1. (8 points) The following code executes on the pipelines computer with branch in the ID stage with data forwarding, load-use stalls, branch flushing, and branch data stalling.

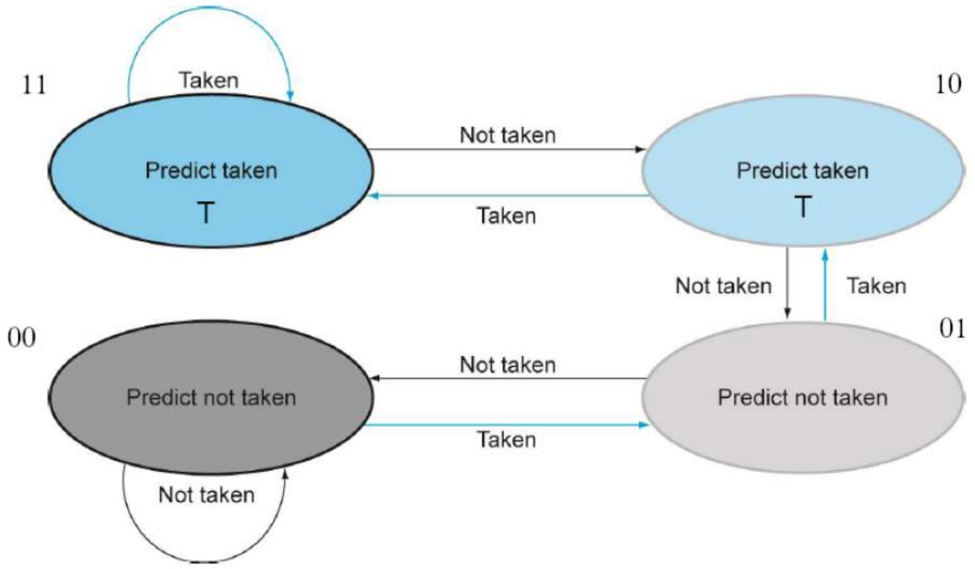
```
100 ADDI X1,X31,#20
104 ADDI X2,X31,#56
108 LDUR X5,[X2,#0]
112 ADDI X2,X2,#8
116 SUBI X1,X1,#1
120 CBNZ X1,#-3
124 ADDI X9,X5,X31
```

(a) (5 points) Compute how many correct/incorrect branch predictions are made using various schemes for branch prediction.

For all branch prediction methods, assume that the branch destination table has already been built and you begin with branch not taken for 1-bit prediction. The start state for 2-bit prediction is indicated below (see also the FSM on the following page).

	Branch Prediction	
Next branch instruction	Number Correct	Number Incorrect
PC+4	1	19
Branch destination	19	1
1-bit start at state 0	18	2
2-bit start at state 00	17	3
2-bit start at state 10	19	1

(b) (3 points) Here is the FSM for the 2-bit branch predictor:



```
100 ADDI X1,X31,#6
104 ADDI X2,X31,#56
108 LDUR X5,[X2,#0]
112 ADDI X2,X2,#8
116 SUBI X1,X1,#1
120 CBNZ X1,#-3
124 ADDI X9,X5,X31
```

Trace the state of the 2-bit branch predictor for line `120 CBNZ` in the code above. Assume that the FSM starts in state 00, and that it generates an output *T* that is high (1) when predicting that the branch should be taken. Note whether each branch prediction was correct (C) or incorrect (I). You should trace the code until it completes execution of line 124.

Current State	00	01	10	11	11	11
T	0	0	1	1	1	1
Correct/**I**ncorrect	I	I	C	C	C	I
Next State	01	10	11	11	11	10