

What is branch hazard? What are its solutions?

→  
 ↓  
 Control hazard  
 or  
 Instruction hazard

Solution

- 1) Pipeline flushing
- 2) delayed branching
- 3) early branching.

Max limit of Amdahl's law

$$L \propto \frac{1}{f}$$

$f_0$  is for serial part.

Gustafson's law

This law states that any sufficiently large problem can be efficiently parallelised; thus the sequential operation will no longer be a bottleneck. It is based on the fixed execution time.

$$S + P = 1$$

↑ serial      ↓ Parallel

For non-parallel (single processor machine)

$$ET = (S + nP)$$

$$\text{Speed up} = \frac{S + nP}{S'(n)}$$

$$= \frac{S + nP}{1}$$

$$S'(n) = S + nP$$

$$S + P = 1$$

$$\Rightarrow P = (1 - S)$$

$$S'(n) = S + nP$$

$$= \cancel{n} + (1 - n)S$$

$$= S + n(1 - S)$$

$$= S + n - nS$$

$$S'(n) = n + (1 - n)S$$

[ Putting  $P = (1 - S)$  ]

Q. det we have 5% serial section 20 PEs (Processing Elements). What will be the speed up according to Amdahl's law and Gustafson's law.

→  $S = 0.05$  ,  $P = \cancel{0.95} 0.95$  ;  $n = 20$

Amdahl's law

$$N = \frac{0.05 + \frac{0.95}{20}}$$

$$= 10.206 \text{ times}$$

Gustafson's

$$S'(n) = 20 + (1-20) \times 0.05$$

$$= 19.05 \text{ times}$$

Gustafson's is better than Amdahl's.

With a careful design it is possible to have a speed-up of 1021, 1020, 1016 in practice with 1024 PEs.

Sun & Ni's Law

$M$  = Memory requirement of the problem

$W$  = Computational work load (CPU power)

$M$  and  $W$  are related to each other

$$W = g(M)$$

$$M = g^{-1}(W)$$

Let, the execution time be  $T$ , which will be a will be again the function of no. of processor  $\phi(n)$ , work load ( $W$ ) and memory ( $M$ )

$$T = f(M)$$

$$\text{Enhanced time} = T \times$$



$$\cancel{T^* = f^* \circ (nM)} \quad T^* = f^* (nM) \quad \boxed{\text{for}}$$

where,  $nM$  is the increased memory capacity of  $n$ -processor machine.

$$f^*(nM) = f(n) \cdot f(M)$$

$$r(n, w) = F(M)$$

$$z \neq f(n) \cdot T(n, w)$$

$$S^*(n) = \frac{T_S + F(n) T_P(n, w)}{T_S + \frac{F(n)}{n} (T_P(n, w))}$$

$$= \frac{\alpha + F(n) (1-\alpha)}{\alpha + \frac{F(n)}{n} (1-\alpha)}$$

Where,  $T = T_s + T_p$  = Total execution time

$T_s$  = Execution time of serial part

$T_p = \dots \dots \dots$  Parallel  $n$

$\alpha$  = Serial # part/fraction of the program

$(1-\alpha) = \text{Parallel}$   $n / n$  of  $n$   $n$

Instructions = 1000

$t_m = 5 \text{ ms}$

$t_c = 100 \text{ ns} = 0.1 \text{ ms}$

Total time =  $t_c + t_m$

10 ins.  $\rightarrow 8$

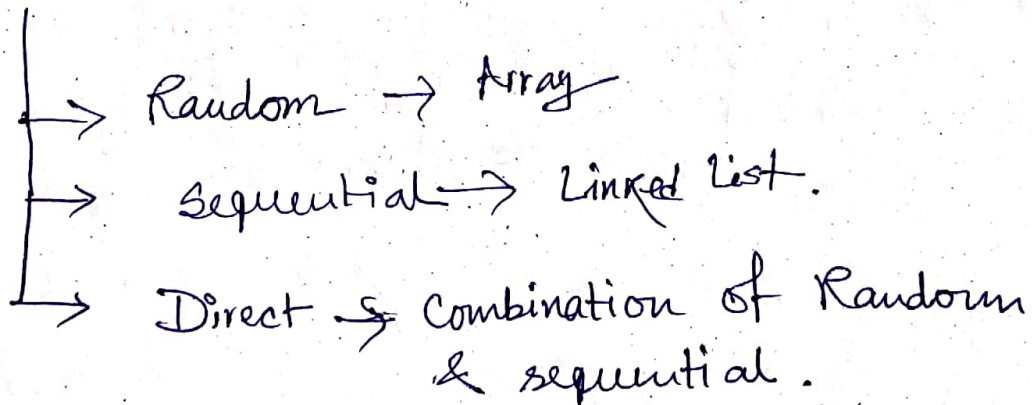
1000 n  $\rightarrow 800$

~~800~~

$$\begin{aligned}\text{Total time} &= 800 \times t_c + 200 (t_c + t_m) \\ &= 800 \times 0.1 + 200 \times 5.1 \\ &= \cancel{4000} + 1100 \text{ ms.}\end{aligned}$$

Q. Diff<sup>n</sup> SRAM vs DRAM

Memory Accessing method



Q. There are four types of accessing method -

① Random

In Random accessing method any location of the memory can be accessed randomly.

as we are calling ~~the~~ array location by using index.

It is not related to any other physical location.

② Sequential

Here the memory