1st August 2022

Parallel Processing &

Pipeline:

To solve a problem: First identify (it) data, then manipulate the data.

DS + Algorithm + Programming Language (HLL)

DS → to store the data

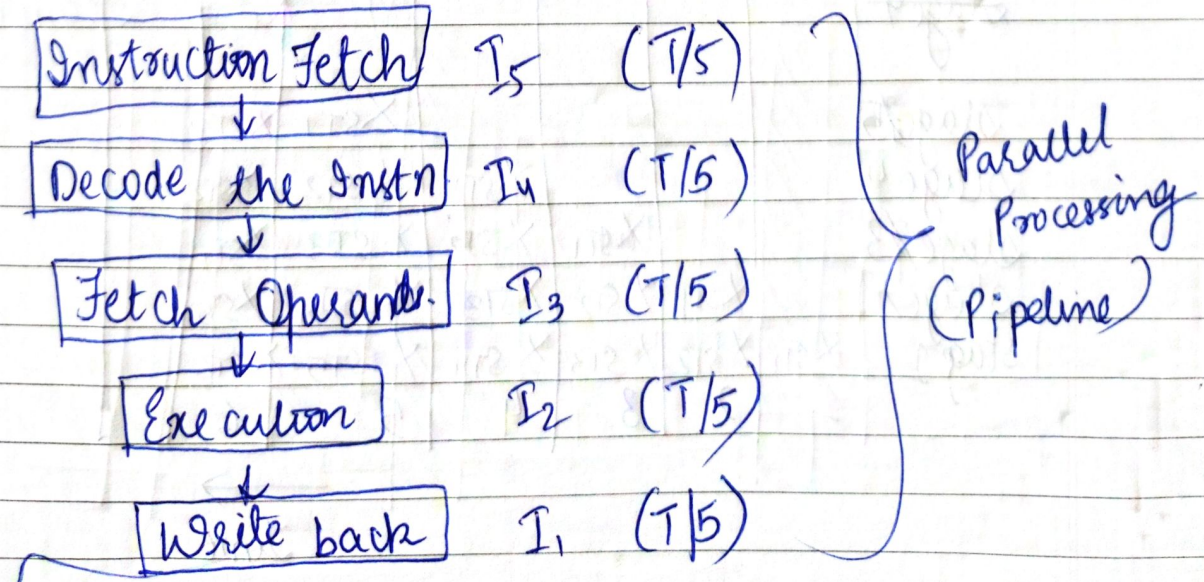
Algorithm → to solve the problem

3rd August 2022

→ DSP is that processor which is customized for mathematical operations.

→ OS given an environment to install & run other applications.

→ Several instructions are run simultaneously.
→ Execution of instructions is a 6-step process.



(single instr. execution)

→ Pipeline makes execution of instrns. faster.
→ Frequency can be increased by increasing no. of stages.

22/08/22

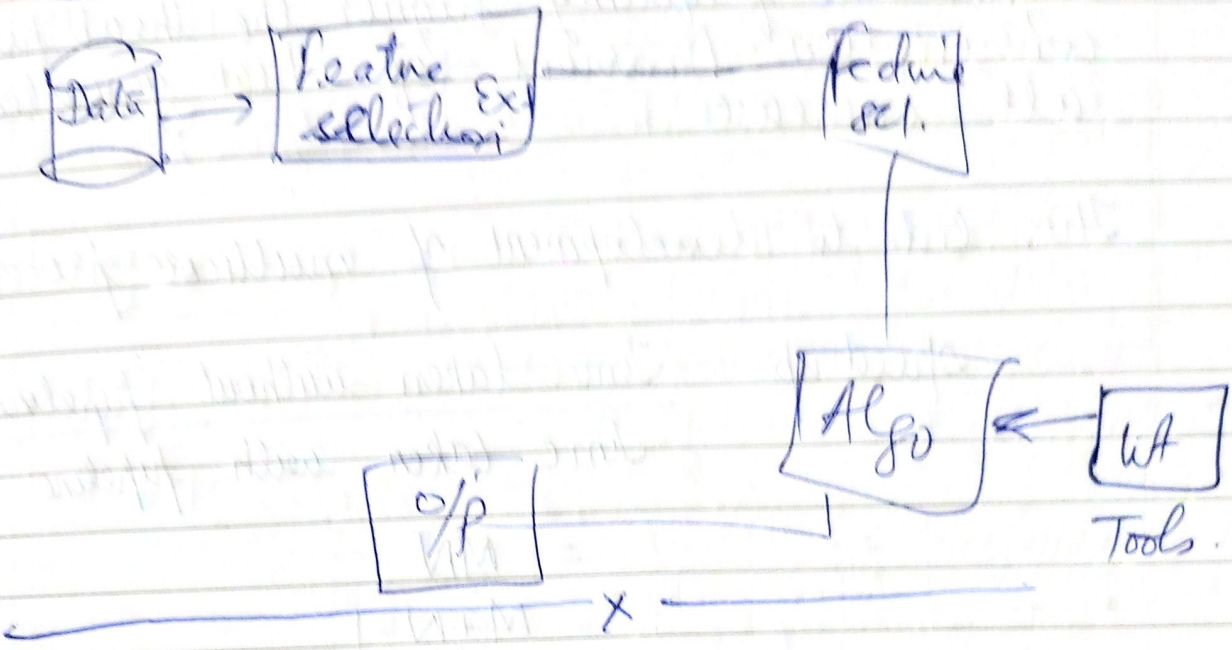
No. of ~~segments~~ / stages = M
No. of tasks = N

Total time taken to complete tasks = ~~2~~ ?
Throughput = ~~2~~ ?

	1	2	3	4	5	6	
~~Stage 5~~							
~~Stage 4~~							
Stage ~~5~~					X_{ST1}	X_{ST2}	
Stage 4				X_{ST1}	X_{ST2}	X_{ST2}	
Stage ~~3~~			X_{ST1}	X_{ST2}	X_{ST3}	X_{ST4}	
Stage 2		X_{ST1}	X_{ST2}	X_{ST3}	X_{ST4}	X_{ST5}	
Stage 1	X_{ST1}	X_{ST2}	X_{ST3}	X_{ST4}	X_{ST5}	X_{ST6}	

→ Time

Framework



ST_1 = Sub task 1
ST_2 = Subtask 2.

Total time = Time to process first + Time taken for (N-1) task

= M+N-1 clock cycles.

Max. throughput = $\lim_{N \to \infty} \frac{Nf}{M+N-1}$

Throughput = $\frac{\text{No. of tasks}}{\text{Total time taken}} = \frac{N}{(M+N-1)}$

$= \frac{Nf}{M+N-1}$

$\lim_{N \to \infty} \frac{Nf/N}{\frac{M}{N} + \frac{N}{N} - \frac{1}{N}}$

$= f$ = frequency

Higher the frequency, higher the throughput

Higher the frequency, higher the heat power consumption. Amount of heat generated all increases

This led to development of multicore processors

$$\text{speed up} = \frac{\text{Time taken without pipeline}}{\text{Time taken with pipeline}}$$

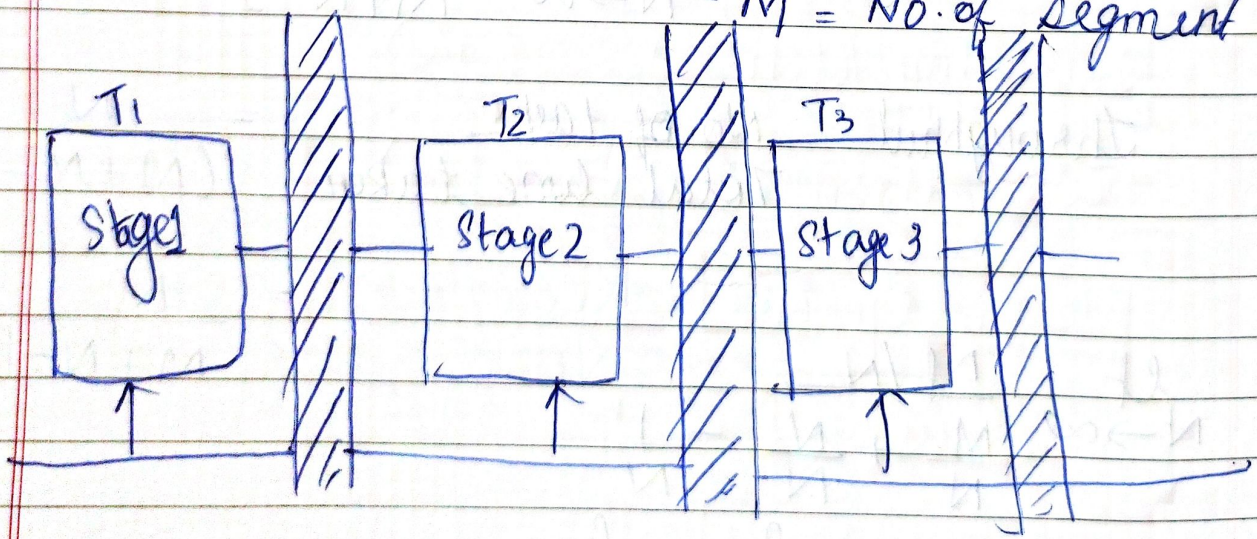
$$= \frac{MN}{M+N-1}$$

$$\text{Max speed up} = \lim_{N \to \infty} \frac{MN}{M+N-1}$$

$$\frac{\frac{MN}{N}}{\frac{M}{N} + \frac{N}{N} - \frac{1}{N}} = \frac{M}{\frac{M}{N} + 1 - \frac{1}{N}}$$

$$\frac{M}{\frac{M+N-1}{N}}$$

$= M$ = No. of segment.



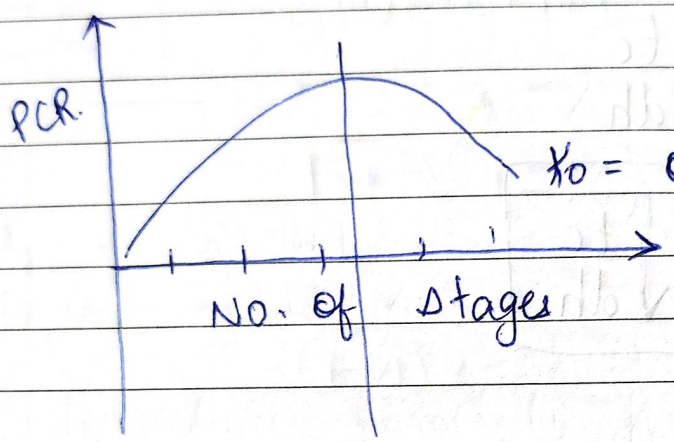
23/08/22

$$f = \frac{1}{\text{latch time} + \max(T_1, T_2, T_3, T_4)}$$

No. of stages:

PCR = Performance cost Ratio = $\frac{\text{Performance}}{\text{cost}} = \frac{f}{c+kh}$

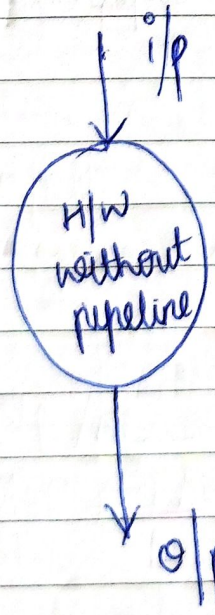
f = frequency of pipeline

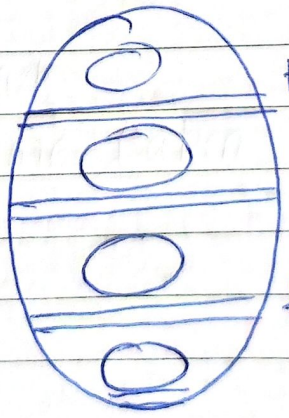


k_0 = optimal no. of stages

If c = cost of digital gates
h = latch cost
k = no. of stages

t = time taken by hardware without pipeline





d = latch time

$$f = \frac{1}{d + \frac{t}{k}}$$

$$\therefore PCR = \frac{f}{c+kh} = \frac{1}{(c+kh)\left(\frac{t}{k}+d\right)}$$

Maximize PCR wrt 'k'

$$k_0 = \frac{d}{dk}(PCR) = 0$$

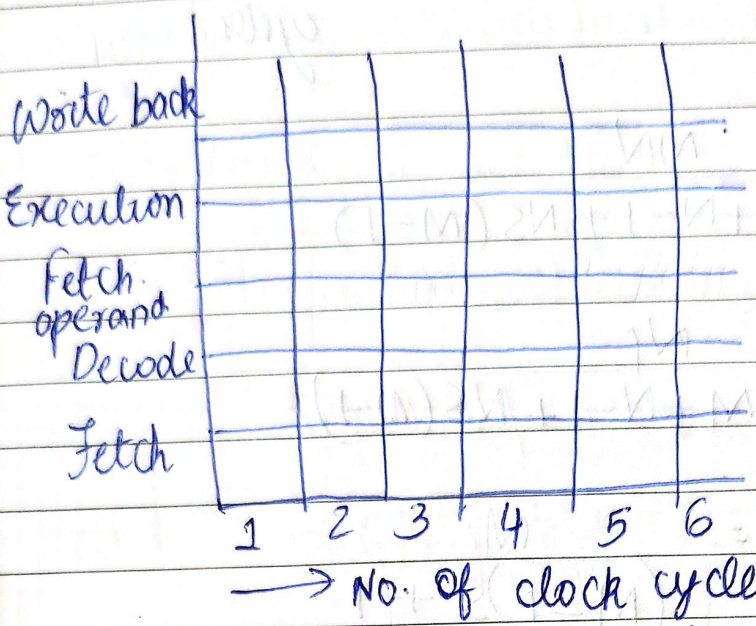
$$k_0^2 = \frac{tc}{dh}$$

$$\boxed{k_0 = \sqrt{\frac{tc}{dh}}}$$

24/08/22

classmate
Date
Page

Instruction pipeline



No. of Instructions = N
No. of segments = M

Probability that a given Instruction will cause branching = S

Without branching = $m+n-1$
$= 5+3+15-1$
$= (28-1)\ 20-1$
$= (27)\ 19$

No. of extra clock cycles due to 1 branch instruction = $(m-1)$
$= 5-1 = 4$

Avg. no. of branching $= \sum_{i=1}^{N_i} 1 \cdot S + 0(i-S)$

$= NS$

=

Time = $M+N-1+NS(M-1)$ clock cycles.

Speed up = $\frac{MN}{M+N-1+NS(M-1)}$

Throughput = $\frac{Nf}{M+N-1+NS(M-1)}$

Max. speed up = $\frac{M}{(M-1)S+1}$

Max. throughput = $\frac{f}{1+S(M-1)}$

S	max. speed up	Max. Throughput
0	5	10×10^6
0.1	3.5	7.14×10^6
0.2	2.8	5.55×10^6
0.3	2.27	4.54×10^6

$M=5$
$f=10MHz$

① $S=0$, max. speedup = $\frac{5}{4\times0+1} = 5$

max. throughput = $\frac{10}{1+}$ =

② $S=0.1$ = $\frac{5}{4\times0.1+1}$ = $\frac{5}{1.4}$ = 3.57

max. throughput = $\frac{10}{1 + 0.1(4)} = \frac{10}{1.4}$

$= 7.14 \times 10^6$

3) $s = 0.2$,

Speed up = $\frac{5}{4 \times 0.2 + 1} = 2.77$

$= 2.8$

throughput = 5.55×10^6.

Avg. no. of instructions executed per instruction cycle = $\frac{N}{\frac{M + N - 1 + NS(M-1)}{M}}$

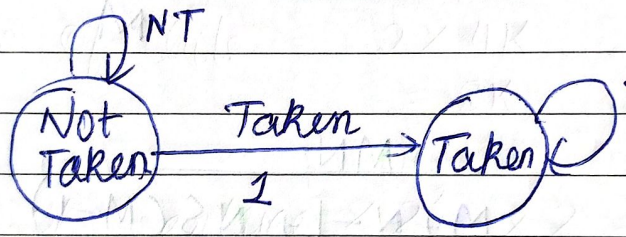
$= \frac{MN}{M + N - 1 + NS(M-1)}$

Branch prediction

* 1-bit branch prediction - 1 bit entry in BHT.
* 2-bit branch prediction
* (m, n) branch predictor
* Tournament branch prediction
* BTB (Branch ?---- Buffer)

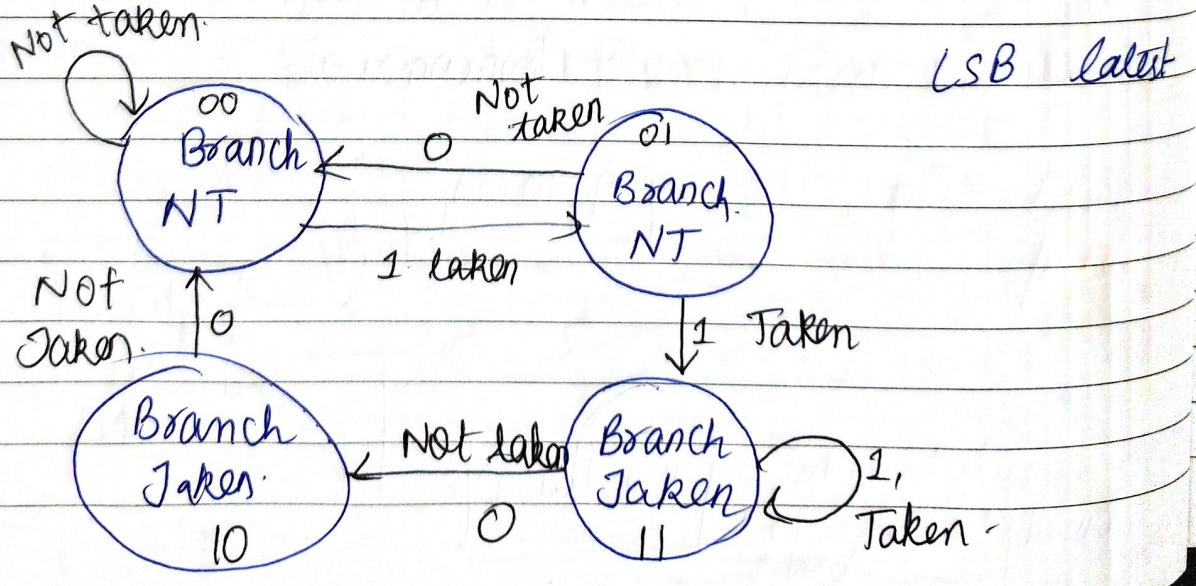
→ Size of predictor is between 8K-32K.
→ 4K bit predictor is as good as infinite predictor

→



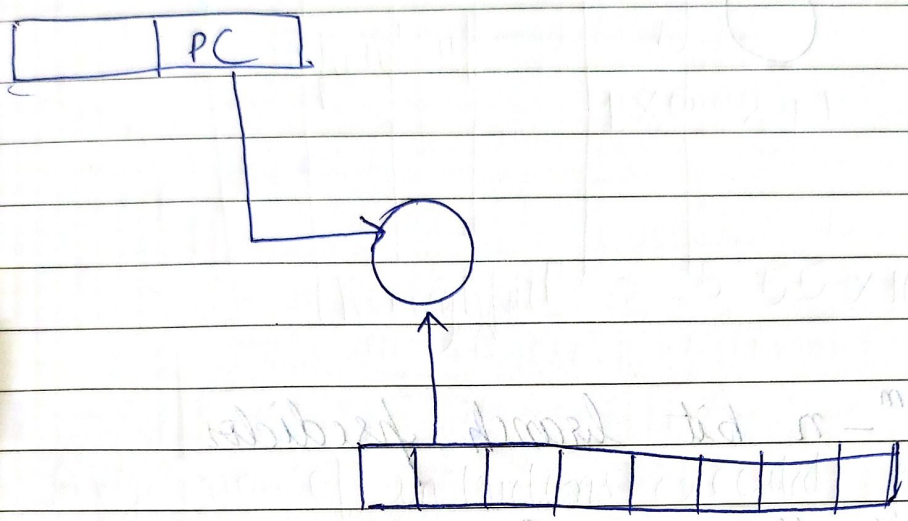
Misprediction 2 times.

2 bit predictor:



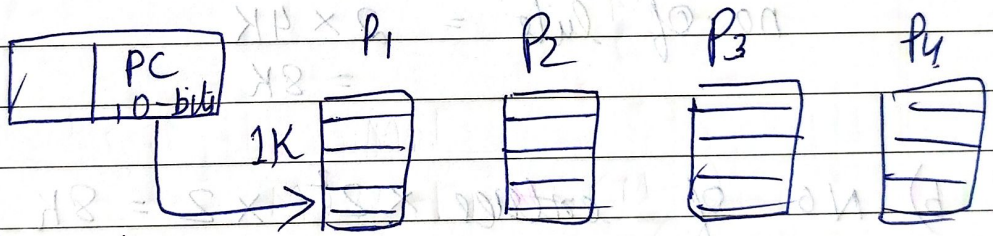
(m, n) branch predictor

→ 1 branch may depend upon other branches.
→ Take a GHT (Global History table) which records history of 'm' branches.

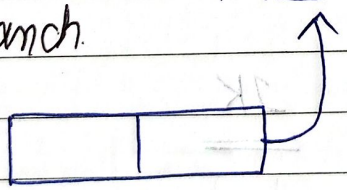


let (m, n) = 2, 2

then



(2, 2) branch predictor.



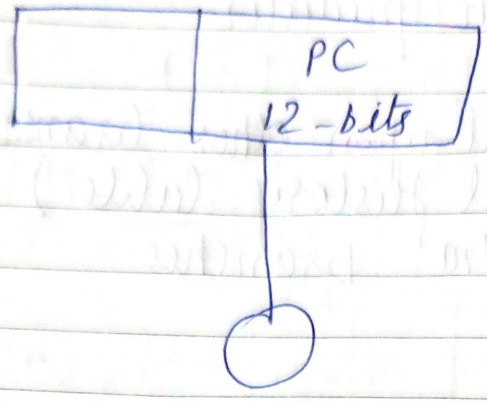
Each predictor is 2 bit.

The predictor is decided by the GHT.

Qn) a) How many bits are in (0, 2) Branch predictor with 4K entries.

b) How many entries are in (2, 2) Branch Predictor with same no. of bits.

Book:
Patterson.



(m, n)

2^m – n bit branch predictor

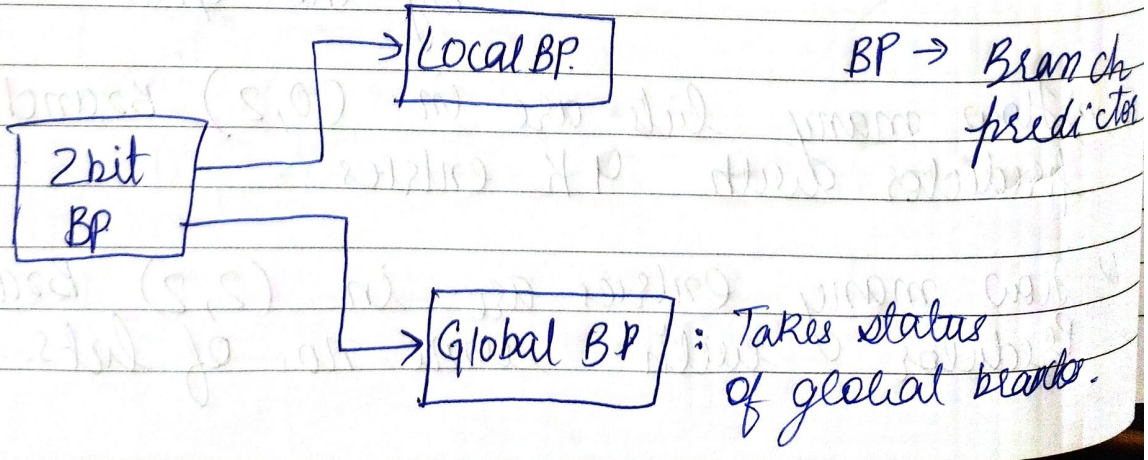
a) No. of predictors = $2^0 = 1$.

no. of bits = $2 \times 4K$
= $8K$

b) No. of entries $\times 2^2 \times 2 = 8K$

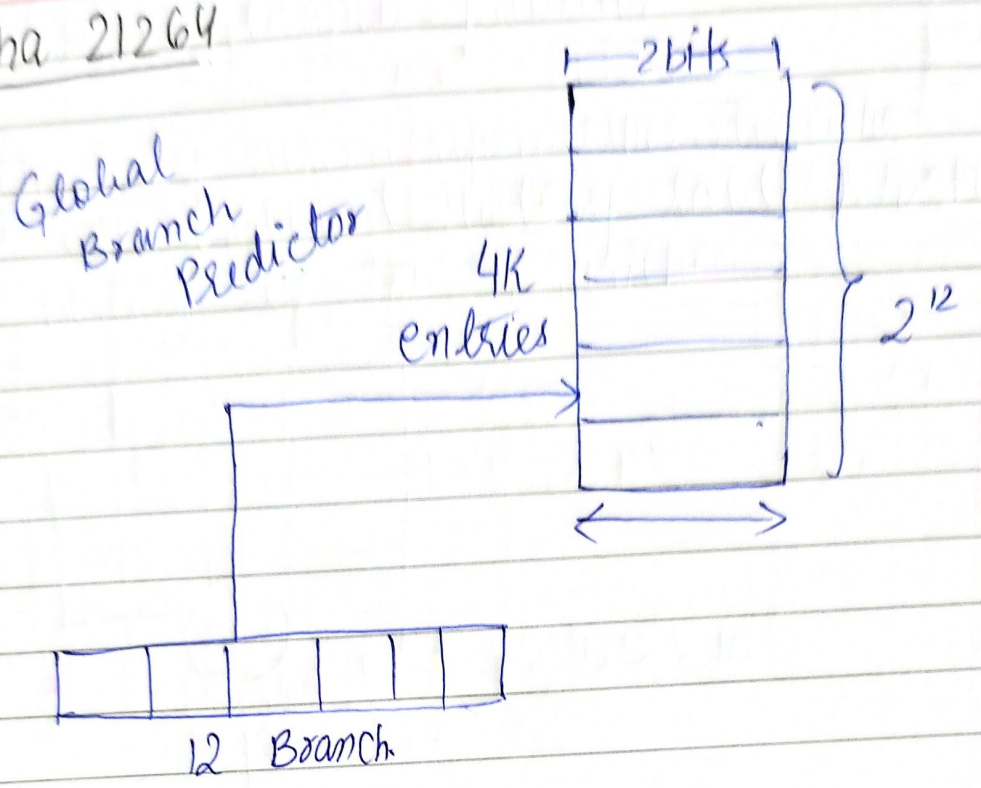
No. of entries = $\frac{8K}{8} = \underline{\underline{1K}}$

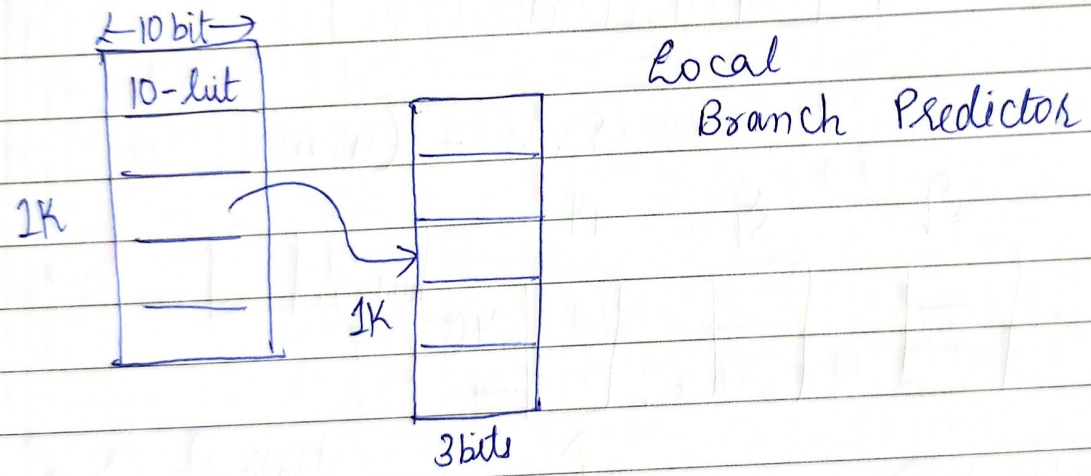
Tournament branch prediction



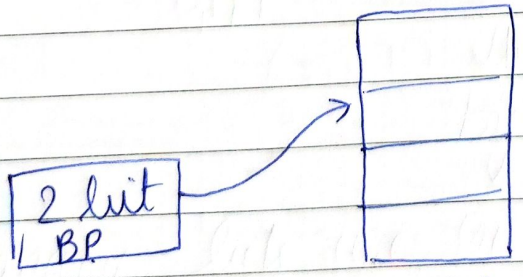
BP → Branch predictor

Alpha 21264





$1K \times 10bit + 1K \times 3bit + 2^{12} \times 2bit + 8K$



$= 10K + 3K + 1K * 8 + 8K$

$10K + 3K + 8K + 8K$

$= 13 + 16$

$= 29$