EEL 5764 HW #3

11/10/19

1 3.11)

```
                    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
lw x1,0(x2)         F D E M - - W
addi x1,x1,1          F - F - - D E M W
sw x1,0(x2)               F - - D E M - - W
addi x2,x2,4                  F D E - - M W
sub x4,x5,x2                    F - - - - D E M W
bnz x4,loop                           F - - D E - M W
                                      F^b   F^a
lw x1,0(x2)                           F F   F
```

a) 3 cycles of branch overhead, as the fetch would start at the 22nd cycle in the case of the branch instruction running, and it otherwise would start at cycle 19.

b) 1 cycles now wasted as we save 2 by being able to fetch at cycle 20.

c) 0 cycles lost as the prediction is correct. As a predicted taken will run during a fetch, then the predicted branch will be the next fetch.

HW #2 (cont.)

2a) 3.4GHz processor. Always-not-taken prediction, 20% of dynamic insts are branch. 80% of all branches are taken. Wrong 16% of time.

Assume ideal CPI = 1

$(.84 \cdot 1) + (.16 \cdot 3) = 1.32$ CPI

$\frac{1 \text{ inst}}{1.32 \text{ cycle}} \cdot \frac{3.4e9 \text{ cycles}}{1 \text{ s}} = \boxed{2.57e9 \text{ inst/s}}$

2 stalls for a misprediction

b) 1.16 CPI as $\Rightarrow (.84 \cdot 1) + (.16 \cdot 2)$

$\frac{1 \text{ inst}}{1.16 \text{ cycle}} \cdot \frac{3.4e9 \text{ cycles}}{1 \text{ s}} = \boxed{2.93e9 \text{ inst/s}}$

Only 1 stall between misprediction and realizing it.

3 Since we don't ever get to "DONE" as we JUMP to loop every time. In the case that R1 is initialized to a number >1 or a number <0, the branch prediction will be wrong twice, then always correct after. If R1 is initialized to $=0$, it will predict wrong once, then be correct always after. If R1 is initalized to $=1$, it will be correct once, then wrong twice, flipping the prediction, and will be correct after that every time. Therefore as the loops continue to run, the prediction accuracy will approach but never reach 100%.