1. Consider the following code segment within a loop body:

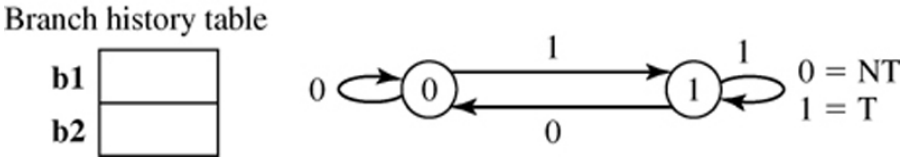
```
if (x is even) then                ;  Branch b1
     increment a                   ;  Branch b1 taken
if (x is a multiple of 10) then    ;  Branch b2
     increment b                   ;  Branch b2 taken
```

Assume the following list of nine values of x is to be processed by nine iterations of this loop:

8,9,10,11,12,20,29,30,31

https://www.apc.com/us/en/country-selector? Note: Assume that predictor entries are updated by each dynamic branch before the next dynamic brach access the predictor.

i) Assume that a 1-bit (history bit) state machine is used as the prediction algorithm for



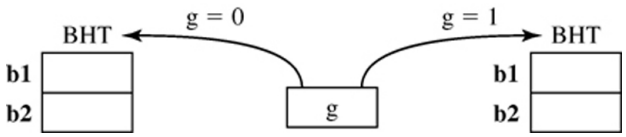
predicting the execution of the two branches in this loop. Indicate the predicted and actual branch directions of the b1 and b2 branch instructions for each iteration of this loop. Assume the initial state of 0, i.e. Not taken, for the predictor.

ii) What are the prediction accuracies for b1 and for b2?

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

iii) What is the overall prediction accuracy?

2. Using the code from Problem 2), assume a two-level branch prediction scheme is used. In addition to the 1-bit predictor, a 1-bit global register g is used. Register g stores the direction of the last branch executed (which may not be the same branch as the branch currently being predicted) and is used to index into two separate 1-bit branch history tables as shown in the following figure.



Depending on the value of g, one of the two BHTs is selected and used to do the 1-bit prediction as in Problem 1).

i) Again, fill in the predicted and actual branch directions of b1 and b2 for nine iterations of the loop. Assume the initial value of g=0, i.e. NT. For each prediction, depending on the current value of g, only one of the two BHTs is accessed and updated. Hence, some of the entries in the following table should be empty.

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

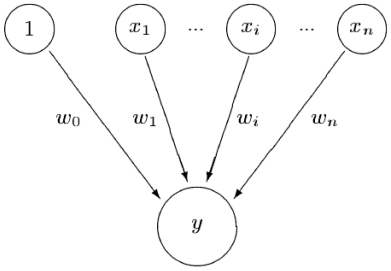
ii) What are the prediction accuracies for b1 and b2?

iii) What is overall prediction accuracy?

iv) What is the prediction accuracy of b2 when g = 0? Explain why.

4. Assume the following branch is predicted by a perceptron predictor such as the one discussed in class and in the paper on Canvas.

BGT R1, 3, TARG ; branch to TARG if R1 > 3



$$y = w_0 + \sum_{i=1}^{n} x_i w_i$$

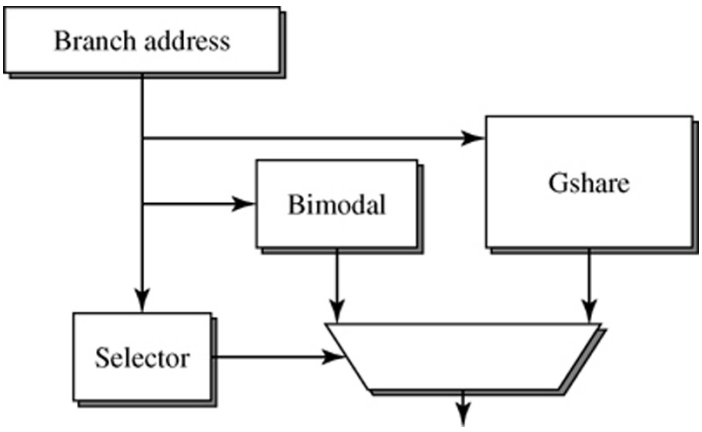
```
for each bit in parallel
    if t = x_i then
        w_i := w_i + 1
    else
        w_i := w_i − 1
    end if
```

Perceptron Model (from Jimenez and Lin)

For the following values of R1, show in a table, the values of the weights, the output Y, the prediction (Taken vs. Not taken) and the actual output. Assume four bits of history are used. Start with a history of 0000.

X	w0	w1	w2	w3	w4	y	Prediction	Outcome
0	0	0	0	0	0			
3								
6								
1								
4								
7								
2								
5								
0								
3								
6								
1								
4								
7								
2								
5								

5. Given a tournament branch predictor with a two-entry bimodal predictor and a gshare-indexed GAg predictor with a 1-bit BHR and two PHT entries and a two-entry selector table (to select between the bimodal and gshare), simulate a sequence of taken and not-taken branches as shown in the rows of the table, record the prediction make by the predictor before the branch is executed as well as any change to the predictor entries after the branch executes. Note, that bits of the branch address are used directly to index into the bimodal and selector tables and the branch address is used along with the BHR (following the gshare indexing method from McFarling 1993) to index into the gshare predictor. Note also the assumption bullet point below that relates to branch address bits and indexing.



Use the following assumptions:

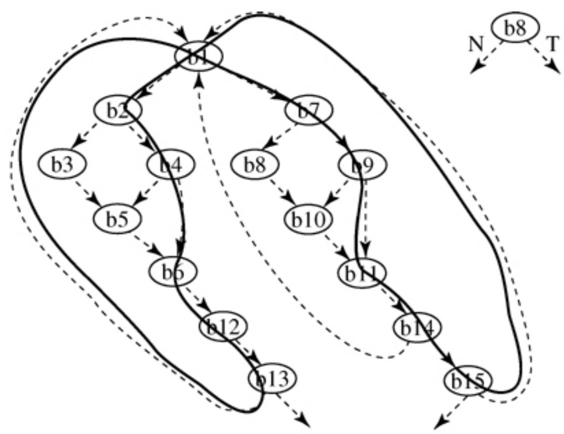
- Instructions are a fixed 4 bytes long; hence, the least-significant bits of the branch address are always 0 and are skipped when indexing into a predictor. Use the next lowest-order bit to index instead.
- Each predictor and selector entry is a simple saturating up-down 2-bit counter with the initial states shown. (Note, this is also called Automaton A2 from Yeh and Patt paper)
- A taken branch (T) increments the predictor entry; a not-taken branch (N) decrements the predictor entry.
- A predictor entry less than 2 (0 or 1) results in a not-taken prediction.
- A predictor entry greater than or equal to 2 (2 or 3) results in a taken prediction (T) prediction.
- A selector value of 0 or 1 selects the bimodal predictor, while a selector value of 2 or 3 selects the gshare predictor.
- Increment or decrement the selector value whenever one predictor (bimodal or gshare) is right and the other is wrong.
- Update both predictors for every branch regardless of which predictor was used to make the “combined” prediction.

					Predictor state after the branch is executed						
		Predicted Outcome			Selector		Bimodal		Gshare		
branch address	branch outcome (T / N)	bimodal	Gshare	Combined	PHT0	PHT1	PHT0	PHT1	BHR	PHT0	PHT1
Initial	N/A	N/A	N/A	N/A	2	0	0	2	0	2	1
0x654	N										
0x780	T										
0x78C	T										
0x990	T										
0xA04	N										
0x78C	N										

ii) Complete the overall branch prediction rates (number of correctly predicted branches / total number of predicted branches) for the bimodal, gshare, and final (combined) predictors.

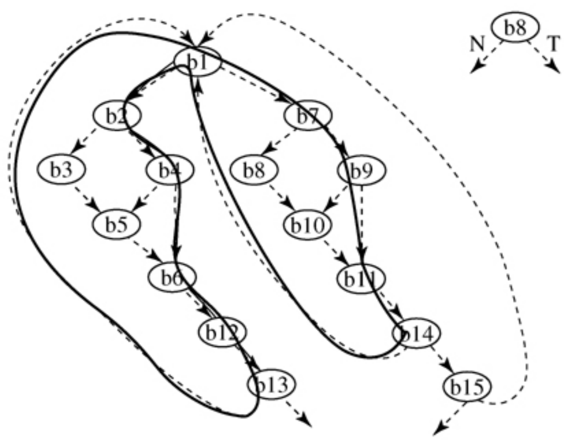
6. The figure shows the control flow graph of a simple program. The CFG is annotated with three different execution paths. For each execution path, circle which branch predictor (bimodal, local, or gselect). Note, local is another way of referring to a PAg predictor that uses separate local histories for each branch. Gselect is GAg prediction using gselect indexing (described in McFarling 1993).

More than one predictor may perform equally well on a particular trace. However, you are to use each of the three predictors *exactly once* in choosing the best predictor for the three traces. *Circle* your choice for each of the three traces. Assume each trace is executed many times and every node in the CFG is a conditional branch. The branch history registers in each of the predictors is limited to 4 bits.



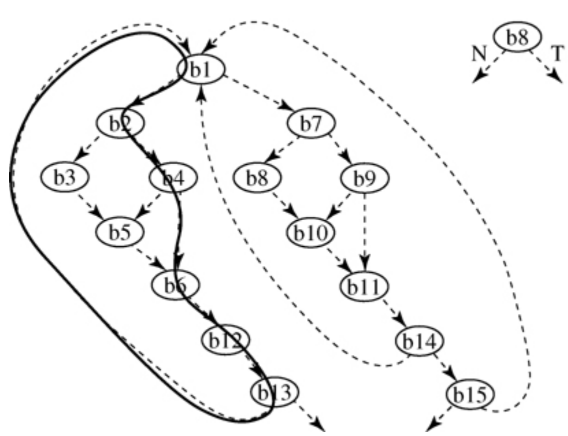
Circle one:

Bimodal
Local
Gselect



Circle one:

Bimodal
Local
Gselect



Circle one:

Bimodal
Local
Gselect