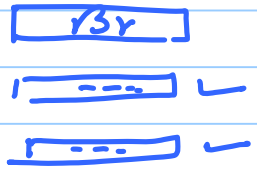
Oct. 18.

Note Title 18-10-2012

Branches: Delayed Branch

Normal Branch.

Delayed Branch



\+ Higher IPC if delay slots are filled

\- Over-emphasis on the compiler.

Predict the direction of the branch:

Taken (T) (NPC ≠ PC+4)

Not Taken (NT) (NPC = PC+4)

IF ID EX MEM WB

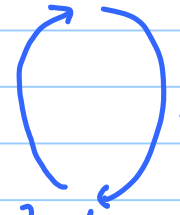
Correct branch prediction → do nothing

Wrong branch " → cancel the incorrectly fetched instructions
fetch from correct address

\+ compiler independent

How do we predict branches?

Typical program



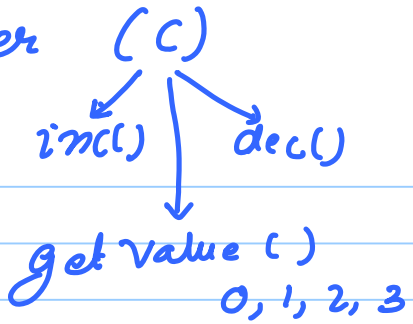
Predict : always taken (mostly correct)
(70% acc.)

1) Coin toss → 50%.

2) always taken → 70%.

2-bit saturating counter (C)

(implemented in HW)




```
C.inc() {
    value = min(value+1, 3);
}

C.dec() {
    value = max(value-1, 0);
}
```

1) Simple Predictor:

Branch prediction (bp)

bp = same output as last branch.

1st pred: x

.

.

(N-1)th pred } T

Nth pred x } NT

1st pred → x } NT

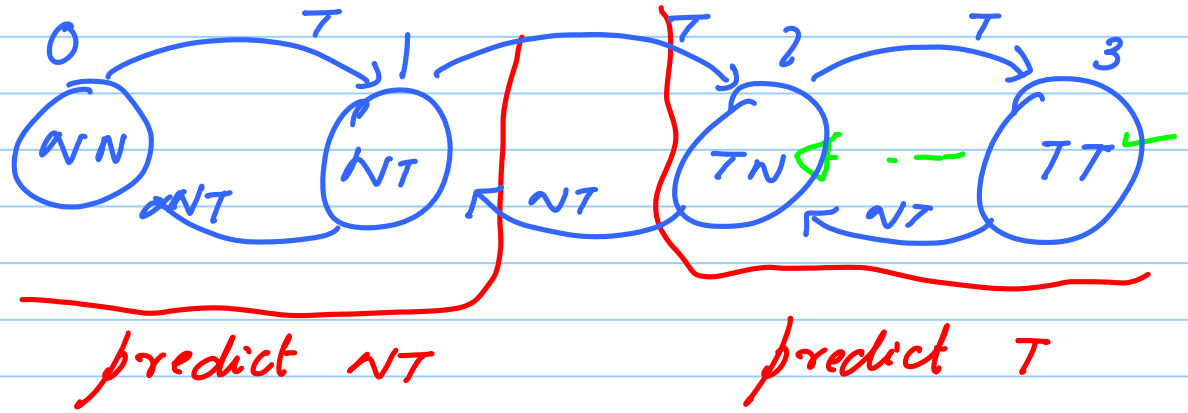
.

.

.

ath pred → x

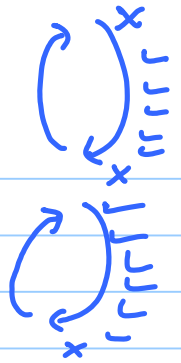
Predictor, which does not take exceptions very seriously.



Branch Predictor → predict (PC) → [T, NT] (IF)
→ Train (PC, [T/NT]) (EX)

2-bit sat cntr. adds some memory to the branch predictor.

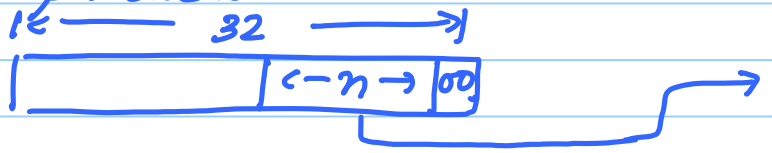
For the example with two loops, makes one less mistake.



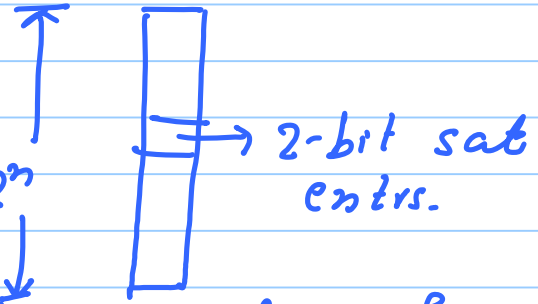
✗ You cannot have only one 2-bit sat cntr. (Collisions)

✗ Ideally, 1 2-bit sat cntr per branch. (Too much space)

1) Bimodal prediction.



(80-90%)



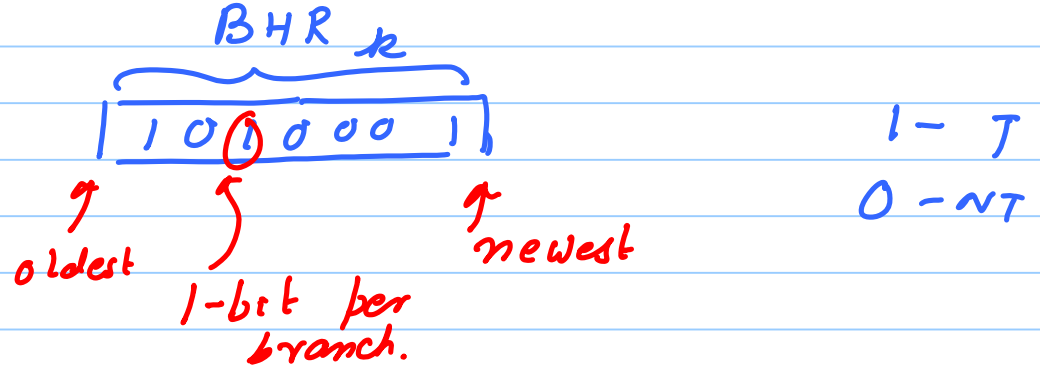
- Taking a look at the past behavior of a branch PC
- Saving the result in the 2-bit cntr.
- Using it for pred.

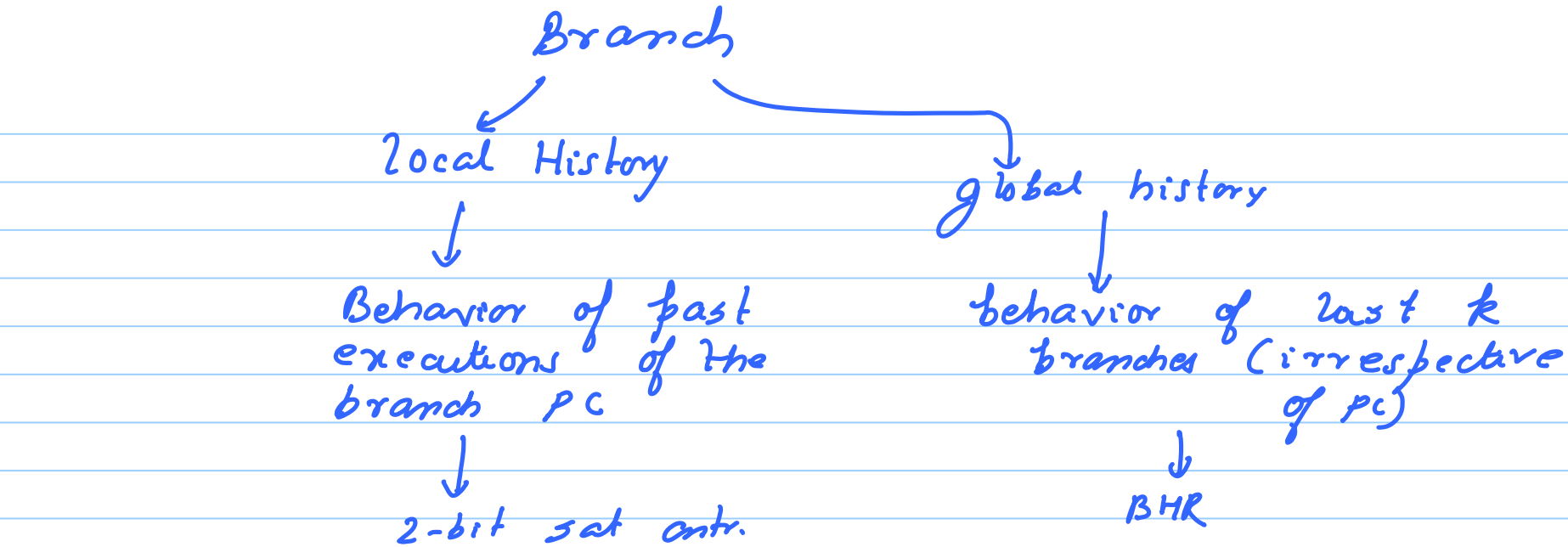
+ reduces collisions.

```
if (flag1 == 0) (B1)
    flag2 = 1;
if (flag2 == 1) (B2)
    printf ("great");
```

(B1 is NT) ⇒ (B2 is NT)

Branch History Register:





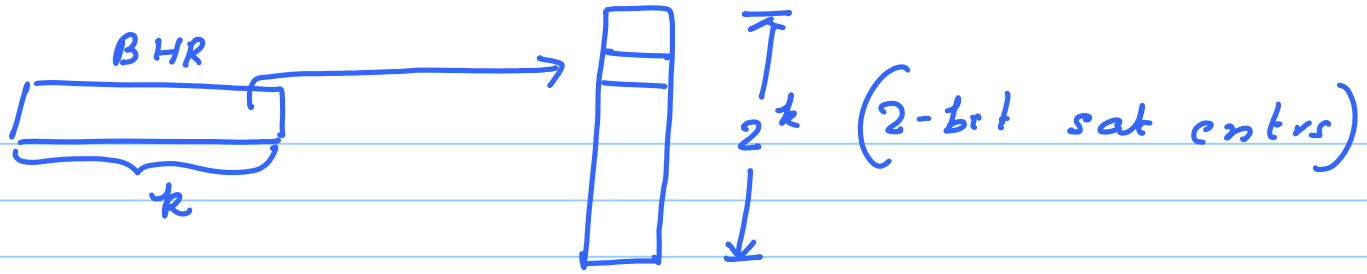
Family of branch preds.

G → global

P → pattern (local)

Gag:

(90%)

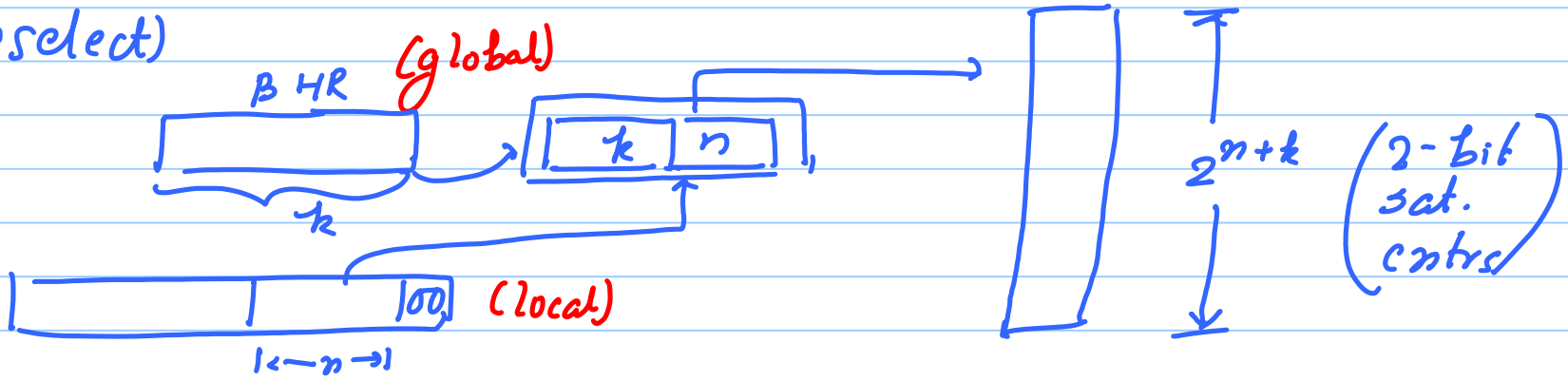


\+ simple

\- not taking local history into account

Gap (gselect)

(95%)

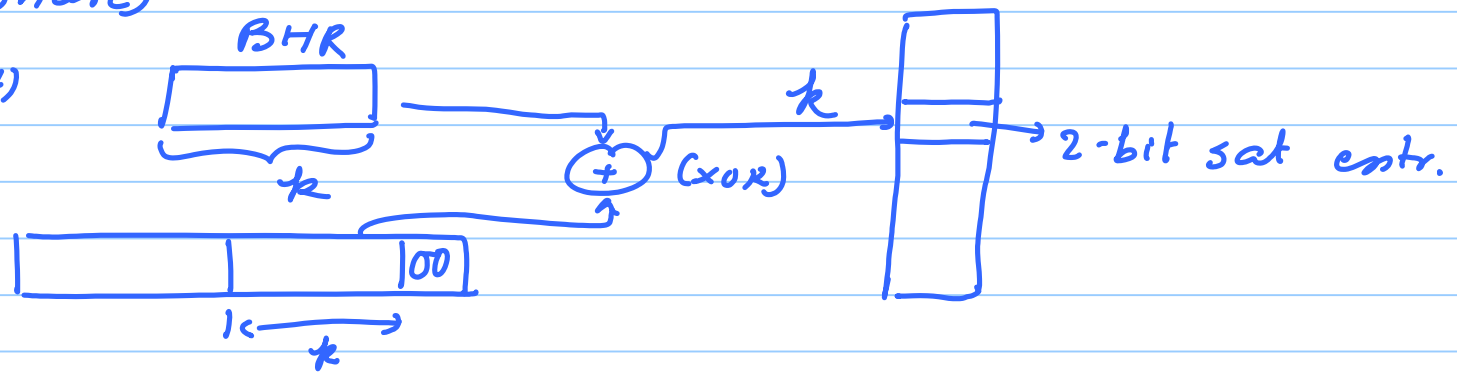


+ combines both global & local

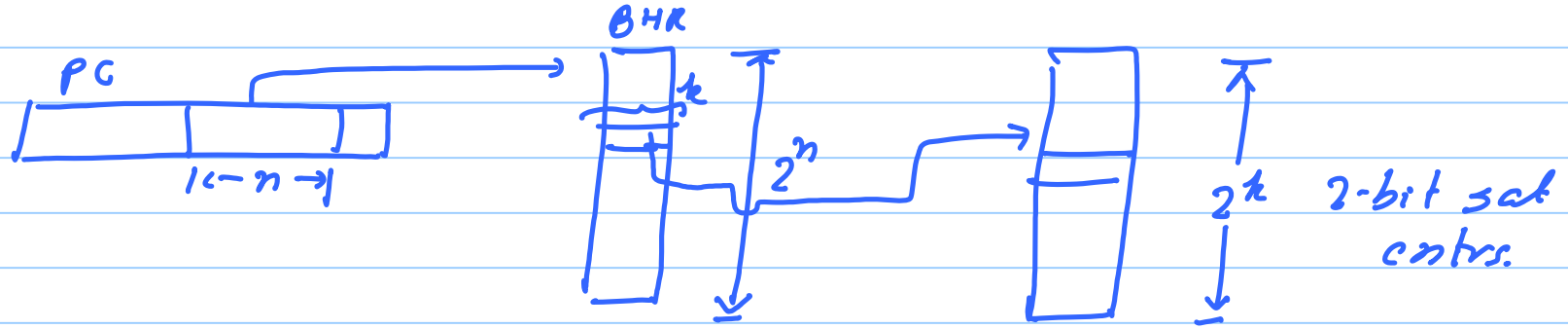
— needs more space (2^{n+k})

(gshare)

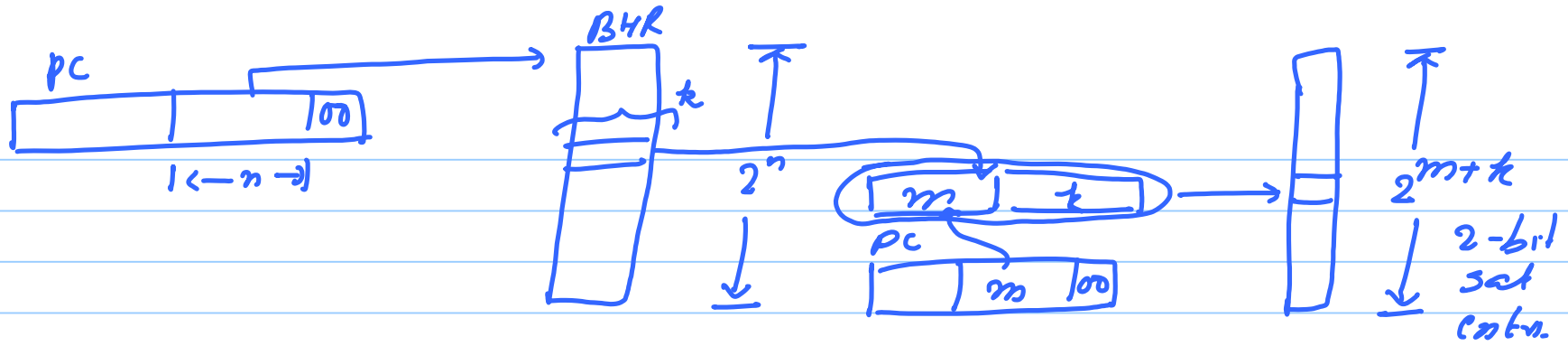
(95.1)



Pag



PAp

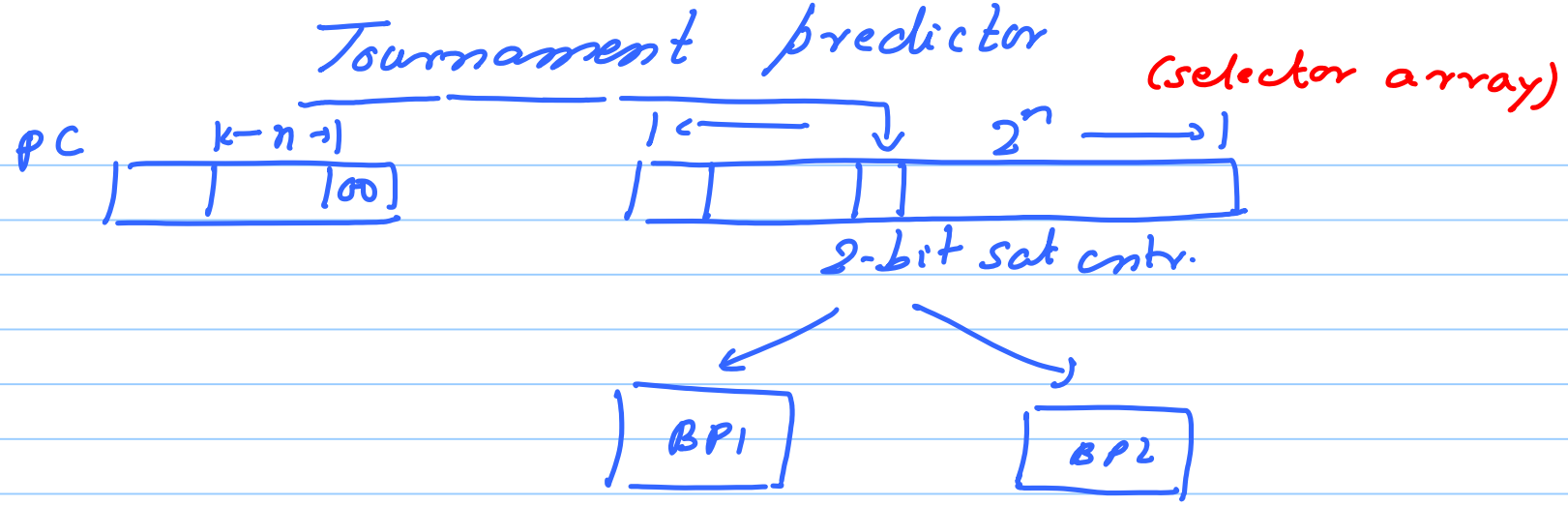


Theory of branch preds:

Prediction is related to compression in information theory

file → zip → file.zip → rar → file.zip.rar.

(Fano's inequality)



Predict:

1) Find 2-bit sat cntr. (c) in the selector array.

2) if (V = value (c))

V = [0, 1) predict BP1.predict()

$v = [2, 3]$ $\quad$ predict BP_2. predict()

Train:

1) Train the selector array.

2) Train BP_1 and BP_2