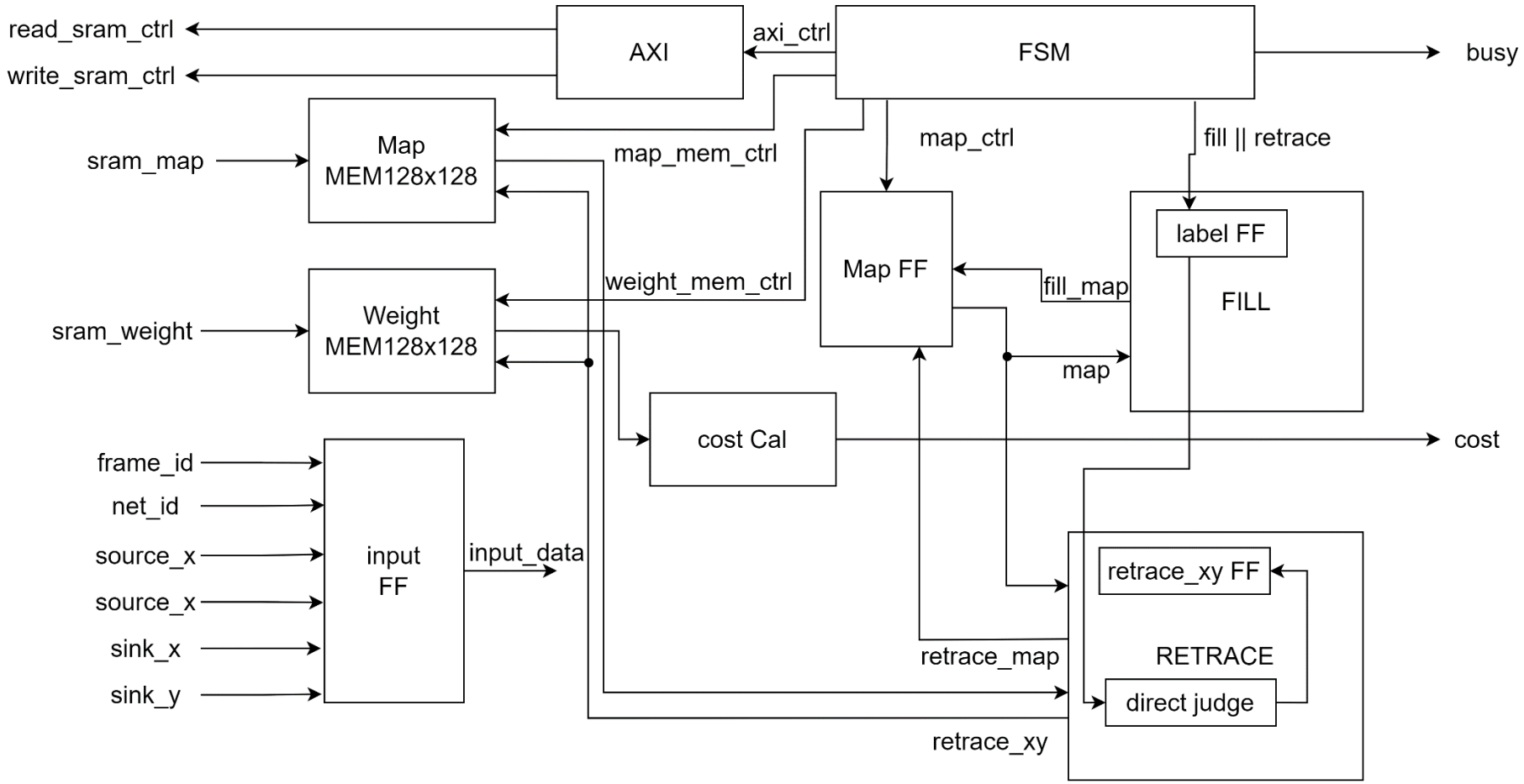
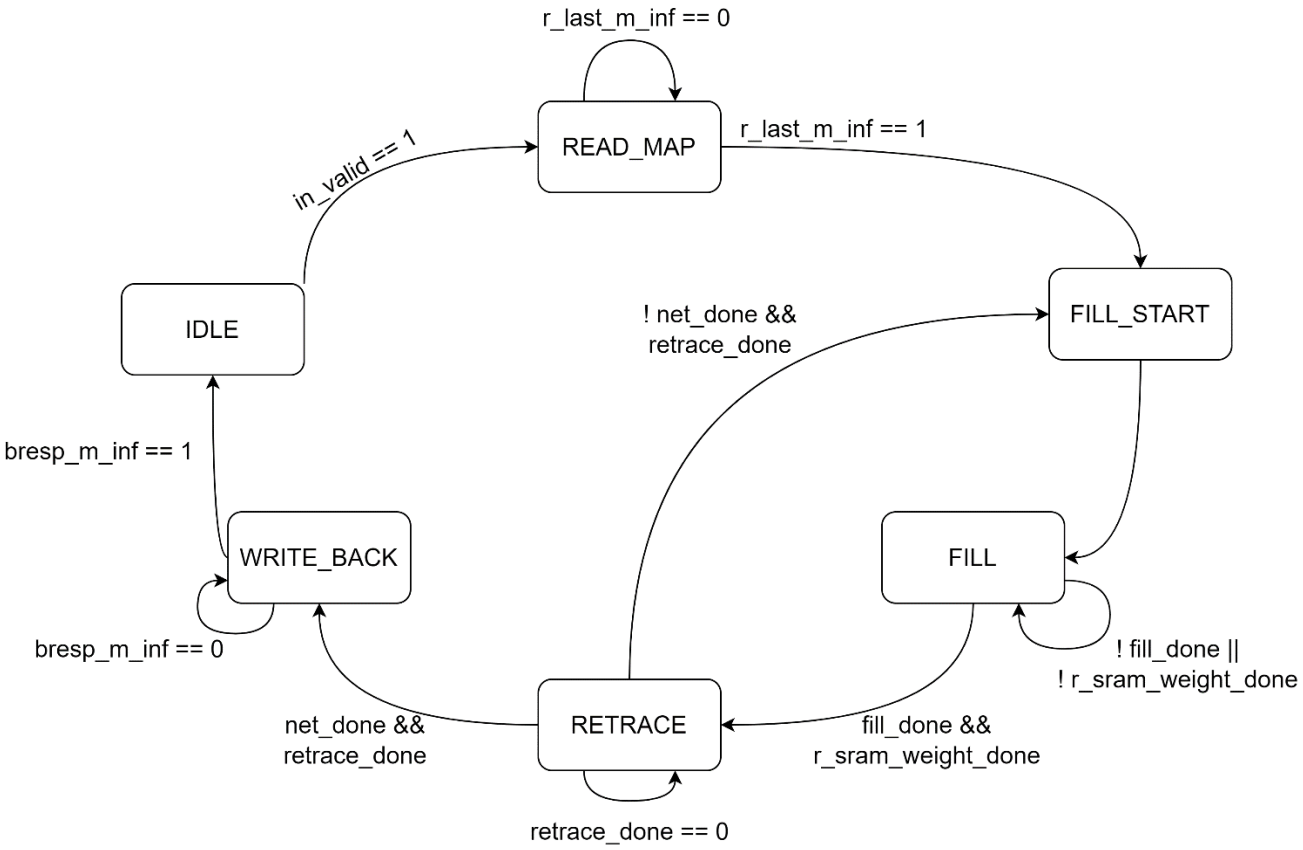
Midterm Project Report

311510170 iclab064

一. 電路架構圖



二. FSM(流程圖)



READ_MAP :
從 map DRAM 中讀出目前 frame_id 的 map 並存進 MAP MEM 內，當 r_last_m_inf 為 1 代表讀完整張 map。

FILL_START：
若有未清空的 label 將 Map FF 還原，並將新的 source 及 sink 在 Map FF 上做標記，方便之後 FILL 的判斷。

FILL：
利用 label FF 記錄目前標記到的 label，並接收 map FF 傳送過來的 map 判斷哪些地方要標記目前的 label。FILL 在 fill_done 拉為 1 且所有 weight 從 weight DRAM 取出存到 weight MEM 後進到 RETRACE state，需要等待 weight 全部取完是因為 retrace state 需要用到 weight 算 cost。

RETRACE ：
label FF 回推之前的 label 傳送給 retrace，讓 retrace 判斷要往回走到 source 的方向，同時更新路徑上的點在 map FF 要改為 lock 的狀態，並將當前 retrace_xy 傳送給 weight MEM 及 map MEM 分別取出 weight 計算 cost 及更新路徑上的點為 net_id。RETRACE 到 source 時

retrace_done 拉為 1，此時需判斷是否還有 net_id 需進行繞線，若有則回到 FILL_START，沒有則進到 WRITE_BACK state。

WRITE_BACK：將 MAP MEM 內的值透過 AXI 寫回 MAP DRAM。寫完最後一筆收到 bresp_m_inf 為 1 時，回到 IDLE state。

三. 優化方法

1. 因為 map FF 為一個 2bit*64*64 的矩陣，因此只要需要用到 map 判斷的地方，都會產生一個很大的 mux，因此若不是同時需要進行判斷，可以共用 mux，減少面積。
2. 在更新 map FF 時電路內部會寫到兩層 for 迴圈，合成軟體可能較難優化，behavior 盡量寫得有一致性，這樣合成時所需花的時間也比較少。
3. 原先 retrace 時，每往回走一個點會需要從 map MEM 中取出一整條 32 個點，改完後再寫入 map MEM，此時需要花讀出跟寫入共 2 cycle，因為一次可以取出 32 個點，因此若是走左、右方向，要更新的點剛好在同一條，可以讀出一次後就一直寫入更新的值即可，此種優化方式可以將 retrace 所需的 cycle 減少約一半。