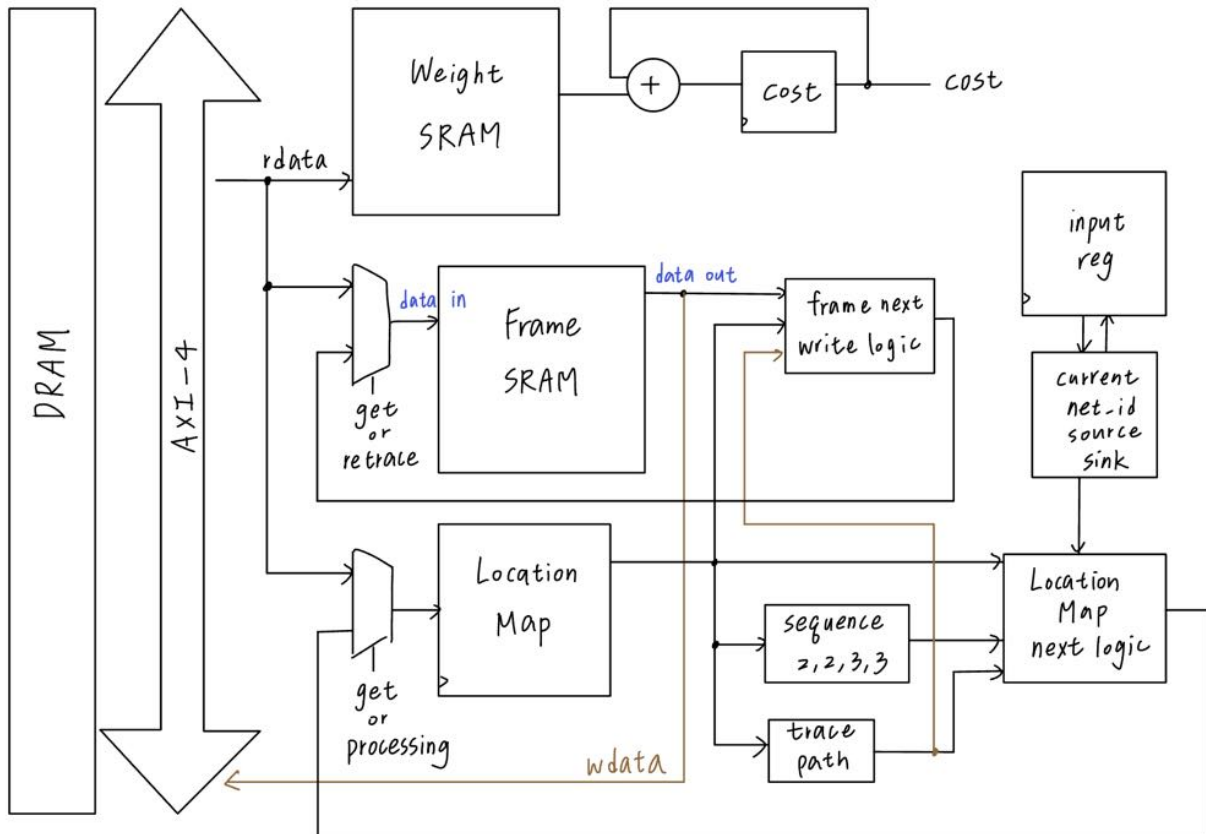


ICLAB Midterm Project Report

Maze Router Accelerator (MRA)

312591003 呂昀庭

一、電路架構



Storage:

本次設計共使用了兩個 128bits x 128 的 SRAM，分別用來儲存從 DRAM 讀出來的整張 Frame 和 Weight。主要的 filling 和 retrace 步驟則是利用 64 x 64 x 2bits 的 registers (Location_map) 進行。

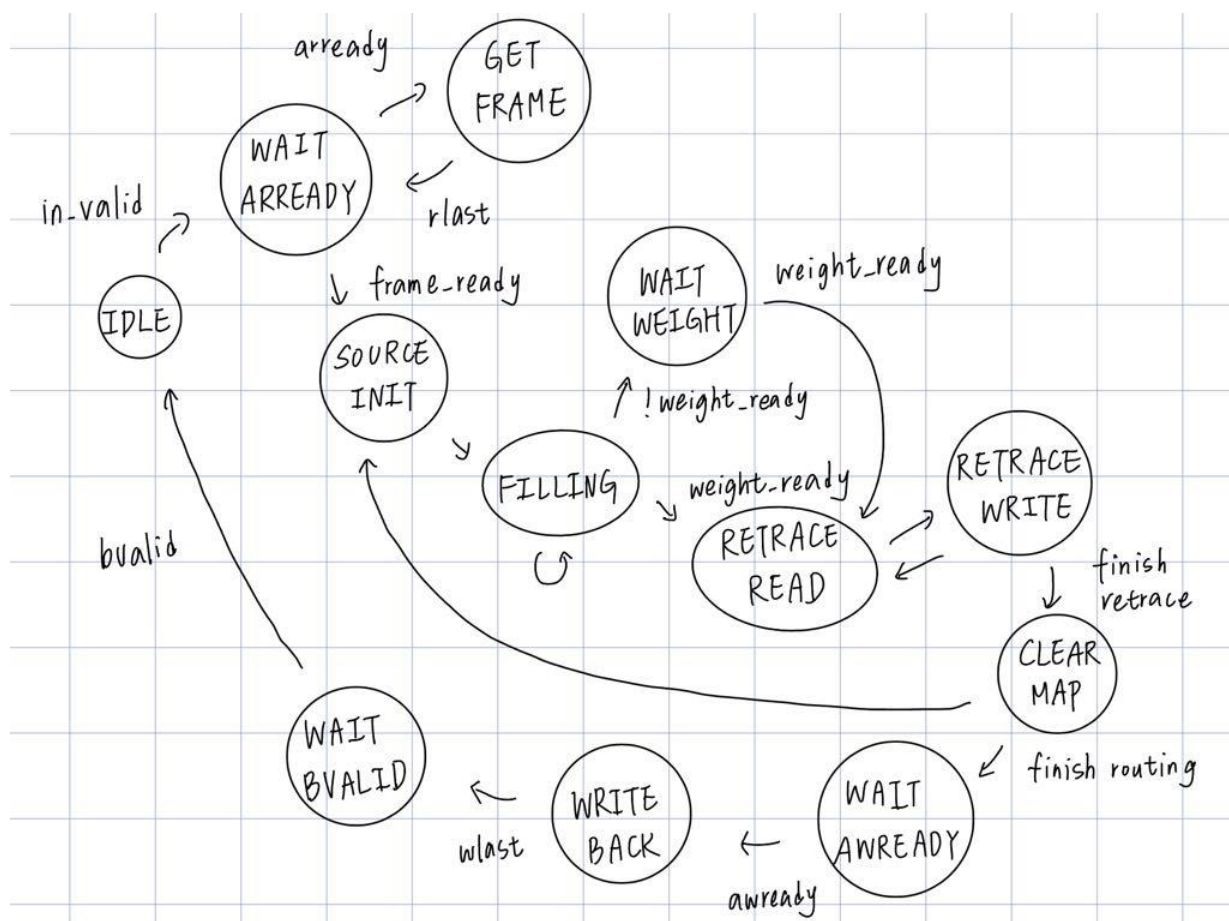
Design:

1. 因為這次 DRAM 的 burst length 為 127，read latency 很長，所以在 input 進入時就可以同時開始讀 DRAM。
2. 先讀 Frame 再讀 Weight，讀 Frame 時除了將整張圖存進 SRAM 之外，同時也將 Location_map 中對應 macro 的位置設為 1。讀 Weight 的同時即可開始第一個 net_id 的 filling。
3. 在開始 filling 前，將目前的 source 設為 2、sink 設為 0。從 source 開始，filling sequence 為 2,2,3,3，直到 sink 不為 0 時停止 filling。若這時 Weight 還沒有讀完，則進入 WAIT_WEIGHT 等待 weight_ready，否則直接進入 retrace 的步驟。
4. 在 retrace 時，走一步需要花兩個 cycle。從 sink 開始往回，RETRACE_READ 時將走到的路徑設為 1 並讀出 Frame SRAM 和 Weight SRAM；RETRACE_WRITE 時依照 net_id 更新走到的路

徑寫回 Frame SRAM，並將對應的 weight 累加到 cost。走到 source 即可停止 retrace。

5. CLEAR_MAP 將剛剛有填入 filling sequence 但是沒有被 retrace 到的地方清空。並判斷是否已 routing 完整張 map。若還沒就繼續 filling, retrace，若完成則可以將 Frame SRAM 中的資料寫回 DRAM。

二、FSM



三、優化方法

1. SRAM 的寬度設為 128bits，和 DRAM data 的寬度相同，可以直接對接。不過可能需要經過一些 dummy logic，否則會有 hold time 的問題。
2. 開 64 x 64 個 registers 直接對整張 map 進行 fill 和 retrace，比較容易，latency 也較短，且 filling sequence 從 1, 2, 3 改成 2, 2, 3, 3，registers 只需要開 2bits 就足夠，CLEAR_MAP 時也只要判斷第 1 個 bit 是否不為零。
3. 原本在 SOURCE_INIT 之前有一個 GET_WEIGHT 的 state，等 Frame 和 Weight 都讀完才開始 filling，後來將讀 Weight 拆散在 SOURCE_INIT, FILLING, WAIT_WEIGHT 都可以做，並多一條判斷訊號來看 weight 是否讀完，就可以邊讀邊 fill，降低 latency。
4. Location_map 利用 IDLE state 進行 reset 歸零。
5. FSM 善用 rlast 訊號，並盡量以 gray code 的方式轉換 state。