HW4 solutions

Problem 1 i)

	8	9	10	11	12	20	29	30	31
b1 predicted	NT	T	NT	T	NT	T	T	NT	T
b1 actual	T	NT	T	NT	T	T	NT	T	NT
b2 predicted	NT	NT	NT	T	NT	NT	T	NT	T
b2 actual	NT	NT	T	NT	NT	T	NT	T	NT

ii)
b1 accuracy = 1 / 9 = 11.1 %
b2 accuracy = 3 / 9 = 33.3 %

iii)
Overall accuracy = 4 / 18 = 22.2 %

Problem 2 i)

	8	9	10	11	12	20	29	30	31
For g=0									
b1 predicted	NT	T	NT		T	T		T	
b1 actual	T	NT	T		T	T		T	
b2 predicted		NT		NT			NT		NT
b2 actual		NT		NT			NT		NT
For g=1									
b1 predicted				NT			NT		NT
b1 actual				NT			NT		NT
b2 predicted	NT		NT		T	NT		T	
b2 actual	NT		T		NT	T		T	

ii)
b1 accuracy = 6 / 9 = 66.7 %
b2 accuracy = 6 / 9 = 66.7 %

iii)
Overall accuracy = 12 / 18 = 66.7 %

iv)
Accuracy of b2 when g=0: 100%. When g=0 that means that the last branch, b1, was not taken. Thus, the value of X is not even and therefore not a multiple of 10. Thus, when g=0, b2 is always not taken. Since its prediction starts as not taken, it is predicted 100% correctly.

Problem 3)

Note, my solution assumes the most recent branch's history is shifted in from the left. Since the problem didn't explicitly state that, it would be OK if you shifted in the bit of history in the opposite direction but your answer would differ because of that.

X	w0	w1	w2	w3	w4	Y	Prediction	Outcome	History before prediction
0	0	0	0	0	0	0	T	NT	0101
3	-1	1	-1	1	-1	3	T	NT	1010
6	-2	0	0	0	0	-2	NT	T	0100
1	-1	-1	1	-1	-1	-3	NT	NT	1001
4	-2	-2	2	0	-2	0	T	T	0010
7	-1	-3	1	1	-3	-1	NT	T	0101
2	0	-4	2	0	-2	-8	NT	NT	1011
5	-1	-5	3	-1	-3	9	T	T	0110
0	0	-6	4	0	-4	-6	NT	NT	1101
3	-1	-7	3	1	-5	-5	NT	NT	1010
6	-2	-8	4	0	-4	14	T	T	0100
1	-1	-9	5	-1	-5	-19	NT	NT	1001
4	-2	-10	6	0	-6	8	T	T	0010
7	-1	-11	5	1	-7	7	T	T	0101
2	0	-12	6	0	-6	-24	NT	NT	1011
5	-1	-13	7	-1	-7	25	T	T	0110

HW4 solutions

Problem 4)
Bimodal, in which we have a two-bit state machine for each branch, will only work for 100% of branches in the third case. In this case, each branch has only one direction (taken/not taken) in the shown execution path and the two-bit state machine (saturating up-down counter) will learn the “mode” or most common behavior and perform accurately for each branch. For the first two cases, it would mispredict b1 since that branch is unbiased (50% taken/50% not taken).

The local predictor, in which we collect a separate 4-bit history for each branch will work for 100% of branches in all three cases. However, the case is the only one in which bimodal succeeds, so we have chosen the bimodal predictor for the third case. In the first and third cases, only b1 is not strongly biased, but in both cases, the behavior of b1 alternates, taken, not taken, taken, not taken, and this pattern can be easily stored in a 4 bit local history.

The global predictor with gselect indexing uses a global 4-bit history combined with some of the address of each branch to index into a predictor. If we consider branches b13 and b15 in both the first and second cases, we will see that the pattern of the last four branches that lead up to those branches is Taken, Taken, Taken, Taken. Thus, without gselect indexing, b13 and b15 would both use the same state machine to make their predictions. HOWEVER, with gselect indexing, we can assume because the branches’ address is combined with the history, they will use different predictor state machines and thus b13 and b15 can make different predictions.

Instead, look at branch b1. b1 has two likely outcomes (taken or not taken). In the first case, i) when b1 should be predicted taken, the last four branches leading up to that point were Taken, Taken, Taken, Not Taken. ii) when b1 should be predicted not taken, the last four branches leading up to that point were Taken, Taken, Taken, Taken. Thus, a different predictor will be used for each of these two different patterns. In the second case, however, i) when b1 should be predicted taken, the last four branches leading up to that point were Taken, Taken, Taken, Not Taken. ii) when b2 should be predicted not taken, the last four branches leading up to that point were Taken, Taken, Taken, Not Taken. In both cases, b1 will use the same prediction state machine. The gselect indexing (combining the history with bit of branch address) doesn’t help in this case, because both instances are for a single branch, b1. Thus b1 will be mispredicted using the gselect predictor for case two.

Case One - Gselect
Case Two - Local
Case Three - Bimodal