What is branch hazard? What are it's solutions?

$\Downarrow$

→ Control hazard
or
Instruction hazard

Solution

1> Pipeline flashing

2> delayed branching

3> early branching.

Max limit of Amdahl's law

$\lim_{n\to\infty} \frac{1}{f}$  'f' 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.

$s+P = 1$

$\underset{\text{serial}}{s} + \underset{\text{Parallel}}{P}$

For non-parallel (single processor machine)

$E.T = (s+nP)$

$$\text{Speed up} = \frac{s+nP}{s+P}$$

$S'(n)$

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

$s'(n) = s+nP$.

$s+P = 1$

$\Rightarrow P = (1-s)$

$s'(n) = s+nP$

$= \cancel{s + n + (1-n)s}$   [Putting $P = (1-s)$]

$= s + n(1-s)$

$= s + n - ns$

$$\boxed{s'(n) = n + (1-n)s}$$

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

→ $S = 0.05$ , $P =$ ~~0.095~~ $0.95$ ; $n = 20$ .

Amdahl's law

$$N = \frac{1}{0.05 + \frac{0.95}{20}}$$

$= 10.286$ times

Gustafson's

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

$= 19.05$ 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~~ ~~is a~~ will be again the function of no. of processor (n), work load (W) and memory (M)

$$T = f(M)$$

Enhanced time $= T^*$

~~$t^* = f^* (nM)$~~ $T^* = f^*(nM)$ ~~for~~

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

$$f^*(nM)$$
$$= F(n) . F(M)$$

$$T(n,W) = F(M)$$

$$f^*(nM)$$
$$= f(n) . T(n.W)$$

$$S^*(n) = \frac{T_s + F(n)\, T_p(n,W)}{T_s + \frac{F(n)}{n}\left(T_p(n,W)\right)}$$

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

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

$T_s$ = Execution time of serial part

$T_p$ = " " " Parallel "

$\alpha$ = Serial ~~A~~ part/fraction of the program

$(1-\alpha)$ = Parallel " / " of " "

Instructions = 1000

$t_m$ = 5 ms

$t_c$ = 100 ns = 0.1 ms

Total time = $t_c + t_m$

10 ins. → 8

1000 " → 800

~~800~~

Total time = $800 \times t_c + 200\,(t_c + t_m)$

$= 800 \times \overset{0.1}{\cancel{5}} + 200 \times 5.1$

= ~~4000 +~~ 1100 ms.

Q. Diffⁿ SRAM VS DRAM

Memory Acessing method

- → Random → Array
- → Sequential → Linked List.
- → Direct → Combination of Random & sequential.

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 array location by using index.

It is not related to any Other physical location.

② Sequential

Here the memory