

CSE140 HW #8

Consider the following code for the following problems

```
loop:
    LW    R4, 0(R3)
    ADDI  R3, R3, 4
    SUBI  R1, R1, 1
b1:
    BEQ   R0, R4, b2      # R0 is zero
    ADDI  R2, R2, 1
b2:
    BNE   R0, R1, loop
    ADDI  R3, R3, 4
```

Assume R3 points to the array with all 0 entries.
R1 = R2 = 5.

1) If there is a branch delay slot and result of R2 stays even after branch is taken. What is the value of R2 when the loop finishes?

2) If there is **no** branch delay slot, what is the value of R2 when the loop finishes?

3) Which behavior makes the most sense?

4) Now assume we use a global branch-predictor with **no** branch delay slot. Assume R1 starts out with value 10. R3 points to the array with all 0 entries. How well does the predictor work?

5) Now assume the array has all 1 entries. How well does the predictor work?

6) Now assume the array has alternating 0's and 1's in the array entries (starting with 0's). How well does the predictor work?

7) Consider the following ROB and how the above code will be executed with branch prediction assuming it is predicted as in (6). Fill the table and trace through how it might work with $n = 4$.

Reorder Buffer (ROB)									
use	ex	op	p1	PR1	p2	PR2	Rd	LPRd	PRd
next to commit	→								

Assume $R1 \rightarrow P0$, $R2 \rightarrow P1$, $R3 \rightarrow P2$, $R4 \rightarrow P3$
 Free physical registers: $P4 - P12$

8) How bits are each column in the ROB table to support standard 32-bit MIPS architecture?

9) What type of hazards are detected by ROB and avoided?