6/1/2018

High performance Computer Architecture :

Moore! 2x transistor 18-24 month Server

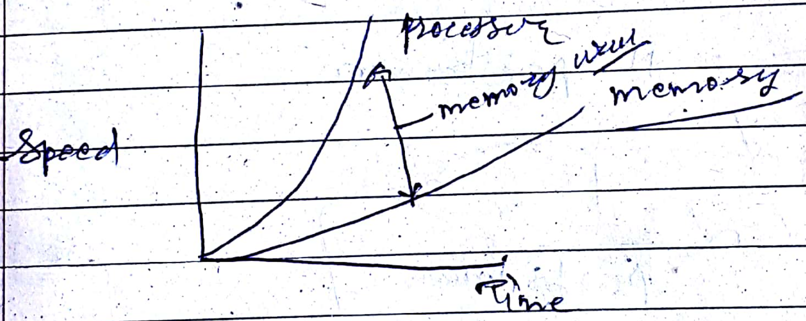
desktops

double processor 2xn 18-24 month -

capacity of memory -

Today	speed	cost	power
	~~cost~~ ↓	↓	
		1x	1x
Tomo	2xp		
	↓		
	1.1x	1/2	1/2

memory wall - difference increasing processor & memory gap



1970

VAX

~~diff~~

we focusing on performance in this subject

~~Sy~~

Syg
(1) ~~bio~~ introduction
(2) - Instruction label parallelism pipelining
(3) Advance TLP
(4) DLP & TLP

→ memory & cache — Jani
other Ganesh

Page No.
Date

~~Ch~~ ~~memo~~ ~~measuring & summerizing~~ performance

Latency → ~~start → done~~

0 — 20 days

~~Ch~~ ~~Through put →~~

Through put → performance

latency → execution time

X Y

P_1 P — Running

t_1 t_2

100 Entire X

10 Ex. Time Y

= N performance

$$\boxed{\text{Execution} \propto \frac{1}{\text{performance}}}$$

$$\text{Throughput} \propto \frac{1}{\text{time/latency}}$$

$$N = \frac{\text{performance } x}{\text{performance } y}$$

$$= \frac{\text{perf } y}{\text{perf } x}$$

$$\frac{P_y}{P_x} = \frac{ET_x}{BT_y} = N$$

Benchmark programme

↳ User Real Application

→ Kernel — core part of Application

Standard Benchmark

Users group

org ← companies

Benchmark Suits — www.spec.org

Spec → SPEC CPU 2006

12 integer programs

17 — floating point programs

gcc

espresso

x y

$$\frac{P_x}{P_y} = \frac{BT_y}{BT_x} =$$

How to summarize performance →

measuring / summarize performance →

~~MCA~~

Page No.
Date

	P1	P2	P3	SRT	
m/c A	~~8~~ 10	~~16~~ 8	~~20~~ 25	~~35~~ 43	2
m/c B	6 12	8 9	~~18~~ 20	~~32~~ 41	1
m/c C	~~7~~ 8	~~9~~ 8	~~19~~ 30	~~30~~ 46	3

3 techniques

(1) Sum of execution time

(2) Sum of weighted execution time

(3).

(1) M/P Ref — SWRT / Best Techn

$\frac{1}{10}$ $\frac{1}{8}$ $\frac{1}{25}$ — 3 — (N)

$12 \times \frac{1}{10}$ $9 \times \frac{1}{8}$ $20 \times \frac{1}{25}$ — 3.125 (3)

$8 \times \frac{1}{10}$ $8 \times \frac{1}{8}$ $30 \times \frac{1}{8}$ (1) 2.0

→ reference machine Ren

geometric mean:→

$GM = \sqrt[n]{\text{product of } n \text{ terms}}$

3, 5, 8

$\sqrt[3]{3 \times 5 \times 9}$

GM and weight

~~14.7~~ 12.59 — (2)

12.92 — (3)

12.42 (1)

SBT, SWBT SIM

	P_1	P_2	P_3
M/C	10	8	25
M/B	12	9	20
M/C	8	8	30

			SBT	SWRT	
5	10	20	35	3.0	10
6	8	18	32	2.9	9.52
7	9	14	30	3.0	7.89

Ans	1	1	1
	1.2	.8	.9
	1.4	.9	.7

SIM $\sqrt{}$ 40×10 = $\sqrt{}$

$$\frac{P_B}{P_A} = \frac{\text{Bx Time A}}{\text{Bx Time B}} = \frac{35}{32} = 1.09$$

$$P_B = 1.09 \times P_A$$

Approach

$$\frac{P_C}{P_A} = \frac{\text{BxT A}}{\text{BxT C}} = 8/8 = 1$$

$$P_C = P_A$$

	A	B	C
P_1	1 sec.	10 s	20
P_2	1000 s	100	20

$gm = \sqrt[2]{1\times 1000}$ $\sqrt[2]{10\times 100}$ $\sqrt[2]{20\times 20}$

$= \sqrt[2]{1000}$ $\sqrt[2]{1000}$ $\underline{\sqrt[2]{400}}$

A = B

$\underline{C = 1.58\ (A, B)}$

P_1 ~~A~~ occurs 100 times for every ~~B~~ $\underline{P_2}$

~~P_2 A~~ $\underline{P_2 + 100 P_1}$

	A	B	C
P_1	100	1000	2000
P_2	1000	100	20

gm $\sqrt[2]{1,0000}$ $\sqrt[2]{1000\times 100}$ $\sqrt{40,000}$

	A	B	C
P_1	1	10	20
P_2	1000	100	20

A = B = 1000, C = 2020

(1) SBT

(2) SWBT — Reference multi Base

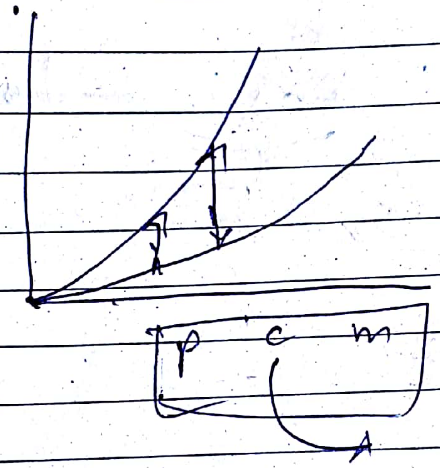
(3) gm

Quantitative principle of Computer Design: →

→ Take Advantage of parallelism → ILP, → DLP, → TLP

→ Locality of reference

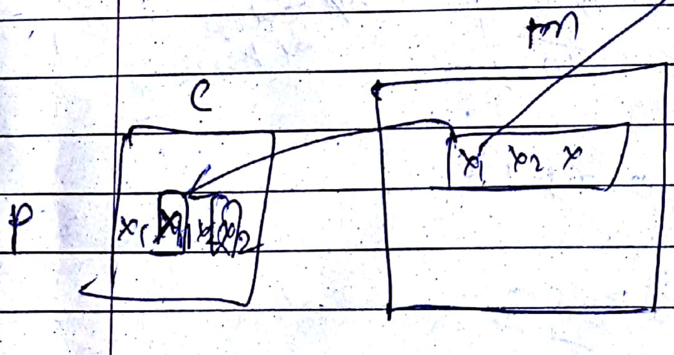
→ Focus on the common case:



Two loc → (1) Spatial locality of reference

(2) Temporal locality of

again



Page No.
Date

S → P_1 90 P_2 10

Focus on Common case:→

Ex. Amdahl's law

State that the perf improvement to be gain using some faster mode of execution is limited by the fraction by time of the time the faster mode is used.

100 Job

80 Jobs   20 Jobs — faster mode of execution

80   5

→ 85

$\frac{100}{85} = 2$

Ex Time new = Ex Timold $\left(1 - \text{Fraction enhanced} + \frac{\text{Fraction enhanced}}{\text{Speedup enhanced}}\right)$

$= 100\left(1 - 0.5 + \frac{0.5}{10}\right)$

$= 100(0.5 + 0.05)$

$= 100(0.55) = 55$

Scanned by CamScanner

Page No.
Date

Speedup :- $\frac{\text{Execution time old}}{\text{execution time new}}$

$$= \frac{\text{Time old}}{\text{Time old}\left(1 - Fe + \frac{Fe}{Sp}\right)}$$

$$= \frac{1}{\left(1 - Fe\right) + \frac{Fe}{Sp}}$$

1) Suppose we are considering enhancement that runs 10 time faster then the orginal machine, but it is usable 40% of time what is speedup, gain by incorporating the enhancement

$$\text{Speedup} = \frac{1}{(1 - 0.4) + \frac{0.4}{10}} = \frac{1}{.6 + 0.06} = \frac{1}{.66}$$

$$= \frac{1}{0.6 + 0.04} = 1.56$$

2) what is the speed up if you make 10% of program runs 90 times faster

3) what is speed up if you make 98% of process 10% faster

Page No. ______
Date

$$\frac{1}{(1-.1)\times\frac{.1}{90}} = \frac{1}{.9+\frac{1}{900}} = 1.10$$

$$\frac{1}{1-0.9+\frac{0.9}{90}} = \frac{1}{1-0.9\times\frac{.9}{10}} = 5.26$$

→ law of diminishing returns →

80% eg. Time 80%.

SE = 2

Impost

$$S = \frac{1}{0.5+\frac{0.5}{2}} = 1.33$$

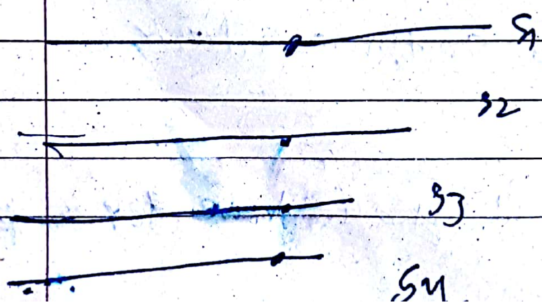
$$\frac{0.5}{} \quad \frac{.28}{} \quad \frac{1}{0.57}$$

$$1-0.33+\cancel{0.7}\ \cancel{0.38}\ \frac{0.33}{2}$$

$= 1.19$

→

$$\frac{1}{1-.67\ \frac{.67}{2}}$$



processor performance equation:-

$$\text{Cpu time} = \text{cpu clock cycle of a program} \times \text{clock cycle time}$$

$$\text{CPI} = \frac{\text{cpu clock of a program}}{\text{instruction count (IC)}}$$

$$\text{cpu clock cycle of programe} = IC \times CPI$$

$$\text{cpu time} = IC \times CPI \times CCT \text{ (clock cycle time)}$$

cycle.cycle 

Ins = 2GHz frequency $= \left(T \propto \frac{1}{F}\right)$

time

suppose a processor takes 2 billion instruction

cpu time = IC × CPI × CCT → H/W Technology

→ user / Compiler

→ computer A

$$= \frac{\text{Instruction}}{\text{program}} \times \frac{\text{clock cycle}}{\text{Instruction}} \times \frac{\text{seconds}}{\text{clock cycle}}$$

$$\text{CPU time} = \frac{\text{seconds}}{\text{program}}$$

(1) different type of instruction.

type-i

type-i+1

,

,

n

cpu clock cycle = $\sum_{i=1}^{n} IC_i \times CPI_i$

$$\text{cpu time} = \sum_{i=1}^{n} IC_i \times CPI_i \times CCT$$

$$\text{overall CPI} = \frac{\sum IC_i \times CPI_i}{IC} = \sum_{i=1}^{n} \frac{IC_i}{IC} \times CPI_i$$

$IC = 1$

Suppose the ~~processor~~ program select 1 billion instruction to execute on a 2 processor (2 GHz proce) and 50% of instruction who CPI is 3 clock cycle and 30% is 4 clock and 20% 5 clock cycle what is cpu time.

$$= \frac{10^9}{2 \times 10^9}\left(\frac{0.5 \times 3}{1} + .3 \times 4 + .2 \times 5\right)$$

$$= 1.5 + 1.2 + 1 = 3.7$$

$$= 3.7 \times 10^8$$

Page No:
Date

$10^9 \times 37 \times 0.5 \times 10^{-9}$

$= 1.85$ sec.

Them = ferold

$$\frac{\text{cpu Time old}}{\text{cpu Time new}} = \frac{Ic_{old}}{Ic_{new}} \times \frac{CPI_{old}}{CPI_{new}} \times \frac{CCT_{old}}{CCT_{new}}$$

some
Energy & power Consumption: →

~~pow~~ p watts $= DP + LP$

$DP \propto$ activity $\times$ capacitance $\times V^2 \times f$

$LP = f(V, I)$

25% $LP \propto V$

Ex → A program runs for 3 second on 100 watt processor and energy

$E = P \times t = 100$ wt $\times 3 = 300$ Joule

CU DFS → Dynamic frequence ~~Stealing~~ scaling.

example. → program x P 100 w T 100 sec.

75 DP 25 LP

old $-100 \times 100 = 10000$

Page No.

Date

1000 w

75 25

375 watt 25

$f = 1 GHz$

f 0.5 GHz

$E = \underline{p \times t} = 62.5 \times 200 = 1.25$

$= 12500$

$E_{new} = 12500$

$E_{old} = 10,000$

$E_{new} = 1.25 \times E_{old}$

DVFS :

P T

100 100

1V — 1ns

0.9V

1.1ns

$75 = (.9)^2 \times 1.1 ns$

~~75~~ 75

25 w

25×0.9

$0.9 f = 1.1 ns$

$75 \times (0.9)^2 \times (0.9) \sin$

$= 54.67$ DP = 85

~~25W~~

29.4

old = 10,00

=

p new = 77w

En = p x t = 77 x 110 = 8470
new

$\boxed{E_{new} = 0.84 \times E_{old}}$

ILP

7/11/2018

(1) Instruction ~~label~~ level parallelism (ILP)
→ pipelining.
→ ~~At~~ Allow instructions to execute in overlap manner

ADD ~~F0~~, R0, R1 R2

RISC

Instruction
| ~~two~~
| 32-bit word → Add R0, R1, R2
64 - double → DAdd R0, R1, R2

(. Load, store Architecture.

x = a+b   LD R0 0(Rx)
          LD R0 0(Ry)

Addressing mode,

16-bit
hould word,

Add R1, R1 R2
store R0 0(R2)

MIPS → multiprocessor without interlocked pipeline stage

Five Stage

- IF →
- ID Instruction decode
- Ex — ALU → AFU
- MEM — LD R0 Rx
- WB → R0 R1+R2

		1cc	2cc	3cc	4cc	5c
PC-	i	IF	ID	Ex	MEM	WB
PC=PC+4	i+1		IF	ID	Ex	MEM
PC=PC+4	I+2			IF		

The objective of pipelining is CPI = 1

			1C	2cc	3cc
PC →	ADD R0 R1 R2 —		IF	ID	Ex

LD R0 0(R2) 1cc 2cc 3cc
IF ID Ex MEM

Page No. ______
Date | | |

L2 20/

BMB2 P_0 P_1 Labey

8C . 21C

Relen ID

① 1cc 2cc 3cc 4cc 5cc

② IF ID Ex MEM WB

③ IMR | Reg | | DR | Rg — buffer

pipeline Register

(4) IF/ID IMR

IMR

Speedup Pipelining: → $\dfrac{\text{execution time unpipelining}}{\text{Ex time pipelining}}$

$$= \frac{\cancel{IC}\ \text{unpipelining}}{\cancel{IC}\ \text{pipelin}} \times \frac{CPI\ \text{unpipe}}{CPI\ \text{pipelin}} \times \frac{\cancel{CCT}\ \text{unpipe}}{\cancel{CCT}\ \text{pipe}}$$

$$= \frac{CPI\ \text{unpipe}}{CPI\ \text{pipelin}} \cong \frac{CPI\ \text{unpipelining}}{1} \cong 1+2$$

λ = pipeline depth

objective

$CPI_{\text{pipelined}} = 1$

Hazard :→ e1

$CPI_{\text{pipelined}} > 1$

(with hazard.

Page No.
Date

Hazards:→

(1) Structural Hazard
(2) data Hazard → Require Data Forwarding.
(3) Control Hazard:

Speedup pipelining with Hazard = $\frac{\text{pipeline depth}}{1 + \text{Stall due to Hazard}}$

Structure Hazard:→ Lack of Resource
& Same Register

I_1	IF	ID	EX	mem	WB
I_2		IF	ID	EX	
I_3					
I_4					

8cc → 11cc
without St (with Stall

Data Hazard:→

DADD	R_1	R_2	R_3	IF	ID	EX	mem	WB
DSUB	R_4,	R_1	R_5		IF	Stall	Stall	IF
AND	R_6	R_1	R_7			IF		IF
OR	R_8	R_1	R_9					
XOR	R_{10}	R_1	R_{11}					

Page No. ____ Date ____

Not work data forwarding

LD	R0	0(R1)	IF	ID	EX	Mem	R0
ADD	R2	R0	R4	IF	ID	EX	mem

Control Hazard ~~→ Ho~~ flushing

	1cc	2cc	3cc	4cc	5cc	6cc
I_1 if (x==0) then	IF	ID	EX	mem	WB	
I_2 ~~x = x + 9~~		~~IF~~	~~ID~~	~~EX~~	~~mem~~	~~WB~~ ~~Imsh~~
I_3 else x = x - 9			IF	ID	EX	mem WB

(1) Flushing

(2) P.U.T Predict untaken used flushing.

(3) Predict taken

(4) Delayed Branch.

if (x==0) then x = x + 9

else x = x - 9

I_4 y = y + 10

I_1	IF	ID
I_4		IF
I_2/I_3			IF I_2/I_3
I_3		

Instruction Scheduling:

I_1 If (x==20) then

I_2 x = x+9

I_3 else x = x-9

predict

I_1 IF ID

Dynamic Branch prediction techniques:

P?

I_1 IF ID

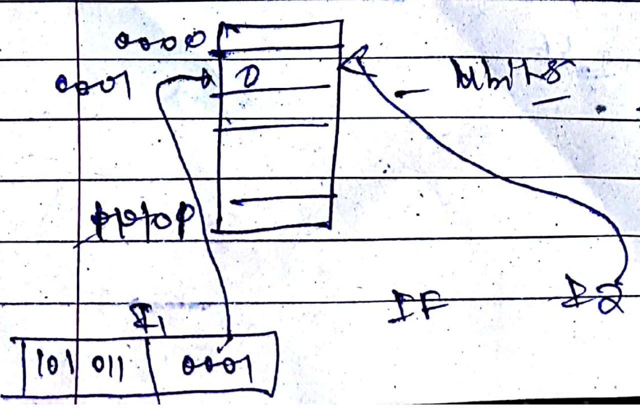
I_2 / I_3

(1) 1-bit predictor
(2) 2-bit predictor
(3) Correlating predictor
(4) Tournament Branch Predictor
(5) Branch Target Buffers
(6) Return Address Stack

Correlated

Local / global predictor
gshare / a select

BHT Branch History Table



0 - NT

1 - T

Page No. ____
Date

IF IR I2

0001

4-bit

memory aliasing → two instruction are pointing same address BHT

while (0)

B1 (0001)

for (i=0; i<10; i++)

{

:

} BR1 (NT)

for (j=0; j<20; j++)

B2 (1,000)

{

:

} BR2 NT

}

0001 | 0
(1,000) | 0

1	2	3	4	5
NT	T	T	T	T
T	T	T	T	T
6	7	8	9	10
T	T	T	T	NT

8

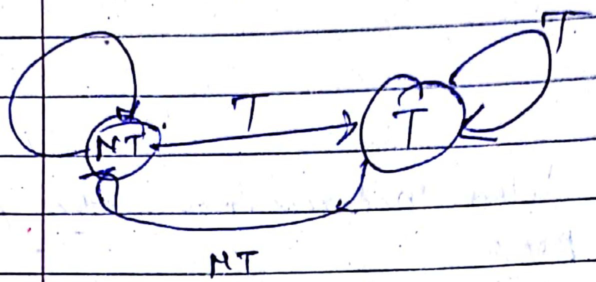
$\frac{8}{10} + \frac{18}{20} = \frac{26}{30} = 87\%$

PVT → $\frac{1}{10} + \frac{1}{20} = \frac{2}{30} = 6\%$

predict on taken

Page No.
Date.

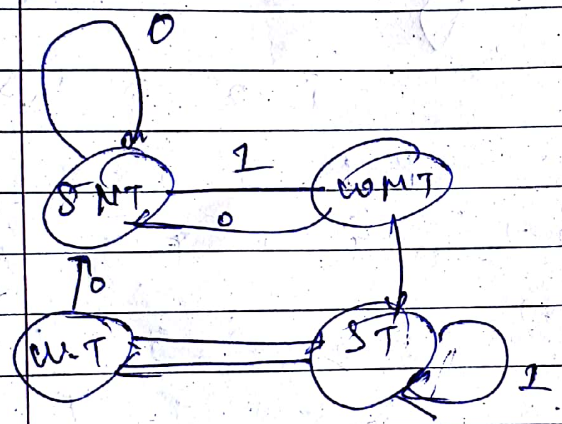
~~2-bit predictor:→~~



2-bit predictor:→

11
10
00
01

0 0 - strong
01/10 weekly
~~10~~ strongly taken



9/10 → 19/20 → 20/3 = 93%

NT T NT T NT T → 1-bit poor
↳ 2-bit better

NT | if (x == 2)
x = 0.

NT | if (y == 2
y = 0

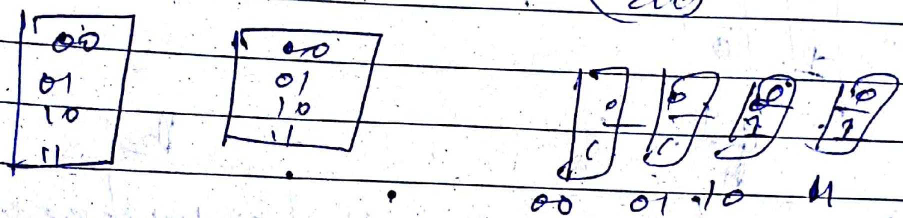
T if (x != y) ← correlating predictor
~~stat1~~ (m, n) predictors
~~el~~ else (m- branches previous
Stat2 n → n-bit inside the buffer

2^m - buffer

each buffer maintain n bits

(1, 2) predictor

(2, 1) 2^2



{
B1 : BNE2
B2 : BNE2
}

~~87~~ 93 – 95

B1 → T, NT, T, NT, T

B2: T T T T NT

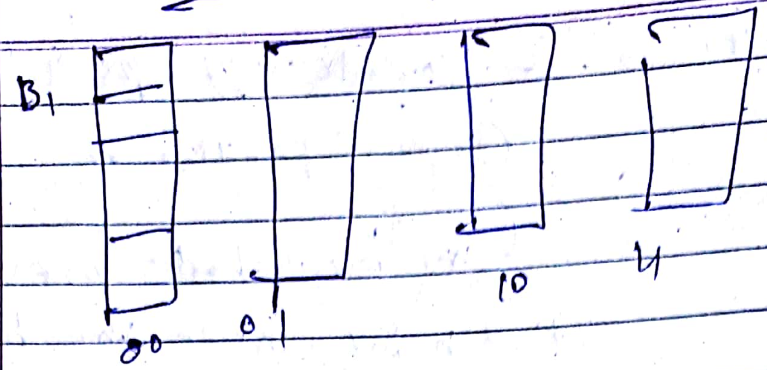
B1

~~# Iteration Predictor | Prediction | outcome~~

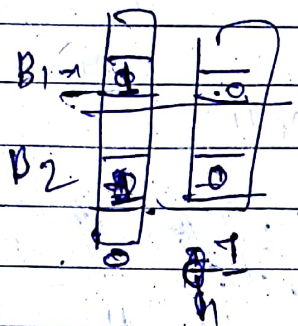
Iter	Hist	Pgr	Pin	outcm	~~#~~ predictor	prediction	outcm
	00	0101010		T			
	11	1101010		NT			T
	01	1		T			T
	11			NT			T
	01			T			NT

Scanned by CamScanner

(1,2) scheme



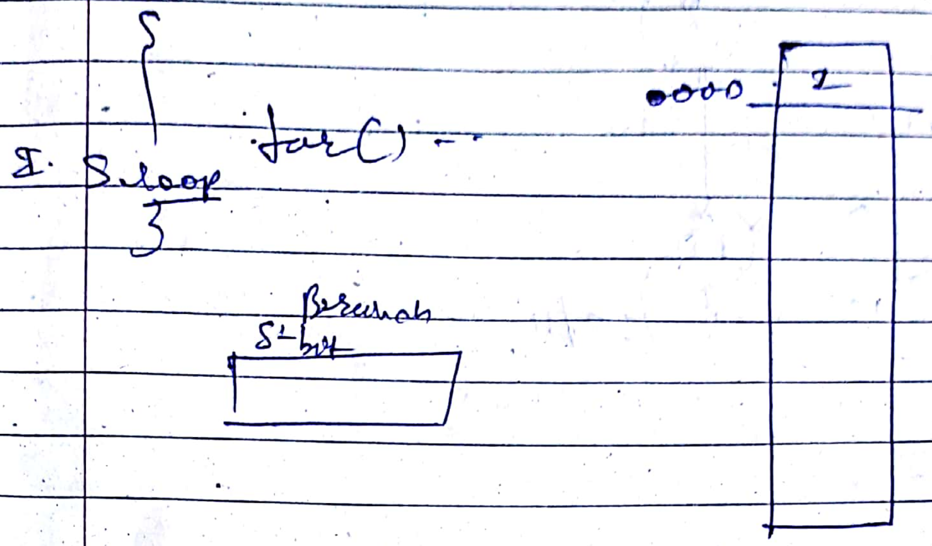
Same (1,2)



B2

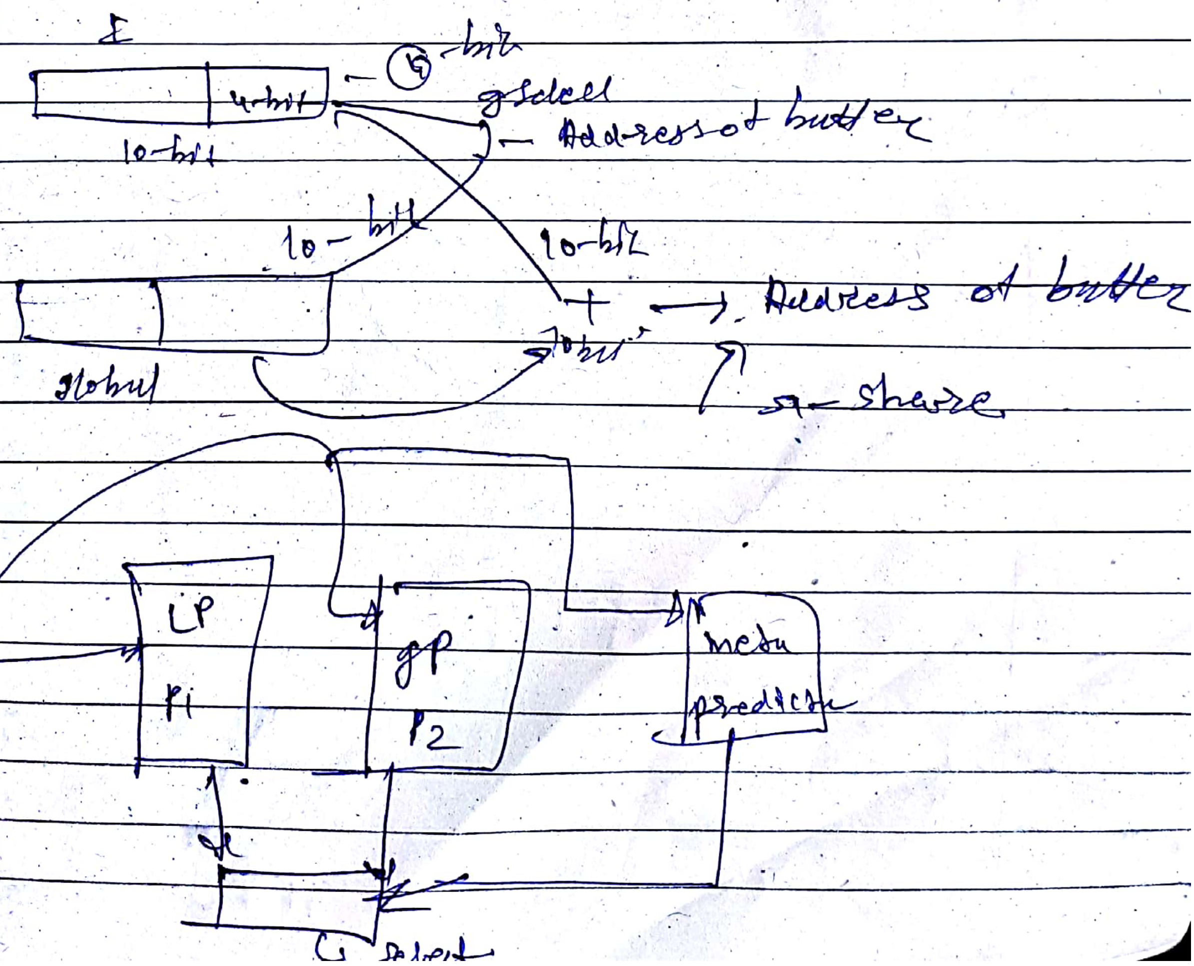
It	history	prediction	prediction	outcome	hist	prediction	prediction	out
1	0	0.1.0	MT	T	1	0/0	0 MT	T
2	1	1.1.0	NT	MT	0			T
				T				T
				MT				MT
				T				

local predictor :→ depends ~~own~~ on own Branch



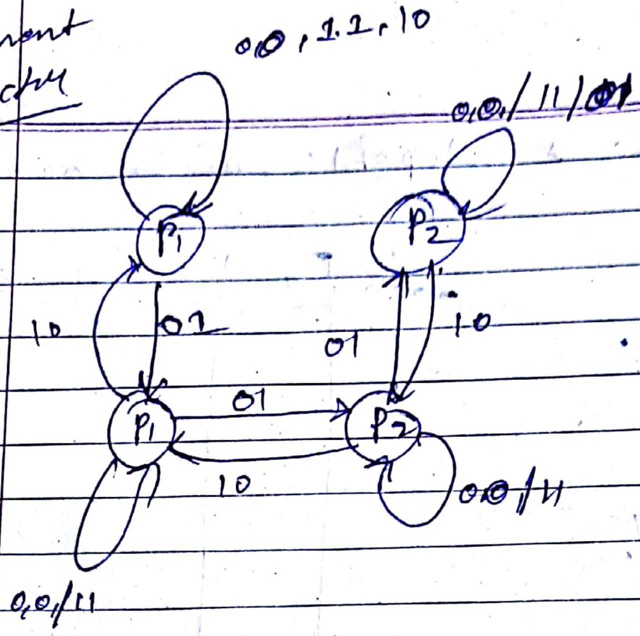
global predictor :→ depends one other Branch

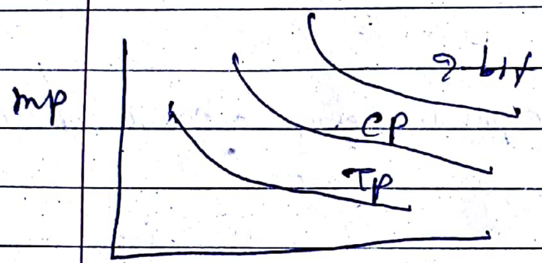
(1) a select
(2) a - share



Page No.
Date

Tournament predictor





→ Advance ILP

Q.