HDL Digital Design (Graduate Level)
Spring 2024

HOMEWORK
REPORT

Must do self-checking before submission:
☑ Compress all files described in the problem into one zip file.
☑ All files can be compiled under ModelSim environment.
☑ All port declarations comply with I/O port specifications.
☑ Organize files according to File Hierarchy Requirement
☑ No waveform files or project file in deliverables

Due Date: 2024/05/15 08:59 a.m.

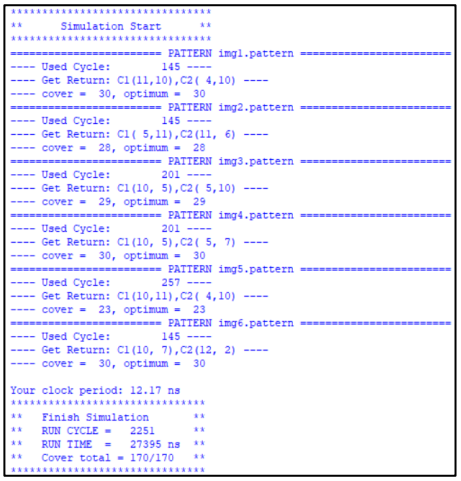
Student name: ___林柏戎___
Student ID: ___Q36114239___

1. Performance & Area table

Gate level simulation all pass (Yes/No)	Yes
Clock period	12.17
RUN TIME	27395
Total logic elements	1220

2. RTL & Gate-level simulation result on the terminal.

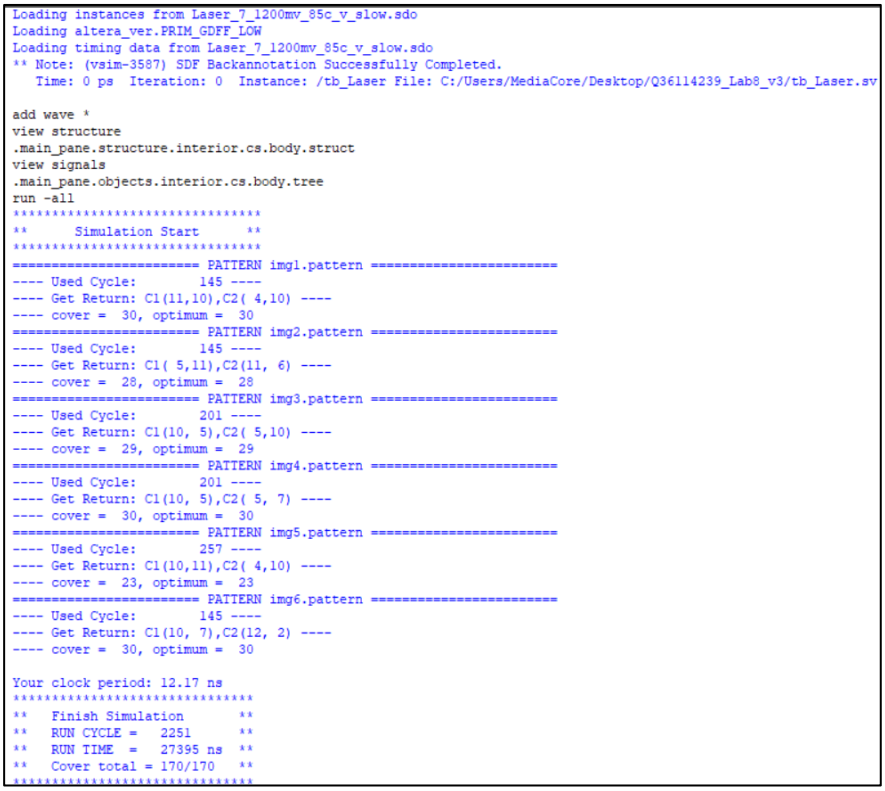
RTL:



```
*******************************
**      Simulation Start      **
*******************************
======================== PATTERN img1.pattern ========================
---- Used Cycle:        145 ----
---- Get Return: C1(11,10),C2( 4,10) ----
---- cover =  30, optimum =  30
======================== PATTERN img2.pattern ========================
---- Used Cycle:        145 ----
---- Get Return: C1( 5,11),C2(11, 6) ----
---- cover =  28, optimum =  28
======================== PATTERN img3.pattern ========================
---- Used Cycle:        201 ----
---- Get Return: C1(10, 5),C2( 5,10) ----
---- cover =  29, optimum =  29
======================== PATTERN img4.pattern ========================
---- Used Cycle:        201 ----
---- Get Return: C1(10, 5),C2( 5, 7) ----
---- cover =  30, optimum =  30
======================== PATTERN img5.pattern ========================
---- Used Cycle:        257 ----
---- Get Return: C1(10,11),C2( 4,10) ----
---- cover =  23, optimum =  23
======================== PATTERN img6.pattern ========================
---- Used Cycle:        145 ----
---- Get Return: C1(10, 7),C2(12, 2) ----
---- cover =  30, optimum =  30

Your clock period: 12.17 ns
*******************************
**   Finish Simulation       **
**   RUN CYCLE =   2251      **
**   RUN TIME  =   27395 ns  **
**   Cover total = 170/170   **
*******************************
```

Gate-level:

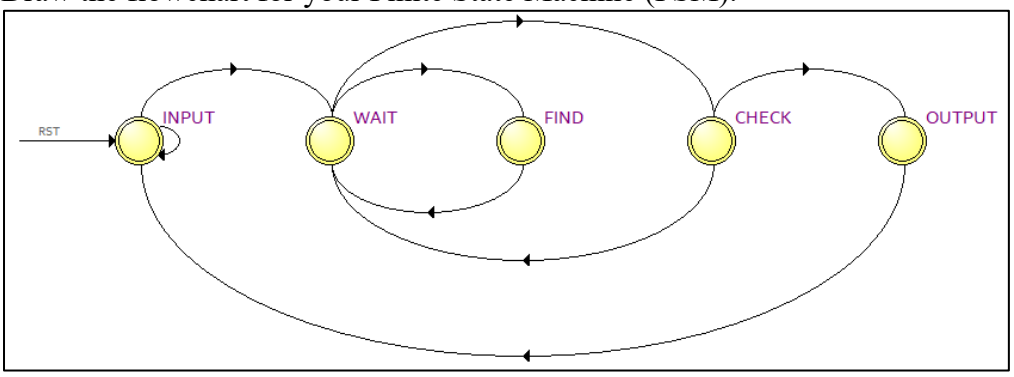


```
Loading instances from Laser_7_1200mv_85c_v_slow.sdo
Loading altera_ver.PRIM_GDFF_LOW
Loading timing data from Laser_7_1200mv_85c_v_slow.sdo
** Note: (vsim-3587) SDF Backannotation Successfully Completed.
   Time: 0 ps  Iteration: 0  Instance: /tb_Laser File: C:/Users/MediaCore/Desktop/Q36114239_Lab8_v3/tb_Laser.sv

add wave *
view structure
.main_pane.structure.interior.cs.body.struct
view signals
.main_pane.objects.interior.cs.body.tree
run -all
*******************************
**      Simulation Start      **
*******************************
======================== PATTERN img1.pattern ========================
---- Used Cycle:        145 ----
---- Get Return: C1(11,10),C2( 4,10) ----
---- cover =  30, optimum =  30
======================== PATTERN img2.pattern ========================
---- Used Cycle:        145 ----
---- Get Return: C1( 5,11),C2(11, 6) ----
---- cover =  28, optimum =  28
======================== PATTERN img3.pattern ========================
---- Used Cycle:        201 ----
---- Get Return: C1(10, 5),C2( 5,10) ----
---- cover =  29, optimum =  29
======================== PATTERN img4.pattern ========================
---- Used Cycle:        201 ----
---- Get Return: C1(10, 5),C2( 5, 7) ----
---- cover =  30, optimum =  30
======================== PATTERN img5.pattern ========================
---- Used Cycle:        257 ----
---- Get Return: C1(10,11),C2( 4,10) ----
---- cover =  23, optimum =  23
======================== PATTERN img6.pattern ========================
---- Used Cycle:        145 ----
---- Get Return: C1(10, 7),C2(12, 2) ----
---- cover =  30, optimum =  30

Your clock period: 12.17 ns
*******************************
**   Finish Simulation       **
**   RUN CYCLE =   2251      **
**   RUN TIME  =   27395 ns  **
**   Cover total = 170/170   **
*******************************
```

3. Draw the flowchart for your Finite State Machine (FSM).



4. Explain how your design works?
   INPUT: 將 input 給的 (x,y) 存到對應的 x_list, y_list，寄存所有點。
   WAIT: 紀錄圓的(x, y)以及記錄哪些點目前有被計算到。如果(x,y)還沒到最後一個點，則會到 FIND 狀態。若到了最後一個點，則會到 CHECK 狀態。
   FIND: 改變 (x, y) 的值，並會回到 WAIT 狀態，紀錄(x, y)以及計算包含的點。
   CHECK: 確認是否(x1 == x_old && y1 == y_old)。若成立，則代表迭代結束，接下來會到 OUTPUT 狀態。
   OUTPUT: 將 DONE 拉高，且回到 INPUT，進入下一個 pattern。

5. What is your strategy to get a batter Performance?
   分成兩個: 縮減 critical path 以及 減少 RUN CYCLE 的方法。
   - 縮減 critical path
     原本最長的 critical path 是因為用大量的 for loop 在計算有幾個點在圓內以及計數 BITCNT 的時候。
     後來分別使用不同的 module 先對比較少的 bit 數做操作，在 top module 上用 generate for 來複製 module，這樣就從 70.00 變到 12.17。

   - 減少 RUN CYCLE
     原本(x,y)是從(0,0)開始搜尋，且下一個點為(1,0)，直到(x,y)數到(15,15)。
     後來發現起始點可以不用從(0,0)開始，可以從(6,7)開始搜尋，且不是一個一個點慢慢跳，會因為不同的狀況，下一個點就可能不會是 x+1 的結果。
     但當找到第一個圓後，需要從(12,2)開始找。

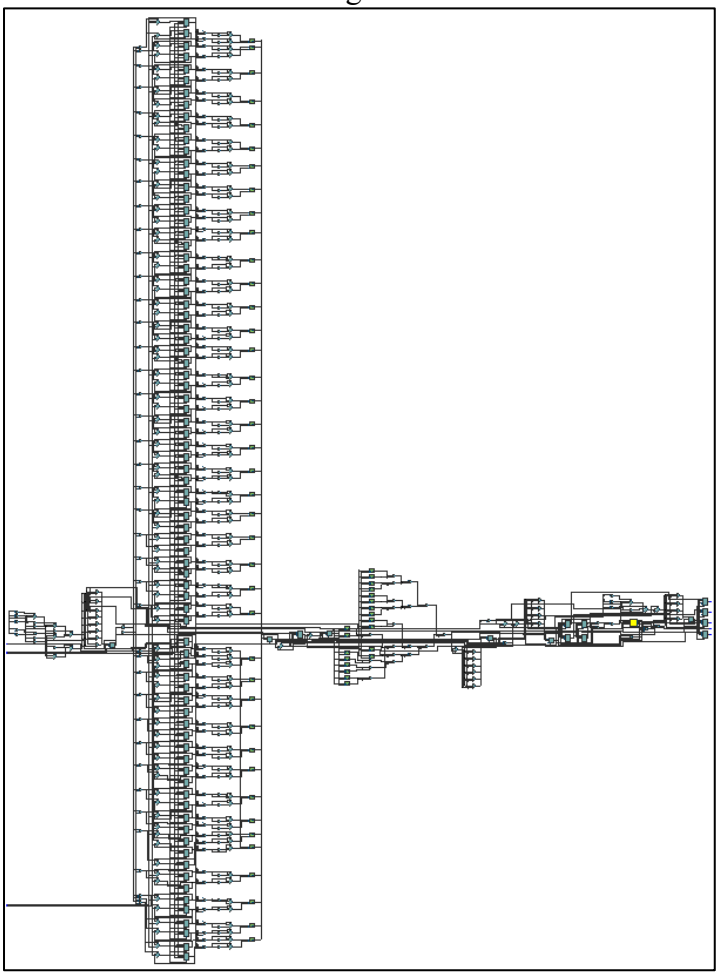
6. Screenshot from Quartus
   - flow summery

Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	1,220 / 114,480 (1 %)
Total registers	500
Total pins	28 / 529 (5 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

- worst-case timing paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	0.013	C2Y[1]~reg0	C2Y[1]	CLK	CLK	12.170	-3.034	3.018
2	0.023	C1Y[1]~reg0	C1Y[1]	CLK	CLK	12.170	-3.034	3.008
3	0.025	C1X[0]~reg0	C1X[0]	CLK	CLK	12.170	-3.042	2.998
4	0.027	C2X[2]~reg0	C2X[2]	CLK	CLK	12.170	-3.050	2.988
5	0.027	C1X[3]~reg0	C1X[3]	CLK	CLK	12.170	-3.050	2.988
6	0.027	C1X[2]~reg0	C1X[2]	CLK	CLK	12.170	-3.050	2.988
7	0.039	C2Y[0]~reg0	C2Y[0]	CLK	CLK	12.170	-3.028	2.998
8	0.047	C2X[3]~reg0	C2X[3]	CLK	CLK	12.170	-3.050	2.968
9	0.049	C1Y[0]~reg0	C1Y[0]	CLK	CLK	12.170	-3.028	2.988
10	0.057	C2X[0]~reg0	C2X[0]	CLK	CLK	12.170	-3.040	2.968
11	0.061	C1X[1]~reg0	C1X[1]	CLK	CLK	12.170	-3.036	2.968
12	0.067	C2X[1]~reg0	C2X[1]	CLK	CLK	12.170	-3.040	2.958
13	0.073	currState.OUTPUT	DONE	CLK	CLK	12.170	-3.044	2.948
14	0.073	C2Y[2]~reg0	C2Y[2]	CLK	CLK	12.170	-3.034	2.958
15	0.081	C1Y[3]~reg0	C1Y[3]	CLK	CLK	12.170	-3.026	2.958
16	0.081	C1Y[2]~reg0	C1Y[2]	CLK	CLK	12.170	-3.026	2.958
17	0.083	C2Y[3]~reg0	C2Y[3]	CLK	CLK	12.170	-3.034	2.948
18	0.991	p2_list[19]	y1[0]	CLK	CLK	12.170	-0.085	11.092
19	0.991	p2_list[19]	y1[1]	CLK	CLK	12.170	-0.085	11.092
20	0.991	p2_list[19]	y1[3]	CLK	CLK	12.170	-0.085	11.092
21	0.991	p2_list[19]	y1[2]	CLK	CLK	12.170	-0.085	11.092
22	1.027	quarter_list[7]	y1[0]	CLK	CLK	12.170	-0.097	11.044
23	1.027	quarter_list[7]	y1[1]	CLK	CLK	12.170	-0.097	11.044

- Schematic view of the design netlist



- Fmax summary

	Fmax	Restricted Fmax	Clock Name	Note
1	82.26 MHz	82.26 MHz	CLK	

7. Lesson learned from this Lab.
   - 這次的 lab 是參考 ic 競賽上的題目，學習到設計一個電路來控制雷射的位置，包括了解如何處理輸入輸出信號，設計雷射的精確定位和控制。
   - 透過計算最佳雷射位置以最大化覆蓋區域，第一次學到關於優化以及迭代解法的實際應用。在少少時間內，設計出一個比較不一樣的計數方法，來加速找出雷射的位置。
   - 用 generate for loop 來複製 module，並達到計數 reg 裡有幾個 1，以及計數 circule 裡有包含幾個 point。