I. Tổng quan hệ thống:

a. FSM:

Có 4 trạng thái chính:

GaRb: 100 001

YaRb: 010 001

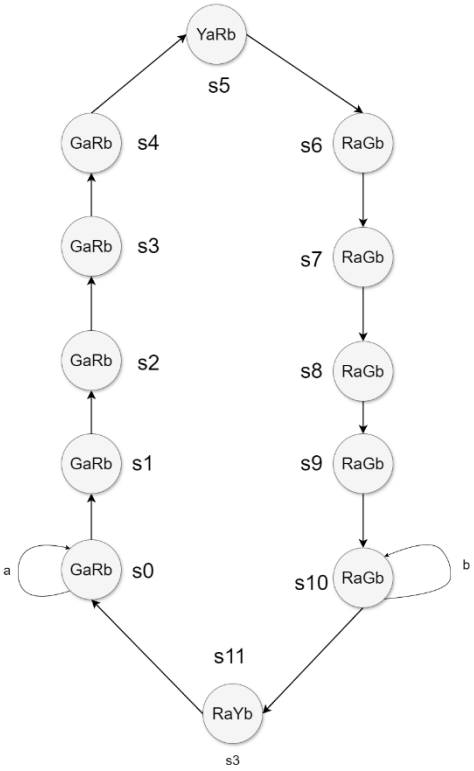
RaGb: 001 100

RaYb: 001 010

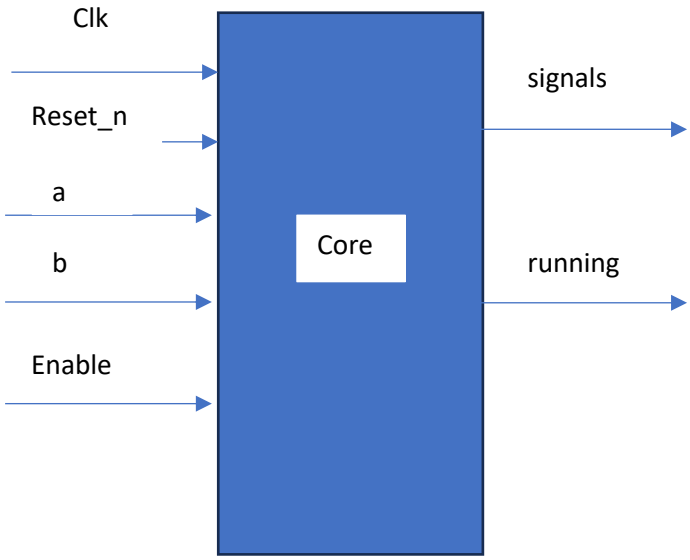
Với (trong FSM):

G là đèn xanh, R là đèn đỏ, Y là đèn vàng.

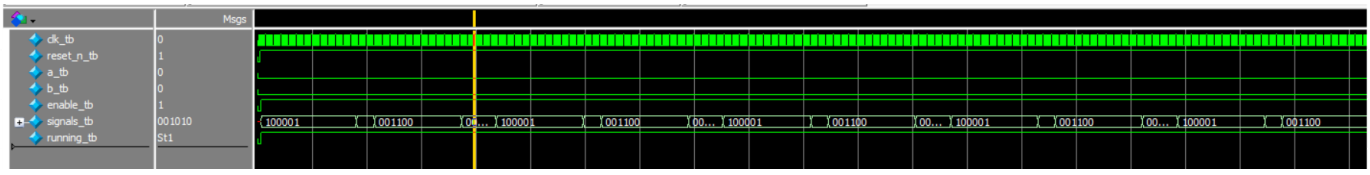
a là lane a, b là lane b



b. Bảng mô tả thiết kế core traffic light controller:



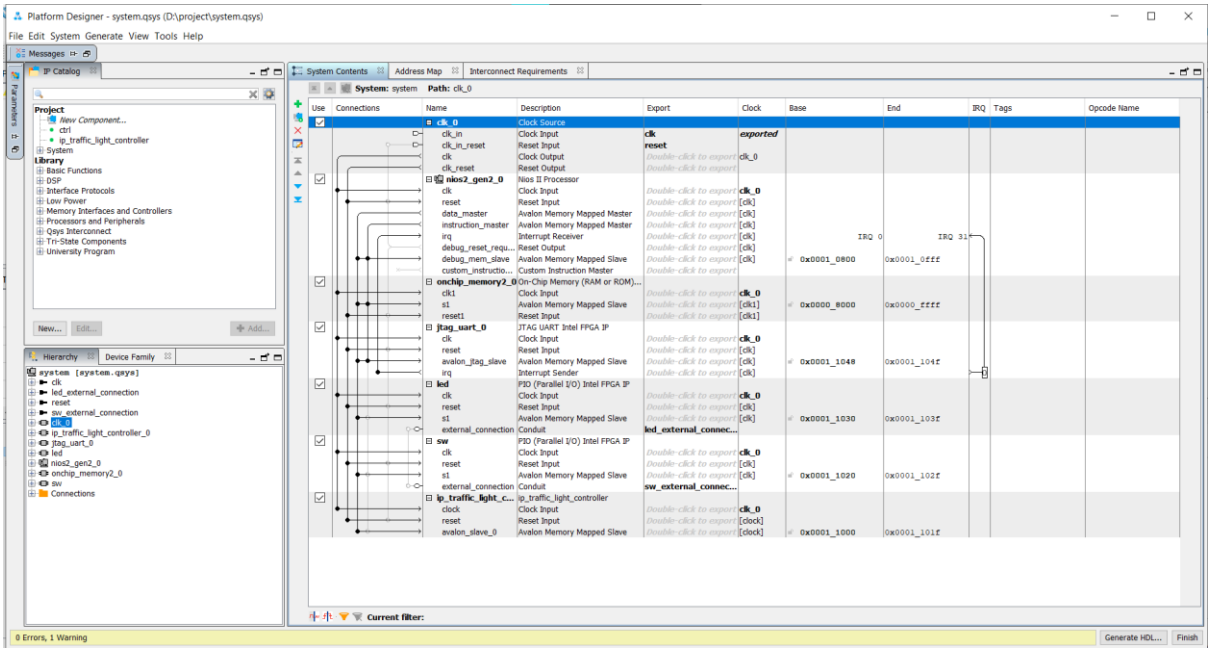
II. Mô phỏng testbench core Traffic light controller:



Nhận xét: Khi a_tb = 0 và b_tb = 0 thì hệ thống chạy đúng với hệ thống mong muốn.

III. Kết quả chạy trực tiếp:

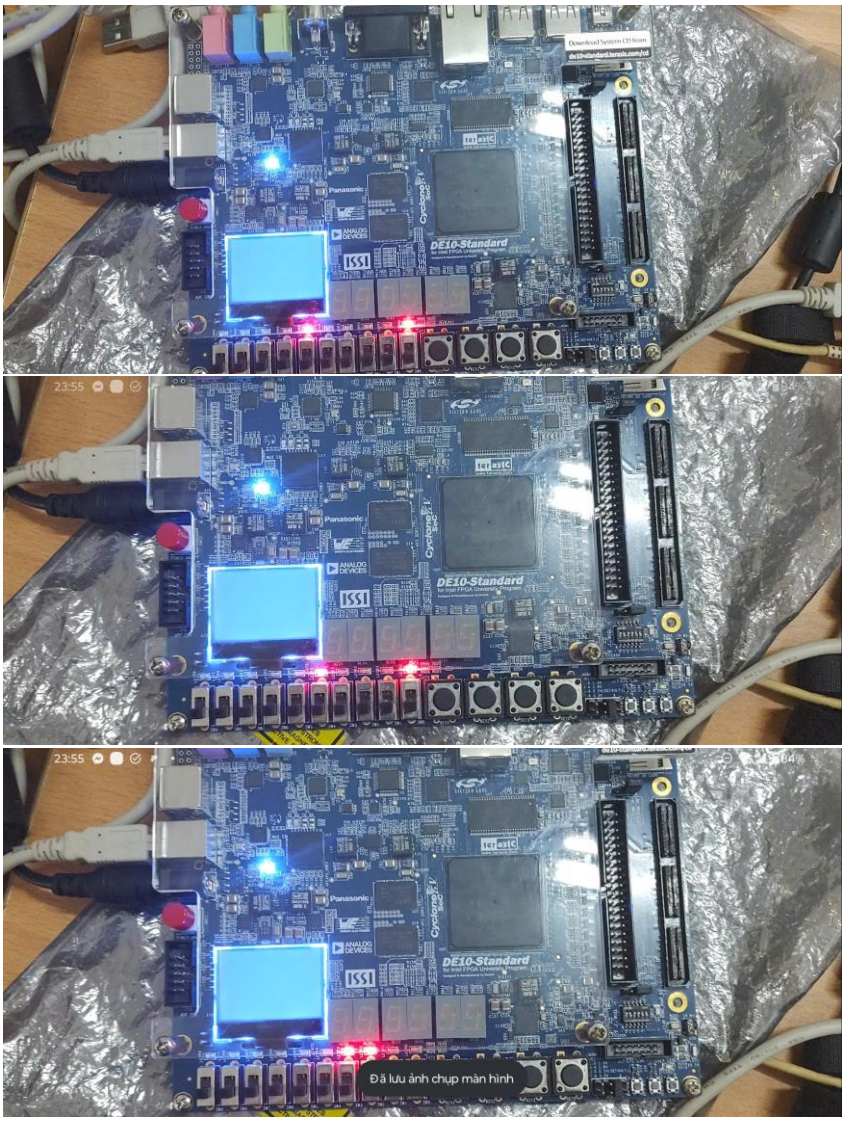
Flatform Designer:



Code eclipse:

```
#include <stdio.h>
#include "system.h"
#include "altera_avalon_pio_regs.h"

// Định nghĩa địa chỉ các thanh ghi CSR
#define ADDR_CONTROL 0
#define ADDR_STATUS 1
#define ADDR_A 2
#define ADDR_B 3
#define ADDR_SIGNALS 4

#define IP 0x11000

// Hàm ghi dữ liệu vào thanh ghi
void write_register(int address, int value) {
    IOWR_32DIRECT(IP, address * 4, value);
}
```

```
// Hàm đọc dữ liệu từ thanh ghi
int read_register(int address) {
   return IORD_32DIRECT(IP, address * 4);
}

// Hàm ghi giá trị vào thanh ghi Control
void set_control(int value) {
   write_register(ADDR_CONTROL, value);
}

// Hàm đọc giá trị từ thanh ghi Status
int get_status() {
   return read_register(ADDR_STATUS);
}

// Hàm ghi giá trị vào thanh ghi A
void set_a(int value) {
   write_register(ADDR_A, value);
}

// Hàm ghi giá trị vào thanh ghi B
void set_b(int value) {
   write_register(ADDR_B, value);
}

// Hàm đọc giá trị từ thanh ghi Signals
int get_signals() {
   return read_register(ADDR_SIGNALS);
}

int main() {
   printf("Traffic Light Controller\n");

   // Bật bộ điều khiển
   set_control(1);
   set_a(0); // Tắt đèn A
   set_b(0); // Bật đèn B
   while(1){
        IOWR(LED_BASE, 0, get_signals());
        usleep(1000);
   }
   return 0;
}
```


DE10-Standard
Đã lưu ảnh chụp màn hình