered to 204; recoverd to 204; unrecoverable*)
2. For each of the following sets of 8 data bits interleaved with 5 parity bits using the Hamming system, is the value valid, or was there an error? If the original unsigned number is recoverable, what is it? If not, why isn't it?
  - A.** 1101011101110 (*error; 255*)
  - B.** 1000100011110 (*valid; 129*)
  - C.** 1000001101110 (*error; unrecoverable*)

3. Answer the questions for each MIPS processor in this table, whose stages have the latencies shown.

	IF	ID	EX	MEM	WB
<b>P1</b>	300 ps	400 ps	350 ps	500 ps	100 ps
<b>P2</b>	200 ps	150 ps	120 ps	190 ps	140 ps

- A.** How long will a clock cycle be, in both a pipelined and nonpipelined system? (*500 ps, 1650 ps; 200 ps, 800 ps*)
  - B.** What will the total latency of a `lw` instruction be, in both a pipelined and nonpipelined system? (*2500 ps, 1650 ps; 1000 ps, 800 ps*)
  - C.** If one could split any single stage into two new stages (each with half the latency) which stage should be split and what would the new period of the processor be, assuming a pipelined system? What would the total latency of `lw` be? (*MEM, 400 ps, 2400 ps; IF, 190 ps, 1140 ps*)
4. For the following questions, consider the MIPS code on the right. Assume that each of these instructions is divided into five stages (IF, ID, EX, MEM, and WB), and that each stage takes exactly one cycle to complete.
 

<code>lw</code>	<code>\$t1, 0(\$s0)</code>
<code>lw</code>	<code>\$t2, 4(\$s0)</code>
<code>add</code>	<code>\$t3, \$t1, \$t2</code>
<code>sw</code>	<code>\$t3, 0(\$s0)</code>
<code>sub</code>	<code>\$t4, \$s4, \$t3</code>
<code>lw</code>	<code>\$t5, 0(\$s1)</code>

    - A.** How many cycles will it take to execute this code, assuming that the system is *not pipelined*? (*30 cycles*)
    - B.** How many cycles will it take, assuming that the system *is pipelined*, but *cannot support forwarding*? (*14 cycles*)
    - C.** How many cycles will it take, assuming that the system *is pipelined*, and *allows forwarding*? (*11 cycles*)
    - D.** How many cycles will it take, assuming that the system *is pipelined*, *allows forwarding*, and a *good compiler has optimized this code (reordering instructions)*? (*10 cycles*)