SRAM_Controller_SPEC

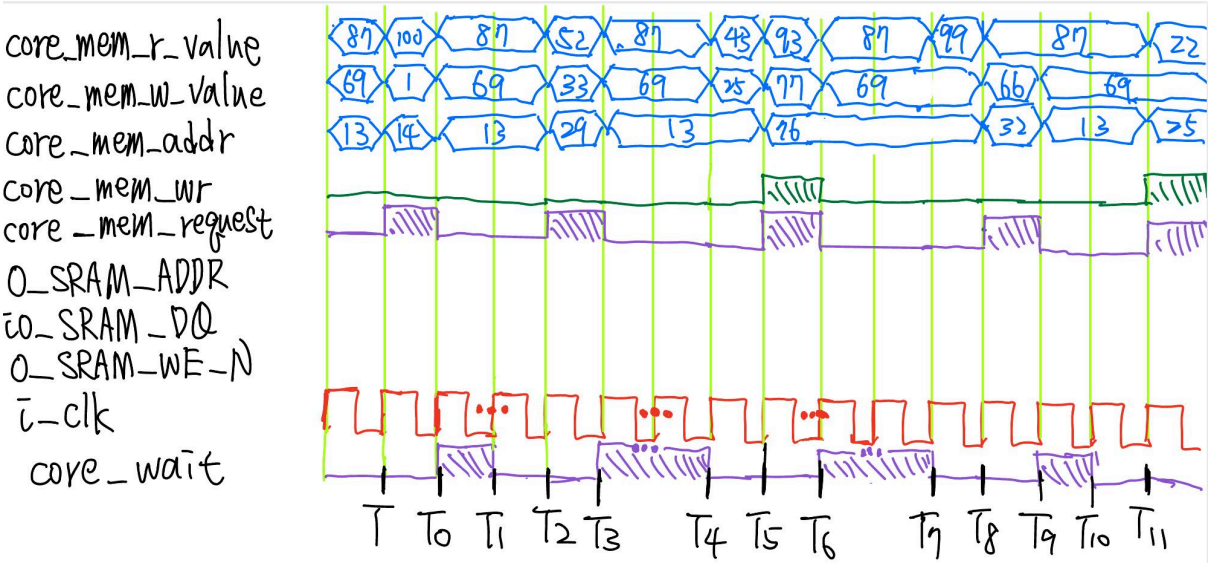
core_mem_r_value[15:0] (O)	使用者從library(mem)讀取的資訊值
core_mem_w_value[15:0] (I)	使用者寫入library(mem)的資訊值
core_mem_addr[19:0] (I)	使用者使用記憶的位址
core_mem_wr (I)	寫入signal low，讀出signal high
core_mem_request (I)	core請求讀寫
core_wait (O)	寫入或讀取後，有latency，告訴core等待。low才能再次讀寫
o_SRAM_ADDR (O)	使用記憶記憶體位址
io_SRAM_DQ (IO)	讀取記憶體值或是寫入記憶體值
o_SRAM_WE_N (O)	記憶體寫入有效訊號，low for write, high for read
o_SRAM_CE_N (O)	低電位表示啟用 SRAM
o_SRAM_OE_N (O)	低電位表示 SRAM 可以驅動資料輸出總線
o_SRAM_LB_N (O)	低電位表示啟用 SRAM 的低 8-bit
o_SRAM_UB_N (O)	低電位表示啟用 SRAM 的高 8-bit

Lab3 的 Memory access 使用連續寫入，連續讀取

```
assign o_SRAM_ADDR = (state_r == S_RECD) ? addr_record : addr_play[19:0];
assign io_SRAM_DQ  = (state_r == S_RECD) ? data_record : 16'dz; // sram_dq as output
assign data_play   = (state_r != S_RECD) ? io_SRAM_DQ : 16'd0; // sram_dq as input
assign o_SRAM_WE_N = (state_r == S_RECD) ? 1'b0 : 1'b1;


assign o_SRAM_CE_N = 1'b0;
assign o_SRAM_OE_N = 1'b0;
assign o_SRAM_LB_N = 1'b0;
assign o_SRAM_UB_N = 1'b0;
```

Final Project 的 寫入讀取的 spec



T ：core要求寫入記憶體，位址14數值1
T0：等記憶體latency，告訴core等待，不要傳送讀寫要求
T1：剛等完記憶體處理，core於posedge clk見到core_wait = 0 才能讀寫，所以再等一周期
T2：core要求寫入記憶體，位址29數值33
T3：等記憶體
T4：剛等完記憶體處理，core於posedge clk見到core_wait = 0 才能讀寫，所以再等一周期
T5：core要求讀取記憶體，位址76
T6：等記憶體
T7：剛等完記憶體處理，core於posedge clk見到core_wait = 0 才能讀寫
T8：讀取值99，知道mem[79]=99，並要求寫入記憶體，位址32數值66
T9：等記憶體
T10: 剛等完記憶體處理，core於posedge clk見到core_wait = 0 才能讀寫，所以再等一周期
T11: : core要求讀取記憶體，位址25

備註:
A. 為了設計可讀性，所以**core_mem_value**分成兩**read and write**兩種。
B. 為更嚴謹要求讀寫記憶體時間點，使用**core_mem_wr**和**core_mem_reques**控制。
C.core_wait就是看**sram**的**latency**多久能做完，我們**lab3**就卡在這裡，實驗抓出**data valid**的時間點即可。