Subject = COA
type = Hanotes
topic = Memory

Average access time

Hierarical

$$T_{avg} = H_1T_1 + (1-H_1)H_2(T_2+T_1) + (1-H_2)(1-H_1)H_3(T_1+T_2+T_3)\ldots$$

$$= \sum_{j=0}^{n}\left[\left[\prod_{i=0}^{j-1}(1-H_i)\right]H_j\left[\sum_{k=0}^{j}T_k\right]\right]$$

Simutaneous acces

$$T_{avg} = H_1T_1 + (1-H_1)H_2T_2 + (1-H_1)(1-H_2)H_3T_3 \ldots$$

$$= \sum_{j=0}^{n}\left[\prod_{i=0}^{j-1}(1-H_i)\right]H_jT_j$$

Global miss Rate = $\dfrac{\text{Number of Miss Operation in CM}}{\text{total number of Memory References generated by CPU}}$

local miss Rate = $\dfrac{\text{Number of Miss Operation in CM.}}{\text{Total NO. of Access to the CM}}$

throughput of memory = $\dfrac{1}{T_{avg}}$ WPS [words per second]

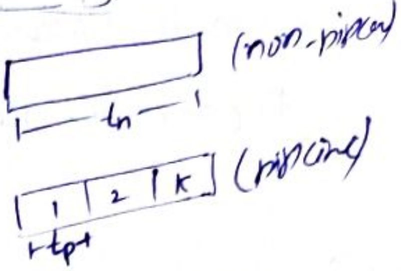
Floating Point IEEE

S	E	M	Value
0/1	E=0	M=0	±0
0/1	E=255	M=0	±∞
0/1	1≤E≤254	M=xxx	$(-1)^S \times 1.M \times 2^{E-127}$ (Implicit normalized)
0/1	E=0	M≠0	$(-1)^S \times 0.M \times 2^{-126}$ (Fractional form)
0/1	E=255	M≠0	NAN

COA

#Pibelining

Speed-up (s) = $\frac{\text{non-pipeline time}}{\text{bibeline time}} = \boxed{\frac{n * t_n}{(k+n-1)\,t_p}}$



when $n >> k$ then

(ideal) Speed up(s) = $\boxed{\frac{t_n}{t_p}}$

when time required to perform 1 task is equal in pipline & non-pipline

$t_n = k * t_p$

(ideal) $\boxed{S = k}$ $\boxed{\eta = \frac{S}{k}}$

→ in general $\left(S = \frac{n * t_n}{(k+n-1)\,t_p} \right)$

→ if max or ideal speed asked
→ if value of n is not given $\left(S = \frac{t_n}{t_p} \right)$

throughput

general

$$\frac{n}{(k+n-1) * t_p}$$

ideal

$$\frac{1}{t_p}$$

$\boxed{t/p = 1/\text{cycle time}}$

frequency = $\frac{1}{\text{cycle time } (T_p)}$

Total no. of cycles = $(k+n-1)$ + total stalls cycle

If branch condition is evaluated in i^{th} phase then no. of stalls cycles because of branch instruction = $i-1$

t_p

Total time = total no. of cycle * cycle time

Pipline Hazerds

① Structural Hazard / Resource conflict

② Data Hazard / Data Dependency

③ control Hazard / Branch Difficulty.

① structural hazard / Resource conflict

⇒ 2 different segments try to use same resource at same time.

Solⁿ ① Increase stall cycles

② Increase number of resources

② Data Hazard / Data Dependency

⇒ Result of an instruction is used as input in next

Solⁿ: ① provided by compiler (Delay Load) (Software)

e.g. R1 ← R2 + R3 ⟹ R1 ← R1 + R3
R5 ← R1 + R4

≡ ← Independent or No. of operation

R5 ← R1 + R4

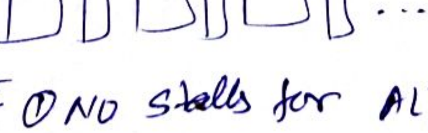
② (Hardware)

(i) Hardware interlock

e.g. IF DA OF Ex WB

IF DA - - OF Ex WB

(ii) Operand forwarding / Bypassing

e.g. 

Note ① No stalls for ALU to ALU dependancy

② stalls for ALU to memory

(or) Mem to ALU

Data Hazards Classification

[RAW, WAW, WAR]

① RAW (Read After write) (true data dependency)

If j reads a source before it is written by i, hence j incorrectly gets old value.

i : R1 ← R2 + R3
j : R5 ← R1 + R4

No. of stall = phase no. of WB − phase no. of OF

WB :- write Back
OF :- Operand fetch.

Soln

① (S/w) Delay load

② (H/w)

(i) H/w interlock

(ii) Operand forwarding

2 stall cycle for ~~both~~ depend instruction, if these are immediately depends

② WAW (Write after write) (~~Anti~~ output data dependency)

'j' writes in a destination before 'i' writes it.

i : R1 ← R2 + R3
j : R1 ← R4 * R5

Soln ① Register Renaming

eg i : R1 ← R2 + R3
R8 ← R4 * R5
← ~~R~~ different Register)

③ WAR (Write after Read) (~~Output~~ Anti data dependency)

'j' writes a destination before 'i' reads it ; hence i incorrectly reads new value.

eg i : R1 ← R2 + R3
j : R2 ← R5 + R7

Soln ① Register Renaming

Pipline efficiency

In ideal case

~~$\eta = \frac{t_n}{t_p}$ (S')~~

~~(100%) when Hazard = S~~

$$\eta = \frac{S}{S'}$$ ← when Hazards

↑ no Hazards

Control Hazards / Branch Difficulty

A general pipline can't detect the branch problem

Solⁿ ① Delayed Branch (S/w)
② Branch prediction (H/w)

cycles per Instruction

$$CPI = \frac{\text{Total cycles}}{\text{no of Instructions}}$$

$$\boxed{CPI = \frac{k + n - 1}{n}}$$

In ideal conditions (k-1 ignored)

$$\boxed{CPI = 1}$$

with hazard

Assume because of hazard total extra stall cycle = x

$$\boxed{CPI = \frac{k+n-1+x}{n}}$$

In ideal condition

$$\boxed{CPI = \frac{n+x}{n}}$$

(or)

$$\boxed{CPI = 1 + \frac{x}{n}}$$

$$\boxed{S_{ideal} = \frac{t_n}{CPI_{avg} \times t_p}}$$

~~time~~

one instruction time execution time = CPI_{avg} ·

execution time of Pipline $= \boxed{CPI \times \max(\ldots stage)}$