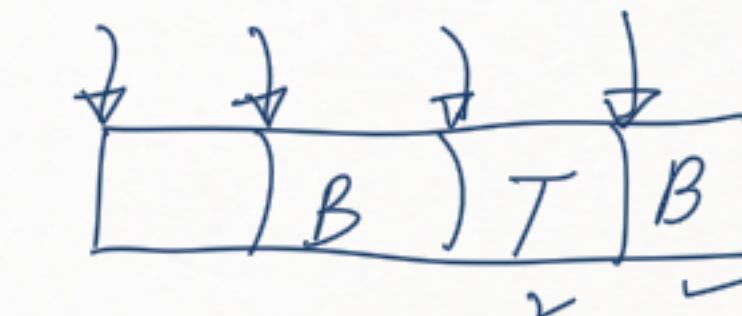
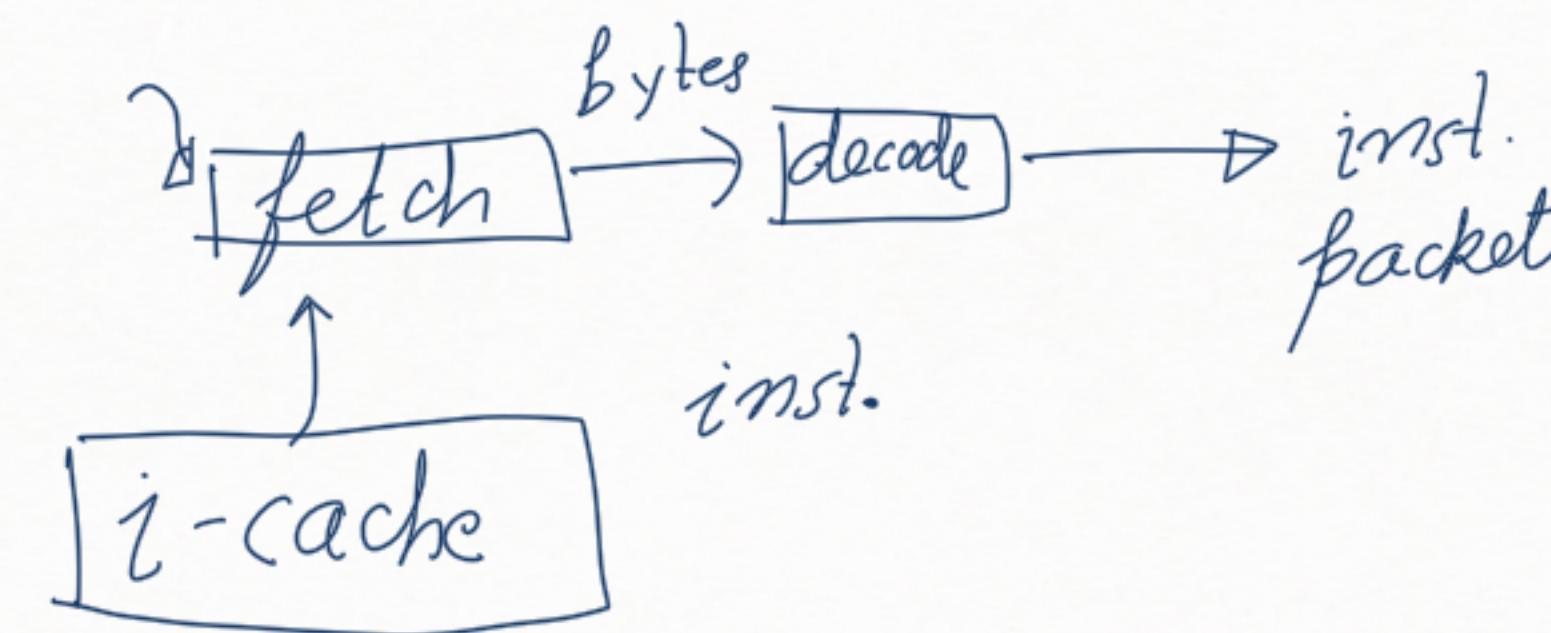


Extra work and discard.

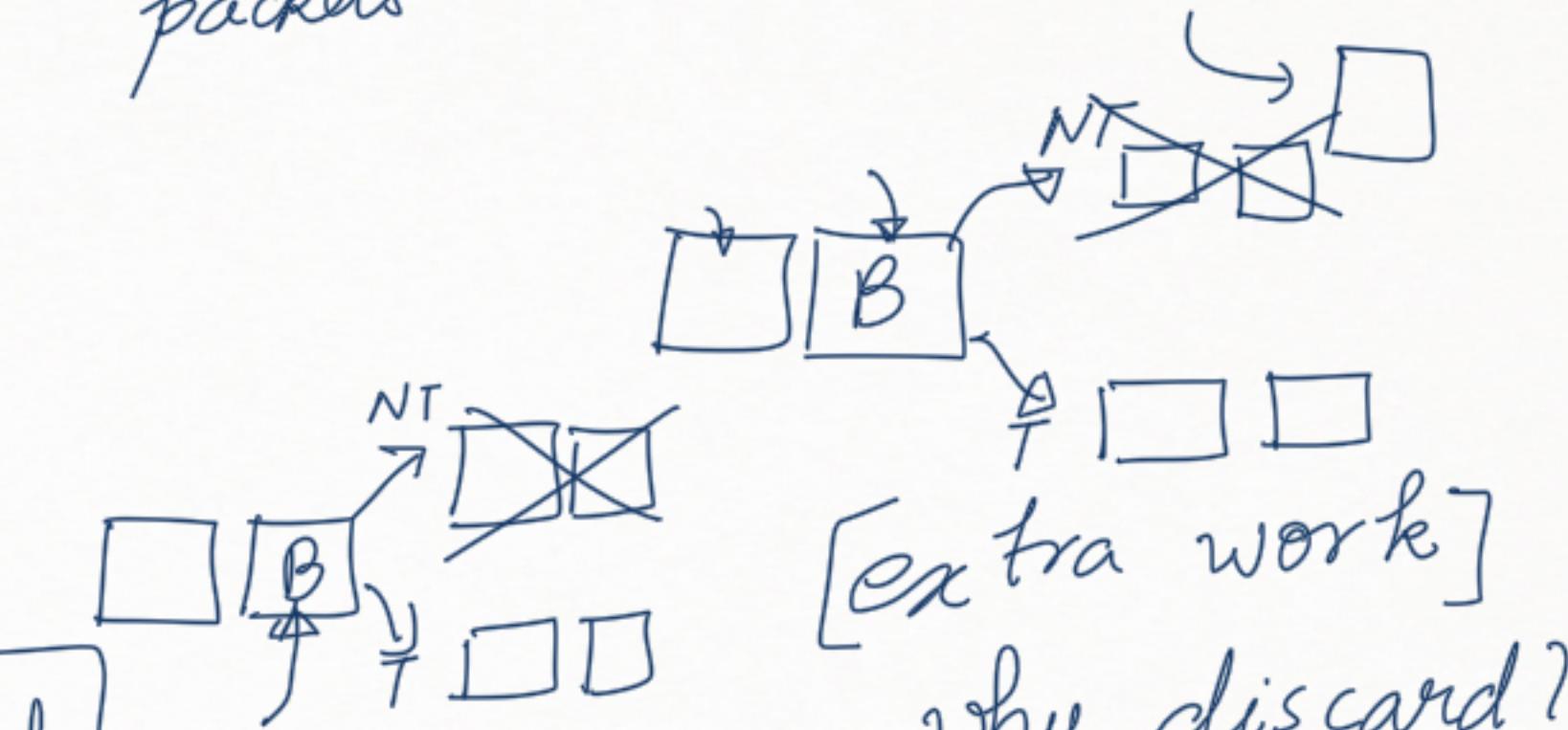
→ Parallelize



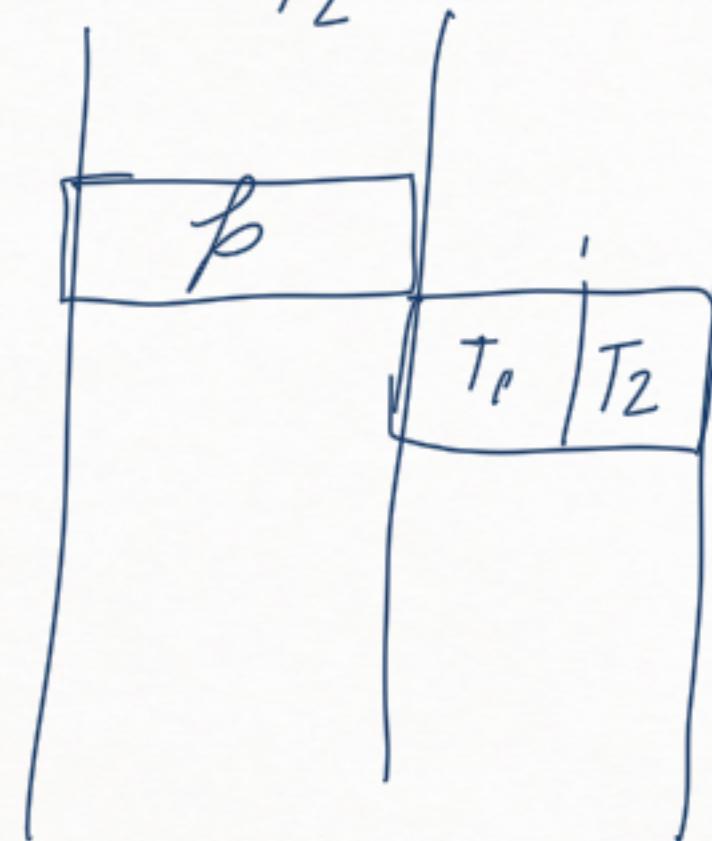
NO



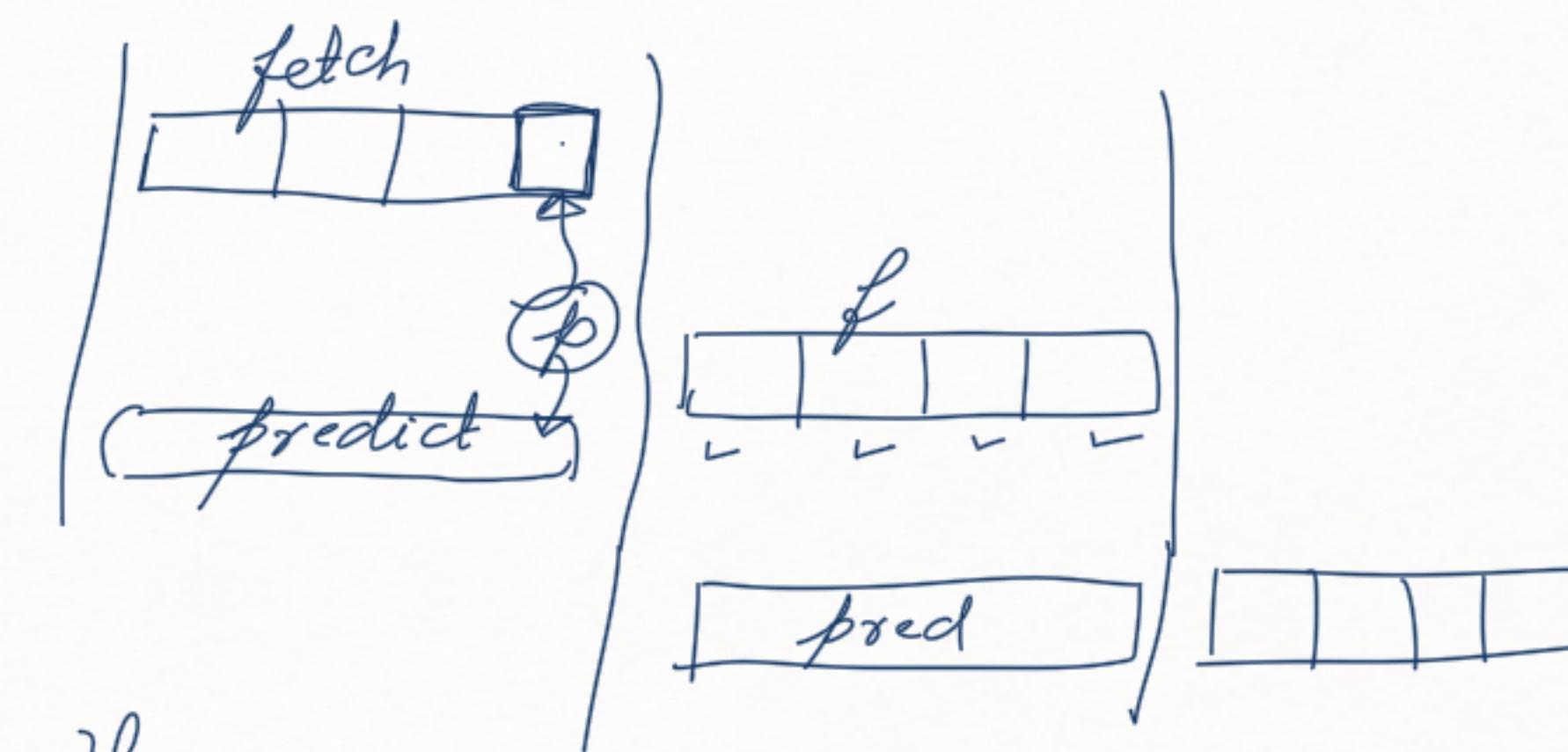
- high fetch BW
1. Do not contiguously place branches
  2. Parallelize: extra work and discard.



$$PC = p \xrightarrow{T_1} \\ \downarrow T_2$$



When I am fetching, all we have is the PC.  
We have not taken a look at the contents.



> Compilers always reorder the code.

`[-g] [-O1] ... -O3`  
precise exceptions  $\xrightarrow{SW}$   $\xrightarrow{HW}$

[  
→ BTB  
→ type  
→ target  
]

[Branch predictor]  
taken or  
not taken

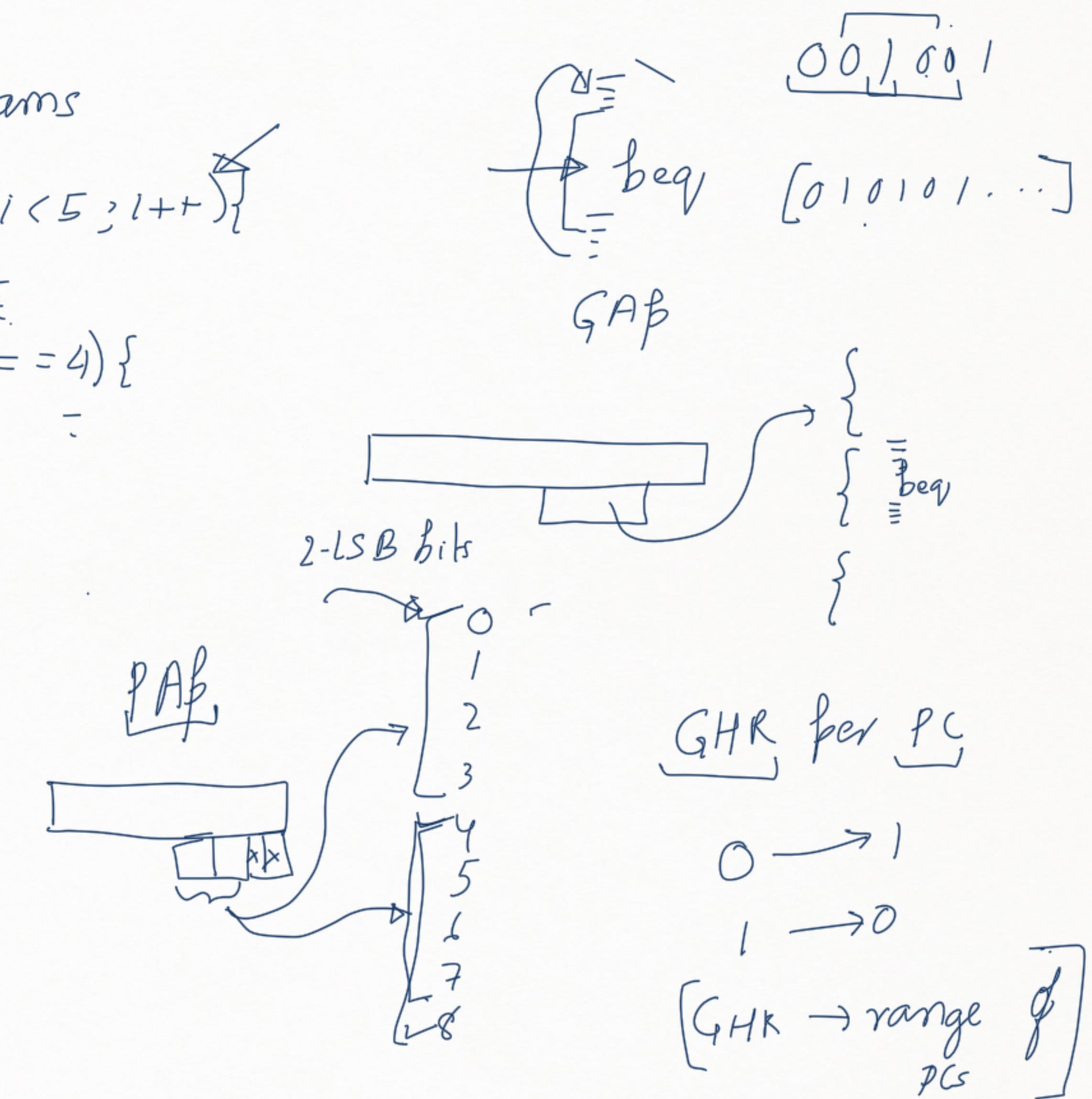
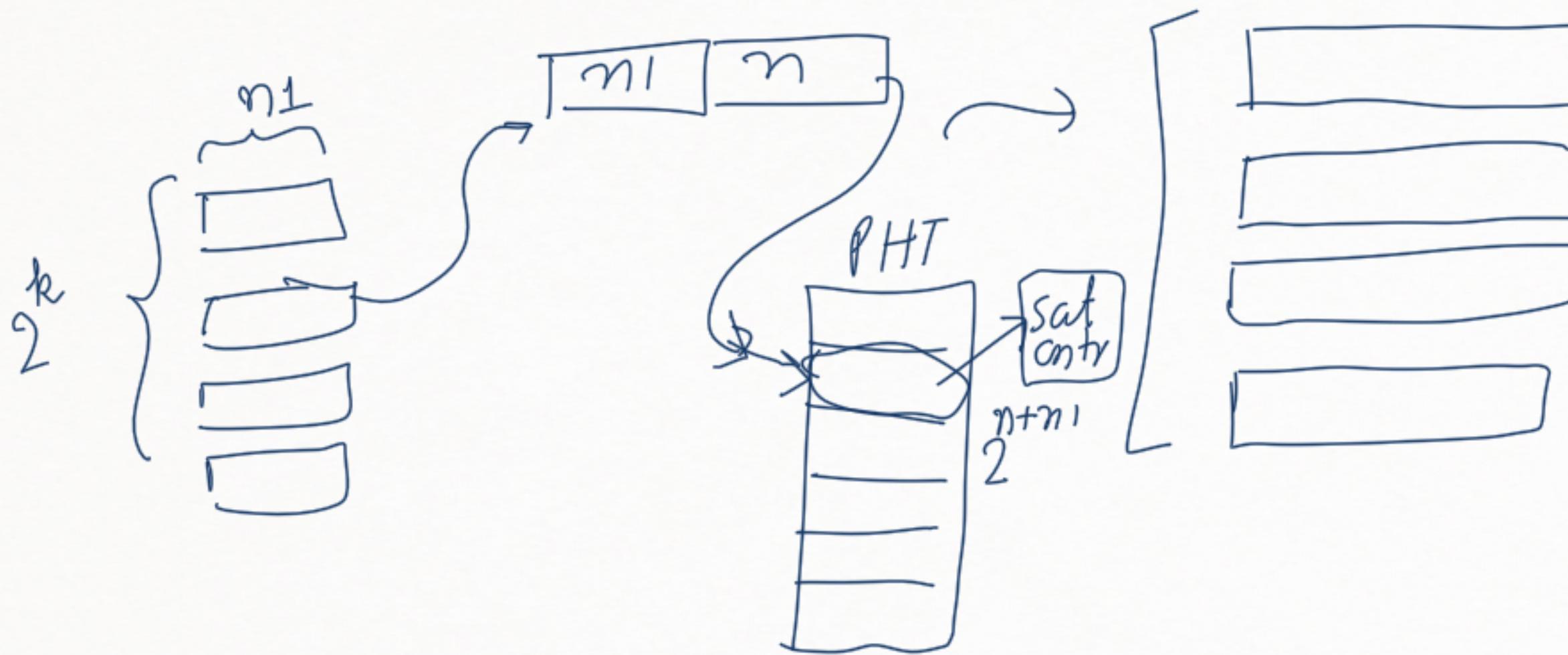
## 2nd Lecture

generic of all programs

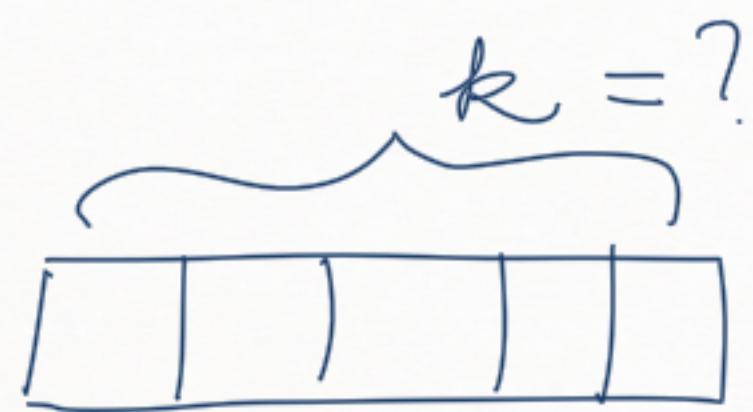
```

for (i=0; i<5; i++) {
    if (i == 4) {
    }
}
  
```

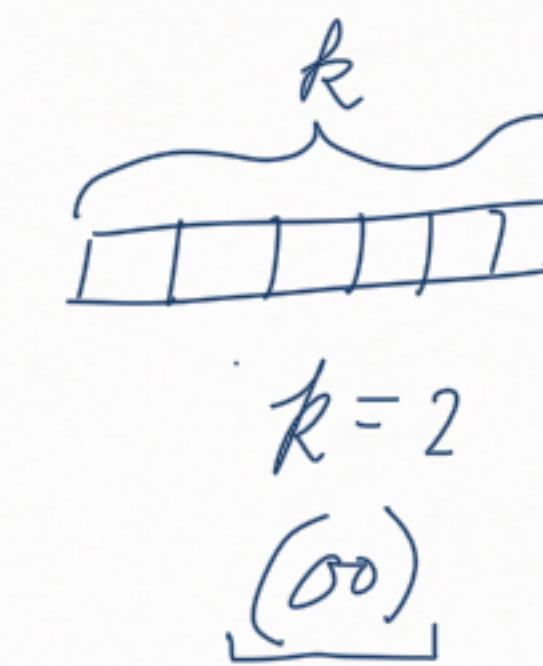
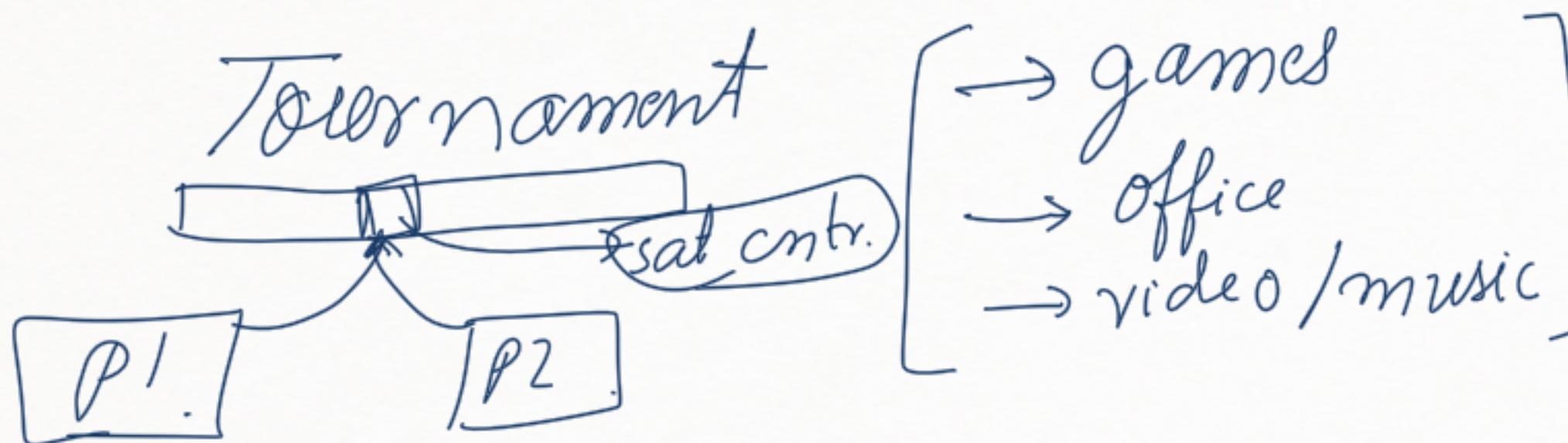
\* GAP vs PAE  
→ global history pollution



if (...) ~  
 if (...) ~  
 if (...) - ~



→ 20 benchmarks



saturating counter  
single PC  
(ignoring aliasing)

GHR → Behavior of the last  $k$  branches



gshare       $\text{XOR} \rightarrow \text{PC bits}$   
+ GHR contents

0 1 0 1 0 1 0

0 1 0 0 0 1 0 1

$\times$   
Sat. counter helps

Boolean sense  $\rightarrow \begin{bmatrix} \log_2 n \\ 2^n \end{bmatrix}$

RAS stack  $\rightarrow$   
LIFO

$\rightarrow$  TAGE  
 $\rightarrow$  hierarchical perceptron predictor

Overfitting  $\rightarrow$  avoided