Feb 11

Fill the slots from target of branch

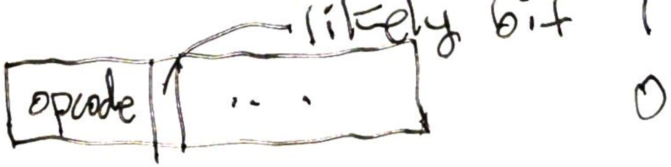
```
BREQZ R1, X              BREQZ-likely R1, Y
                   →     SUB R4, R5, R6

X: SUB R4, R5, R6
Y: ADD ...               Y: ADD . . .
```

squash "the slot instruction" if BR not taken

2nd version that assumes BR not often taken

```
BREQZ-unlikely R1, X  |  if BR is taken,
[XOR ...              |  squash the slot
:
X:
```



Methods to set likely bit

Predict always taken

Backward Taken (loops)/Forward not taken

Heuristics Ball/Larus

Profiling: code.c

compile with instrumentation code added

↓

run → profile - what each BR does

compile -profile

Dynamic prediction needed

(Reducing control flow stalls)

3 W's

Predict

1) whether branch is taken or not

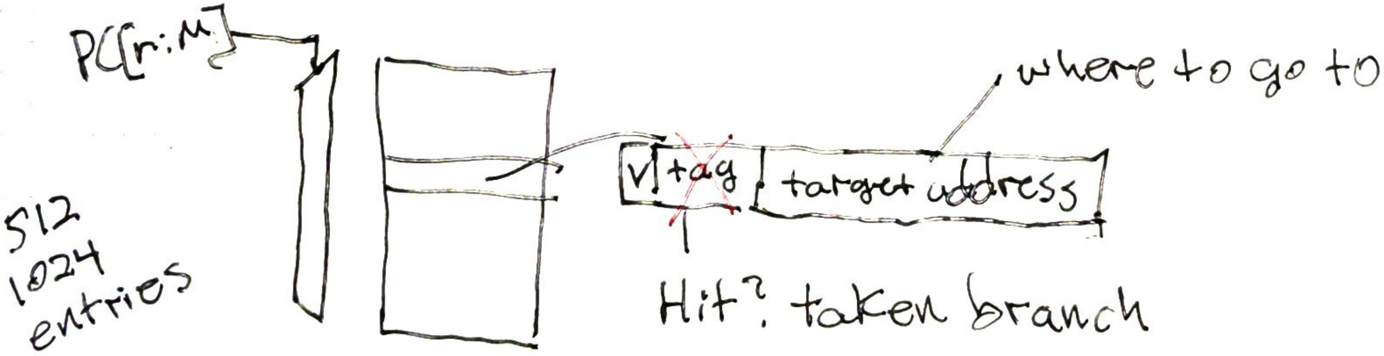
2) Where branch goes (target address)

3) When branch occurs

Taken branch known

IF1 IF2 ID1 ID2 EX1 EX2 . . .

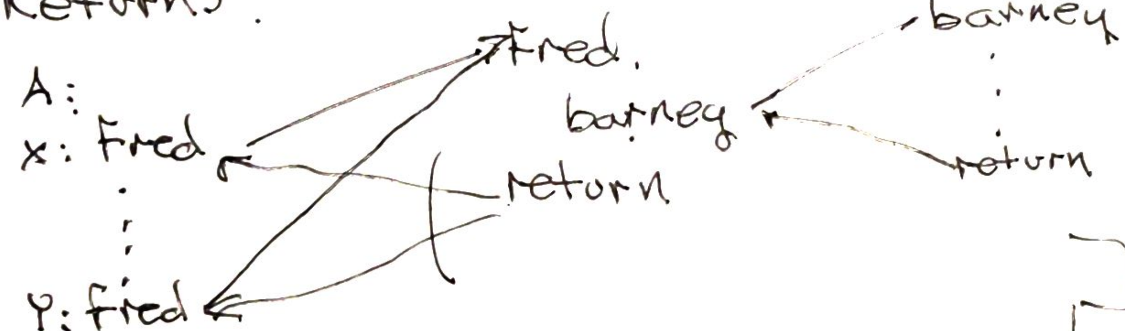
what is known in IF1? Branch target Buffer



Problem kind of branch

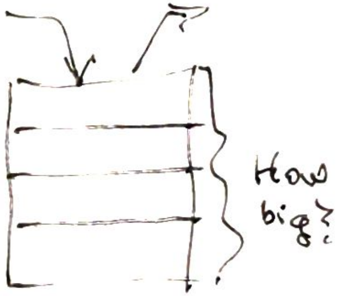
PC doesn't correlate with the target addr.

Returns!



("return addr stack")

Hardware stack in IFI



on call inst, push return addr

on a return, IFI pop return addr

Valid	return bit	target address

BTB entry

=0 use

=1 use top of hardware stack

Experimental results show

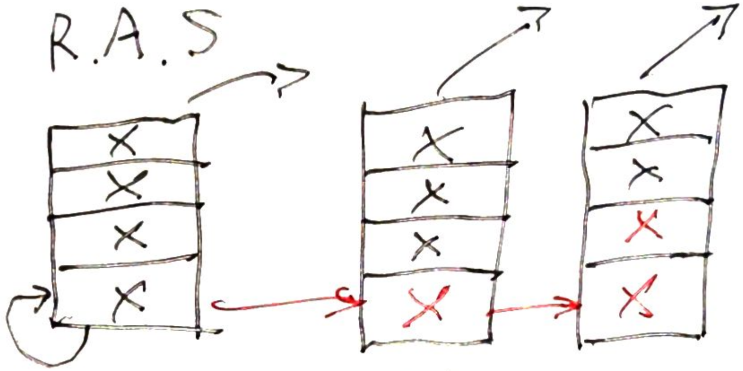
① shallow call chain ≤ 4

② very deep - recursion

Tail recursion

```
→fibbo:
   .
   .
   .
  call Fibbo
x:
```

R.A.S



~~copy~~ on pop — returns after recursion

always

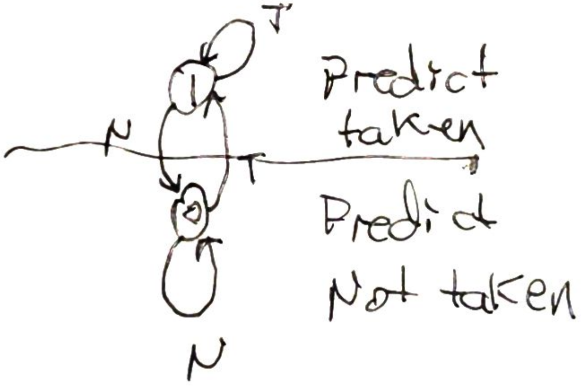
BTB uses valid bit to predict whether

"Predict what happened last time"

Taken last time → taken again

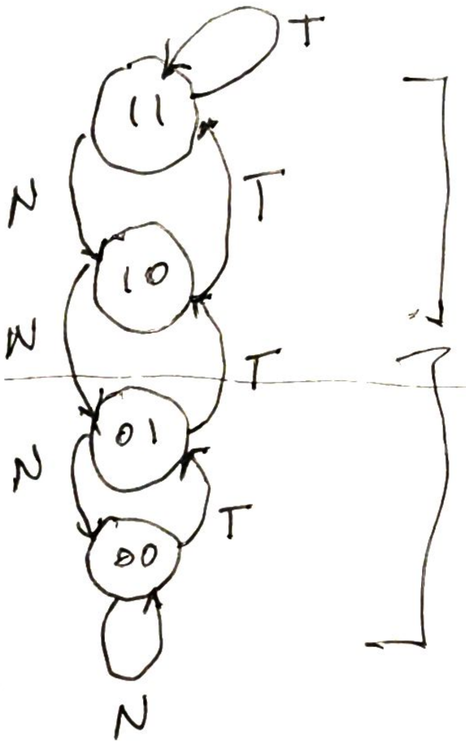
BTB

Prediction Table



2 bits of info?

Smith n-bit bimodal counter



Predict taken

Predict not taken

bad situation for Dr Smith

TNTNT...

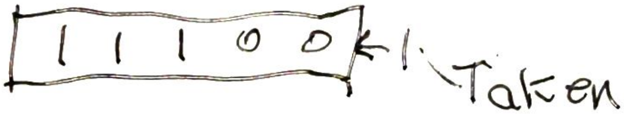
loop closing branch: saturate at 11

conditional that's rare 00→01→10→01→
T T N

~ 85% accuracy

Improving on Smith Counter

History shift register



① Global history, or

② Per branch history