Problem M3.2: Branch Prediction [? Hours]

```
loop:
     LW   R4, 0(R3)
     ADDI R3, R3, 4
     SUBI R1, R1, 1
b1:
     BEQZ R4, b2
     ADDI R2, R2, 1
b2:
     BNEZ R1, loop
```

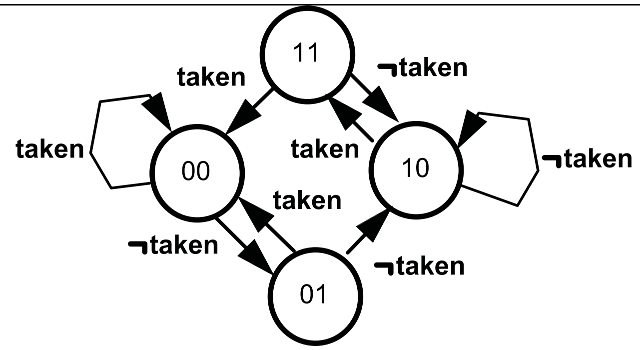

Figure M3.6-A: BP bits state diagram

Problem M3.2.A — Program

R2 contains the number of non-zero entries in the first n elements of array.

Problem M3.2.B — 2-bit branch prediction

There are 7 mispredicts (shown in bold italics).

System State		Branch Predictor		Branch Behavior	
PC	R3/R4	b1 bits	b2 bits	Predicted	Actual
b1	4/1	**10**	10	N	N
b2	4/1	*10*	**10**	N	***T***
b1	8/0	**10**	*11*	N	***T***
b2	8/0	*11*	**11**	N	***T***
b1	12/1	**11**	*00*	N	N
b2	12/1	*10*	**00**	T	T
b1	16/0	**10**	*00*	N	***T***
b2	16/0	*11*	**00**	T	T
b1	20/1	**11**	*00*	N	N
b2	20/1	*10*	**00**	T	T
b1	24/0	**10**	*00*	N	***T***
b2	24/0	*11*	**00**	T	T
b1	28/1	**11**	*00*	N	N
b2	28/1	*10*	**00**	T	T
b1	32/0	**10**	*00*	N	***T***
b2	32/0	*11*	**00**	T	***N***

Table M3.2-1

Problem M3.2.C — Branch prediction with one global history bit

There are 9 mispredicts (shown in bold italics).

System State			Branch Predictor				Behavior	
PC	R3/R4	history bit	b1 bits		b2 bits			
			set 0	set 1	set 0	set 1	Predicted	Actual
b1	4/1	**1**	10	**10**	10	10	N	N
b2	4/1	**0**	10	*10*	**10**	10	N	***T***
b1	8/0	**1**	10	**10**	*11*	10	N	***T***
b2	8/0	1	10	*11*	11	**10**	N	***T***
b1	12/1	1	10	**11**	11	*11*	N	N
b2	12/1	0	10	*10*	**11**	11	N	***T***
b1	16/0	1	10	**10**	*00*	11	N	***T***
b2	16/0	1	10	*11*	00	**11**	N	***T***
b1	20/1	1	10	**11**	00	*00*	N	N
b2	20/1	0	10	*10*	**00**	00	T	T
b1	24/0	1	10	**10**	*00*	00	N	***T***
b2	24/0	1	10	*11*	00	**00**	T	T
b1	28/1	1	10	**11**	00	*00*	N	N
b2	28/1	0	10	*10*	**00**	00	T	T
b1	32/0	1	10	**10**	*00*	00	N	***T***
b2	32/0	1	10	*11*	00	**00**	T	***N***

Table M3-2-2

Problem M3.2.D — Branch prediction with two global history bits

There are 7 mispredicts (shown in bold italics).

System State			Branch Predictor								Behavior	
PC	R3/R4	history	b1 bits				b2 bits					
		bits	set 00	set 01	set 10	set 11	set 00	set 01	set 10	set 11	Predicted	Actual
b1	4/1	**11**	10	10	10	**10**	10	10	10	10	N	N
b2	4/1	**01**	10	10	10	*10*	10	**10**	10	10	N	***T***
b1	8/0	**10**	10	10	**10**	10	10	*11*	10	10	N	***T***
b2	8/0	11	10	10	*11*	10	10	11	10	**10**	N	***T***
b1	12/1	11	10	10	11	**10**	10	11	10	*11*	N	N
b2	12/1	01	10	10	11	*10*	10	**11**	10	11	N	***T***
b1	16/0	10	10	10	**11**	10	10	*00*	10	11	N	***T***
b2	16/0	11	10	10	*00*	10	10	00	10	**11**	N	***T***
b1	20/1	11	10	10	00	**10**	10	00	10	*00*	N	N
b2	20/1	01	10	10	00	*10*	10	**00**	10	00	T	T
b1	24/0	10	10	10	**00**	10	10	*00*	10	00	T	T
b2	24/0	11	10	10	*00*	10	10	00	10	**00**	T	T
b1	28/1	11	10	10	00	**10**	10	00	10	*00*	N	N
b2	28/1	01	10	10	00	*10*	10	**00**	10	00	T	T
b1	32/0	10	10	10	**00**	10	10	*00*	10	00	T	T
b2	32/0	**11**	**10**	10	*00*	10	10	00	10	**00**	T	***N***

Table M3.2-3

Problem M3.2.E — Analysis I

The first thing to notice is that the more history bits we have, the longer it takes to get any correct prediction since we have to "train" the predictor. These start-up costs go up as the number of history bits increase.

Another thing to notice is that the single history bit does not help at all (even after we get into a steady-state phase). In both the single history bit and no history cases, the b2 branch is predicted correctly once we get past the start-up phase (since b2 is always taken). The single bit of history does not help since this history is too "nearsighted". The second history bit captures the alternating pattern of the b1 branch, and hence does not mispredict once it gets past the start-up phase. For a large n then, the 2-bit history predictor is the best.

The final point of observation is that all the predictors mispredict the fall-through case (the last b2 branch).

Problem M3.2.F — Analysis II

When the input is random, no prediction scheme will help predict whether b1 is taken or not. All three schemes will eventually predict b2 as always taken. However, the more history bits are used, the more sets need to be trained to predict the always taken for b2. Thus, the more history bits used, the more mispredicts of branch b2 will occur initially. The answer does not depend on the size of n. However, as n gets large, the start-up costs become insignificant among the three schemes.

The moral of the problem is that history bits are useful if there is a pattern among a sequence of branches. The longer this pattern is, the more history bits are needed to be able to recognize this pattern. If the pattern is not recognized, then global history bits can hurt because it take longer to train the branches that can be predicted correctly.

Problem M3.3: Branch Prediction [? Hours]

Problem M3.3.A

Cycle	Instruction Fetched	Branch Prediction	Prediction Correct?	Branch Predictor State		
				Branch History	Last Branch Taken Predictor	Last Branch Not Taken Predictor
0	-	-		T	TW	TW
1	1	T	N			
2	2					
3	4					
4	5	T	Y			
5	6			NT	NTR	
6	2					
7	3					
8	4					
9	5	T	Y			
10	6					
11	1	T	N			
12	2					
13	4			T		TR
14	5	NT	N			
15	6			NT		TW
16	2					
17	3					
18	4					
19	5	T	Y			
20	6					
21	1	T	N			
22						
23				T		TR
24						
25				NT		TW
26						

Problem M3.3.B

Cycle	Instruction Fetched	Branch Prediction	Prediction Correct?	Branch Predictor State		
				Branch History	Last Branch Taken Predictor	Last Branch Not Taken Predictor
0	-	-		T	TW	TW
1	1	T	N			
2	2			T		
3	4					
4	5	T	Y			
5	6			NT	NTR	
6	2					
7	3					
8	4					
9	5	T	Y			
10	6			T		
11	1	NT	Y			
12	2			NT		
13	3					TR
14	4					
15	5	T	Y		NTR	
16	6			T		
17	1	NT	Y			
18				NT		
19						TR
20						
21					NTR	
22						
23						
24						
25						
26						

Problem M3.4: Branch Prediction [? Hours]

Problem M3.4.A

	Predicted Taken?	Actually Taken?	Pipeline bubbles
BEQZ/ BNEZ	Y	Y	3
	Y	N	6
	N	Y	6
	N	N	0
J	Always taken (No lookup)	Y	3
JR	Always taken (No lookup)	Y	6

Problem M3.4.B

	BTB Hit?	(BHT) Predicted Taken?	Actually Taken?	Pipeline bubbles
Conditional Branches	Y	Y	Y	1
	Y	Y	N	6
	Y	N	Y	Cannot occur
	Y	N	N	Cannot occur
	N	Y	Y	3
	N	Y	N	6
	N	N	Y	6
	N	N	N	0

Problem M3.4.C

TIME →

Address	Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0x1000	BEQZ R5, NEXT	A	P	F	B	I	J	R	E															
0x1014	ADDI R1, R1, #1								A	P	F	B	I	J	R	E								
0x1018	SLTI R2, R1, 100									A	P	F	B	I	J	R	E							
0x101C	BNEZ R2, LOOP										A	P	F	B	I	J	R	E						
0x1000	BEQZ R5, NEXT												A	P	F	B	I	J	R	E				
0x1014	ADDI R1, R1, #1														A	P	F	B	I	J	R	E		

Problem M3.4.D

(Valid) V	Entry PC	Predicted Target PC
1	0x101c	0x1000
1	0x1000	0x1014

BTB

	...	
BR1	0	0
	...	
BR2	0	0
	...	

BHT