

$$\text{Floating point number} = (-1)^{\text{S}} (1 \cdot \text{Fraction}) \times 2^{\text{Exp-Bias}}$$

$$= \text{IEEE 754 single precision floating point}$$

Int number = signed magnitude

1's complement

2's complement \rightarrow all modern computer users.

biased

32768

R

$$(-25) \rightarrow \begin{array}{c} 1000 \\ \text{16-bit} \\ \hline 00011001 \end{array} = \frac{1 \text{ 0001111100111}}{15611}$$

1000	+7	-7	+0	-0
sign mag	0111	1111	0000	1000
1	0111	1000	0000	1111
2	0111	1001	0000	0000
biased integer	1110	0000		
	\downarrow $+7 = 14$			

bias (add 7 to all)

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

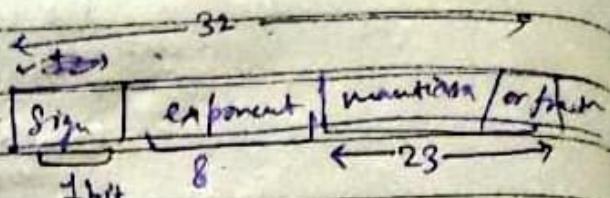
32768

-32768 to 32767

16 67 2272

TEESTI

$$\text{sign} = 1$$



$$2 \times 0.75 = 1.50$$

0.11

$$\text{Floating point normalized} = (-1)^s (1.\text{Fraction}) \times 2^{\text{Exponent}}$$

$$\text{single precision} = 127 = \text{bias}$$

$$= (-1)^1 (1 \cdot 1) \times 2$$

$$= (-1)(1 \cdot 1) \times 2$$

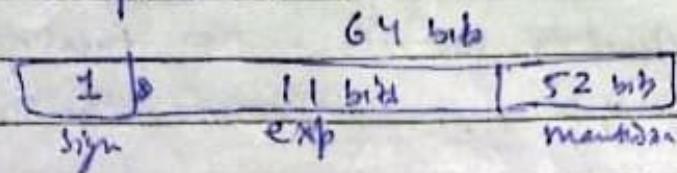
\rightarrow Exp - 128

Exp = 12.6

A handwritten musical staff on lined paper. It features a bass clef at the top left, followed by a box containing two sharps, indicating a key signature of B major. The time signature is common time, represented by a 'C'. The melody begins with a note labeled 'B', followed by two notes labeled 'F', then a note labeled 'B', and finally a note labeled 'G'. After a short vertical line, there is a rest, followed by five '0's representing rests or silence.

BF 00000

double precision :-



$$A = \{ A_1, b \}$$

$$B = \{ : y_1, y_2 \}$$

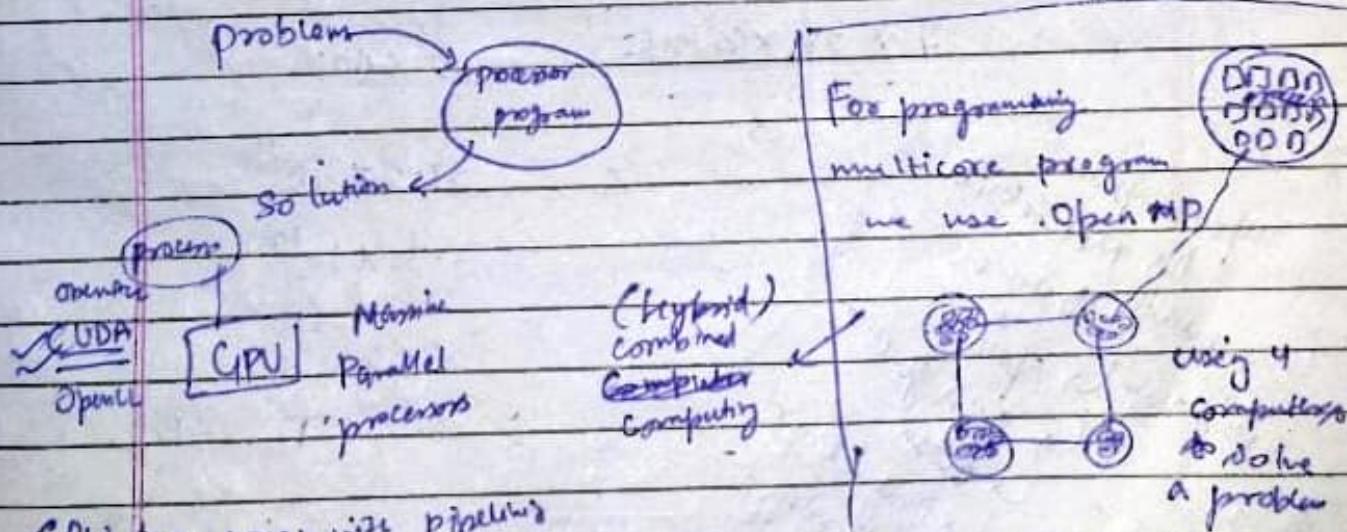
$$AXB = \{ (a_{11}), (a_{12}) \\ (b_1), (b_2) \}$$

table is a relation.

$$R = \{ (a_{11}) \\ (b_1) \}$$

Roll no.	Marks
a	2
b	1

Parallel Distributed Computing



CPU does work with pipeline

GPU does not have core with pipeline

array addition
 $\text{for } (i=0; i<=255; i++)$
 $c[i] = a[i] + b[i]$

if we have 256 cores
 then 1 addition per core

5×10^8 cells

$$\text{If } 2.00 \text{ fp operations per cells} \Rightarrow \text{total operations} = 5 \times 10^8 \times 2.00 \\ \hookrightarrow \text{floating point}$$

$$\Rightarrow 10^{11} \text{ operations}$$

If a computer operates at 1 Gflops .

$$= 10^9 \text{ operations per sec.} \\ \Rightarrow \text{fixed time} = \frac{10^{11}}{10^9} = 100 \text{ sec}$$

: 7 days 1 minute interval.

$$7 \times 24 \times 60 \times 100$$

$$= 1.008 \times 10^6 \text{ sec} = 10^6 \text{ sec}$$

≈ 11.57 day

Increase # cells or use faster computer or both

$$\underbrace{7 \times 24 \times 60 \times 10^{11}}_{B} = 5 \times 60 \cdot$$

$$A = 33.6 \times 10^{11}$$

$$= 3.36 \times 10^{12}$$

$$= 3360 \cdot 4 \text{ flops}$$

$$\approx 3.36 \text{ Teraflops}$$

supercomputer
by Metacogized
Computer Defense
by Space
DNA genetics
green design
dry design
by Compaq

Grand challenge Problem \rightarrow cannot be solved in reasonable amount of time with current computers

Hybrid computing

parallel Computer — using more than 1 computer processor to solve a problem.

to improve performance of your system by performing several tasks ~~at~~ simultaneously

Parallelism in conventional computer may be in different form

1 — I/O

— multiple functional units

— pipelining / overlapping instr.

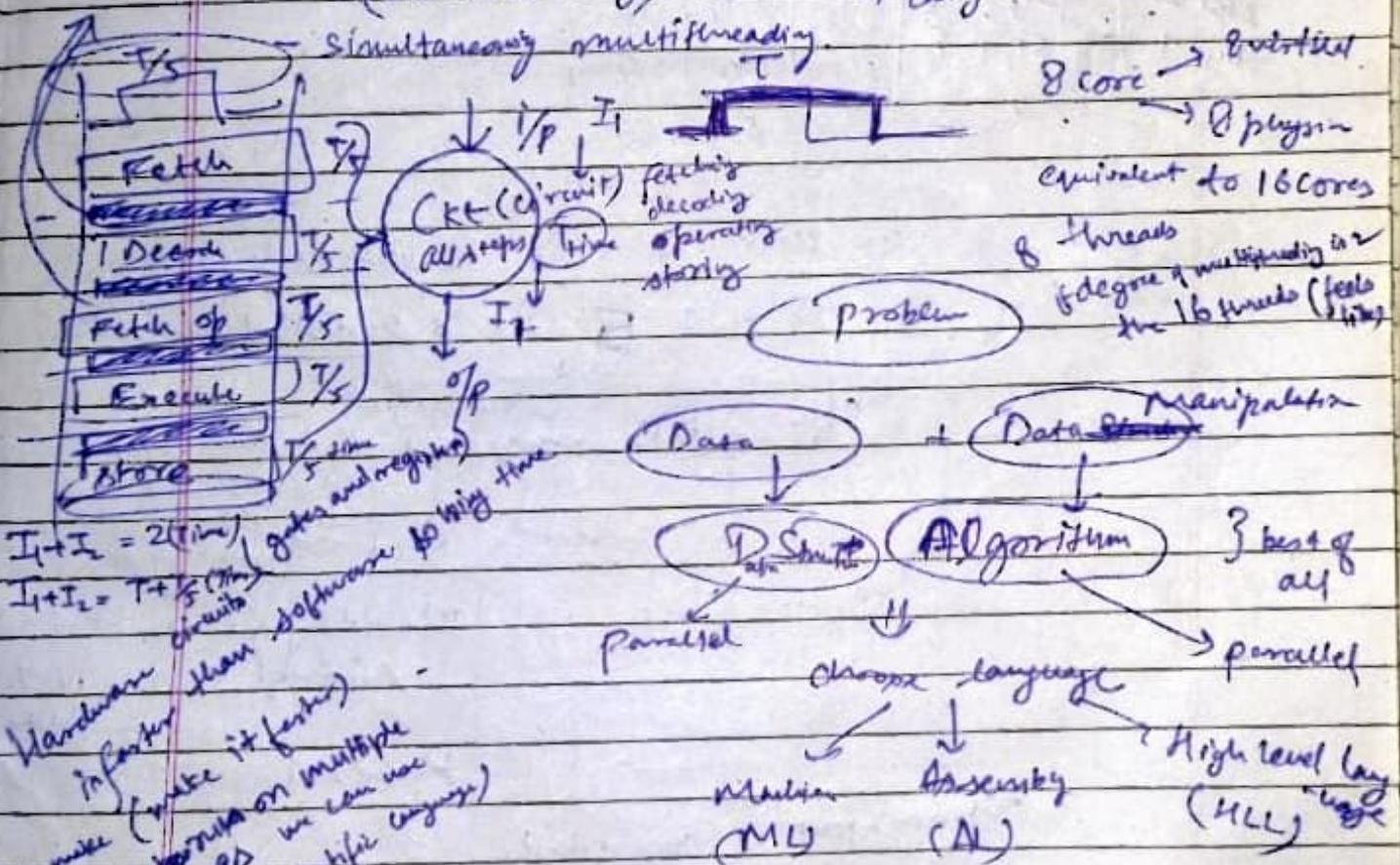
— faster hardware (CSA)

— high speed memory (associative memory) (CLA) carry save adder

— simultaneous multithreading (CLAA) carry look ahead adder

— simultaneous multithreading

pipelining
(memory)
latency

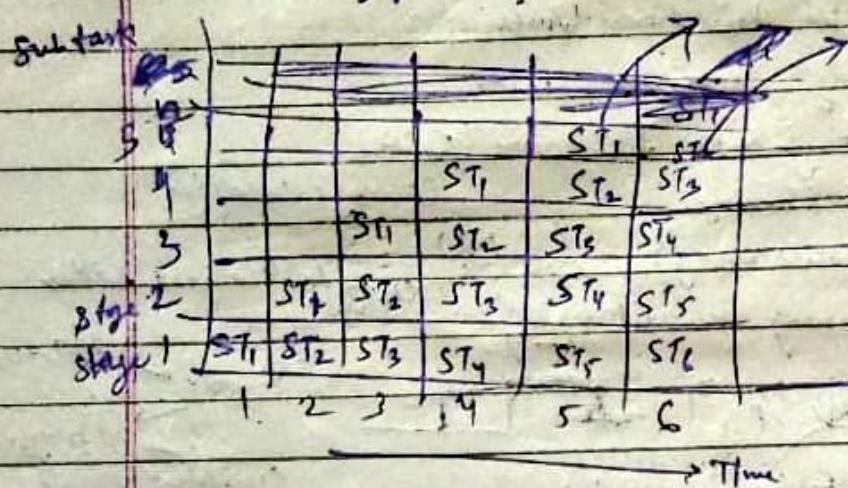


not customized for particular system or special purpose

Pipeline :- no. of segments / stages = M
no. of tasks = N

Total time taken to complete Tasks = ?

Throughput = ?



In clock 2 \Rightarrow 2 subtasks in progress

In clock 5 \Rightarrow 1 subtask 1 is completed
 $= M$

after 5 clocks, ~~every~~^{pipeline is full} 1 task is over
on every clock.

rest = $N-1$ tasks

$$\begin{aligned} \Rightarrow \text{Total time} &= \text{Time to process first task} \\ &+ \text{Time to process } N-1 \text{ tasks} \\ &= M+N-1 \end{aligned}$$

$$\begin{aligned} \text{Throughput} &= \frac{\text{No. of tasks}}{\text{Total time tasks}} = \frac{Nf}{(M+N-1) \text{ clock cycle}} \\ &= \frac{Nf}{M+N-1} \xrightarrow{\text{freq. of pipeline}} \frac{Nf}{M+N-1-N} = f_{\text{freq.}} \end{aligned}$$

higher the frequency, higher the throughput

$$\begin{aligned} \text{Max throughput } f_t &= \frac{Nf}{M+N-1} = f \\ \Rightarrow f &= \frac{Nf}{M+N-1-N} = \frac{Nf}{N+1-M} = f_{\text{freq.}} \end{aligned}$$

$$f = 3 \text{ MHz} = 3 \times 10^6 \text{ Hz} \Rightarrow \text{throughput} = 3 \times 10^6 \text{ Tasks/clk}$$

Page No.: 1 / 1 = sec

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

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

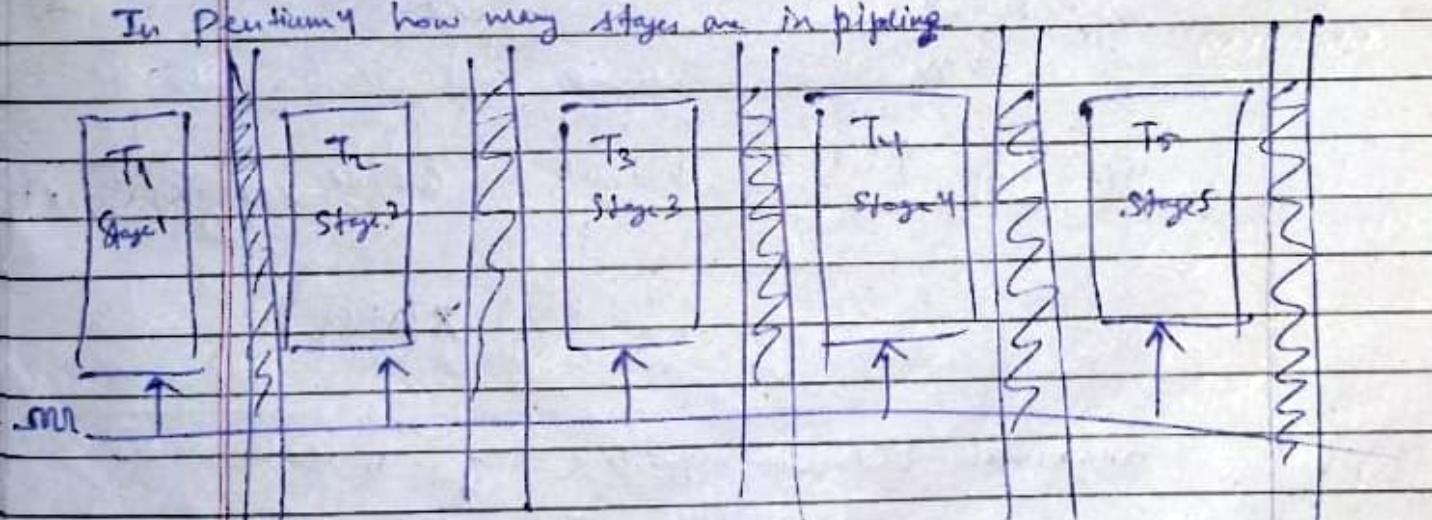
$$= \frac{MN}{M+N-1}$$

$$\text{MAX speedup} = \lim_{N \rightarrow \infty} \frac{MN}{M+N-1} = M = \text{no. of segments}$$

higher the no. of segments/stage higher the speedup
 and also freq will increase

as time taken for each stage to complete ^{out} is reduced \Rightarrow freq increases

In Pentium how many stages are in pipelining.



↳ Latch (time)
 to pipeline

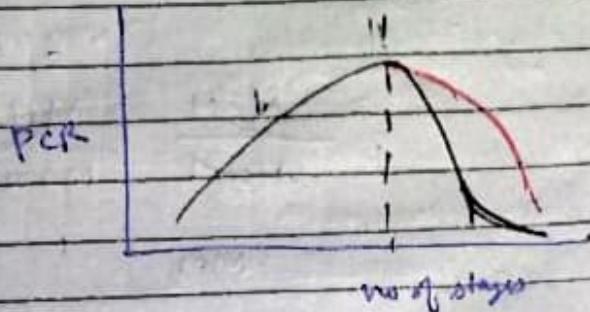
$$\text{freq} = \frac{1}{t_{lat} + \max(T_1, T_2, T_3, T_4)}$$

(Latching)

Performance cost



$$PCR = \text{Performance per Rate} = \frac{\text{Performance}}{\text{Cost}} = \frac{f}{c+kh}$$

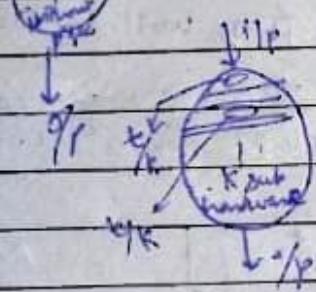
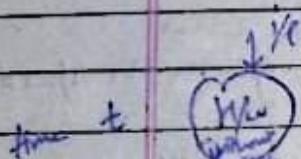


c_f = Cost of digital gates

h = Latch cost

k = no. of stages, d = Latch time

t = time taken by hardware without pipeline.



$$\text{freq} = \frac{1}{t_{\text{data}} \cdot d + t_k} = f$$

$$\Rightarrow PCR = \frac{k}{kd + t} \times \frac{1}{c + kh} = \frac{1}{(d + t_k) \times (h + c_k)}$$

maximize PCR with respect to k

$$= \frac{\partial}{\partial k} \frac{k}{(kd + t)(c + kh)}$$

$$= \frac{d}{dk} \frac{k}{k^2(dh) + (dc + ht)k + ct} = \cancel{\frac{1}{(k^2(dh) + (dc + ht)k + ct)}} \star k$$

$$= \frac{1}{k^2(dh) + (dc + ht)k + ct} + \frac{-k}{(k^2(dh) + (dc + ht)k + ct)^2} \star (2Kdh + dc + ht)$$

$$= \frac{k^2(dh + 2(d + h)t) + ct}{(k^2(dh + (dc + ht)k + ct)^2}$$

$$= \frac{ct - k^2 dh}{(k^2 dh + (dc + ht)k + ct)^2} = 0$$

$$\Rightarrow k^2 dh = ct$$

$$K = \sqrt{\frac{ct}{dh}}$$

$$\frac{d^2 PCR}{dt^2} \rightarrow \frac{ct - k^2 dh}{(k^2 dh + (dc + ht)k + ct)^2}$$

$$\frac{d^2 PCR}{d^2 K} = \frac{-2kdh}{(k^2 dh + (dc + ht)k + ct)^2} \Rightarrow \frac{(ct - k^2 dh)}{(k^2 dh + (dc + ht)k + ct)} \times \cancel{2kdh} \times \cancel{(dc + ht)}$$

$$= (-2kdh)(k^2 dh + (dc + ht)k + ct) \\ - 2(ct - k^2 dh)(2kdh + dc + ht)$$

$$= -2 \left[kdh(k^2 dh + (dc + ht)k + ct) \right. \\ \left. + (ct - k^2 dh)(2kdh + dc + ht) \right]$$

$$= -2 \left[k^3 dh^2 + k^2 dh (dc + ht) + ct \cancel{k^2 dh^2} \right. \\ \left. + \cancel{2kdhct} + dc^2 t + ht^2 c \right]$$

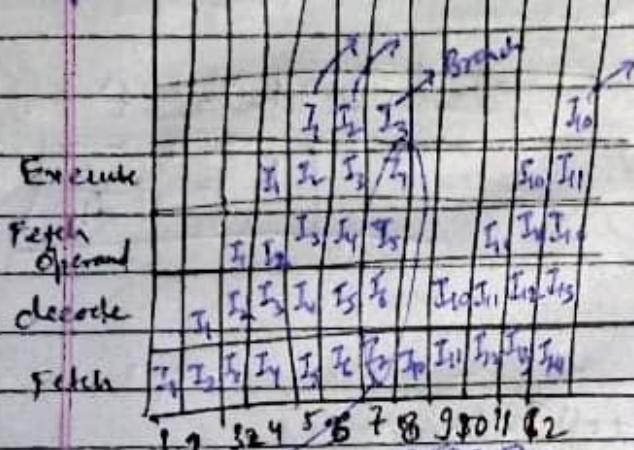
$$\cancel{dk^2 dh^2} - k^2 dh + -k^2 dh^2 t$$

$$\frac{d^2 PCR}{d^2 K} = -2 \left[-k^3 dh^2 + 3kdhct + dc^2 t + ht^2 c \right]$$

Instruction pipeline :-

$I_3 \rightarrow$ branch
Jump to

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



probability that a given instruction will cause branching = S

Waves

1 2 3 4 5 6 7 8 9 10 11 12

① extra cycles

→ no. of clock cycles

i) Speed up

ii) throughput

iii) Avg. no. of Instructions Executed per Instruction cycle

One branch will cause delay = M-1 delay.
Avg. no. of Branches = $\sum_{i=1}^N 1 \cdot S + 0 \cdot (1-S) = \sum_{i=1}^N S$ ↑ close cycles

⇒ total delay caused by all instructions = NS(M-1)

$$\Rightarrow \text{Total time} = T_{\text{pipeline}} + \frac{\text{delay}}{\text{Branch}}$$

$$= N + M - 1 + NS(M-1)$$

$$(i) \text{ speed up} = \frac{N + M - 1 + NS(M-1)}{NM} = \frac{NM}{M + N - 1 + NS(M-1)}$$

$$(ii) \text{ throughput} = \frac{Nf}{N + (M-1) + NS(M-1)}$$

$$\text{new speedup} = \frac{fT}{N+100} = \frac{NM}{M \cdot fN - 1 + NS - NS}$$

$$= \frac{M}{\frac{N+100}{N} - \frac{1}{N} + \frac{NS}{N} - \frac{NS}{N}} = \frac{M}{\frac{1+MS}{S}} = \frac{M}{\frac{1+MS}{1+S(M-1)}}$$

$$\text{Max throughput} = \text{latency} \cdot f$$

$$N \rightarrow \infty \quad \frac{N}{N} + \frac{(M-1)}{N} + \frac{N(M-1)}{N} =$$

$$= 1 \cdot f$$

$$1 + S(M-1)$$

$$\text{if no branching } S=0 \quad \text{max thru.} = f$$

$$\text{and max speedup} = M$$

\Rightarrow as $S \uparrow$ throughput and speedup \downarrow

freq =

$$\text{Latency Time} + \text{Max (Stage } T_1, T_2, T_3, \dots, T_M \text{)}$$

In one Instruction cycle we ~~complete~~ only 1 instruction

\Rightarrow 1 instruction cycle = M clock cycles

$$\Rightarrow \text{avg total I.C.} = \frac{\text{time per}}{M} = N + M - 1 + NS(M-1)$$

$$\text{avg I.C.} = \frac{\text{instructions executed}}{\text{per I.C.}} = \frac{N(M-1) + NS(M-1)}{NM}$$

$$N + M - 1 + NS(M-1)$$

freq of pipeline = f

$$Q \quad \text{No. of instructions} = N, \text{No. of stages} = M = \text{Speed up.}$$

probability the a given instruction is unconditional branch instruction = p

conditional ?? " = q

a conditional branch will cause branch delay = R

calculation: ① total time ② speedup ③ throughput

④ max speedup ⑤ max throughput.

⑥ Avg. no. of instructions calculated per Instruction cycle.

one branch will cause delay = M-1

$$\text{avg no. of branches} = (P + QR)N = \text{unconditional + conditional}$$

total delay caused by all branches = N(P+QR)(M-1)

① Total time = $T_{\text{pipeline}} + T_{\text{branch delay}}$

$$= N + M - 1 + N(P+QR)(M-1)$$

$$② \text{Speed up} = \frac{NM}{N + M - 1 + N(P+QR)(M-1)}$$

$$\textcircled{3} \quad \text{Throughput} = \frac{Nf}{N + (M-1) [1 + N(P+QR)]}$$

$$\begin{aligned}\textcircled{4} \quad \text{max throughput} &= \frac{f}{N \rightarrow \infty} \frac{NM}{M+M-1+N(P+QR)(M-1)} \\ &= \frac{NM/f}{\frac{M}{N} + \frac{N-1}{N} + \frac{N}{N}(P+QR)(M-1)} \\ &= \frac{M}{1 + 1 - 1 + (P+QR)(M-1)} \\ &= \frac{M}{(P+QR)(M-1) + 1}\end{aligned}$$

$$\textcircled{5} \quad \text{Max throughput} = \frac{f}{N + (M-1) + N(P+QR)(M-1)}$$

$$= \frac{f}{1 + (P+QR)(M-1)}$$

\textcircled{6} \quad 1 instruction cycle = M clock cycles

$$\text{total time per IC} = \frac{N + M - 1 + N(P+QR)(M-1)}{M}$$

$$\text{avg instruction executed per IC} = \frac{NM}{M + N - 1 + N(P+QR)(M-1)}$$

\Rightarrow replaced S with $(P+QR)$

$$\text{Clocks per instruction} = \frac{\text{Time}}{\text{avg. time}} = \frac{M + N - 1 + N(P+QR)(M-1)}{N}$$

$$= 1 + \frac{M-1 + (P+QR)(M-1)}{N}$$

$$\text{CPI } N \rightarrow \infty \Rightarrow 1 + \frac{N}{(P+QR)(M-1)}$$

Branch Prediction :-

if prediction is 90% correct
then 90% time don't have to
flush or stall

Speculative calculation :-

we use Branch History Table (BHT)

if BHT bit is 1

↳ then branch Yes

else if BHT bit = 0

↳ then branch No

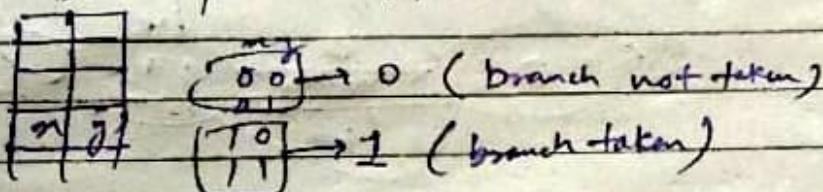
IPC

BHT

Predict.

1 bit Branch predictor → store only last instance of branch by
for a particular instruction.

2 bit Branch predictor → store last 2 instances of branches



3 bit Branch predictor

0	0	0	0	not taken	0, 1, 2, 3
0	0	1	0	taken	
0	1	0	0		
0	1	1	1		
1	0	0	0		
1	0	1	0		
1	1	0	0		
1	1	1	1		

n-bit branch predictor

0 to $2^n - 1 \rightarrow$ not taken

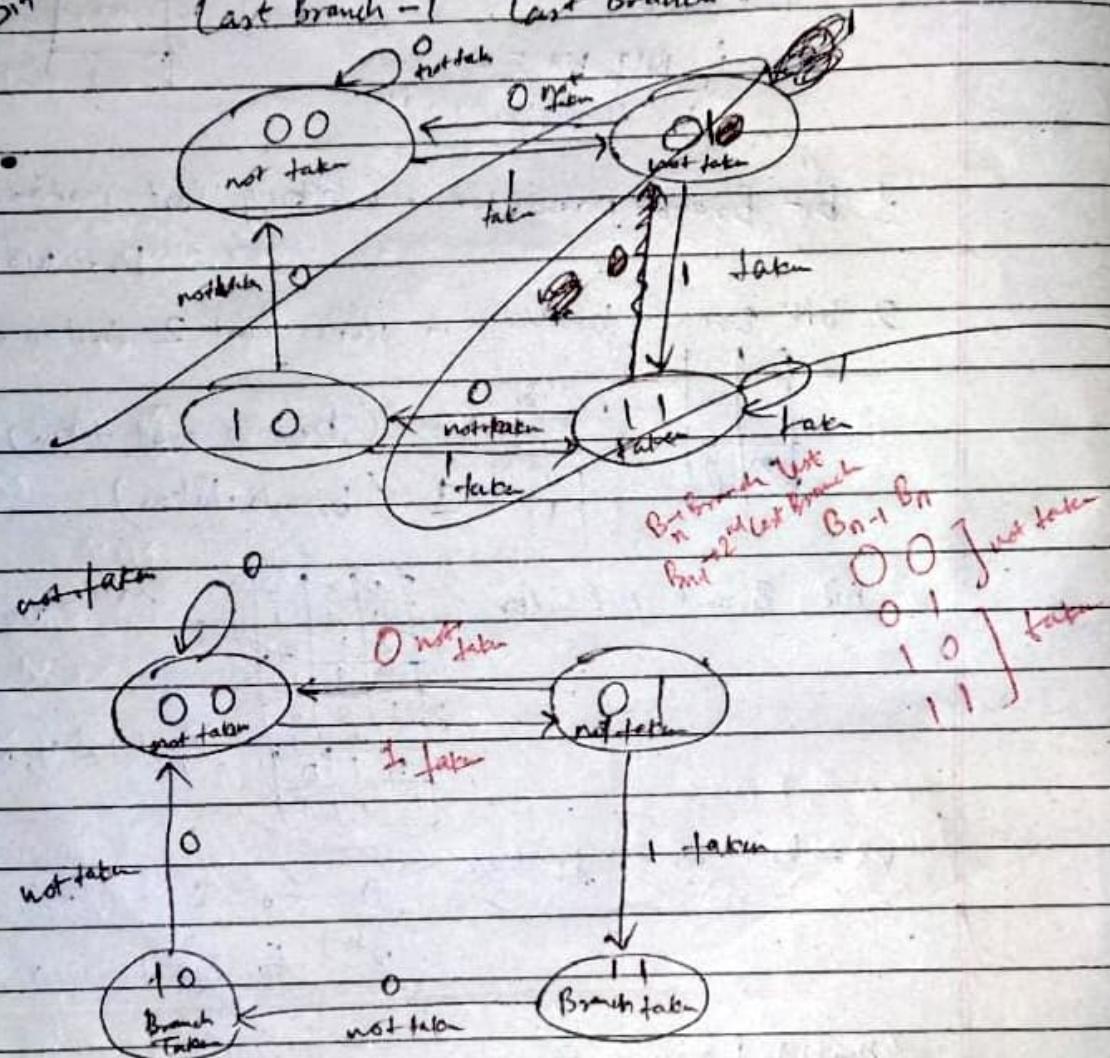
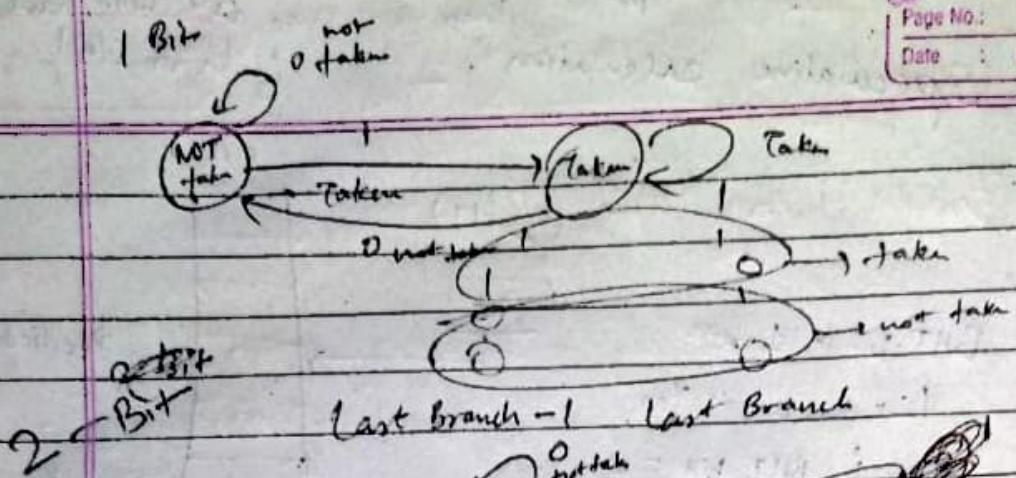
2^n to $2^n - 1 \rightarrow$ taken

(m,n) branch Predictor

Tournament Branch Predictor

BTB (Branch Target Buffer)

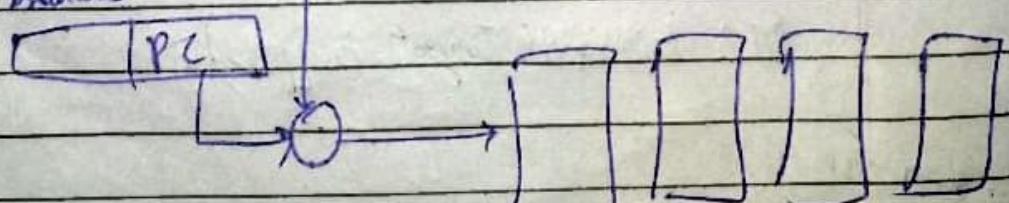
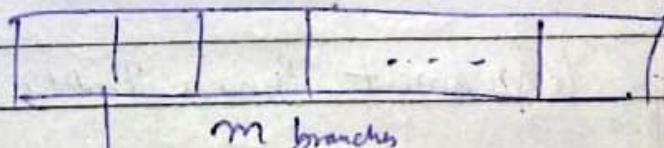
8086-186P



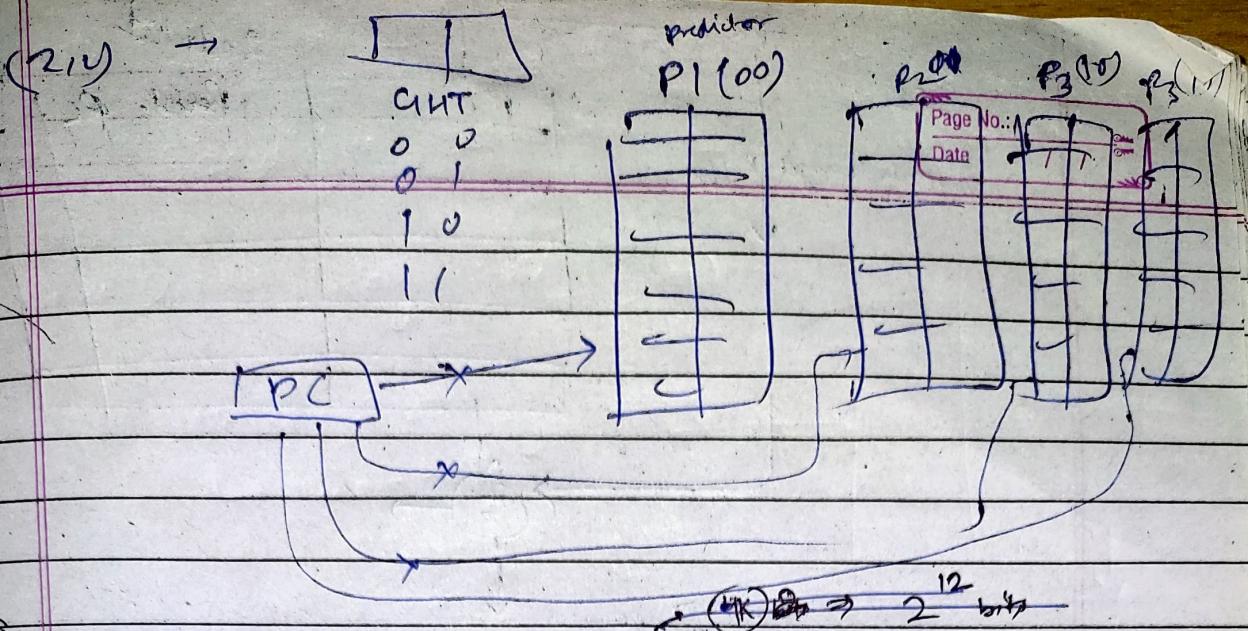
Branch Predictor

(Global history till) GHT

~~records history of in branches~~



g share
g select



- Q How many bits are with (0, 2) branch pred with 4K entries per bit
- Q How many entries are in (0, 2) Branch Pred with same no. of bits

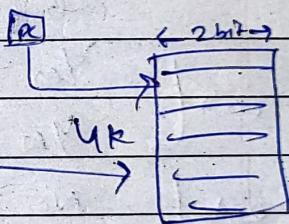
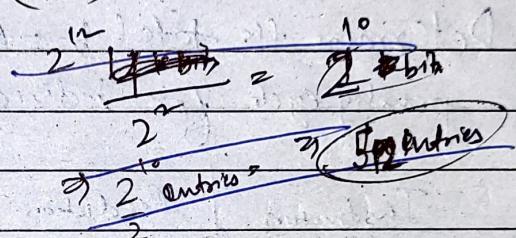
(0, 2) Branch Predictor

$$2^0 = 1 \text{ predictor of } 2 \text{ bits}$$

$\rightarrow 8 \text{ K bits}$

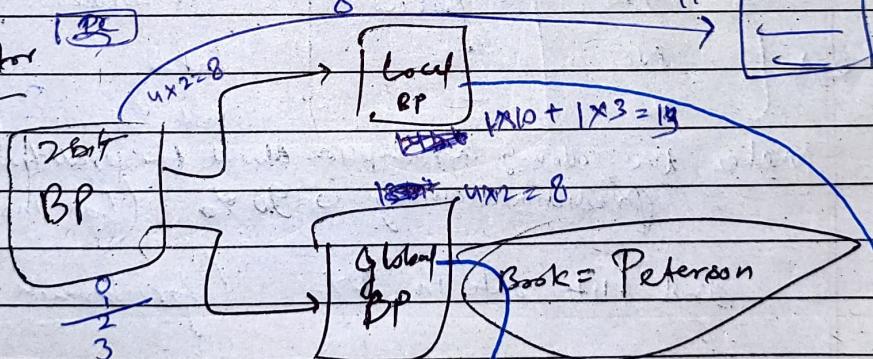
$$\text{no. of entries} \times 2^2 \times 2 = 8 \text{ K}$$

$$\text{no. of entries} = \frac{8 \text{ K}}{8} = 1 \text{ K}$$



Tournament Branch Predictor

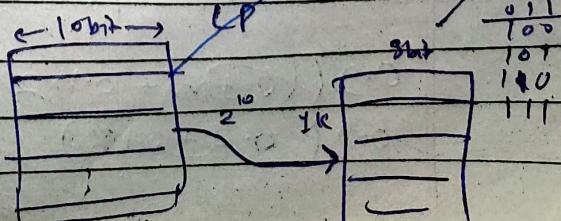
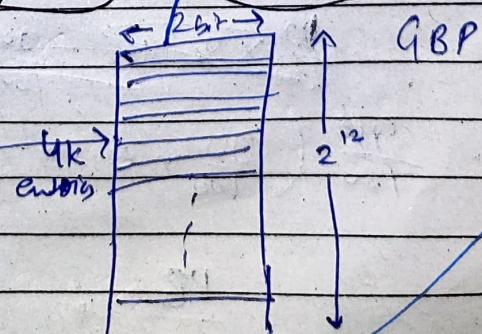
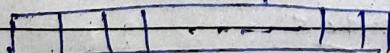
$$\text{Total bits } 8 + 10 + 8 = 26 \text{ bits}$$

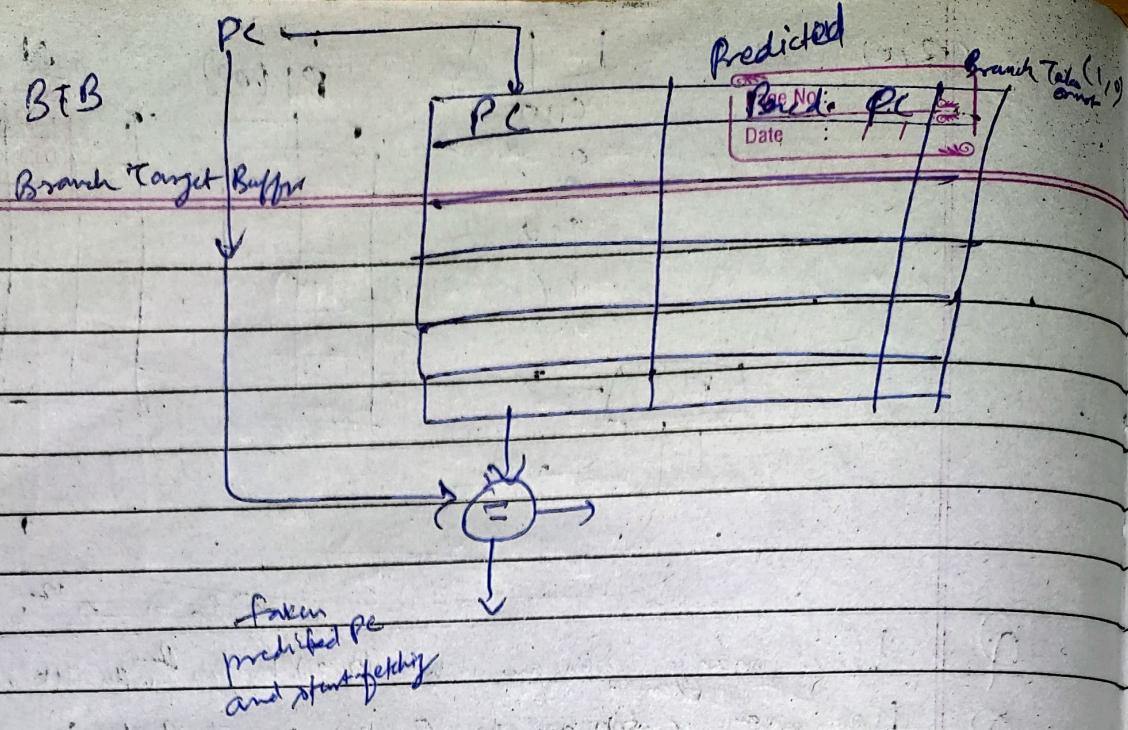


g Alpha System

$$\text{Size} = 30 \text{ K entries}$$

~~Alpha~~ Alpha 21264





Q Determine the total branch penalty for a BTB assuming the penalty cycle for individual prediction as follows.

Instruction in Buffer	Prediction	Actual Redirection	Penalty Cycle
0 → [Yes]	taken	taken	0
0 → [Yes]	not taken	not taken	2
0 → [No]	taken	taken	2
0 → [No]	not taken	not taken	0

Make the following assumption about the prediction accuracy and hit rate
Prediction accuracy is 90% (for instructions in the buffer)

and Hit rate is 90% (for branch predicted taken)

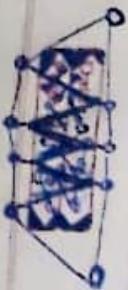
$$\frac{10}{100} \times 2 + \frac{90}{100} \times 0$$

$$\frac{90}{100} \left(\frac{90}{100} \times 0 + \frac{10}{100} \times 2 \right) + \frac{10}{100} \left(\frac{10}{100} \times 2 \right)$$

$$\frac{18}{100} + \frac{10}{100} = \frac{28}{100}$$

$$0.9 \left((0.9 \times 0 + 0.1 \times 2) + 2 \left(0.9 \times 0 + 0.1 \times 2 \right) \right) + 0.1 \times 2 \\ 0.9 (0.2 + 1.8) = 0.9 \times (0.9 \times 0 + 0.1 \times 2) + 0.1 \times 2 \\ = 0.9 \times 0.2 + 0.2 = 0.18 + 0.2 = 0.38$$

Bergman's condition of parallel
homologous member condition



$$\begin{array}{ll} I_1 & I_2 \\ T_1 & T_2 \\ I_3 & \left[\begin{array}{c} I_{12} \\ I_{13} \\ I_{23} \end{array} \right] \\ I_4 & \left[\begin{array}{c} N_{12} \\ N_{13} \\ N_{23} \end{array} \right] \\ I_5 & \left[\begin{array}{c} I_1 \\ I_2 \\ I_3 \end{array} \right] \end{array}$$

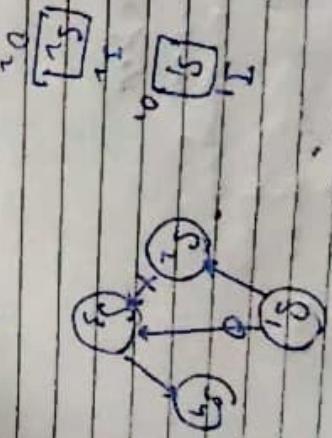
$$\begin{array}{ll} I_6 & I_7 \\ I_8 & I_9 \\ I_{10} & I_{11} \\ I_{12} & I_{13} \end{array}$$

Data Dependence

$$S_1 \xrightarrow{\text{Data}} S_2 \xrightarrow{\text{Data}} S_3 \xrightarrow{\text{Data}} S_4$$

$$\begin{array}{ll} S_1 : & C = D \times E \\ S_2 : & n = G + C \\ S_3 : & A = B + C \\ S_4 : & C = L + M \end{array}$$

$$\begin{array}{l} S_1 \parallel S_2 \\ S_2 \parallel S_3 \\ S_3 \parallel S_4 \end{array}$$



Program Flowchart

$$\begin{array}{ll} S_1 & \text{Load } R_1, A \\ S_2 & \text{Add } R_1, R_2 \\ S_3 & \text{Mov } R_1, R_3 \\ S_4 & \text{Stop } R_1 \end{array}$$

Bernstein's Conditions in Operating System

Bernstein's Conditions are the conditions applied on two statements S1 and S2 that are to be executed in the processor. It states that three conditions that are explained below must be satisfied for two successive statements S1 and S2 to be executed concurrently and still produce the same result.

The intersection between read-write set, write-read set and write-write set of S1 and S2 must be null.

The above statement can be expressed in the form of expressions as following:

1. $R(S1) \cap W(S2) = \{ \}$
2. $W(S1) \cap R(S2) = \{ \}$
3. $W(S1) \cap W(S2) = \{ \}$

The read and write set for the statement $c = a - b$;

$$\begin{aligned} R(c=a-b) &= \{a, b\} \\ W(c=a-b) &= \{c\} \end{aligned}$$

The read and write set for the statement $w = c + 1$;

$$\begin{aligned} R(w=c+1) &= \{c\} \\ W(w=c+1) &= \{w\} \end{aligned}$$

The read and write set for the statement $x = x + 2$;

$$\begin{aligned} R(x=x+2) &= \{x\} \\ W(x=x+2) &= \{x\} \end{aligned}$$

Example-1:

Let,

$$\begin{aligned} S1 : a &= x + y \\ S2 : b &= z + 1 \end{aligned}$$

Check whether two statements S1 and S2 satisfies Bernstein's conditions.

Solution:

$$\begin{aligned} R(S1) &= \{x, y\} \\ W(S1) &= \{a\} \\ R(S2) &= \{z\} \\ W(S2) &= \{b\} \\ \\ 1. \quad R(S1) \cap W(S2) &= \{x, y\} \cap \{b\} = \{ \} \\ 2. \quad W(S1) \cap R(S2) &= \{a\} \cap \{z\} = \{ \} \\ 3. \quad W(S1) \cap W(S2) &= \{a\} \cap \{b\} = \{ \} \end{aligned}$$

Therefore S1 ans S2 can be executed concurrently i.e., Bernstein's conditions are satisfied.

Example-2:

Let,

$$\begin{aligned} S1 : a &= x + y \\ S2 : b &= z + 1 \\ S3 : c &= a - b \end{aligned}$$

Check whether two statements S2 and S3 satisfies Bernstein's conditions.

Solution:

$$\begin{aligned} R(S2) &= \{z\} \\ W(S2) &= \{b\} \\ R(S3) &= \{a, b\} \\ W(S3) &= \{c\} \\ \\ 1. \quad R(S2) \cap W(S3) &= \{z\} \cap \{c\} = \{ \} \\ 2. \quad W(S2) \cap R(S3) &= \{b\} \cap \{a, b\} = \{b\} \\ 3. \quad W(S2) \cap W(S3) &= \{b\} \cap \{c\} = \{ \} \end{aligned}$$

Therefore S2 ans S3 cannot be executed concurrently i.e., Bernstein's conditions are not satisfied.