

# Digital System Design

## HW06 : Picorv32\_knn

105060016 謝承儒 電資 20

目的：

將每張圖

PCPI：

### 一. Input、Output、Others (FSM)

#### 1. Input：

A. [31:0] pcpi\_rs1, pcpi\_rs2：輸入的 2 個數字。

B. pcpi\_insn\_valid：代表是否可以開始計算。

C. [31:0] mem\_rdata：

藉由上一個 cycle 的 addr，從記憶體得到的值

#### 2. Output：

A. [31:0] pcpi\_rd：回傳值

B. pcpi\_wait：若為 1，代表還沒算完。

C. pcpi\_wr：若為 1，就將現在的 pcpi\_rd 傳出去。

D. pcpi\_ready：若為 1，代表已經算完。

E. [31:0] mem\_addr：利用 addr 可以在下個 cycle 得到記憶體的值。

#### 3. Others：

A. [2:0] state, next\_state

B. [31:0] step, next\_step

C. [31:0] text\_pixel, train\_pixel

D. [31:0] distance, next\_distance

E. [31:0] smallest\_dis[4:0], next\_smallest\_dis[4:0]

F. [31:0] smallest\_label[4:0], next\_smallest\_label[4:0]

G. [31:0] return\_value

### 二. 設計過程

1. 在軟體部分(C 語言)呼叫硬體部分的 function--

hard\_knn\_pcpi(int,int)，後面放的兩個整數分別存入 pcpi\_rs1、pcpi\_rs2。

2. 進入 FSM，經過 950 張後的運算後，回傳 5 個最接近的 class。

3. 在軟體計算哪個 class 最高票，將它印出來。

### 三. FSM 中 state 的轉換

#### 1. START :

每次呼叫硬體 function 時，都會進入這個初始 state，  
pcpi\_insn\_valid=1，則下個 state 為 TEST，而且將 addr 設為  
被測試圖片的記憶體位址，如此下一個 cycle 就能取得相對應的  
值。

若輸入的 pcpi\_rs2 > 999 代表已經和 950 張比對完成，下個  
state 為 RETURN。

#### 2. TEST :

將 mem\_rdate 存入 test\_pixel，下個 state 為 TRAIN，並  
將 addr 設為對比圖片的記憶體位址。

#### 3. TRAIN :

將 mem\_rdate 存入 train\_pixel，下個 state 為  
COMPARE。

#### 4. COMPARE :

將 distance 加上  $(\text{test\_pixel} - \text{train\_pixel})^2$  存入  
next\_distance。

將 step+1 後存入 next\_step，代表往下一個 pixel 繼續計  
算。

若 step==3072 代表已經和這張對比圖片比完，則進入  
CHOOSE\_SMALL。反之，則進入 TEST 繼續比對。

#### 5. CHOOSE\_SMALL :

將 distance 和 smallest\_dis 這 5 個一一比較，[0]最小、[4]  
最大。要是比 smallest\_dis 小，就將 distance 放入相應的  
next\_smallest\_dis，並把原本裏頭的移到後頭。

下個 state 為 END。

#### 6. RETRUN :

將 return\_value 設成 smallest\_label，並把相對應的  
smallest\_die 設為 MAX\_INT(重置)。

下個 state 為 END。

重複 5 次這個動作，將前 5 個 label 傳回軟體。

#### 7. END :

將 pcpi\_wr、pcpi\_ready 皆設為 1，把 pcpi\_rd 送出去。

#### 四．遇到的問題

1. 在 COMPARE 之後，直接進入 TEST 會差 1 記憶體位址，

因為 step 還沒+1。

在 addr 設定時，主動+1。

#### 五．討論

1. How do you partition your algorithm?

計算 distance、選擇最接近的 label 在硬體做。Vote 在軟體做。

2. Profile both software and hardware versions. Which part do you think is the bottleneck of your (software/hardware) design?

設計如何在硬體儲存前 5 個最接近的 label。但後來曉得硬體的 call function 是重複呼叫同一個，而不是如軟體般重新建立一個新的。所以可以在 function 開變數儲存值。

#### 六．資料來源

1. 老師的 PPT