Homework#2 solutions

2.16

2.16.1 The opcode would expand from 7 bits to 9.

The `rs1`, `rs2` , and rd fields would increase from 5 bits to 7 bits.

2.16.2 The opcode would expand from 7 bits to 9.

The`rs1` and `rd` fields would increase from 5 bits to 7 bits. This change does not affect the `imm` field *per se*, but it might force the ISA designer to consider shortening the immediate field to avoid an increase in overall instruction size.

2.16.3 * Increasing the size of each bit field potentially makes each instruction longer, potentially increasing the code size overall.

* However, increasing the number of registers could lead to less register spillage, which would reduce the total number of instructions, possibly reducing the code size overall.

2.39

2.39.1 No. The resulting machine would be slower overall.
Current CPU requires (num arithmetic * 1 cycle) + (num load/store * 10 cycles) + (num branch/jump * 3 cycles) = $500 * 10^6 * 1 + 300 * 10^6 * 10 + 100 * 10^6 * 3 = 3800 * 10^6$ cycles.
The new CPU requires (.75 * num arithmetic * 1 cycle) + (num load/store * 10 cycles) + (num branch/jump * 3 cycles) = $375 * 10^6 * 1 + 300 * 10^6 * 10 + 100 * 10^6 * 3 = 3675 * 10^6$ cycles.
However, given that each of the new CPU's cycles is 10% longer than the original CPU's cycles, the new CPU's $3675 * 10^6$ cycles will take as long as $4042.5 * 10^6$ cycles on the original CPU.

2.39.2 If we double the performance of arithmetic instructions by reducing their CPI to 0.5, then the the CPU will run the reference program in (500 * .5) + (300 * 10) + 100 * 3 = 3550 cycles. This represents a speedup of 1.07.

If we improve the performance of arithmetic instructions by a factor of 10 (reducing their CPI to 0.1), then the the CPU will run the reference program in (500 * .1) + (300 * 10) + 100 *3 = 3350 cycles. This represents a speedup of 1.13.

2.40

2.40.1 Take the weighted average: 0.7 * 2 + 0.1 * 6 + 0.2 * 3 = 2.6

2.40.2 For a 25% improvement, we must reduce the CPU to 2.6 * .75 = 1.95. Thus, we want 0.7 * x + 0.1 * 6 + 0.2 * 3 < = 1.95. Solving for x shows that the arithmetic instructions must have a CPI of at most 1.07.

2.40.3 For a 50% improvement, we must reduce the CPU to 2.6 * .5 = 1.3. Thus, we want 0.7 * x + 0.1 * 6 + 0.2 * 3 < = 1.3. Solving for x shows that the arithmetic instructions must have a CPI of at most 0.14