0=not taken

1=taken

**Branch history**

**register**

(binary)

**Branch pattern**

**history**

(decimal)

**Outcome**

**Prediction**

**Branch**

Branch outcome

shifted in

Index: 3 2 1 0

Value:

0

0

0

0

X[i]

Initial state

X[i] > 0

0

0

0

3

0

1

0

2

0

1

0

1

0

1

0

1

1

1

1

0

0

0

1

1

1

1

0

0

0

0

1

1

0

1

0

0

0

0

1

1

0

1

0

0

0

0

0

0

1

0

1

0

1

0

2

0

1

0

3

0

1

0

0

3

0

1

-1

2

-3

4

-5

6

-7

8

-9

10

X[0]≤0

0

1

1

0

0

0

0

1

0

0

0

0

0

0

0

0

miss

X[1]≤0

X[2]≤0

X[8]≤0

X[9]≤0

miss

X[3]≤0

X[4]≤0

miss

X[5]≤0

X[6]≤0

X[7]≤0

**Algorithm for the branch prediction:**

For each iteration of the branch

1. Look up the prediction for the next iteration of the branch in the branch pattern history entry specified by the current value of the branch history register. Branch patter values of 0 and 1 (strongly not taken and not taken) give a prediction of 0  
   (not taken) and 2 and 3 (taken and strongly taken) give a prediction of 1 (taken).
2. Evaluate the outcome of the branch, (not taken = 0, taken = 1)
3. Update the pattern history state indexed by the current value of the history register by adding or subtracting 1 (depending on the outcome) with saturation arithmetic.
4. Shift in the outcome of the branch into the branch history register.