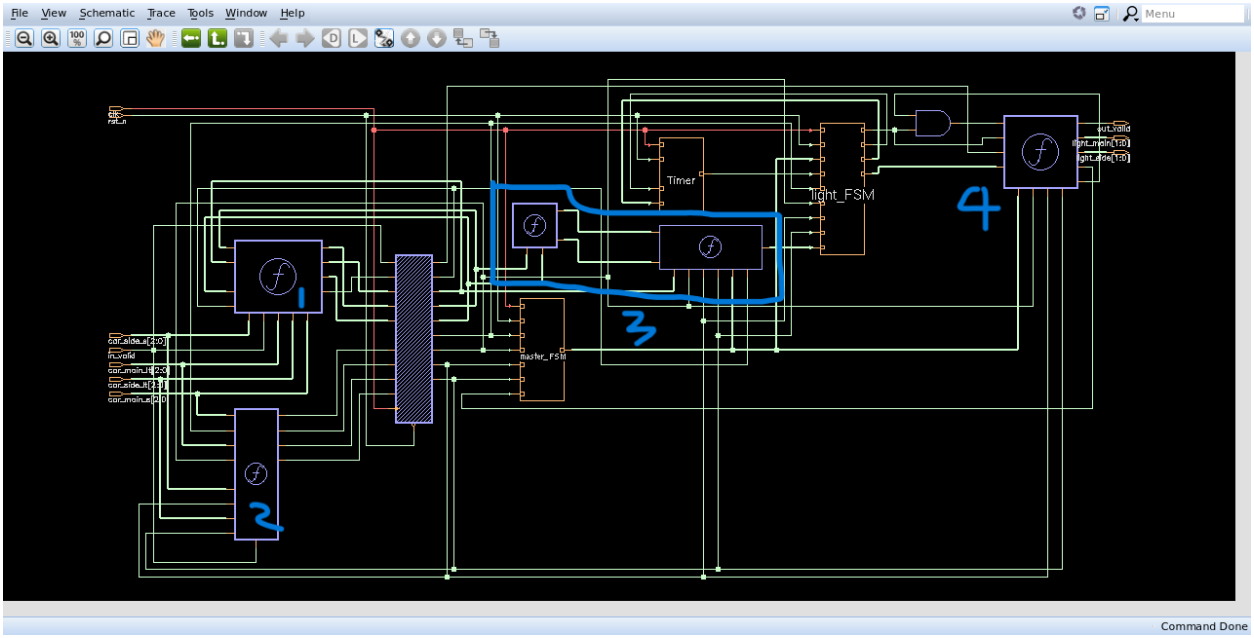
數電HW3



此電路由兩個FSM: light_fsm和master_fsm組成， light_fsm共分GREEN, YELLOW, RED, LEFT 4個狀態，負責控制燈號的變化，master_fsm分為 MAIN_S, SIDE_S, MAIN_LT, SIDE_LT, DEFAULT, IDLE，負責控制輸出。其中light_fsm的輸出只會在 MAIN_S, SIDE_S, MAIN_LT, SIDE_LT 被master_fsm 使用，因此 light_fsm 只有考慮這四個狀態下的情況，以此簡化其判斷條件。

TL

輸入資料儲存(上圖1、2)

為了節省儲存輸入值所需的ff數量，當每筆資料輸入時，可以先計算每筆資料是否等於零(mains_nz ,mainlt_nz ,sides_nz ,sidelt_nz)，以及其綠燈/左轉燈要維持的秒數是最小值的幾倍(mains_comp, mainlt_comp, sides_comp, sidelt_comp)，再將其用ff儲存以便之後使用。

red_tval_comb(上圖3)

得到輸入資料後，還需要計算在目前的master_state下，從紅燈轉換到綠/左轉燈時，要load到timer鐘的時間值為多少。這是因為這時的綠/左轉燈是所有條件中，唯一需要利用輸入來計算燈號維持時間的情況，因此將這部分特別拉出來處理，來幫助我們簡化其他條件下的判斷。

TL_comb(上圖4)

TL還會輸出change_dir_ok來幫助master_fsm判斷是否轉態，SIDE_S, MAIN_LT, SIDE_LT這三個狀態是在紅燈的timer數完時轉態，MAIN_S則是還要考慮只有car_main_s大於零的情況。而DEFAULT只會維持1cycle故change_dir_ok必為1，IDLE則要維持到IN_VALID拉起後的下一個cycle才能轉態，所以會需要用一個ff把in_valid存起來。

master_fsm

master_fsm 的狀態轉換會先由light_fsm輸出的change_dir_ok決定，若change_dir_ok = 0，則維持原狀態，否則會根據目前狀態以及各個輸入是否為零，來決定接下來跳到哪個狀態。

light_fsm

各腳位功能:

腳位類型	名稱	用途
輸出	ltload	傳送load訊號給timer，決定是否要更新燈號維持時間。在燈號轉換時才會啟用
輸出	ltval	輸出燈號維持時間值給timer
輸出	lfsmdone	輸出timer的ldone訊號，此訊號會被用來算出change_dir_ok
輸出	light_state	輸出燈號，在DEFAULT, IDLE以外的狀態時，light_state的值會直接作為輸出
輸入	master_state	由master_fsm輸出的master_state，用於判斷light_state的下一個狀態
輸入	red_tval	由TL輸出，左轉燈/綠燈的維持時間
輸入	ldone	由timer輸出，表示timer是否已數完
輸入	mainlt_nz	輸入資料運算結果，表示car_main_lt是否為零
輸入	mains_nz	輸入資料運算結果，表示car_main_s是否為零

輸入	sides_nz	輸入資料運算結果，表示car_side_s是否為零
輸入	sidelt_nz	輸入資料運算結果，表示car_side_lt是否為零

還可以再優化的地方

demo1 上傳結束後，我發現程式中有許多重複的邏輯，但兩者不會同時被使用，例如以下的兩組casez，他們並不會同時assign red_tval的值，因此可以將此硬體的輸入、輸出加上mux，讓它能夠重複利用，以減少面積。

另外由於這次作業中用到兩個狀態機，兩者的轉態條件會互相影響。因此light_fsm的轉態應該也可以像master_fsm依靠change_dir_ok控制一樣，先依目前的條件判斷是否要變換燈號，再利用此訊號判斷下一周期的燈號，可以簡化next_light_state的判斷。

```
casez(sides_comp)
    2'b1?:begin
      red_tval_sides = 2;
    end

    2'b01:begin
      red_tval_sides = 5;
    end

    2'b00: begin
      red_tval_sides = 8;
    end

    default: begin
      red_tval_sides = 1'bx;
    end
endcase

casez(mainlt_comp)
  2'b1?:begin
    red_tval = 2;
  end

  2'b01:begin
    red_tval = 5;
  end

  default: begin
    red_tval = 8;
  end
endcase
```