

1.

二. 若 BTB 在 ID-stage 階段才能查的話, 則 IF-stage 依舊是會讓 PC+4, 失去預測的效果

(b)

第三種情況：此 pc-address 為 branch 指令且先前做過，但其 index 與後面的 branch 指令衝突，而被擠掉，故 miss

(c)

if (br == 0)

$$PC_{src} = D / 5$$

(d)

PC	22 bit	10 bit
----	--------	--------

$$\Rightarrow \text{size} = 1024 \times (3 + 22 + 32) \text{ bits}$$

(e)

當 branch 猜錯時, 需往前清空 IF/ID 內之值故需設  $IF.Flush = 1$  \*

(請翻面繼續作答)

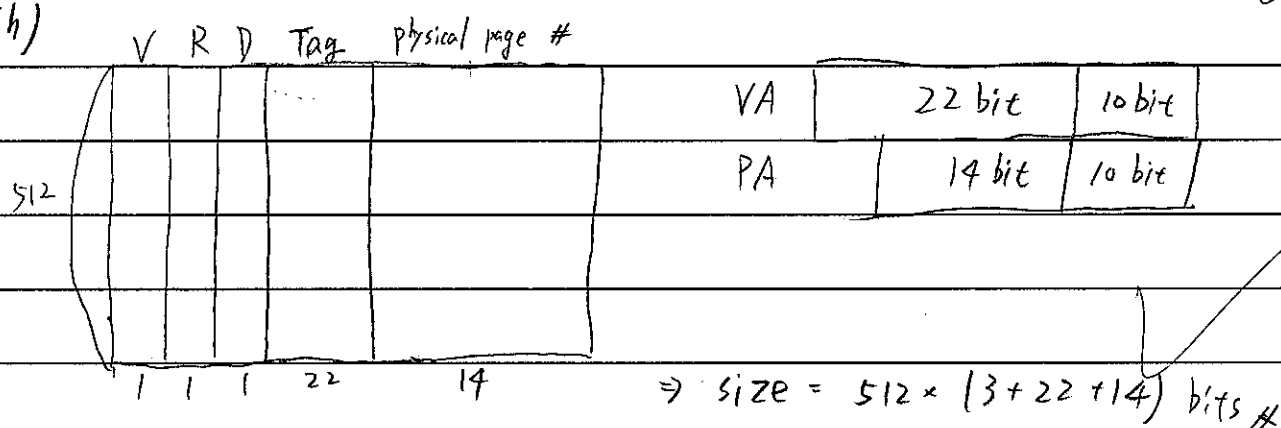
(f)

將 virtual address 轉成 physical address 以使用 data cache 存取資料

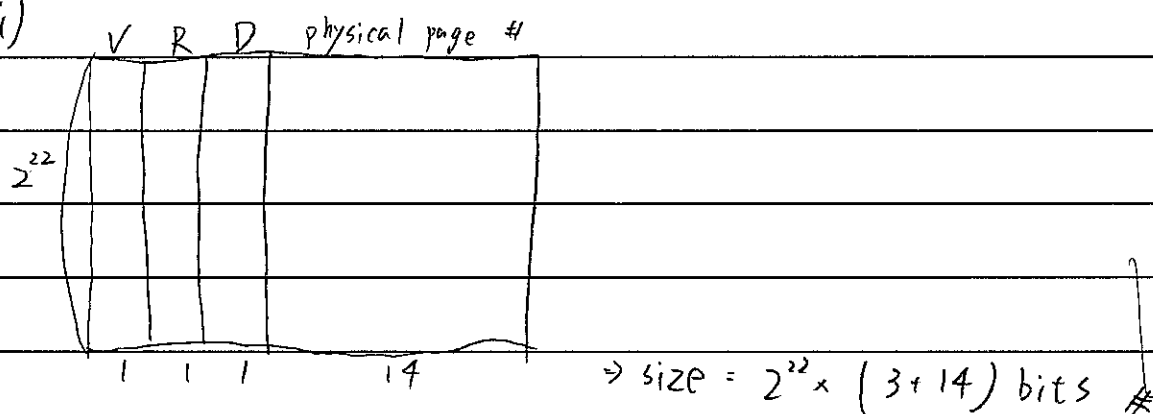
(g)

no,  $\because$  ALU 計算出的值還未穩定, 一定要透過 EX/MEM 裡儲存穩定的值後, 才能送往 TLB 查詢

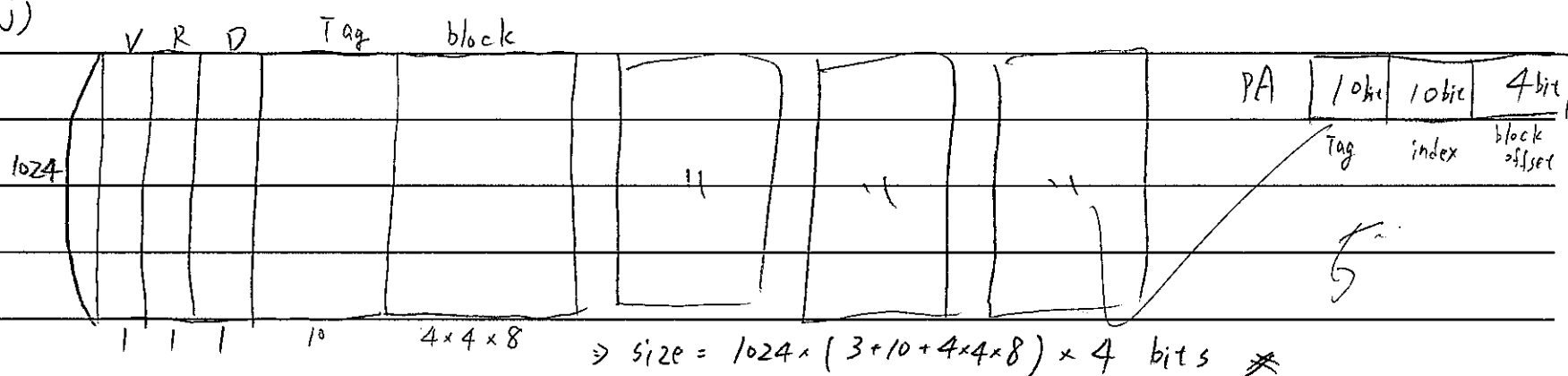
(h)



(i)



(j)



(k)

add \$S1, \$S2, \$S3 此時 forward A = 00, forward B = 00

add \$S2, \$S3, \$S4 此時 forward A = 00, forward B = 11

addi \$S3, \$S4, 5 此時 forward A = 10, forward B = 10

add \$S4, \$S3, \$S2 此時 forward A = 00, forward B = 00

sub \$S5, \$S1, \$S2 此時 forward A = 01, forward B = 01

add \$S2, \$S4, \$S4 此時 forward A = 00, forward B = 11

lw \$S7, 8(\$S0) 此時 forward A = 00, forward B = 00

add \$6, \$S5, \$S5 此時 forward A = 01, forward B = 01

add \$S0, \$S7, \$S7 此時 forward A = 00, forward B = 00

add \$S0, \$S7, \$S7 此時 forward A = 01, forward B = 01

1. 101 學年度 第 一 學期 期 末 考 工 三 系 姓名 林 秉 弘 學號 B9929018

(1) MEM-stage 的 forwarding

if (EX/MEM.ResWrt and (EX/MEM.Rd  $\neq$  0) and (EX/MEM.Rd = ID/EX.Rs))

forward A = 10;

if (EX/MEM.ResWrt and (EX/MEM.Rd  $\neq$  0) and (EX/MEM.Rd = ID/EX.Rc))

forward B = 10;

WB-stage 的 forwarding

if (MEM/WB.ResWrt and (MEM/WB.Rd  $\neq$  0) and not (EX/MEM.ResWrt and (EX/MEM.Rd  $\neq$  0) and (EX/MEM.Rd = ID/EX.Rs)) and (MEM/WB.Rd = ID/EX.Rs))

forward A = 0;

if (MEM/WB.ResWrt and (MEM/WB.Rd  $\neq$  0) and not (EX/MEM.ResWrt and (EX/MEM.Rd  $\neq$  0) and (EX/MEM.Rd = ID/EX.Rc)) and (MEM/WB.Rd = ID/EX.Rc))

forward B = 0;

30  
2.

(a) F, block 增加太多時反而會使 miss rate 上升

(b) T

(c) T, data 在 memory 內

(d) F, TLB 為 PT 的一部份, 故 TLB Hit, Pt 必也 Hit

(e) F, 雖然解決不一致的問題, 但卻使 cycle time 增長許多, 除非有外加 write buffer 來縮短時間, 否則用 write-back 較好

(f) F, 切很多的 stage 可能會造成 寫入外所需的時間變超多, 導致 performance 下降, 應切剛好即可

(g) T, 若 branch 指令很多的話也是有可能 I-cache miss 較高

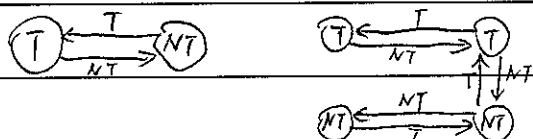
(h) F, load-use 仍有 1 個 stall cycle

(i) F, 只有 1 個 stall cycle 在第 1 個指令與第 2 個指令之間

DU F

3.

(a) 透過 1 個 predictor 或 2 個 predictor 還有 BTB 去猜這次遇到 branch 指令時 PC 該更新為 PC+4 還是 TA, 而不是死板板的一直 PC+4, 此猜測過程稱 Dynamic branch prediction



(請翻面繼續作答)

長庚大學期中、期末考試答案用紙

科目 \_\_\_\_\_

學年度 第 \_\_\_\_\_ 學期 \_\_\_\_\_ 考 \_\_\_\_\_ 系 姓名 \_\_\_\_\_ 學號 \_\_\_\_\_

3.

(b)

在更新值時 (sw), 若在 cache 取 hit 則直接更新 cache 內之值, 並將 dirty bit 設為 1, 不用同時更新 memory 內之值。等到那條 cache line 要被 replace 時, 則去看 dirty bit 若為 0 則直接丟掉, 若為 1 則需往 memory 更新後才丟掉 \*