מבנה מחשבים 046267

תרגול מס' 4

2-Level
Branch Prediction

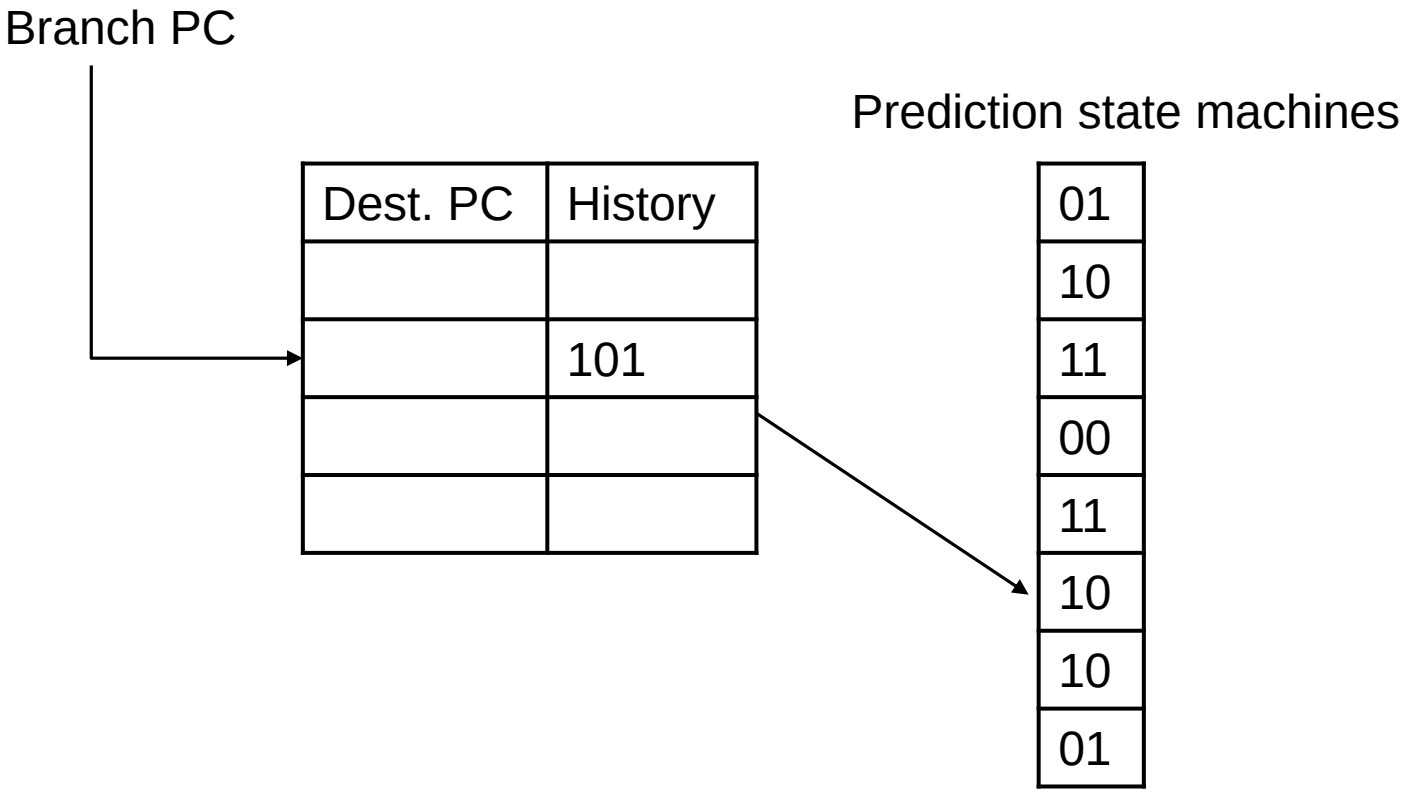
מוטיבציה

- חזאי הסיעוף שנלמד עד כה לא מבחין ב"הקשר" של פקודת קפיצה מסוימת – איך הגענו אליה
- לעיתים הרצף של הקפיצות משפיע:
  - אם יש לולאה קצרה בתוך לולאה ארוכה, התחקות אחרי ההיסטוריה עשויה לעלות על התבנית
  - סדרה של הוראות סיעוף התלויות אחת בשנייה (למשל switch-case) - ניתן אולי ללמוד על התלות
  - הגעה לתנאי מנתיבים שונים בתוכנית
- התאמה בין רצף הקפיצות לחיזוי עובד סטטיסטית...

קישור בין רצף לחיזוי

- ישנם שני סטים – טבלאות:
  - סט של היסטוריות הכרעות סיעוף – History (מסיכה של n ביטים – 0/1 עבור n הכרעות סיעוף)
  - סט של מצבים (מכונת מצבים) - State
- ישנו מיפוי בין שני הסטים – על כן נקרא **2-level**
- <u>במקרה הפשוט</u>:
  - במקום להחזיק במכונת מצבים אחת לכל הוראת סיעוף, מחזיקים 2^n מכונות מצבים
  - בוחרים במכונת חיזוי מבין 2^n בהתאם לערך מסיכת ההיסטוריה באורך n

דוגמה פשוטה


Branch PC
Prediction state machines
Dest. PC
History
101
01
10
11
00
11
10
10
01

איפה ה- catch?

- רוצים להחזיק יותר מכונות מצבים ← יותר זיכרון
- מנגנון מסובך יותר ← עשוי לקחת יותר זמן ולצרוך יותר משאבים

- למרות זאת, במעבדים מודרניים המחיר משתלם:

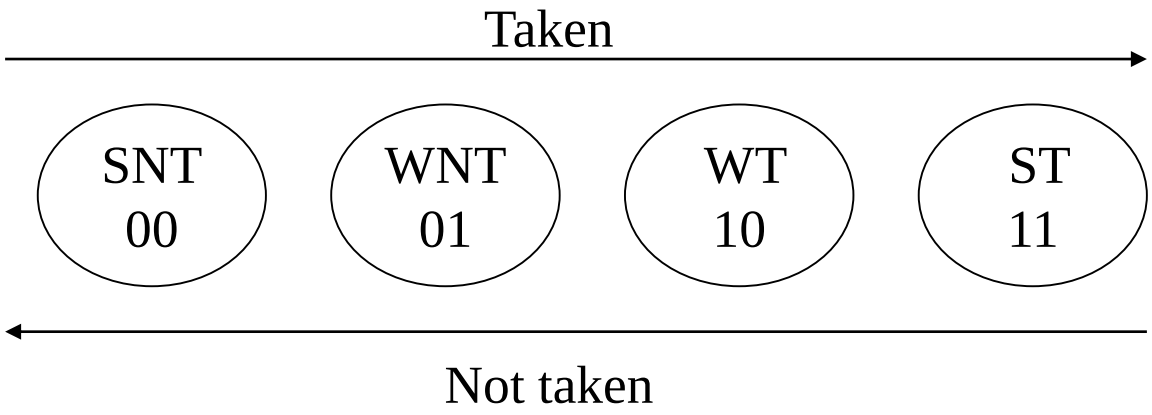
  על מנת לתכנן עבור מחזור שעון קצר יותר
  ← pipeline עמוק יותר
  ← כל חיזוי שגוי "יקר" יותר (יותר פקודות נזרקות)

BHR: Branch History Register

- בכל רגע נתון מחזיקים ב-**shift register** בן n ביטים אשר מציין את ההיסטוריה (0 ל-not taken ו-1 ל-taken), למשל עבור 5 ביטים:

איך זה עובד?

- ישנה טבלה של מכונות מצבים, למשל 2 bit counter:



- אם יש לנו **BHR** בן n ביטים, אז גודל הטבלה 2^n
- בוחרים בכניסה המתאימה ע"י ערך ה-BHR, חוזים לפי הערך שנמצא שם, ולבסוף כשיודעים את התוצאה מעדכנים את אותה המכונה בהתאם

Example: The initialization phase

First level: shift register that keeps the history of the n last branches

1	1	1	0

ST	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

We update the table based on current state (0) and modify the state AFTER the outcome is known

Second level: prediction table that predicts for every history state if the direction should be taken or not taken

```
x=0;
for (i=100; i!=0; i--)
    for (j=4; j!=0;j--)
        x=x+j+i;
```

```
100     sub r10,r10,r10
104     movi r9, 100
108 L1:movi r8,4
112 L2:add r10,r10,r9
116     add r10,r10,r8
120     subi r8,r8,1
124     if (r8 != r0) L2
128     subi r9,r9,1
132     if (r9 != r0) L1
```

2 1 3

Example

BHR	History
14	1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0
13	1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0
11	1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0
7	1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0
14	1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0

- אם נסתכל בערכי BHR האפשריים עבור תוכנית זו, נבחין שעבור הערכים 11,13,14 הסיעוף יילקח תמיד ועבור הערך 7 הוא לא יילקח
- אם רצף של היסטוריות חוזר על עצמו והחיזוי לא משתנה עבור כל אחת מתבניות היסטוריה, נאמר שהמערכת במצב יציב

סוגי BHR

- <u>Global BHR</u>

ניתן להחזיק BHR אחד גלובלי (עבור כל פקודות הסיעוף)
ואז ההיסטוריה היא כללית
(עשוי להיות יעיל בתפיסת תלויות בין הוראות branch)

- <u>Per Branch (Local) BHR</u>

לחילופין, BHR לכל הוראת סיעוף ואז ה-BHR לוקאלי
(עשוי לעלות על תבנית חוזרת של הוראות branch - למשל לולאות מקוננות)

סוגי טבלת מכונות החיזוי

- <u>Global Table</u>

ניתן להחזיק בטבלה אחת גלובלית ובכך לחסוך במקום ולאפשר אולי BHR-ים ארוכים יותר (שימו לב שהטבלה גדלה אקספוננציאלית)

- <u>Per Branch (Local) Table</u>

להחזיק טבלה לוקאלית לכל הוראת branch

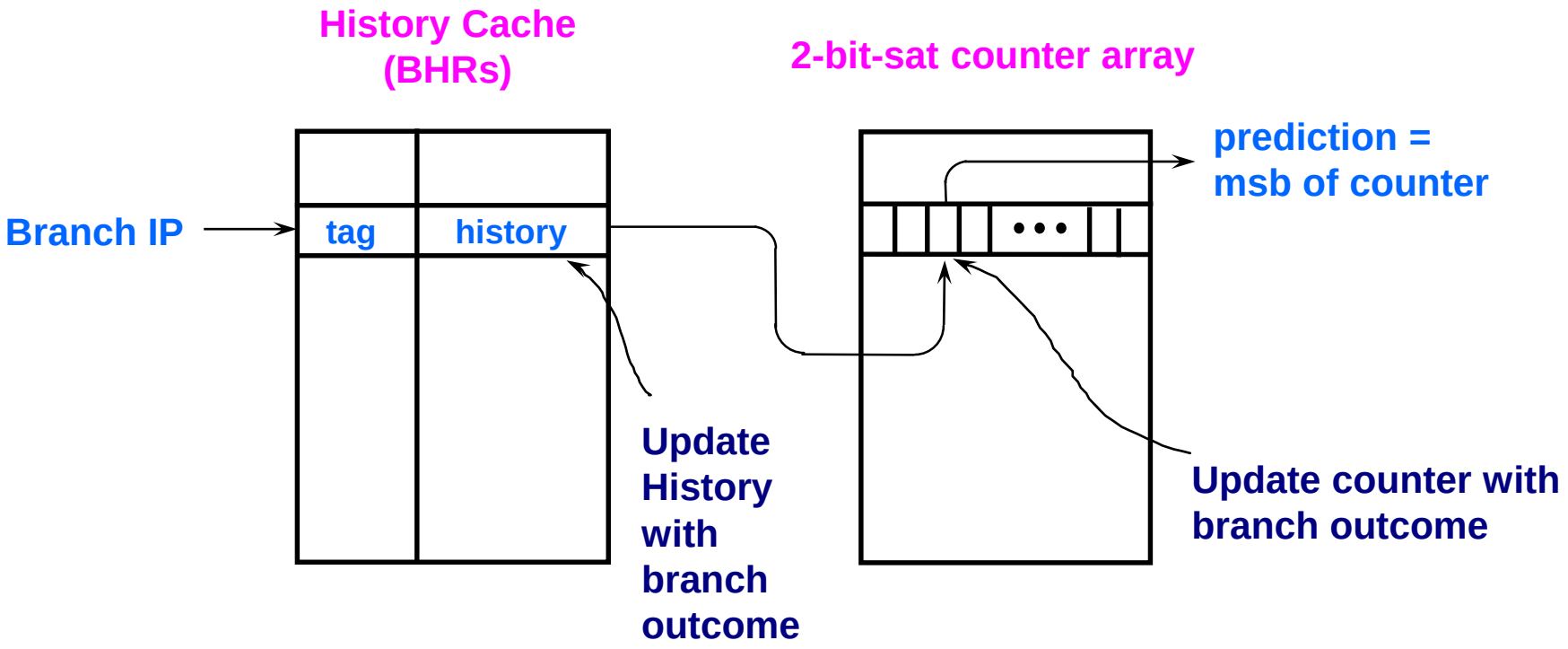
L2 Predictor

- במקרה של היסטוריה ו/או טבלאות חיזוי לוקאליות, נחזיק טבלה עם הערכים המתאימים

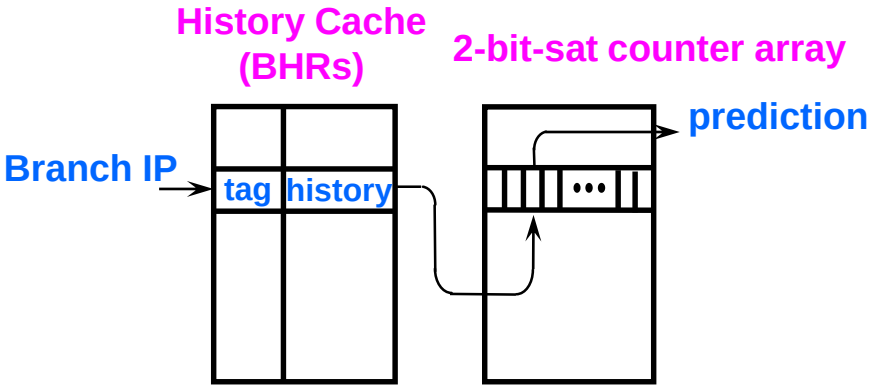
- הכניסה המתאימה תחזיק את הכתובת של הוראת ה-branch (שדה ה-tag)
  - אם ההיסטוריה לוקאלית אז את ההיסטוריה של אותה פקודה
  - אם טבלאות החיזוי לוקאליות אז את הטבלה המתאימה לפקודה

- הרחבת העיקרון של BTB רגיל

ובמילים אחרות...
Local Predictor / Local Counter Array



טבלה ו-BHR **לוקלים**



- נניח שברשותנו טבלת היסטוריות בת 1024 כניסות, ואנו לוקחים היסטוריה של 4 הוראות.
- כמו כן נניח שכתובת היא בת 32 סיביות וכל ההוראות הן aligned כך שניתן להשמיט את שתי הסיביות ה-lsb.
- **מהו גודל החזאי?**

The predictor size:

#entries * (tag_size + target_size + history_size + 2*2 $^{\text{history\_size}}$)

#entries = 1024

tag_size *or* target_size = 32 – 2 = 30 bit , history_size = 4

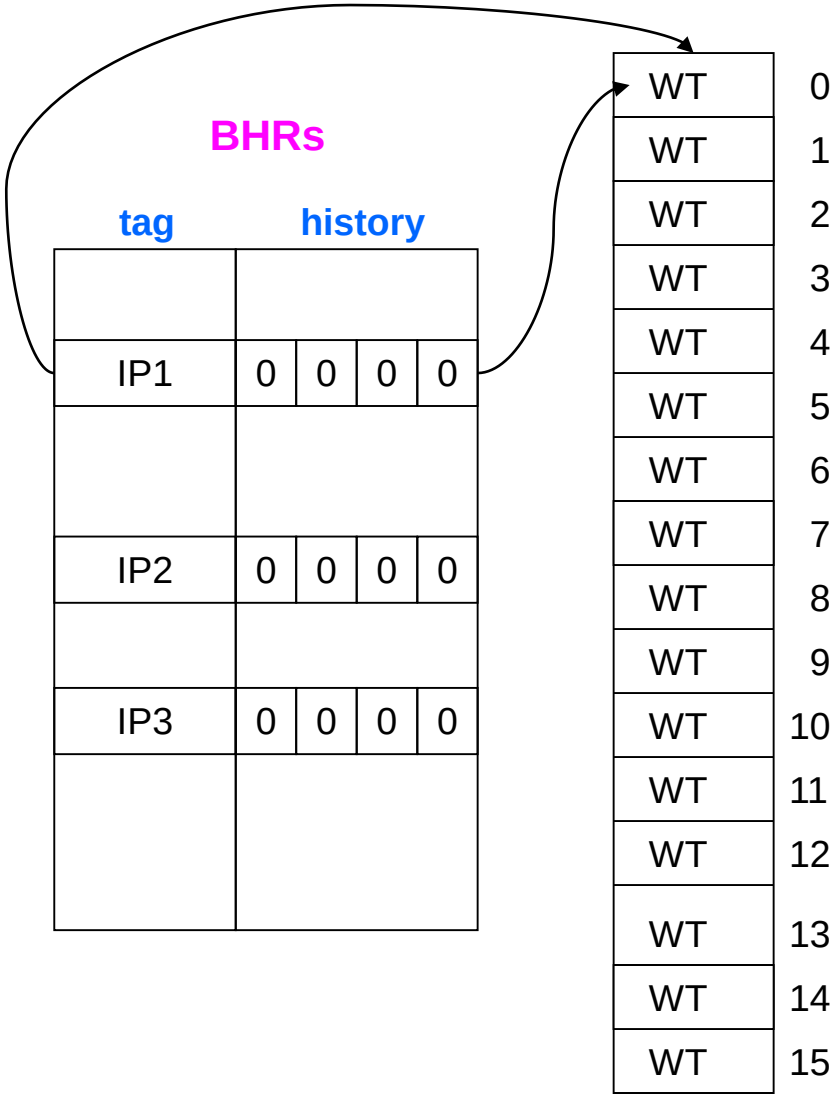
=> size= 1024 * (30 + 30 + 4 + 2*2^4) = 96 K bits

קטע קוד

```
for (i=100; i>0; i--)
     for (j=2; j<5; j++)
          if (i%j == 0) ...
```

```
     addi      r5, r0,  5
     addi      r1,r0,100
L1:  addi      r2, r0,  2
L2:  mod       r3, r1, r2
     bne       r3, r0, ENDIF
...
ENDIF:addi     r2, r2,  1
     bne       r2, r5, L2
     subi      r1, r1,  1
     bne       r1, r0, L1
```

דוגמת הרצה:

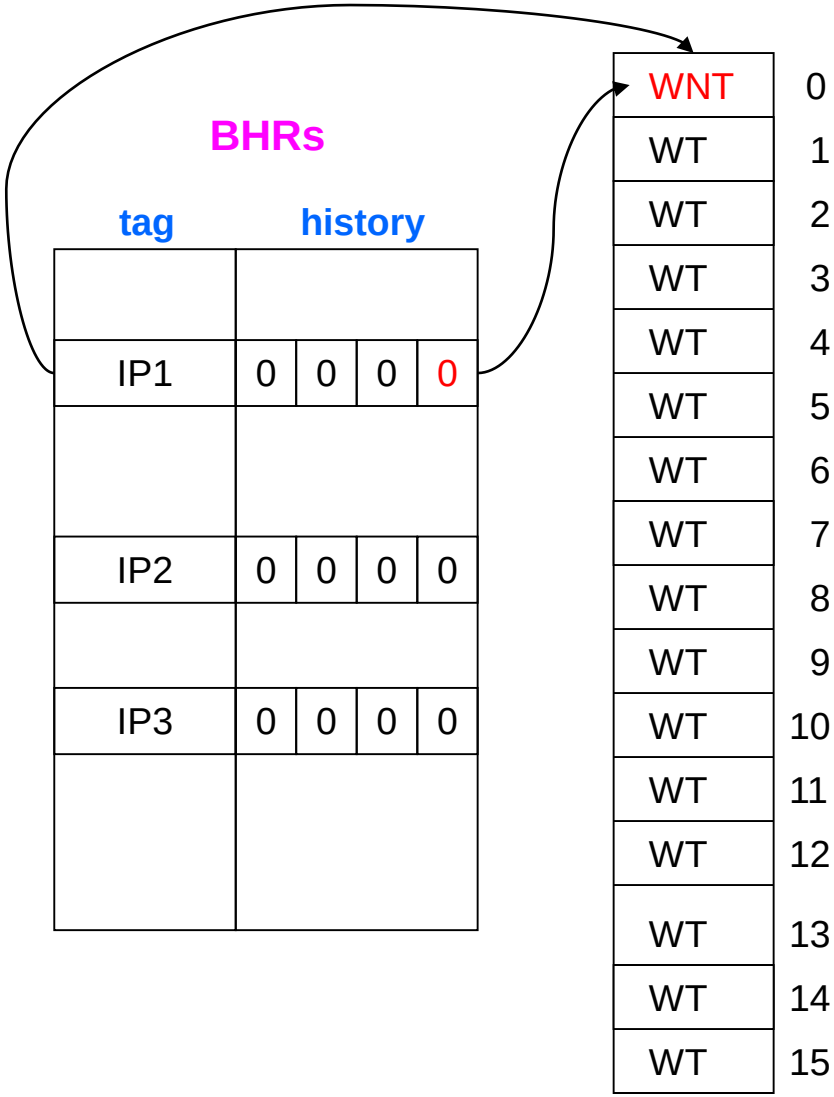


r1	**100**
r2	**2**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

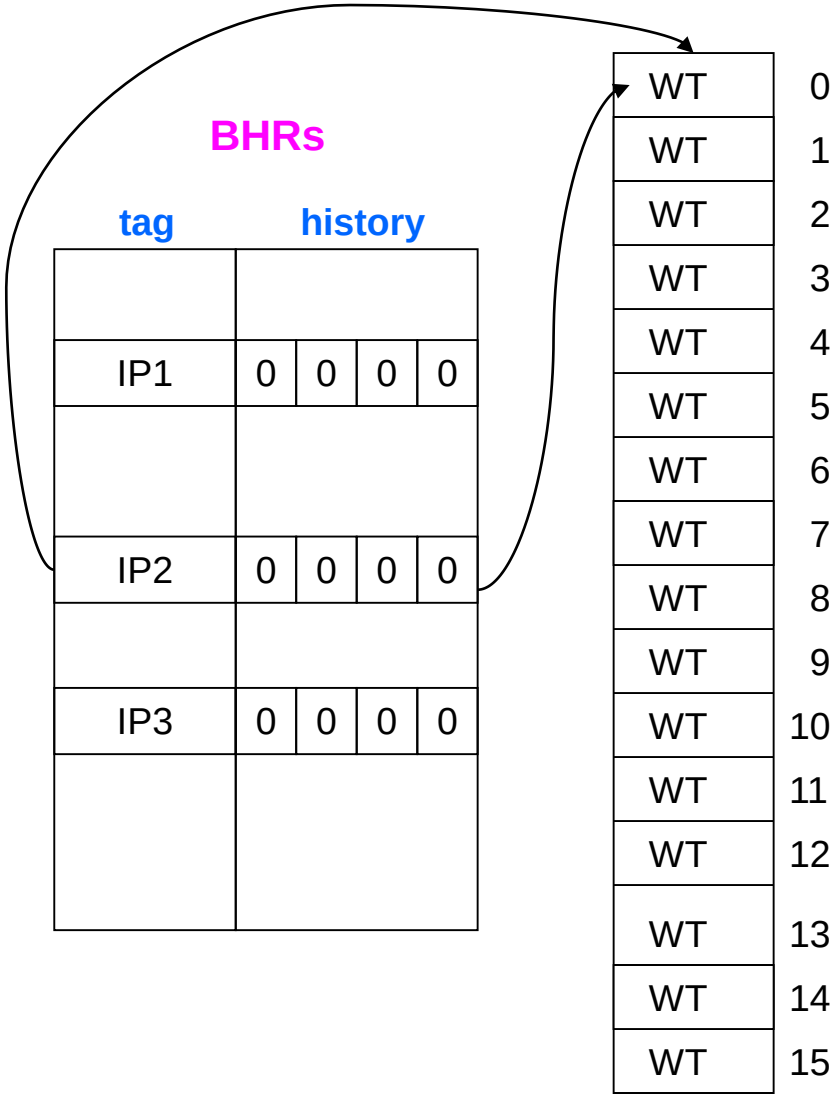
IP1 IP2 IP3

טעות בחיזוי (לא הייתה קפיצה)



r1	**100**
r2	**2**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                 . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```



r1	**100**
r2	**3**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                   . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

חיזוי נכון (הייתה קפיצה)

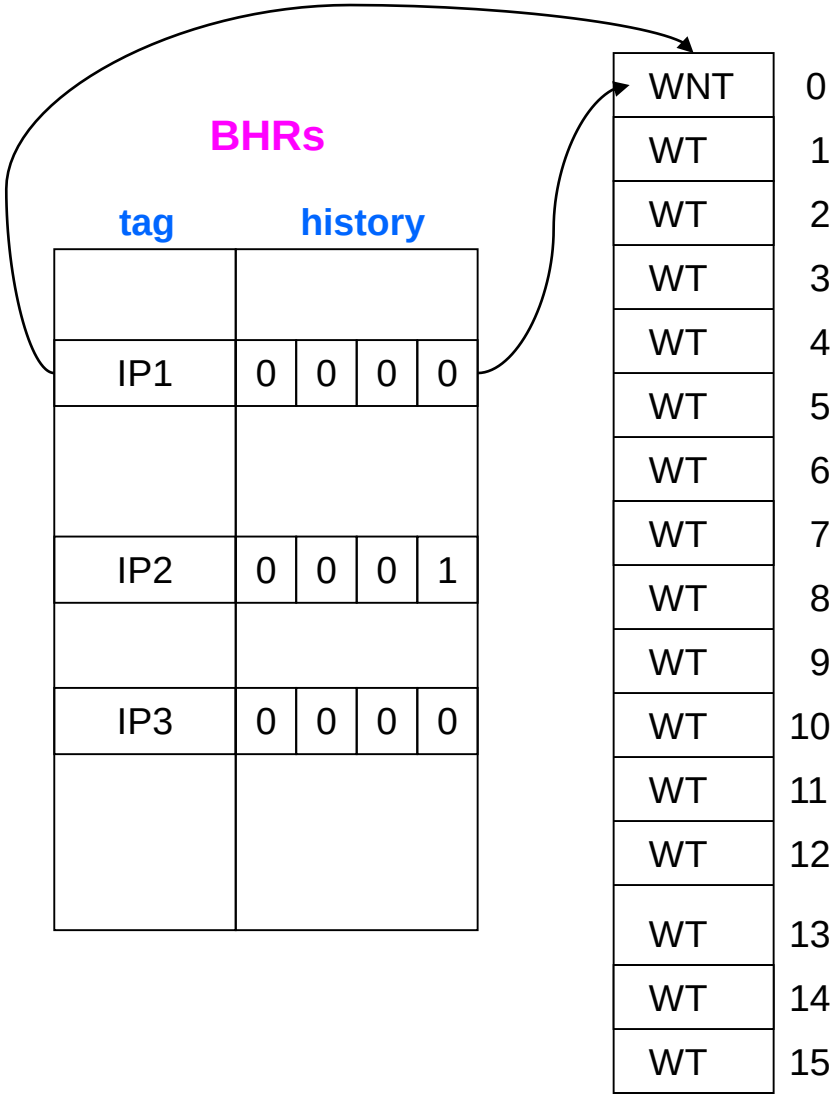
BHRs

tag	history			
IP1	0	0	0	0
IP2	0	0	0	1
IP3	0	0	0	0

ST	0
WT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**3**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```



r1	**100**
r2	**3**
r3	**1**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

טעות בחיזוי (הייתה קפיצה)

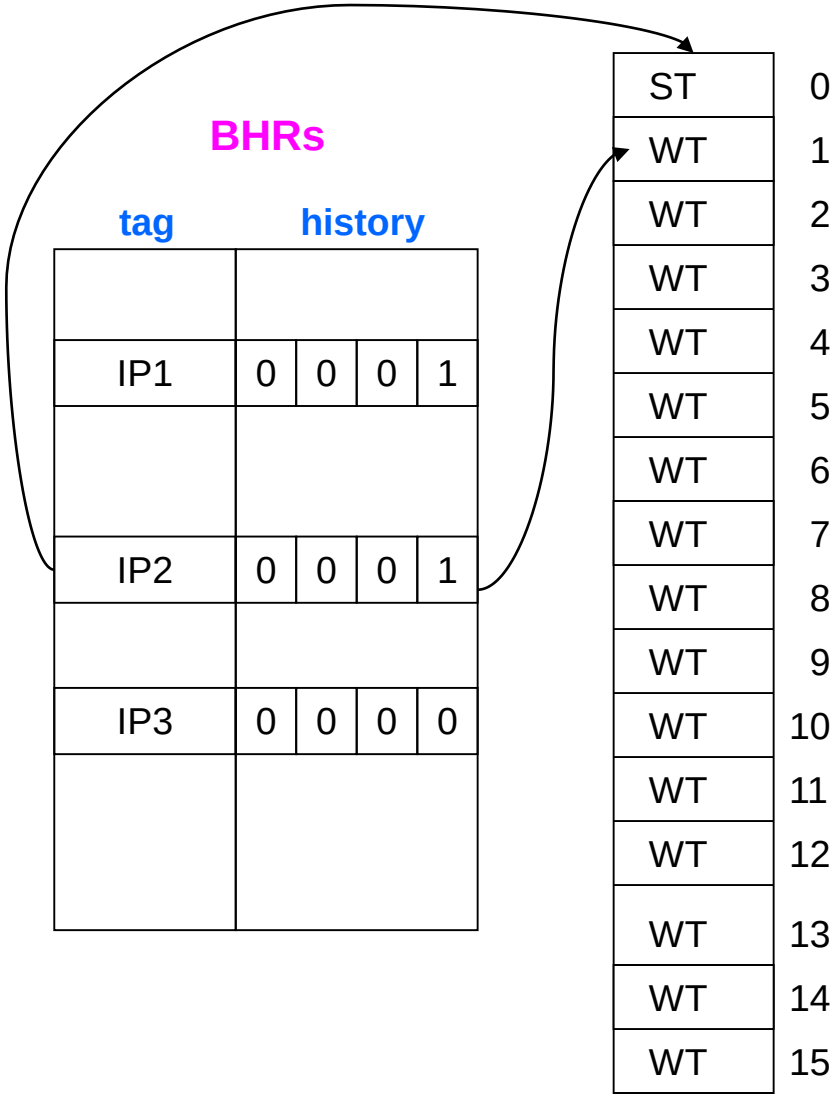
BHRs

tag	history			
IP1	0	0	0	1
IP2	0	0	0	1
IP3	0	0	0	0

WT	0
WT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**3**
r3	**1**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                 . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```



r1	**100**
r2	**4**
r3	**1**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
L2:  Mod    r3, r1, r2
     Bne    r3, r0, IF
                 . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
     Bne    r1, r0, L1
```

חיזוי נכון (הייתה קפיצה)

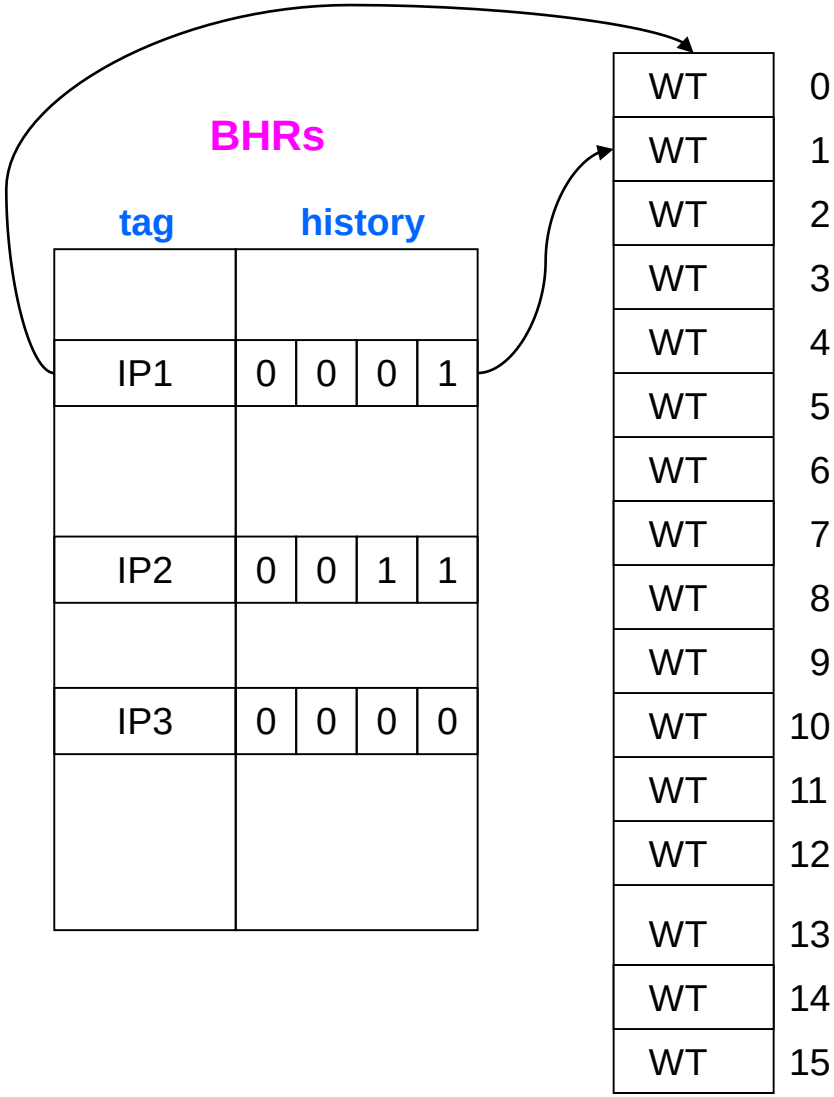
BHRs

tag	history			
IP1	0	0	0	1
IP2	0	0	1	1
IP3	0	0	0	0

ST	0
ST	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**4**
r3	**1**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
L2:  Mod    r3, r1, r2
     Bne    r3, r0, IF
                . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
     Bne    r1, r0, L1
```



r1	**100**
r2	**4**
r3	**0**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
L2:  Mod    r3, r1, r2
     Bne    r3, r0, IF
                 . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
     Bne    r1, r0, L1
```

טעות בחיזוי (לא הייתה קפיצה)

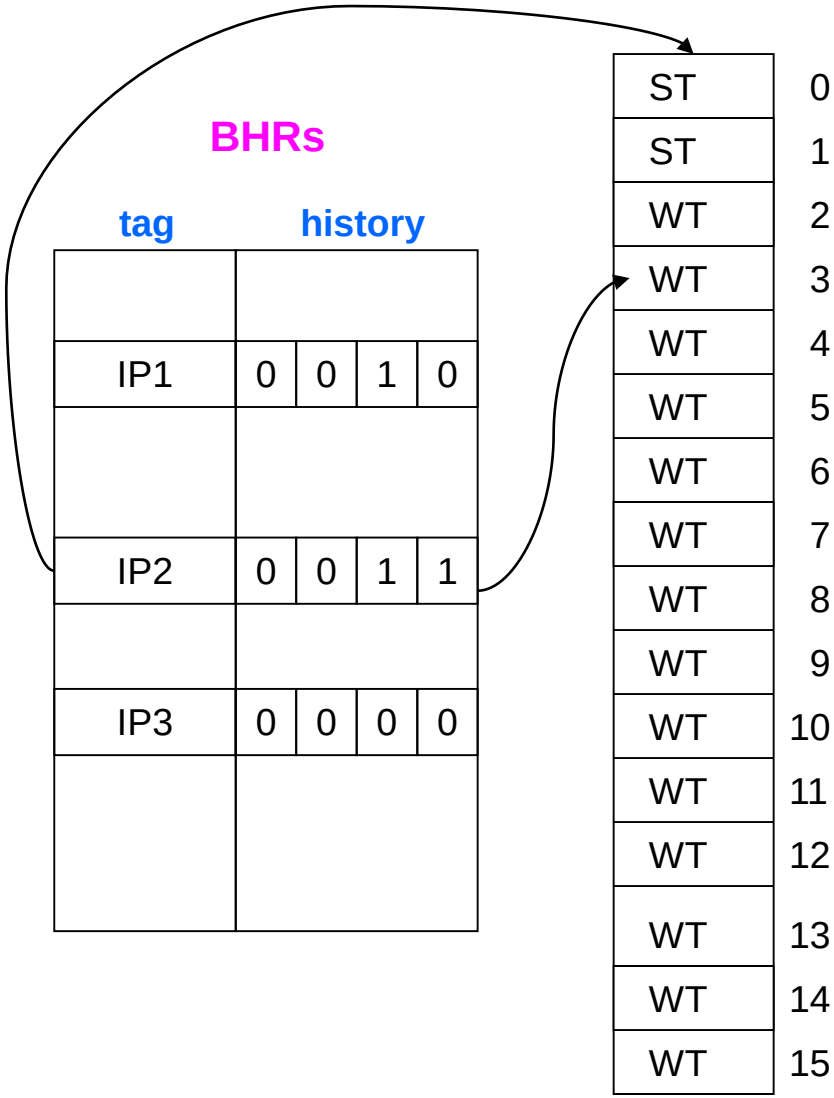
BHRs

tag	history			
IP1	0	0	1	0
IP2	0	0	1	1
IP3	0	0	0	0

WT	0
WNT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**4**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                 . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```



r1	**100**
r2	**5**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
L2:   Mod    r3, r1, r2
      Bne    r3, r0, IF
                   . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

טעות בחיזוי (לא הייתה קפיצה)

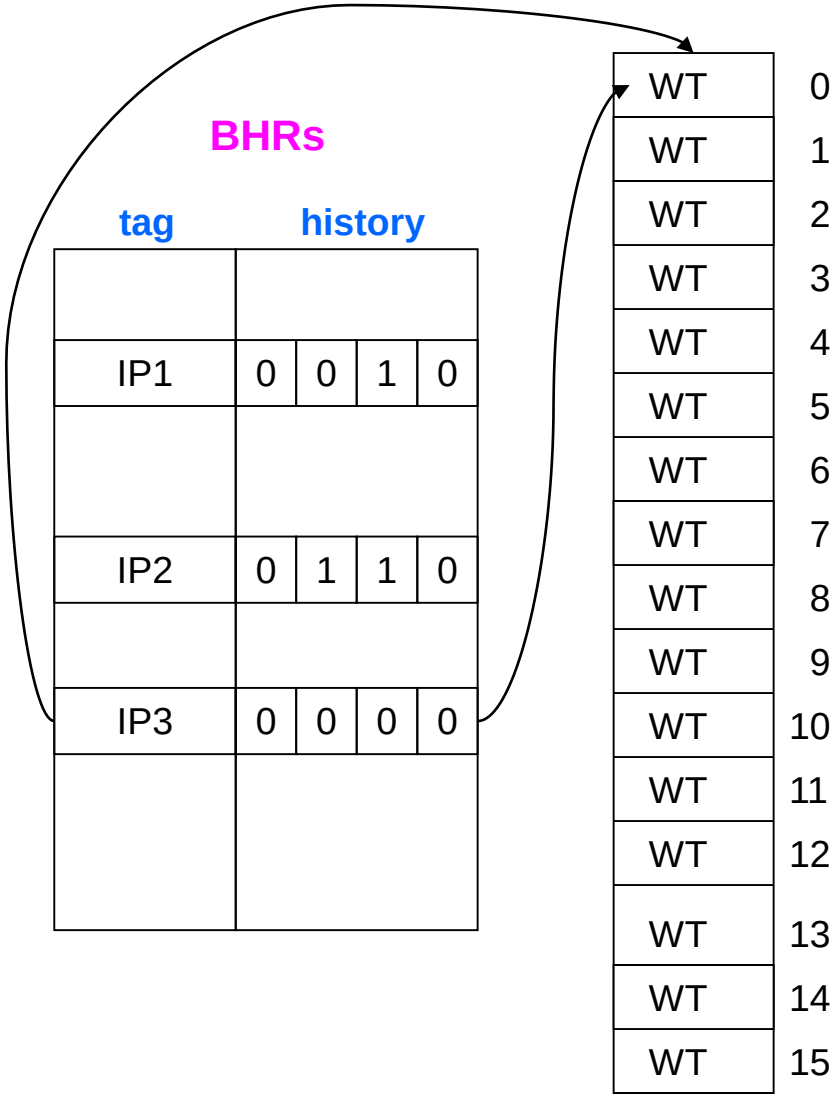
BHRs

tag	history			
IP1	0	0	1	0
IP2	0	1	1	0
IP3	0	0	0	0

ST	0
ST	1
WT	2
WNT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**5**
r3	**0**

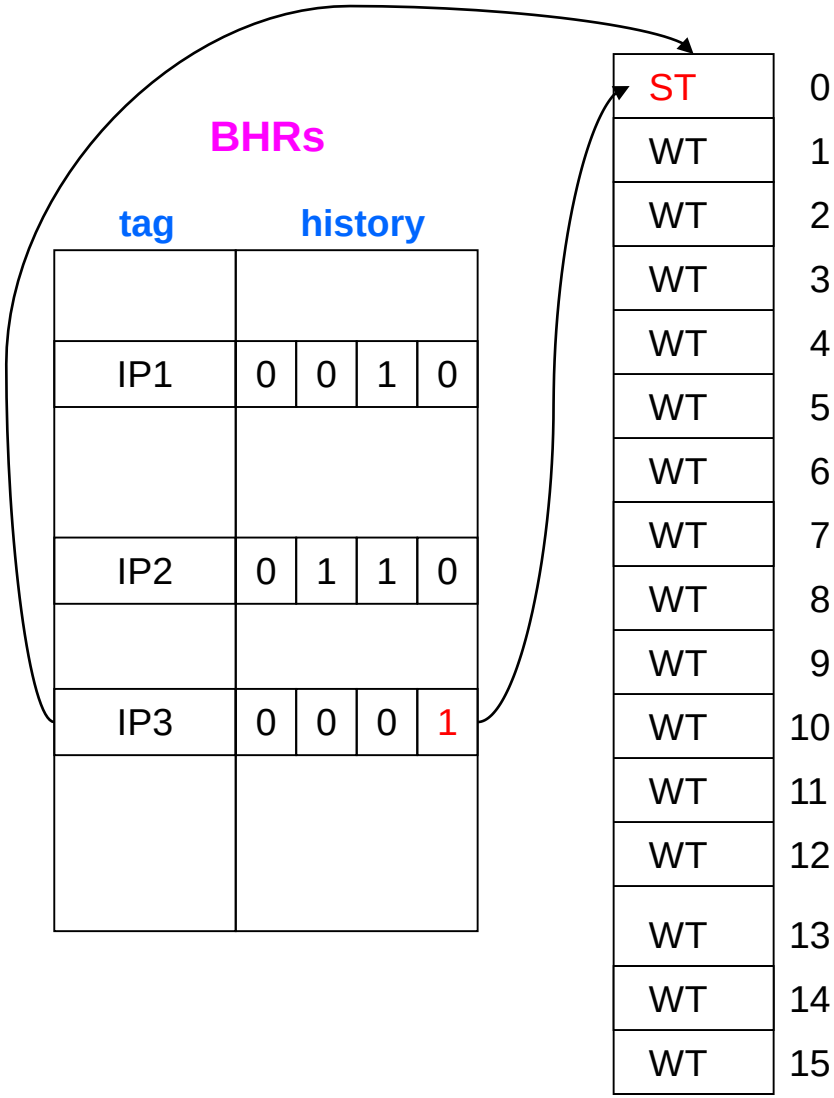
```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
L2:  Mod    r3, r1, r2
     Bne    r3, r0, IF
                . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
     Bne    r1, r0, L1
```



r1	**99**
r2	**5**
r3	**0**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
L2:  Mod    r3, r1, r2
     Bne    r3, r0, IF
                 . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
→    Bne    r1, r0, L1
```

חיזוי נכון (הייתה קפיצה)



r1	**99**
r2	**5**
r3	**0**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
L2:  Mod    r3, r1, r2
     Bne    r3, r0, IF
                . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
→    Bne    r1, r0, L1
```

טבלה ו-BHR **גלובליים**

- נשים לב שכעת אנו משתמשים ב- BHR יחיד, וכן בטבלה בודדת.

- אם נניח שבדיקת כתובת הקפיצה מתבצעת בנפרד ב- BTB (כמו שבעצם הנחנו גם קודם), וכן שאנו רוצים היסטוריה של 4 הוראות.
  מהו גודל הזיכרון הדרוש?
  גודל החזאי:

#entries * (tag_size + target_size) + (history_size + 2*2 $^{history\_size}$)

history_size = 4

→ size= 1024 * (30 + 30) + (4+2*2^4) = 60Kb + 36 bits

vs. 96Kb for local BHR and predictors tables

→ Save ~36Kb

דוגמת הרצה:

Global table

WT	0
WT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

0	0	0	0

r1	**100**
r2	**2**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

IP1

IP2

IP3

טעות בחיזוי (לא הייתה קפיצה)

Global table

WNT	0
WT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

0	0	0	0

r1	**100**
r2	**2**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

Global table

WNT	0
WT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

0	0	0	0

r1	**100**
r2	**3**
r3	**0**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
     Mod    r3, r1, r2
L2:  Bne    r3, r0, IF
                  . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
     Bne    r1, r0, L1
```

טעות בחיזוי (הייתה קפיצה)

Global table

WT	0
WT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

0	0	0	1

r1	**100**
r2	**3**
r3	**0**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
     Mod    r3, r1, r2
L2:  Bne    r3, r0, IF
                 . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
     Bne    r1, r0, L1
```

Global table

WT	0
WT	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

0	0	0	1

r1	**100**
r2	**3**
r3	**1**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

חיזוי נכון (הייתה קפיצה)

Global table

WT	0
ST	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

0	0	1	1

r1	**100**
r2	**3**
r3	**1**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

Global BHR

0	0	1	1

Global table

WT	0
ST	1
WT	2
WT	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**4**
r3	**1**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

חיזוי נכון (הייתה קפיצה)

Global table

WT	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

0	1	1	1

r1	**100**
r2	**4**
r3	**1**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

Global BHR

0	1	1	1

Global table

WT	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	6
WT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**4**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

טעות בחיזוי (לא הייתה קפיצה)

Global table

WT	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	6
WNT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

Global BHR

1	1	1	0

r1	**100**
r2	**4**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

Global BHR

1	1	1	0

Global table

WT	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	6
WNT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WT	14
WT	15

r1	**100**
r2	**5**
r3	**0**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
     Mod    r3, r1, r2
L2:  Bne    r3, r0, IF
                  . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
     Bne    r1, r0, L1
```

טעות בחיזוי (לא הייתה קפיצה)

Global BHR

1	1	0	0

Global table

WT	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	6
WNT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WNT	14
WT	15

r1	**100**
r2	**5**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
→     Bne    r2, r5, L2
      Subi   r1, r1,  1
      Bne    r1, r0, L1
```

Global BHR

1	1	0	0

Global table

WT	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	6
WNT	7
WT	8
WT	9
WT	10
WT	11
WT	12
WT	13
WNT	14
WT	15

r1	**99**
r2	**5**
r3	**0**

```
      Addi   r5, r0,  5
      Addi   r1,r0,100
L1:   Addi   r2, r0,  2
      Mod    r3, r1, r2
L2:   Bne    r3, r0, IF
                  . . .
IF:   Addi   r2, r2,  1
      Bne    r2, r5, L2
      Subi   r1, r1,  1
→     Bne    r1, r0, L1
```

חיזוי נכון (הייתה קפיצה)

Global table

WT	0
ST	1
WT	2
ST	3
WT	4
WT	5
WT	6
WNT	7
WT	8
WT	9
WT	10
WT	11
ST	12
WT	13
WNT	14
WT	15

Global BHR

1	0	0	1

r1	**99**
r2	**5**
r3	**0**

```
     Addi   r5, r0,  5
     Addi   r1,r0,100
L1:  Addi   r2, r0,  2
     Mod    r3, r1, r2
L2:  Bne    r3, r0, IF
                 . . .
IF:  Addi   r2, r2,  1
     Bne    r2, r5, L2
     Subi   r1, r1,  1
→    Bne    r1, r0, L1
```

דוגמא:

```
for (i=100; i>0; i--)      ← 0
    for (j=2; j<6; j++)    ← 1
        switch (i%j) {
        case 0: … break;   ← 2
        case 1: … break;   ← 3
        case 2: … break;   ← 4
        case 3: … break;   ← 5
        case 4: … break;   ← 6
        }
```

התנהגות ה-branch-ים השונים

- TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
  TTTTTTTTTTTTTN **- 0**
- TTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTT
  NTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTT
  TNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNT
  TTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTN
  TTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTNTTTN **- 1**
- TTTTNTTTTNTTNTNTTTTNNNTTTTTTNTNTTNTTNTTNTTTTNNNTTTTTNTTTTNTNNTNTTTTTNNTTTTTTNTNNTNT
  TNTTTTTTTNNNTTTTNNTTTTNTTNTNTTTTTNNTNTTTTNTNTTNTTNTTTTTTNNNNTTTTTNTTTTNTTNTNNTTTTN
  NTTTTTTNTNTTNTNNTTTTTTTNNNTTTTTNTTNTNTTNTNTTTTTNNTTTTTNNTNTTNTTNTTTTTTTNNNNTTTTNTTT
  TNTTNTNTTTTNNNTTTTTTNTNTTNTTNTTNTTTTNNNTTTTTNTTTTNTNNTNTTTTTNNTTTTTTNTNNTNTTNTTTTTT
  TNNNTTTTNNTTTTNTTNTNTTTTTNNTNTTTTNTNTTNTTNTTTTTTNNNNTTTTTNTTTTNTTNTNN **- 2**
- NNNNTTTTNTTTTNTTNTNTTTTNNNTTTTTTNTNTTNTTNTTNTTTTNNNTTTTTNTTTTNTNNTNTTTTTNNTTTTTTNT
  NNTNTTNTTTTTTTNNNTTTTNNTTTTNTTNTNTTTTTNNTNTTTTNTNTTNTTNTTTTTTNNNNTTTTTNTTTTNTTNTNN
  TTTTNNTTTTTTNTNTTNTNNTTTTTTTNNNTTTTTNTTNTNTTNTNTTTTTNNTTTTTNNTNTTNTTNTTTTTTTNNNNTTT
  TNTTTTNTTNTNTTTTNNNTTTTTTNTNTTNTTNTTNTTTTNNNTTTTTNTTTTNTNNTNTTTTTNNTTTTTTNTNNTNTTN
  TTTTTTTNNNTTTTNNTTTTNTTNTNTTTTTNNTNTTTTNTNTTNTTNTTTTTTNNNNTTTTTNTTTTNTT **- 3**
- TTTTTNNNTTTTTTTTTNTTTTNTTTTNTNTTTTTTTTNTTNTTTTTNTTTTTNNTTTTTTTTTTNTNTTNTTTTTTNTTTTTT
  TTNNTNTTTTTTTTTTTTNNTTTTNTTTTTNTTTTNTTTTTTNTNTTTTTTNTTNTTTTTTTTTNTNNTTTTTTTTTTTNTTTTNN
  TTTTTNTTTTTTTTNTTNTNTTTTTTTTTTNNTTTTTTTTNTNTTTTNTTTTTTNTTTTTNTTNTTNTTTTTTTTTTTNNNTTTT
  TTTTTNTTTTNTTTTNTNTTTTTTTTTNTTNTTTTTNTTTTTNNTTTTTTTTTTTNTNTTNTTTTTTNTTTTTTTTTNNTNTTTTTT
  TTTTTNNTTTTNTTTTTNTTTTNTTTTTTNTNTTTTTTTNTTNTTTTTTTTTTNTNNTTTTTTTTT **- 4**
- TTTTTTTTTTNNTTTTTTTTTTTTTTTNTTTTNTTTTTTTTTTTNTTTTTTTTTNTTTTTTNTTTTTTTTTTTTTNTTNTTTTTTTTTT
  TTTTTNNTTTTTTTTTTTTTTTNTTTTNTTTTTTTTTTTNTTTTTTTTTNTTTTTTNTTTTTTTTTTTTTNTTNTTTTTTTTTTTTTTT
  NNTTTTTTTTTTTTTTTNTTTTNTTTTTTTTTTTNTTTTTTTTTNTTTTTTNTTTTTTTTTTTTTNTTNTTTTTTTTTTTTTTTTNNTTT
  TTTTTTTTTTTNTTTTNTTTTTTTTTTTNTTTTTTTTTNTTTTTTNTTTTTTTTTTTTNTTNTTTTTTTTTTTTTTTTNNTTTTTTTT
  TTTTTTNTTTTNTTTTTTTTTTTNTTTTTTTTTNTTTTTTNTTTTTTTTTTTTTNTTNTTTTTT **- 5**
- TTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTT
  TTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTT
  TTTNTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTT
  TTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTT
  TTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTTTTTTTTTTTTTTTTTNTTTTT **- 6**

תוצאות חיזוי נכון (היסטוריה של 5 הוראות)

2 bit counter:
0 : 0.98
1 : 0.7475
2 : 0.605
3 : 0.61
4 : 0.7675
5 : 0.8725
6 : 0.9475
avg: 0.7672

local history & tables:
0 : 0.99
1 : 0.995
2 : 0.6025
3 : 0.605
4 : 0.76
5 : 0.885
6 : 0.95
avg:

global history & table:
0 : 0.99
1 : 0.655
2 : 0.565
3 : 0.77
4 : 0.7675
5 : 0.88
6 : 0.965
avg:

local history
global table:
0 : 0.98
1 : 0.98
2 : 0.6275
3 : 0.64
4 : 0.785
5 : 0.865
6 : 0.9275
avg:

global history
local tables:
0 : 0.99
1 : 0.7975
2 : 0.5725
3 : 0.74
4 : 0.7975
5 : 0.9475
6 : 0.9975
avg:

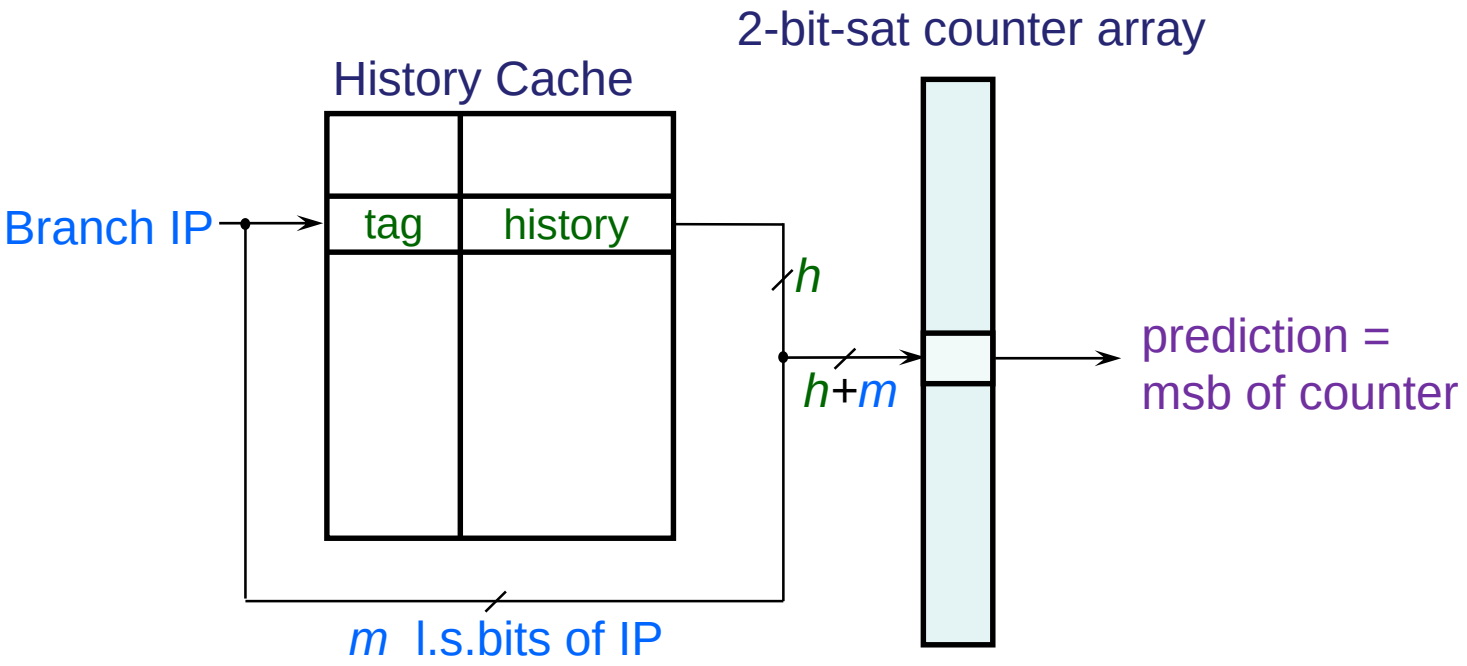
instructions	1,000,000
pipe length	5
penalty	3
branch probability	5

	all wrong	bimodal	local	global hist	perfect
hit rate	0	0.7672	0.8072	0.816	1
cycles	16,000,005	4,492,005	3,892,005	3,760,005	1,000,005
speed up		356.2%	115.4%	103.5%	376.0%

טבלת חיזוי גלובלית

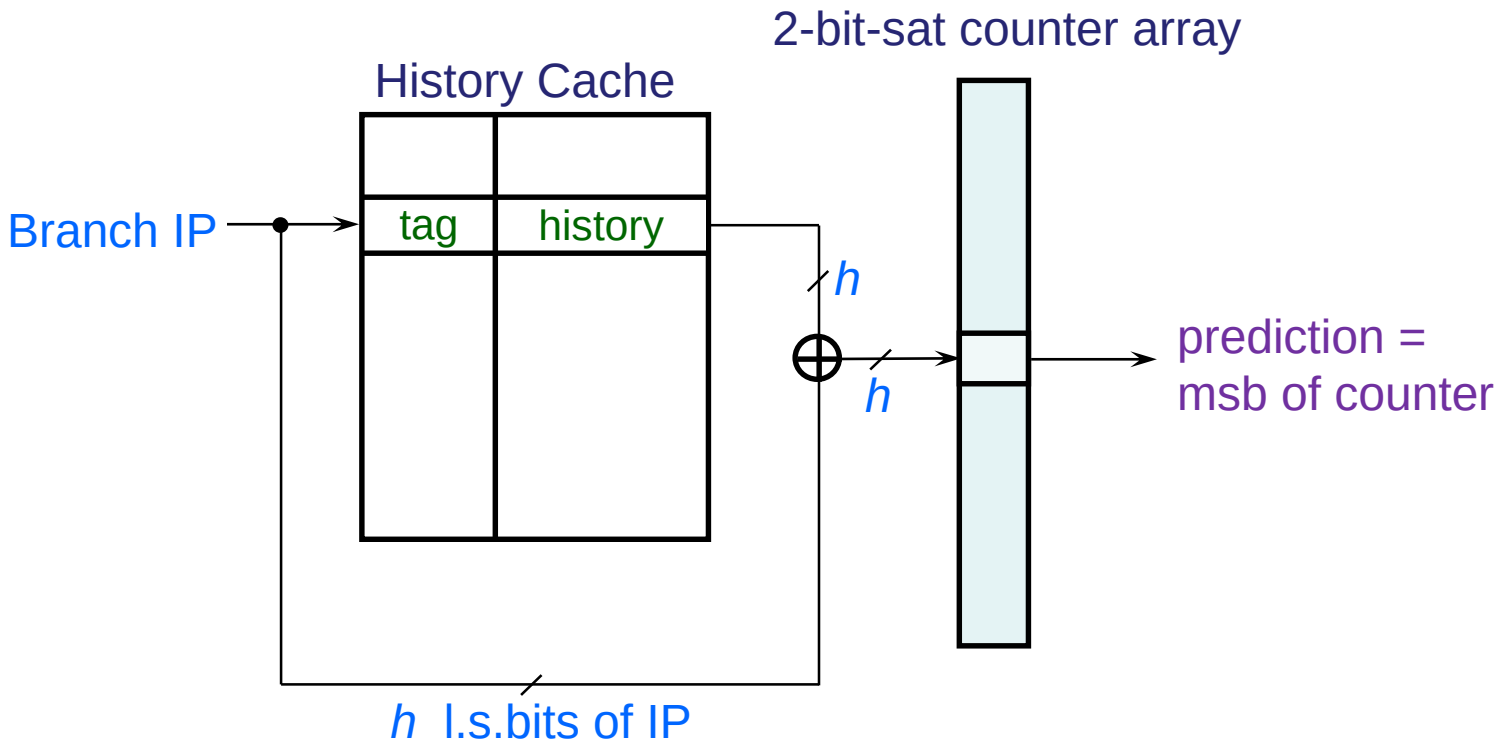
- החיסרון של טבלה גלובלית עשוי להיות בעיית ההתנגשויות בין הוראות branch שונות
- מצד שני שימוש בטבלאות לוקליות יקר
- ניתן להשתמש בחלק מהביטים של כתובת פקודת הסיעוף ל"ערבול" הגישה לטבלת מכונות המצבים הגלובלית
  - פיזור המכונות של פקודות שונות בטבלה הגלובלית
  - הקטנת ההסתברות להתנגשויות
- ישנן 3 שיטות עיקריות לערבול הגישה לטבלת החיזוי הגלובלית:
  - **Lselect** : שרשור ביטים מכתובת הפקודה לתבנית ההיסטוריה הלוקלית ליצירת אינדקס הגישה לטבלה ← הגדלת טבלת החיזוי
  - **Lshare** : XOR של ביטים מכתובת הפקודה עם ביטים מההיסטוריה הלוקלית
  - **Gshare** : כמו Lshare רק עם ביטים של היסטוריה גלובלית

Local Predictor: *Lselect*



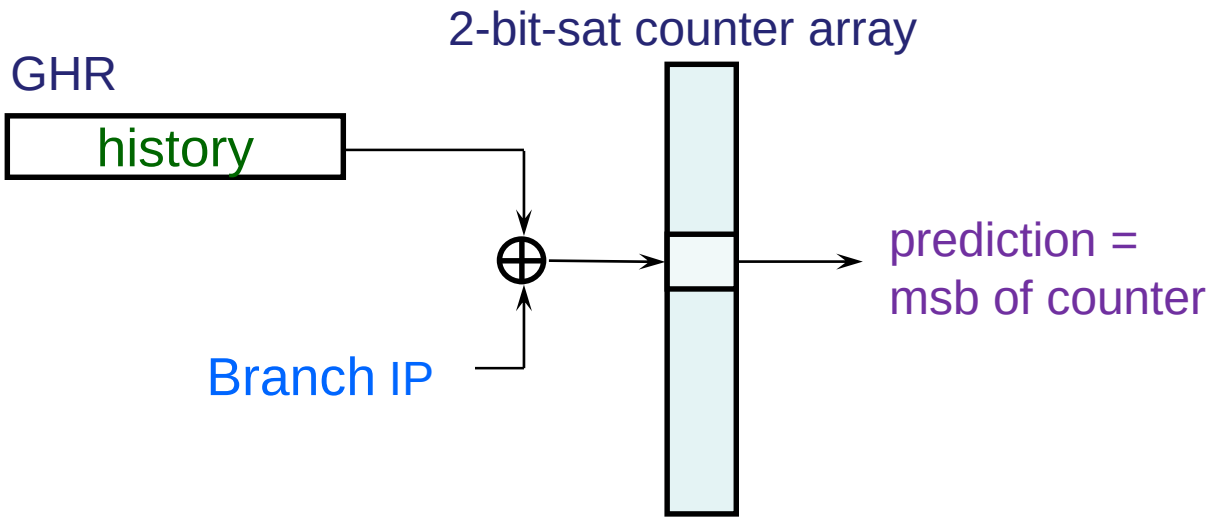
Counters table size: $2 \times 2^{\text{history\_size} + m}$

Local Predictor: *Lshare*



Counters table size: $2 \times 2^{\text{history\_size}}$

Global Predictor: *Gshare*

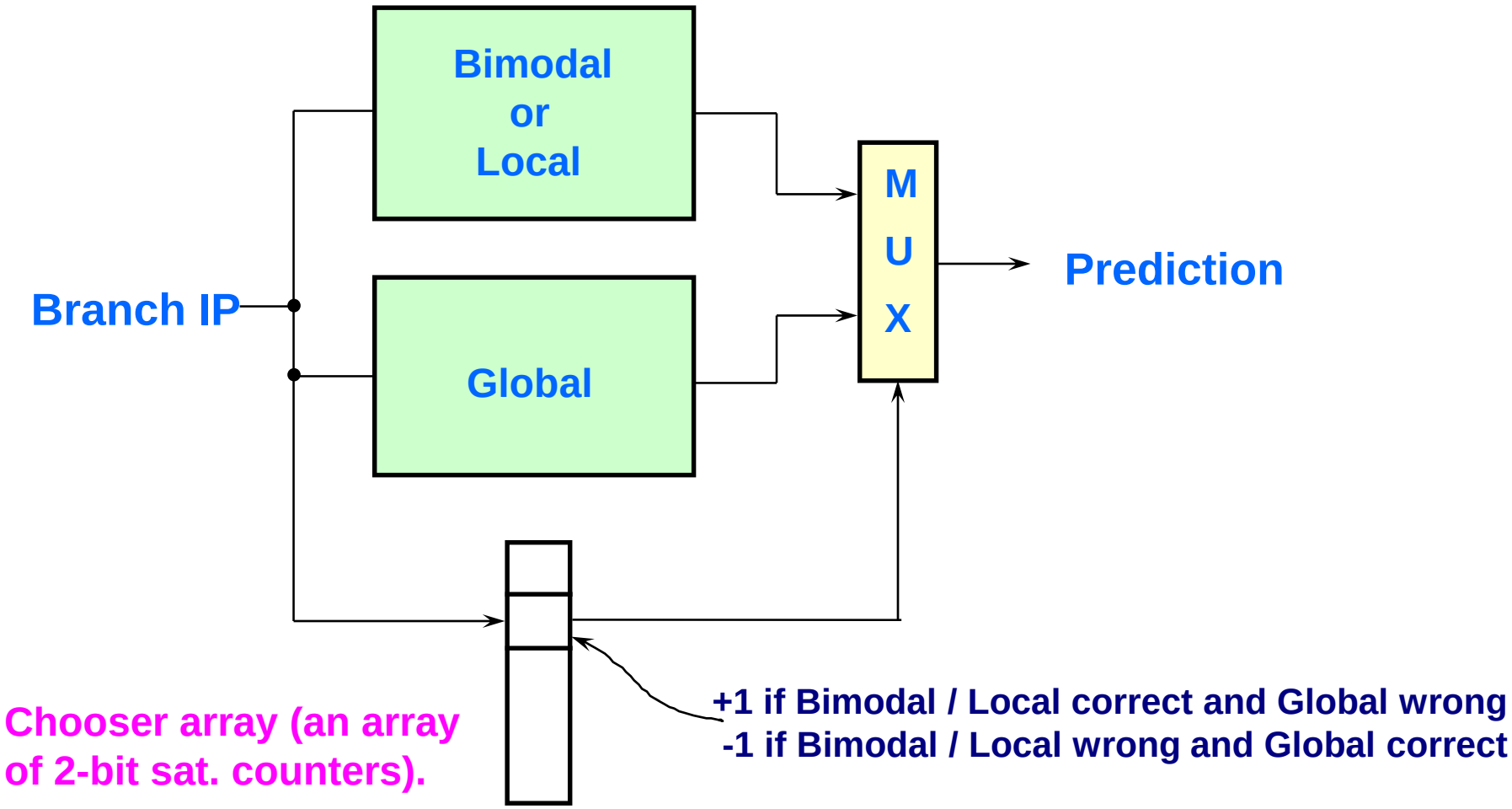


Counters table size: $2 \times 2^{\text{history\_size}}$

ראי ראי שעל הקיר, מהי השיטה הטובה בעיר?

- למעשה, לא קל לקבוע איזה שיטה תתנהג באופן הטוב ביותר ולכל שיטה יש את הרגעים שלה...
- מעבדים מודרניים מחזיקים ב-Chooser, רכיב זה עוקב אחר הצלחות של מספר שיטות שונות (למשל 2) ובוחר בפעם הבאה את השיטה שלדעתו תיתן חיזוי מדויק יותר (סוג רמה שלישית)
- יש לשים לב ששיטות שונות (אפילו 1-level) מכסות מקרים שונים

Chooser



- The chooser may also be indexed by the GHR