

6/11/2018

Page No. _____
Date _____

High performance Computer Architectures

Model architecture 10.24 month Server

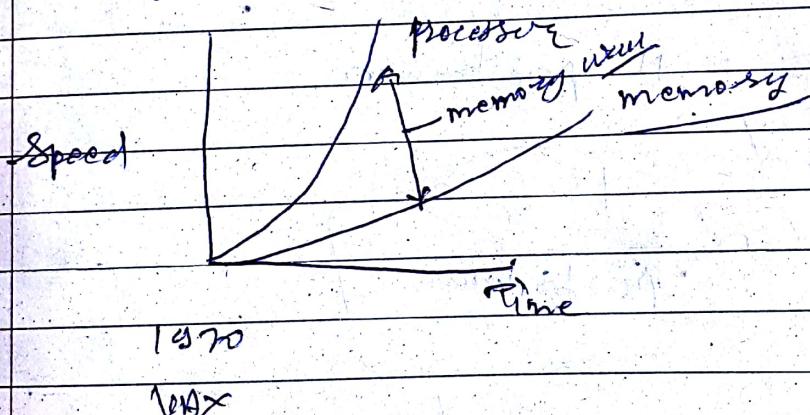
position []

doubt processor gen 10.24 month

capacity of memory -

Today	speed	cost	power
	1x	↓	
Tomorrow	2x P	1x	1x
	↓	y_2	y_2

memory will difference increasing processor & memory gap



Q1 we focus on performance in this subject

- Q1
- (1) Introduction,
 - (2) → Instruction level parallelizing ability
 - (3) Advance TLP

Q2 DLP & TLP

→ memory & cache

some points

Ques 1) Measuring performance

(a) Measuring or summarizing performance

Latency \rightarrow Start \rightarrow done.

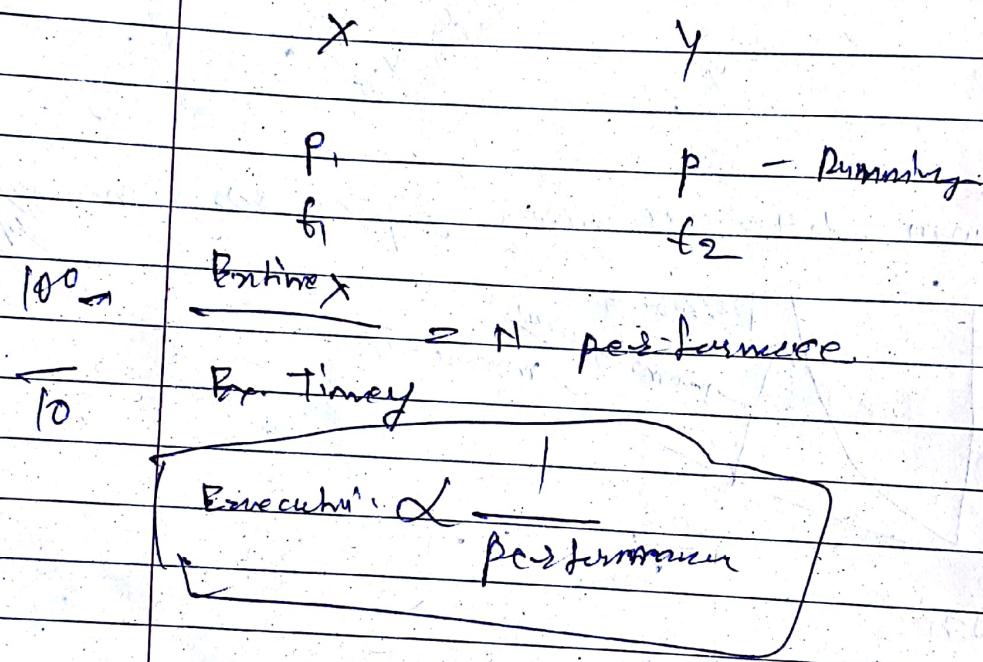
$t_1 - t_0 = \text{Latency}$

(b) Throughput \rightarrow

$\frac{1}{f}$

Throughput \rightarrow performance.

Latency \rightarrow execution time



Throughput \propto 1

time latency

$$H = \frac{\text{Performance}_X}{\text{Performance}_Y}$$

\propto Performance_X

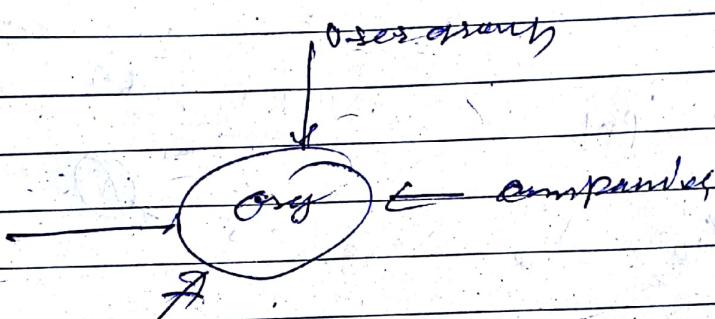
\propto ~~Performance_Y~~

\propto ~~latency_X~~

$$\frac{P_x}{P_{xy}} = \frac{E_{Tx}}{B_{Tx}} = N$$

Benchmarking programme

- ↳ User Real Application
- Kernel - core part of Application
- Standard Benchmark



Benchmarking units - www.spec.org

Spec → SPEC CPU 2006

X	Y	$\frac{P_x}{P_{xy}} = \frac{B_{Tx}}{B_{Ty}} =$	10 - integer programs
			17 - floating point programs
P _x	B _{Ty}	$\frac{P_x}{P_{xy}} = \frac{B_{Tx}}{B_{Ty}} =$	gcc
			espresso

How to summarize performance →

measuring / summarizing performance →

intel

	P ₁	P ₂	P ₃	SRT
M/C A	8/10	16/8	20/8	38.43 - ②
M/C B	6/12	8/9	18/8	32.41 ①
M/C C	7/8	9/8	19/30	36.46 ③

3 techniques

- (1) Sum of execution time
- (2) Sum of weighted execution time
- (3) *Weighted*

(4) MRP Ret

SWRT / best

$$Y_{10} \quad Y_8 \quad Y_{25} - 3$$

$$\frac{12 \times 1}{10} \quad 9 \times 1 \quad 20 \times \frac{1}{8} - 3.125$$

$$8 \times \frac{1}{10} \quad 8 \times \frac{1}{8} \quad \cancel{8 \times \frac{1}{10}} \quad 30 \times \frac{1}{8} - 2.0$$

reducing machine per

geometric mean

$$\sqrt[n]{\text{product of numbers}}$$

$$31.510$$

$$3 | 3 \times 5 \times 9$$

sum avoid weight

$$14 + 12.59 - ④$$

$$12.92 = ⑤$$

$$12.42 ①$$

SBT, SWBT sm

Page No. _____
Date _____

	P_1	P_2	P_3
MC	10	8	25
MBS	12	9	20
MC	8	8	30

	<u>SBT</u>	<u>SWBT</u>
5	10 20	35
6	8 18	32
7	9 14	30
		3.0
		10
		9.52
		7.89

75 9 1 7
10.2 8 9
1.4 9 7

sm \int 40×10 $= \int$

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

$$P_B = 1.09 \times P_A$$

Approach

$$\frac{P_C}{P_A} = \frac{\text{Bx T A}}{\text{Bx T C}} = \frac{35}{18} = 2$$

$$P_C = P_A$$

A B C

P_1 1 sec. 10_s 20

P_2 1000 s 100 20

$$g_m = \frac{2}{\sqrt{100}} = \frac{2}{\sqrt{10 \times 100}} = \frac{2}{\sqrt{2000}}$$

$$= \frac{2}{\sqrt{1000}} = \frac{2}{\sqrt{100}} = \frac{2}{\sqrt{400}}$$

$$A = B$$

$$C = 1.58 (A/B)$$

P_1 occurs 100 Times for every $B \times P_2$

~~$P_1 + P_2 + 100 P_1$~~

A B C
 ~~P_1~~ 100 1000 2000

P_2 1000 100 20

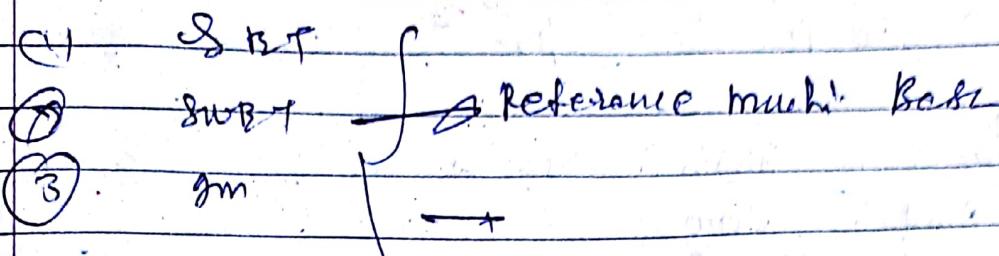
$$g_m = \frac{2}{\sqrt{10000}} = \frac{2}{\sqrt{1000 \times 100}} = \frac{2}{\sqrt{20000}}$$

A B C

P_1 1 10 20

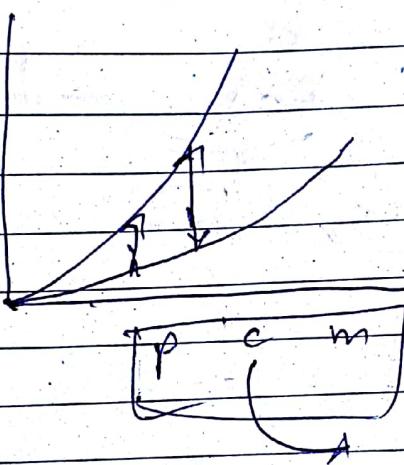
P_2 1000 100 20

$$A = B = 1000, C = 2020$$



Quantitative principle of Computer design →

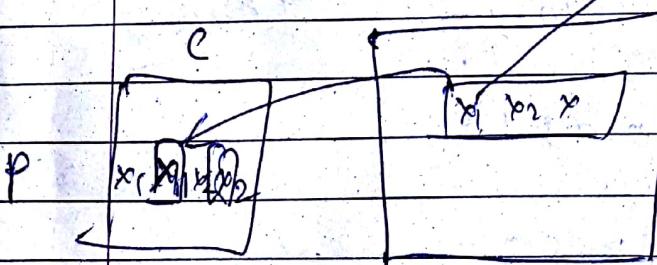
- Take Advantage of parallelism of IIP
- Velocity of reference → DLP
- Focus on the common case → TLP



Two slot → (1) Spatial availability of reference

(2) Temporal closest at

t_m



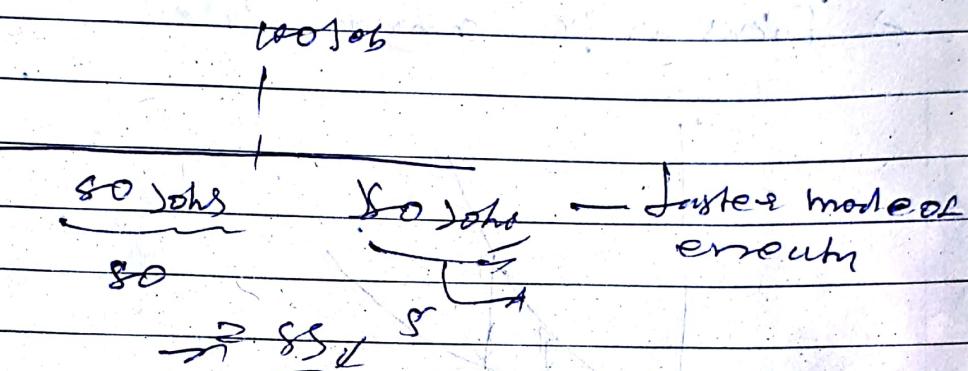
~~8~~ → ~~P_i~~ ~~90~~

~~10~~ ~~P_m~~

Focus on Gantinon case:-

Analysing story

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



$$\frac{100}{88} \geq 2$$

$$\text{By Time Anew} \geq \text{By Timold}$$

(1) Function enhanced
 + Process enhanced
 $\geq 100 (1 - 0.5 + 0.05)$ speedup enhanced
 $\geq 100 (0.5 + 0.05)$

$$\geq 100 (0.55) \geq 55$$

$$\geq 100 (0.85) \geq 85$$

Speedup = $\frac{\text{Execution time old}}{\text{Execution time new}}$

Time old

$$\text{Time old} \left(1 - \frac{\text{Fe}}{\text{SP}} \right)$$

$$\left(1 - \frac{\text{Fe}}{\text{SP}} \right) + \frac{\text{FP}}{\text{SP}}$$

Suppose we are considering enhancement that runs 10 times faster than the original machine, but it is unable to ~~use~~ of time what is speedup, gain by incorporating the enhancement

$$\text{Speedup} = \frac{1}{\left(1 - 0.4 \right) + \frac{0.6}{10}} = \frac{1}{0.6 + 0.06} = \frac{1}{0.66} = 1.56$$

(2) What is true Speed up if you make 10% of program runs 10 times faster

(3) What is Speed up if you make 90% of process fast. ~~10 times~~

$$\frac{1}{(1-0.1) \times \frac{1}{90}} = \frac{1}{0.9 + \frac{1}{900}} = 1.10$$

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

\rightarrow law of diminishing returns \rightarrow

say ex. Time 80%.

$$SB = 2$$

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

$$\begin{array}{r} 0.5 \quad | \quad 0.25 \\ \hline 0.5 + \frac{0.25}{2} = 0.33 \end{array}$$

$$= 1.19$$

$$\begin{array}{r} 1 \\ \hline 1-0.67 \quad -0.67 \\ \hline 2 \end{array}$$

S1

S2

S3

S4

processor performance: equation:-

$CPU \cdot \text{time} = \text{CPU clock cycle of a program} \times \text{clock time}$

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

$\text{CPU clock cycle} = IC \times CPI$
of program

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

cycle cycle 
 Instruction frequency = $(TCK \cdot \frac{1}{F})$
 Time

Suppose a processor takes 1 billion instructions

$CPU \text{time} = IC \times CPI \times CCT \rightarrow \text{H/W Technology}$

↳ User's Compiler

→ Computer

= Instructions \times clock cycle \times Second
program Instruction clock cycle

$CPU \text{time} = \frac{\text{Second}}{\text{Program}}$

(4) different type of instruction.

type-i

type-ii

n

$$\text{CPU clock cycle} = \frac{\sum_{i=1}^n I_{C_i} \times CPI_i}{I_C}$$

$$\text{CPU time} = \frac{\sum_{i=1}^n I_{C_i} \times CPI_i \times CCT}{I_C}$$

$$\text{overall CPI} = \frac{\sum_{i=1}^n I_{C_i} \times CPI_i}{I_C} = \frac{\sum_{i=1}^n \frac{I_{C_i}}{I_C} \times CPI}{I_C}$$

Suppose the processor takes 1 billion instruction to execute on a processor (289) and 50% of instruction whose

CPI is 3 clock cycle and 30% is in clock and 20% Sclock cycle

what is CPU time.

$$\begin{aligned}
 &= \frac{0.5 \times \frac{3}{1} + 0.3 \times 4 + 0.2 \times 5}{10} \\
 &= 0.5 + 1.2 + 1 = 3.7 \\
 &= 3.7 \times 10^8
 \end{aligned}$$

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

$$\approx 1.85 \text{ sec.}$$

$$\frac{\text{CPU Time Old}}{\text{CPU Time New}} \approx \frac{T_{\text{Old}}}{T_{\text{New}}} \times \frac{CPI_{\text{Old}}}{CPI_{\text{New}}} \times \frac{CCT_{\text{Old}}}{CCT_{\text{New}}}$$

Some energy & power Consumption: -

$$\text{gross p worth} = \Delta p + Lp$$

$$\text{OpA activity} \times \text{Capacitance} \times V^2 \times f$$

$$L_P = f(V, I)$$

9st: CP \propto V

Expt A - program runs for 3 seconds
on 100 watt processor and energy

$$B = \mu_0 I t = 100 \text{ nT} \times 3 = 300 \text{ Jc}$$

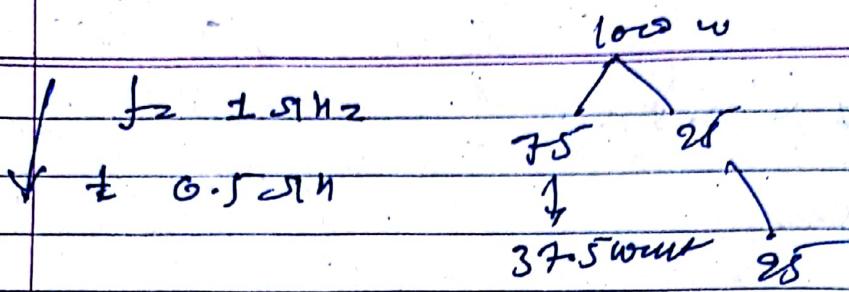
(v) DFS \rightarrow dynamic frequency scaling.

~~252P~~ 23 LP

old $- 100 \times 100 = 10000$

Page No.

Date



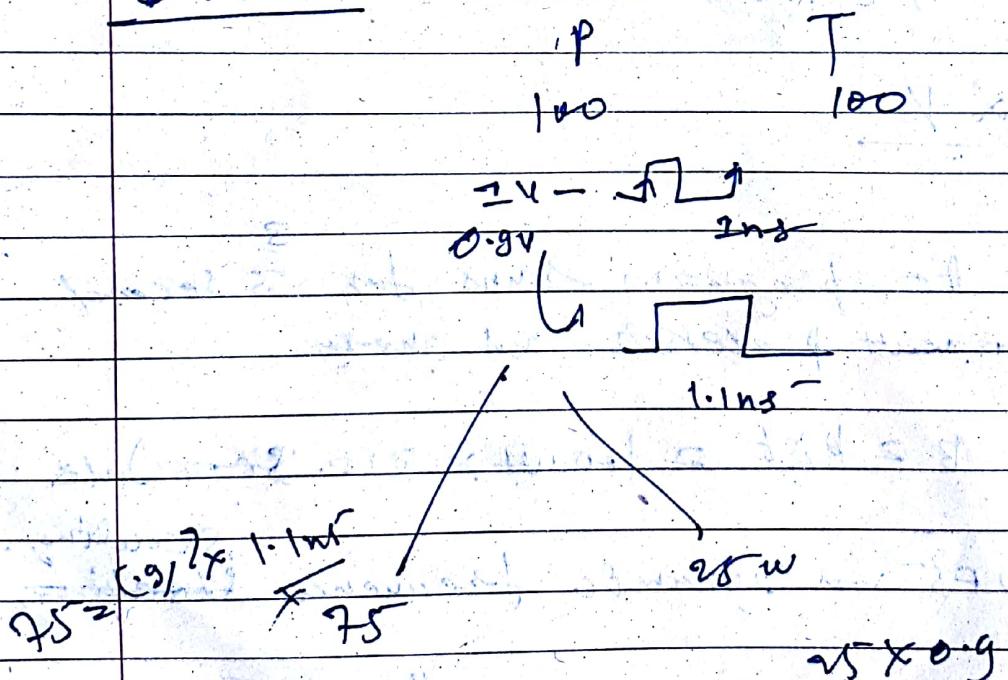
$$E = \frac{P \times t}{t} = 62.5 \times 200 = 1.25 \\ = 12500$$

$$E_{new} = 12500$$

$$E_{old} = 10000$$

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

DVFS : \rightarrow



$$0.9 f = 1.1 \text{ ms}$$

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

$$\geq 84.67 \text{ W} = 85$$

29. V

$$w/d = 10/00$$

2
pnew = 72w

$$E_n = p \times t = 72 \times 110 = 38470$$

new

| $B_{new} = 0.84 \times B_{old}$ |

ILP

~~7/11 23/18~~

(1) Instruction ~~set~~ level parallelism (ILP)
→ pipelining.

→ Allows instructions to execute in overlap manner

ADD fo, R₀, R₁, R₂

RISC

Instruction

| two

32-bit word → Add R₀, R₁, R₂

64-bit double → 2Add R₀, R₁, R₂

C. closed store Architecture

nzadd L2 R₀ o(Cpx)

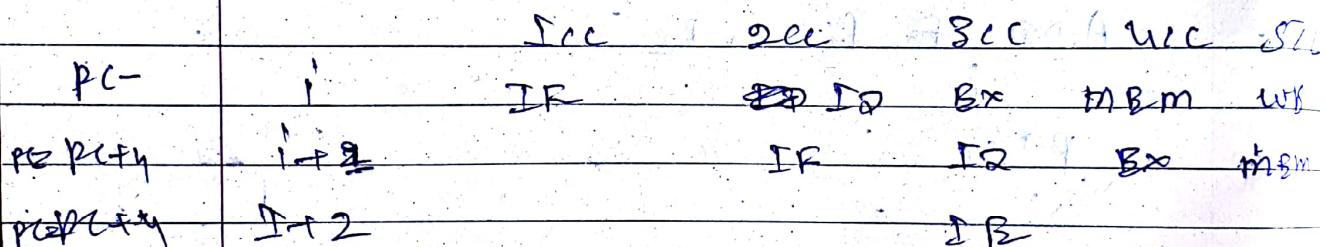
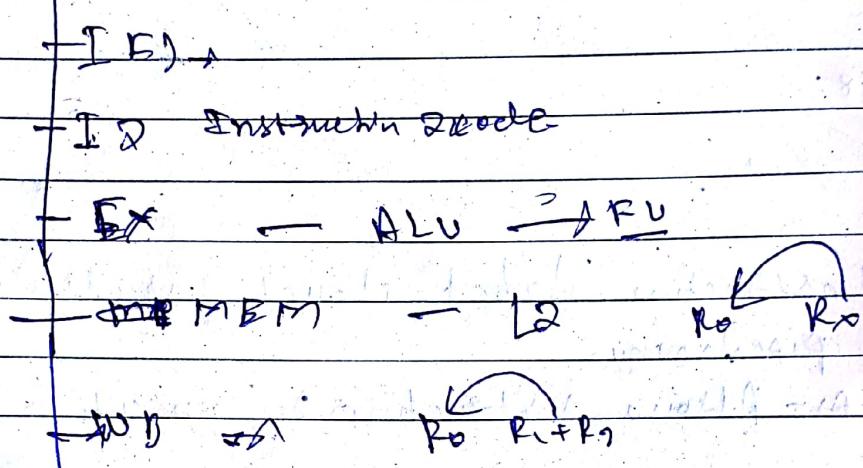
L2 no o(Cpy)

Addressing mode,

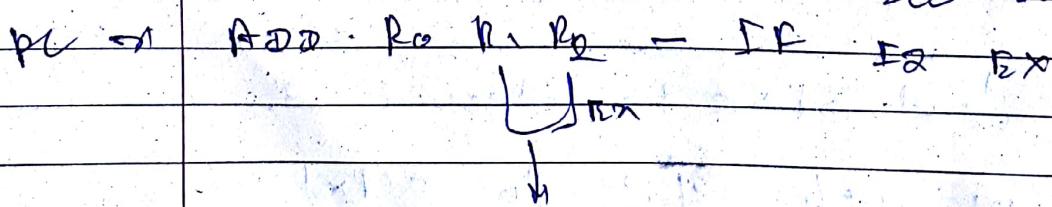
Add R₁ R₂
store R₀ O(R₂)

MIPS → multiprocessor without interleaved pipeline Stage

FIVE Stage



The objective of pipelining is CP \leq 9



ADD R₀ R₁ R₂ → IF ID EX

IF ID EX

PC I₂ R₀ R₁ R₂ R₃ R₄ R₅ R₆ R₇ R₈ R₉ R₁₀ R₁₁ R₁₂ R₁₃ R₁₄ R₁₅ R₁₆ R₁₇ R₁₈ R₁₉ R₂₀ R₂₁ R₂₂ R₂₃ R₂₄ R₂₅ R₂₆ R₂₇ R₂₈ R₂₉ R₃₀ R₃₁ R₃₂ R₃₃ R₃₄ R₃₅ R₃₆ R₃₇ R₃₈ R₃₉ R₄₀ R₄₁ R₄₂ R₄₃ R₄₄ R₄₅ R₄₆ R₄₇ R₄₈ R₄₉ R₅₀ R₅₁ R₅₂ R₅₃ R₅₄ R₅₅ R₅₆ R₅₇ R₅₈ R₅₉ R₆₀ R₆₁ R₆₂ R₆₃ R₆₄ R₆₅ R₆₆ R₆₇ R₆₈ R₆₉ R₇₀ R₇₁ R₇₂ R₇₃ R₇₄ R₇₅ R₇₆ R₇₇ R₇₈ R₇₉ R₈₀ R₈₁ R₈₂ R₈₃ R₈₄ R₈₅ R₈₆ R₈₇ R₈₈ R₈₉ R₉₀ R₉₁ R₉₂ R₉₃ R₉₄ R₉₅ R₉₆ R₉₇ R₉₈ R₉₉ R₁₀₀ R₁₀₁ R₁₀₂ R₁₀₃ R₁₀₄ R₁₀₅ R₁₀₆ R₁₀₇ R₁₀₈ R₁₀₉ R₁₁₀ R₁₁₁ R₁₁₂ R₁₁₃ R₁₁₄ R₁₁₅ R₁₁₆ R₁₁₇ R₁₁₈ R₁₁₉ R₁₂₀ R₁₂₁ R₁₂₂ R₁₂₃ R₁₂₄ R₁₂₅ R₁₂₆ R₁₂₇ R₁₂₈ R₁₂₉ R₁₃₀ R₁₃₁ R₁₃₂ R₁₃₃ R₁₃₄ R₁₃₅ R₁₃₆ R₁₃₇ R₁₃₈ R₁₃₉ R₁₄₀ R₁₄₁ R₁₄₂ R₁₄₃ R₁₄₄ R₁₄₅ R₁₄₆ R₁₄₇ R₁₄₈ R₁₄₉ R₁₅₀ R₁₅₁ R₁₅₂ R₁₅₃ R₁₅₄ R₁₅₅ R₁₅₆ R₁₅₇ R₁₅₈ R₁₅₉ R₁₆₀ R₁₆₁ R₁₆₂ R₁₆₃ R₁₆₄ R₁₆₅ R₁₆₆ R₁₆₇ R₁₆₈ R₁₆₉ R₁₇₀ R₁₇₁ R₁₇₂ R₁₇₃ R₁₇₄ R₁₇₅ R₁₇₆ R₁₇₇ R₁₇₈ R₁₇₉ R₁₈₀ R₁₈₁ R₁₈₂ R₁₈₃ R₁₈₄ R₁₈₅ R₁₈₆ R₁₈₇ R₁₈₈ R₁₈₉ R₁₉₀ R₁₉₁ R₁₉₂ R₁₉₃ R₁₉₄ R₁₉₅ R₁₉₆ R₁₉₇ R₁₉₈ R₁₉₉ R₂₀₀ R₂₀₁ R₂₀₂ R₂₀₃ R₂₀₄ R₂₀₅ R₂₀₆ R₂₀₇ R₂₀₈ R₂₀₉ R₂₁₀ R₂₁₁ R₂₁₂ R₂₁₃ R₂₁₄ R₂₁₅ R₂₁₆ R₂₁₇ R₂₁₈ R₂₁₉ R₂₂₀ R₂₂₁ R₂₂₂ R₂₂₃ R₂₂₄ R₂₂₅ R₂₂₆ R₂₂₇ R₂₂₈ R₂₂₉ R₂₃₀ R₂₃₁ R₂₃₂ R₂₃₃ R₂₃₄ R₂₃₅ R₂₃₆ R₂₃₇ R₂₃₈ R₂₃₉ R₂₄₀ R₂₄₁ R₂₄₂ R₂₄₃ R₂₄₄ R₂₄₅ R₂₄₆ R₂₄₇ R₂₄₈ R₂₄₉ R₂₅₀ R₂₅₁ R₂₅₂ R₂₅₃ R₂₅₄ R₂₅₅ R₂₅₆ R₂₅₇ R₂₅₈ R₂₅₉ R₂₆₀ R₂₆₁ R₂₆₂ R₂₆₃ R₂₆₄ R₂₆₅ R₂₆₆ R₂₆₇ R₂₆₈ R₂₆₉ R₂₇₀ R₂₇₁ R₂₇₂ R₂₇₃ R₂₇₄ R₂₇₅ R₂₇₆ R₂₇₇ R₂₇₈ R₂₇₉ R₂₈₀ R₂₈₁ R₂₈₂ R₂₈₃ R₂₈₄ R₂₈₅ R₂₈₆ R₂₈₇ R₂₈₈ R₂₈₉ R₂₉₀ R₂₉₁ R₂₉₂ R₂₉₃ R₂₉₄ R₂₉₅ R₂₉₆ R₂₉₇ R₂₉₈ R₂₉₉ R₃₀₀ R₃₀₁ R₃₀₂ R₃₀₃ R₃₀₄ R₃₀₅ R₃₀₆ R₃₀₇ R₃₀₈ R₃₀₉ R₃₁₀ R₃₁₁ R₃₁₂ R₃₁₃ R₃₁₄ R₃₁₅ R₃₁₆ R₃₁₇ R₃₁₈ R₃₁₉ R₃₂₀ R₃₂₁ R₃₂₂ R₃₂₃ R₃₂₄ R₃₂₅ R₃₂₆ R₃₂₇ R₃₂₈ R₃₂₉ R₃₃₀ R₃₃₁ R₃₃₂ R₃₃₃ R₃₃₄ R₃₃₅ R₃₃₆ R₃₃₇ R₃₃₈ R₃₃₉ R₃₄₀ R₃₄₁ R₃₄₂ R₃₄₃ R₃₄₄ R₃₄₅ R₃₄₆ R₃₄₇ R₃₄₈ R₃₄₉ R₃₅₀ R₃₅₁ R₃₅₂ R₃₅₃ R₃₅₄ R₃₅₅ R₃₅₆ R₃₅₇ R₃₅₈ R₃₅₉ R₃₆₀ R₃₆₁ R₃₆₂ R₃₆₃ R₃₆₄ R₃₆₅ R₃₆₆ R₃₆₇ R₃₆₈ R₃₆₉ R₃₇₀ R₃₇₁ R₃₇₂ R₃₇₃ R₃₇₄ R₃₇₅ R₃₇₆ R₃₇₇ R₃₇₈ R₃₇₉ R₃₈₀ R₃₈₁ R₃₈₂ R₃₈₃ R₃₈₄ R₃₈₅ R₃₈₆ R₃₈₇ R₃₈₈ R₃₈₉ R₃₉₀ R₃₉₁ R₃₉₂ R₃₉₃ R₃₉₄ R₃₉₅ R₃₉₆ R₃₉₇ R₃₉₈ R₃₉₉ R₄₀₀ R₄₀₁ R₄₀₂ R₄₀₃ R₄₀₄ R₄₀₅ R₄₀₆ R₄₀₇ R₄₀₈ R₄₀₉ R₄₁₀ R₄₁₁ R₄₁₂ R₄₁₃ R₄₁₄ R₄₁₅ R₄₁₆ R₄₁₇ R₄₁₈ R₄₁₉ R₄₂₀ R₄₂₁ R₄₂₂ R₄₂₃ R₄₂₄ R₄₂₅ R₄₂₆ R₄₂₇ R₄₂₈ R₄₂₉ R₄₃₀ R₄₃₁ R₄₃₂ R₄₃₃ R₄₃₄ R₄₃₅ R₄₃₆ R₄₃₇ R₄₃₈ R₄₃₉ R₄₄₀ R₄₄₁ R₄₄₂ R₄₄₃ R₄₄₄ R₄₄₅ R₄₄₆ R₄₄₇ R₄₄₈ R₄₄₉ R₄₅₀ R₄₅₁ R₄₅₂ R₄₅₃ R₄₅₄ R₄₅₅ R₄₅₆ R₄₅₇ R₄₅₈ R₄₅₉ R₄₆₀ R₄₆₁ R₄₆₂ R₄₆₃ R₄₆₄ R₄₆₅ R₄₆₆ R₄₆₇ R₄₆₈ R₄₆₉ R₄₇₀ R₄₇₁ R₄₇₂ R₄₇₃ R₄₇₄ R₄₇₅ R₄₇₆ R₄₇₇ R₄₇₈ R₄₇₉ R₄₈₀ R₄₈₁ R₄₈₂ R₄₈₃ R₄₈₄ R₄₈₅ R₄₈₆ R₄₈₇ R₄₈₈ R₄₈₉ R₄₉₀ R₄₉₁ R₄₉₂ R₄₉₃ R₄₉₄ R₄₉₅ R₄₉₆ R₄₉₇ R₄₉₈ R₄₉₉ R₅₀₀ R₅₀₁ R₅₀₂ R₅₀₃ R₅₀₄ R₅₀₅ R₅₀₆ R₅₀₇ R₅₀₈ R₅₀₉ R₅₁₀ R₅₁₁ R₅₁₂ R₅₁₃ R₅₁₄ R₅₁₅ R₅₁₆ R₅₁₇ R₅₁₈ R₅₁₉ R₅₂₀ R₅₂₁ R₅₂₂ R₅₂₃ R₅₂₄ R₅₂₅ R₅₂₆ R₅₂₇ R₅₂₈ R₅₂₉ R₅₃₀ R₅₃₁ R₅₃₂ R₅₃₃ R₅₃₄ R₅₃₅ R₅₃₆ R₅₃₇ R₅₃₈ R₅₃₉ R₅₄₀ R₅₄₁ R₅₄₂ R₅₄₃ R₅₄₄ R₅₄₅ R₅₄₆ R₅₄₇ R₅₄₈ R₅₄₉ R₅₅₀ R₅₅₁ R₅₅₂ R₅₅₃ R₅₅₄ R₅₅₅ R₅₅₆ R₅₅₇ R₅₅₈ R₅₅₉ R₅₆₀ R₅₆₁ R₅₆₂ R₅₆₃ R₅₆₄ R₅₆₅ R₅₆₆ R₅₆₇ R₅₆₈ R₅₆₉ R₅₇₀ R₅₇₁ R₅₇₂ R₅₇₃ R₅₇₄ R₅₇₅ R₅₇₆ R₅₇₇ R₅₇₈ R₅₇₉ R₅₈₀ R₅₈₁ R₅₈₂ R₅₈₃ R₅₈₄ R₅₈₅ R₅₈₆ R₅₈₇ R₅₈₈ R₅₈₉ R₅₉₀ R₅₉₁ R₅₉₂ R₅₉₃ R₅₉₄ R₅₉₅ R₅₉₆ R₅₉₇ R₅₉₈ R₅₉₉ R₆₀₀ R₆₀₁ R₆₀₂ R₆₀₃ R₆₀₄ R₆₀₅ R₆₀₆ R₆₀₇ R₆₀₈ R₆₀₉ R₆₁₀ R₆₁₁ R₆₁₂ R₆₁₃ R₆₁₄ R₆₁₅ R₆₁₆ R₆₁₇ R₆₁₈ R₆₁₉ R₆₂₀ R₆₂₁ R₆₂₂ R₆₂₃ R₆₂₄ R₆₂₅ R₆₂₆ R₆₂₇ R₆₂₈ R₆₂₉ R₆₃₀ R₆₃₁ R₆₃₂ R₆₃₃ R₆₃₄ R₆₃₅ R₆₃₆ R₆₃₇ R₆₃₈ R₆₃₉ R₆₄₀ R₆₄₁ R₆₄₂ R₆₄₃ R₆₄₄ R₆₄₅ R₆₄₆ R₆₄₇ R₆₄₈ R₆₄₉ R₆₅₀ R₆₅₁ R₆₅₂ R₆₅₃ R₆₅₄ R₆₅₅ R₆₅₆ R₆₅₇ R₆₅₈ R₆₅₉ R₆₆₀ R₆₆₁ R₆₆₂ R₆₆₃ R₆₆₄ R₆₆₅ R₆₆₆ R₆₆₇ R₆₆₈ R₆₆₉ R₆₇₀ R₆₇₁ R₆₇₂ R₆₇₃ R₆₇₄ R₆₇₅ R₆₇₆ R₆₇₇ R₆₇₈ R₆₇₉ R₆₈₀ R₆₈₁ R₆₈₂ R₆₈₃ R₆₈₄ R₆₈₅ R₆₈₆ R₆₈₇ R₆₈₈ R₆₈₉ R₆₉₀ R₆₉₁ R₆₉₂ R₆₉₃ R₆₉₄ R₆₉₅ R₆₉₆ R₆₉₇ R₆₉₈ R₆₉₉ R₇₀₀ R₇₀₁ R₇₀₂ R₇₀₃ R₇₀₄ R₇₀₅ R₇₀₆ R₇₀₇ R₇₀₈ R₇₀₉ R₇₁₀ R₇₁₁ R₇₁₂ R₇₁₃ R₇₁₄ R₇₁₅ R₇₁₆ R₇₁₇ R₇₁₈ R₇₁₉ R₇₂₀ R₇₂₁ R₇₂₂ R₇₂₃ R₇₂₄ R₇₂₅ R₇₂₆ R₇₂₇ R₇₂₈ R₇₂₉ R₇₃₀ R₇₃₁ R₇₃₂ R₇₃₃ R₇₃₄ R₇₃₅ R₇₃₆ R₇₃₇ R₇₃₈ R₇₃₉ R₇₄₀ R₇₄₁ R₇₄₂ R₇₄₃ R₇₄₄ R₇₄₅ R₇₄₆ R₇₄₇ R₇₄₈ R₇₄₉ R₇₅₀ R₇₅₁ R₇₅₂ R₇₅₃ R₇₅₄ R₇₅₅ R₇₅₆ R₇₅₇ R₇₅₈ R₇₅₉ R₇₆₀ R₇₆₁ R₇₆₂ R₇₆₃ R₇₆₄ R₇₆₅ R₇₆₆ R₇₆₇ R₇₆₈ R₇₆₉ R₇₇₀ R₇₇₁ R₇₇₂ R₇₇₃ R₇₇₄ R₇₇₅ R₇₇₆ R₇₇₇ R₇₇₈ R₇₇₉ R₇₈₀ R₇₈₁ R₇₈₂ R₇₈₃ R₇₈₄ R₇₈₅ R₇₈₆ R₇₈₇ R₇₈₈ R₇₈₉ R₇₉₀ R₇₉₁ R₇₉₂ R₇₉₃ R₇₉₄ R₇₉₅ R₇₉₆ R₇₉₇ R₇₉₈ R₇₉₉ R₈₀₀ R₈₀₁ R₈₀₂ R₈₀₃ R₈₀₄ R₈₀₅ R₈₀₆ R₈₀₇ R₈₀₈ R₈₀₉ R₈₁₀ R₈₁₁ R₈₁₂ R₈₁₃ R₈₁₄ R₈₁₅ R₈₁₆ R₈₁₇ R₈₁₈ R₈₁₉ R₈₂₀ R₈₂₁ R₈₂₂ R₈₂₃ R₈₂₄ R₈₂₅ R₈₂₆ R₈₂₇ R₈₂₈ R₈₂₉ R₈₃₀ R₈₃₁ R₈₃₂ R₈₃₃ R₈₃₄ R₈₃₅ R₈₃₆ R₈₃₇ R₈₃₈ R₈₃₉ R₈₄₀ R₈₄₁ R₈₄₂ R₈₄₃ R₈₄₄ R₈₄₅ R₈₄₆ R₈₄₇ R₈₄₈ R₈₄₉ R₈₅₀ R₈₅₁ R₈₅₂ R₈₅₃ R₈₅₄ R₈₅₅ R₈₅₆ R₈₅₇ R₈₅₈ R₈₅₉ R₈₆₀ R₈₆₁ R₈₆₂ R₈₆₃ R₈₆₄ R₈₆₅ R₈₆₆ R₈₆₇ R₈₆₈ R₈₆₉ R₈₇₀ R₈₇₁ R₈₇₂ R₈₇₃ R₈₇₄ R₈₇₅ R₈₇₆ R₈₇₇ R₈₇₈ R₈₇₉ R₈₈₀ R₈₈₁ R₈₈₂ R₈₈₃ R₈₈₄ R₈₈₅ R₈₈₆ R₈₈₇ R₈₈₈ R₈₈₉ R₈₉₀ R₈₉₁ R₈₉₂ R₈₉₃ R₈₉₄ R₈₉₅ R₈₉₆ R₈₉₇ R₈₉₈ R₈₉₉ R₉₀₀ R₉₀₁ R₉₀₂ R₉₀₃ R₉₀₄ R₉₀₅ R₉₀₆ R₉₀₇ R₉₀₈ R₉₀₉ R₉₁₀ R₉₁₁ R₉₁₂ R₉₁₃ R₉₁₄ R₉₁₅ R₉₁₆ R₉₁₇ R₉₁₈ R₉₁₉ R₉₂₀ R₉₂₁ R₉₂₂ R₉₂₃ R₉₂₄ R₉₂₅ R₉₂₆ R₉₂₇ R₉₂₈ R₉₂₉ R₉₃₀ R₉₃₁ R₉₃₂ R₉₃₃ R₉₃₄ R₉₃₅ R₉₃₆ R₉₃₇ R₉₃₈ R₉₃₉ R₉₄₀ R₉₄₁ R₉₄₂ R₉₄₃ R₉₄₄ R₉₄₅ R₉₄₆ R₉₄₇ R₉₄₈ R₉₄₉ R₉₅₀ R₉₅₁ R₉₅₂ R₉₅₃ R₉₅₄ R₉₅₅ R₉₅₆ R₉₅₇ R₉₅₈ R₉₅₉ R₉₆₀ R₉₆₁ R₉₆₂ R₉₆₃ R₉₆₄ R₉₆₅ R₉₆₆ R₉₆₇ R₉₆₈ R₉₆₉ R₉₇₀ R₉₇₁ R₉₇₂ R₉₇₃ R₉₇₄ R₉₇₅ R₉₇₆ R₉₇₇ R₉₇₈ R₉₇₉ R₉₈₀ R₉₈₁ R₉₈₂ R₉₈₃ R₉₈₄ R₉₈₅ R₉₈₆ R₉₈₇ R₉₈₈ R₉₈₉ R₉₉₀ R₉₉₁ R₉₉₂ R₉₉₃ R₉₉₄ R₉₉₅ R₉₉₆ R₉₉₇ R₉₉₈ R₉₉₉ R₁₀₀₀ R₁₀₀₁ R₁₀₀₂ R₁₀₀₃ R₁₀₀₄ R₁₀₀₅ R₁₀₀₆ R₁₀₀₇ R<

L2

20,

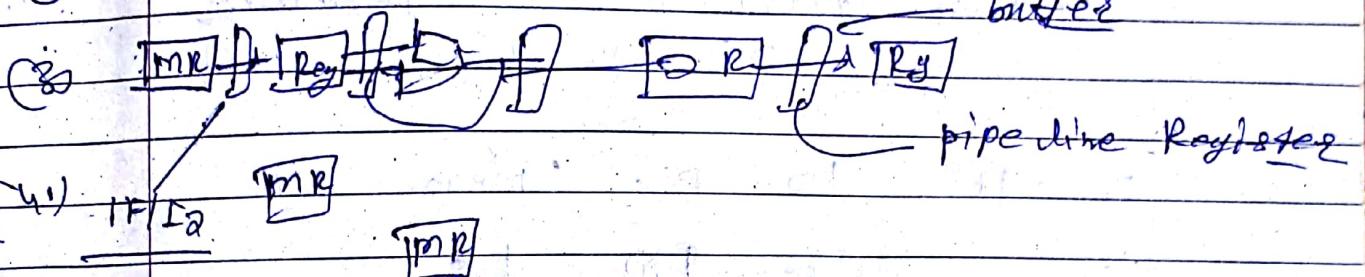
BMBZ Po R. Jabsay

DC - 2IC ~~dt~~

Rolen I2

(1) 1CC 2CC 3CC 4CC 5CC

(2) SF ID BX PEM WY



Speedup

Pipeline time Execution time un-pipelined

Ex time Pipeline -

$$= \frac{IC_{unpipelined}}{IC_{pipeline}} \times \frac{CPD_{unpipe}}{CPD_{pipeline}} \times \frac{CCT_{unpipe}}{CCT_{pipeline}}$$

$$= \frac{CPD_{unpipeline}}{CPD_{pipeline}} \approx \frac{CCT_{unpipeline}}{CCT_{pipeline}}$$

objective

$$CPD_{pipeline} \geq 1$$

$$= \frac{1}{1 + \frac{1}{D}}$$

 $D = \text{pipeline depth}$ Mazard \rightarrow if

$$CPD_{pipeline} > 1$$

(with hazard)

Hazards →

- (1) Structural hazard
- (2) data hazard → require Data forwarding
- (3) Control hazard

Speedup pipelining with hazard = pipeline depth

latency due to
hazard

Structure hazard: → lack of resource
or same register

I ₁	-JR	I ₂	Bx	mem	WB
I ₂		IR	EX	Bx	

I₃

I₄

src → ucc

with ~~for~~
SN

Data hazard →

ADD R₁ R₂ R₃ FF S2 R₄ mem WB

→ SUB R₁ R₂ R₃

IR ~~S2~~ ~~S2~~ ~~WB~~ FF

AND R₆ R₁ R₇

~~IR~~

OR R₈ R₁ R₉

XOR R₁₀ R₁ R₁₁

Hot word Data Forwarding

Page No.	_____
Date	-

LD R₀ o(R_n) IF I₂ BX Mem $\xrightarrow{R_0}$

ADD R₂ R₀ R_y

$\xrightarrow{I_1}$ IF I₂ $\xrightarrow{R_0}$ mem

Control hazard is due to flushing

I₁ if ($n = 0$) then $\xrightarrow{I_2}$ BX mem WB
 $\xrightarrow{I_2} n = n + q$ IF I₂ BX mem WB flush
 else $n = n - q$ IF I₂ BX mem sw

(a) Flushing

(b) P.V.T Predict untaken ~~→~~ used flushing,

(c) Predict taken

(d) Delayed Branch

if ($n = 0$) then $n = n + q$

else $n = n - q$

I₄ $y = y + 10$

I₁ IF I₂

I₄ IF

I₂/I₃

I₃

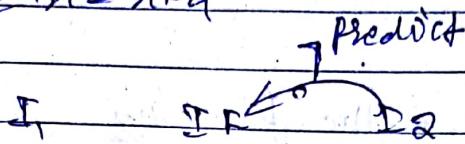
IF

Instruction Scheduling:

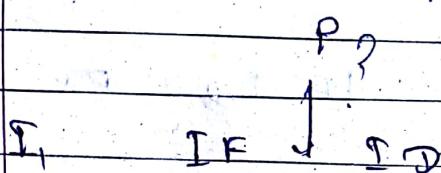
$t_1 \quad \text{if } (x=20) \text{ then}$

$t_2 \quad \quad \quad x = x + 9$

$t_3 \quad \text{else } x = x - 9$



Dynamic Branch prediction Techniques:

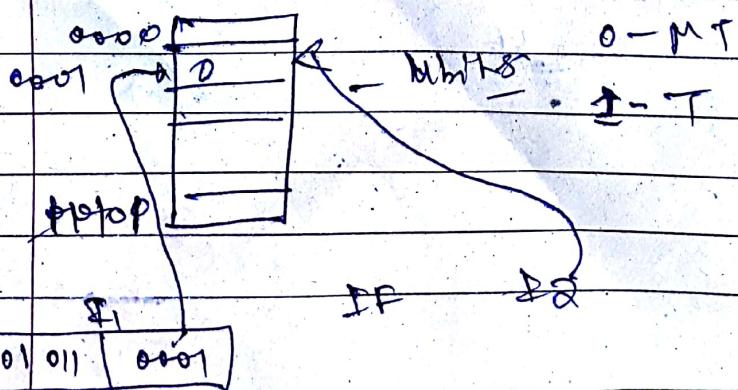


t_2/t_3

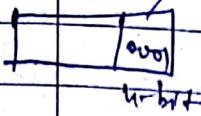
- (1) 1-bit predictor
- (2) 2-bit predictor
- (3) Correlating predictor
- (4) Tournament Branch Predictor
- ~~(5) Branch Target Buffer~~
- ~~(6) Return Address Stack~~

Local / global predictor
tag share / a select

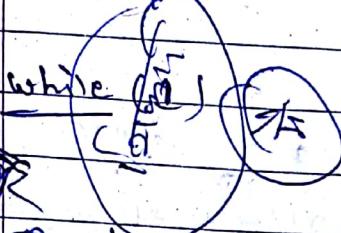
BHT Branch history table



IP →



Memory addressing → two instruction are pointed
same address BN7



B1 (000)

JMP (120), ADD (110), LDR (100),

0001 0

(1000) 0

→ BR (NT), P NT T T T T

JMP 520, ADD 520, LDR 520, T T T T T T

T 82 B2 (1,000) 6 7 0 9 10
= 3 MT 1 1 1 1 MT

8

$$\frac{P}{T_0} = \frac{12}{20} = \frac{26}{80} = 0.75$$

PVT →

T10 + T20

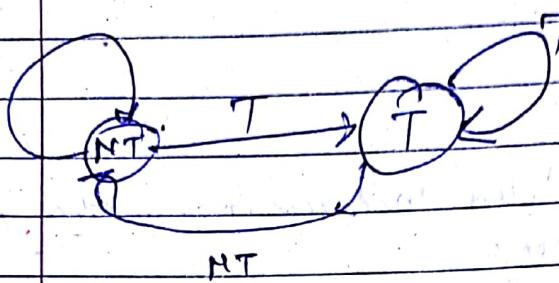
5

22/80 = 0.75

67 67

Product of Jycles

~~2-bit predictor~~ →



2-bit Predictor →

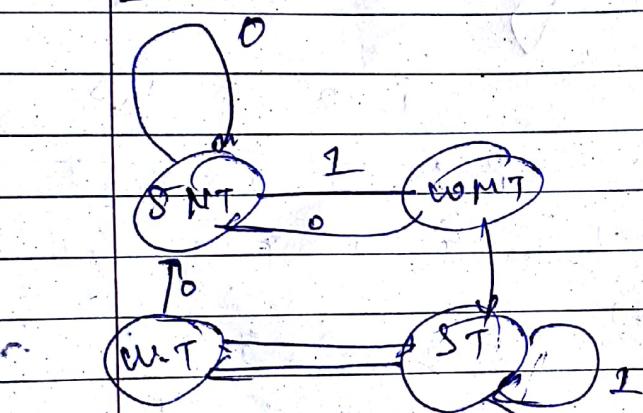
11
10
00
01

00-store

01/¹⁰ weekly

10 → 8 songs

Jaun



$$\frac{9}{10} \times \frac{19}{20} \rightarrow \frac{20}{3} = 93\%$$

NT → NT T NT T → 1-bit poor
(4 2-bit buffer)

NT if ($x \geq 2$)

if ($y \geq 2$)

if ($y \geq 2$)
if ($y \geq 2$)

T If $(x_1 = y)$: ← (obscuring predictor
start)

else

start

(min) predictor

in branches previous

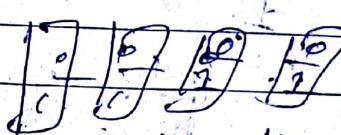
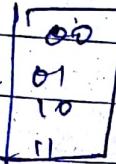
$n \rightarrow n$ -bit Inside the buffer

2^m buffer

each buffer maintains n bits

(2, 2) predictors

(2, 2) 2⁴



00 01 10 11

B₁: BME2

B₂: BM^E2

3

B₁ → T, NT, T, NT, T

87-93-95

B₂: T T T T NT

B₁

Headline Predictor | Predictor outcome
Headline for Run outcome # Predictor Predictor outcome

00 0101010

T

T

11 1101010

NT

T

01

F

T

11

NT

F

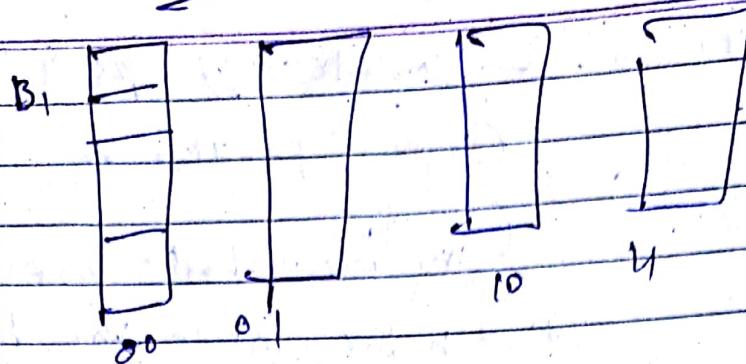
01

T

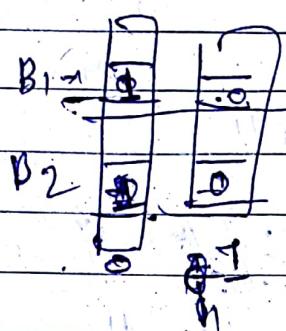
NT

1.2) Summ

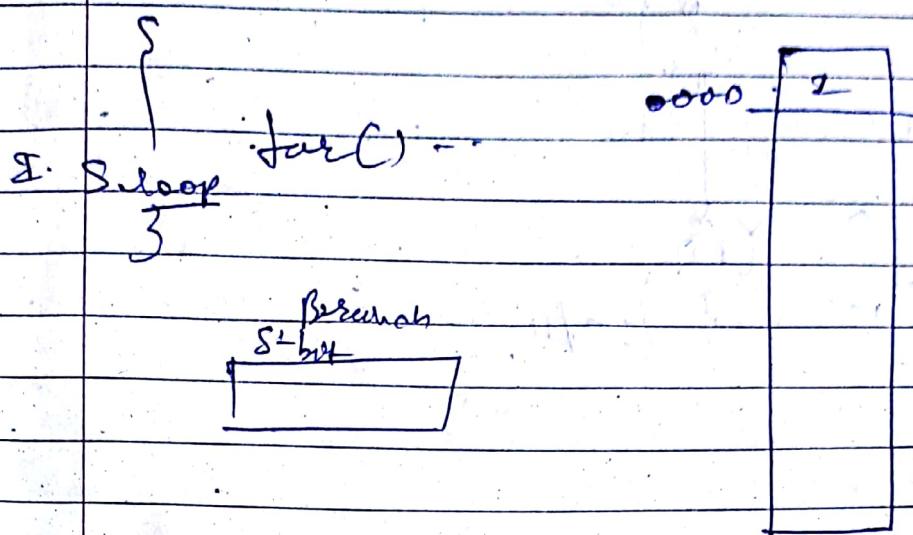
Date _____



Same ... (1,2)

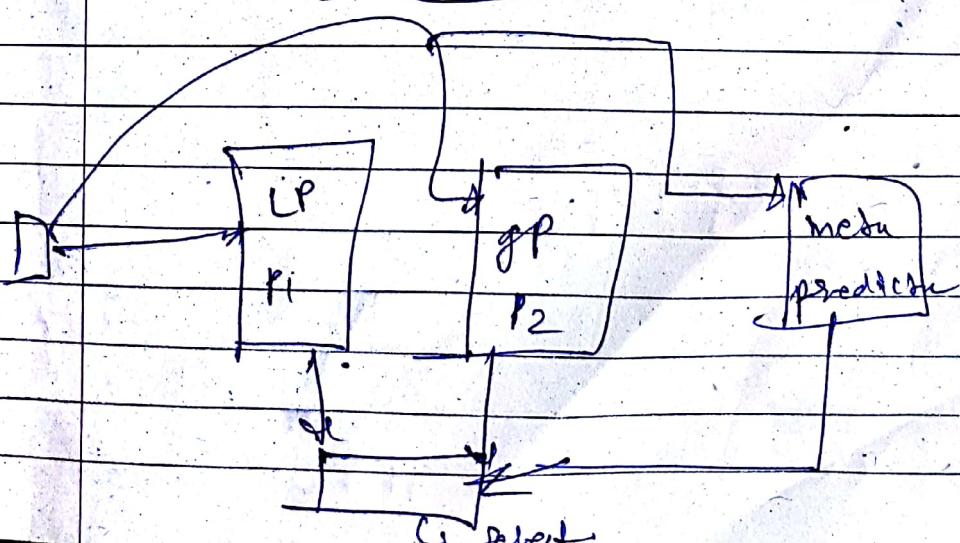
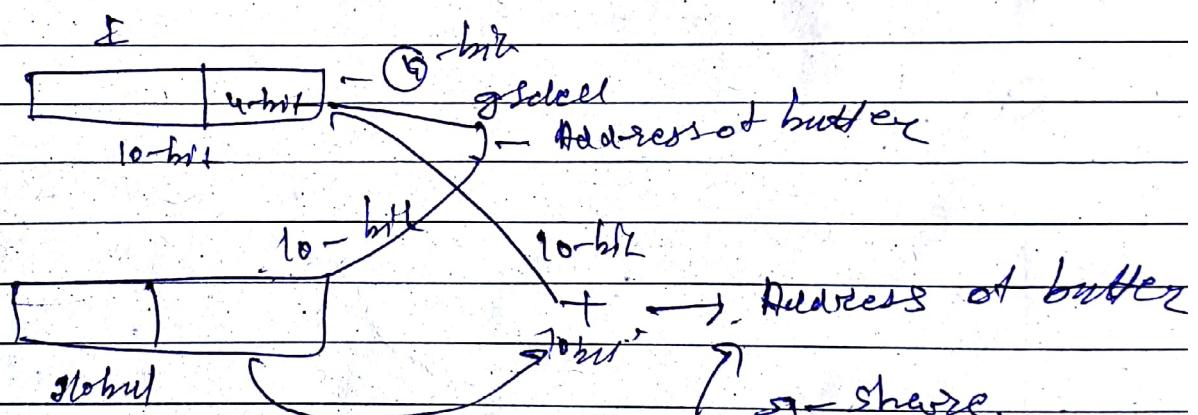


Local predictor: → depends ~~on~~ on own branch



Global predictor: → depends on other branch

- ① Select
- ② Share

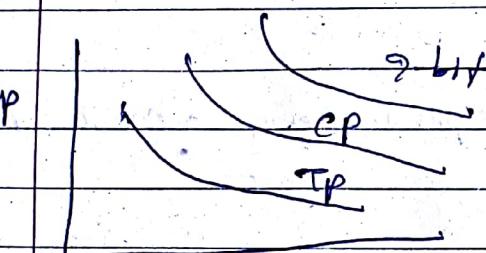
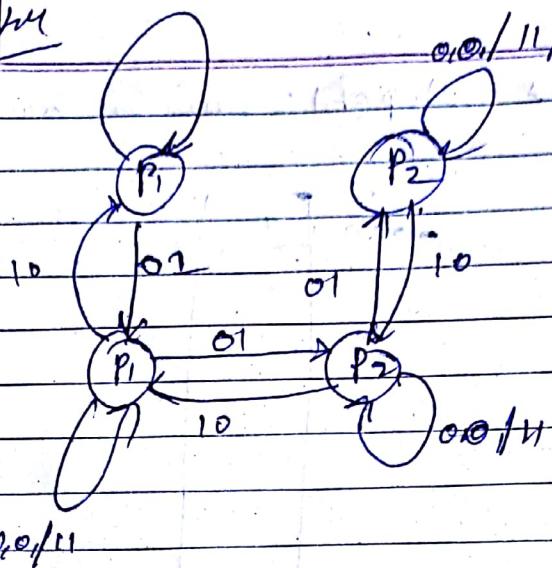


Tournament
predicates

00, 1, 2, 10

Page No. _____

Date _____



Admissible IP

as