

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



THIẾT KẾ LUẬN LÝ VỚI VERILOG HDL

Đề tài 5

Door lock

GVHD: Thầy Ngô Đức Minh
SV: Lê Khắc Minh Đăng - MSSV: 1810109
Huỳnh Phúc Khánh - MSSV: 1810226
Nguyễn Trần Quang Minh - MSSV: 1811083

TP. HỒ CHÍ MINH, THÁNG 12/2019



Mục lục

1	Giới thiệu	2
1.1	Đề tài 5: Door lock	2
1.2	Các công cụ được sử dụng	2
2	Thiết kế và hiện thực	4
2.1	Các modules	4
2.2	Mô hình cây các thành phần	4
2.3	Sơ đồ khối các kết nối	5
2.4	Máy trạng thái	5
3	Giả lập và kiểm thử	6
3.1	Giả lập	6
3.2	Kết quả kiểm thử trên board	9
4	Kết luận	12



1 Giới thiệu

1.1 Đề tài 5: Door lock

Đề tài được chọn để hiện thực trong bài báo cáo này là đề tài 5: Door lock. Tổng quát, yêu cầu của đề tài như sau: Hiện thực một mạch khóa cửa có dùng máy trạng thái có những chức năng như sau:

- Mật khẩu là một số thập lục phân ba chữ số từ 0 đến F.
- Các chữ số được nhập vào thông qua các SWs.
- Mỗi lần chỉ có một chữ số được nhập, KEY1 được dùng để xác nhận nhập một chữ số. Chữ số được nhập phải hiện ra trên LED 7 đoạn.
- Mạch hoạt động theo 2 chế độ: Set và Verify. Ở chế độ Set, người dùng sẽ cài đặt mật khẩu. Ở chế độ Verify, người dùng nhập mật khẩu vào và hệ thống sẽ kiểm tra nó. Hai chế độ có thể được chuyển qua lại thông qua KEY2.
- Trong chế độ Verify: Nếu mật khẩu nhập vào là chính xác, đèn LED xanh sẽ chớp nháy với tần số 5 Hz. Nếu mật khẩu là sai, đèn LED đỏ sẽ chớp nháy với tần số 1 Hz. Sau đó, người dùng có thêm 2 lần thử. Nếu trong 2 lần này, người dùng nhập vào mật khẩu đúng thì số lần thử sẽ trở lại như ban đầu. Ngược lại, hệ thống sẽ bị khóa và chỉ có thể hoạt động lại nếu người dùng nhấn nút reset (KEY3)

1.2 Các công cụ được sử dụng

Ngôn ngữ Verilog HDL

Verilog là ngôn ngữ mô tả phần cứng (Hardware Description Language) được sử dụng trong việc thiết kế các hệ thống số, các mạch tích hợp: như bộ nhớ RAM, bộ vi xử lý,... [6]

Verilog lần đầu được giới thiệu vào năm 1984 bởi công ty Gateway Design Automatic. Verilog không được chuẩn hóa và đều được chỉnh sửa ở hầu hết các phiên



bản sau từ năm 1984 đến năm 1990. Năm 1995 Verilog chính thức được chuẩn hóa bởi tổ chức IEEE (Institute of Electrical and Electronics Engineers).

Verilog là ngôn ngữ khá tương đồng với ngôn ngữ lập trình C. Tuy nhiên, nó vẫn có những ưu điểm vượt trội hơn so với C đặc trưng cho việc thiết kế phần cứng như Verilog có thể thực hiện các câu lệnh tuần tự hoặc song song.

Phần cứng: Board DE2i-150

Trong đề tài, nhóm sử dụng Board DE2i-150 của Altera. Bo mạch có cấu tạo bao gồm [2]:

- CPU : Intel® Atom™ Dual Core Processor N2600 (1M Cache, 1.6GHz).
- Altera Cyclone IV 4CX150.
- Hai thanh SSRAM 2MB và hai thanh SDRAM 64MB.
- Bốn nút bấm (KEY0-KEY3).
- 18 Switches.
- 18 LEDs đỏ và 9 LEDs xanh.
- Các LEDs 7 đoạn.
- Một bộ đếm 50MHz cho việc tính clock.
- Các thành phần khác.

Phần mềm: Quartus, ModelSim và draw.io

Quartus là công cụ phần mềm được phát triển bởi Altera, cung cấp môi trường cho các thiết kế SOPC (System On A Programmable Chip). Đây là phần mềm tích hợp đầy đủ phục vụ cho thiết kế logic với các linh kiện logic khả trình PLD (Programmable Logic Device) của Altera, gồm các dòng APEX, CyClone, FLEX, MAX, Stratix,... [3]



ModelSim là chương trình mô phỏng và debug cho các thiết kế ASIC và FPGA rất mạnh mẽ và hiệu quả. ModelSim cung cấp cho người Waveform của mạch FPGA được thiết kế trên Quartus. Từ đó giúp người thiết kế kiểm soát được giá trị output và phát hiện các sai phạm của mạch [4].

draw.io là 1 trang web hỗ trợ công cụ vẽ các mô hình, biểu đồ đơn giản, gọn, nhẹ và miễn phí [5].

2 Thiết kế và hiện thực

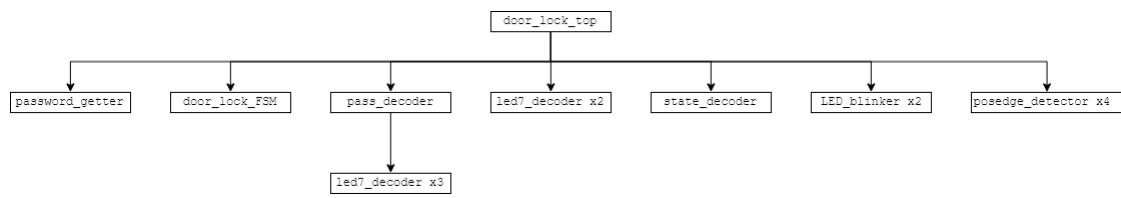
2.1 Các modules

Thiết kế Door lock bao gồm 8 modules:

- *posedge_detector()* dùng để phát hiện cạnh lên cho các KEYS.
- *led7_decoder()* dùng để giải mã tín hiệu hexadecimal ra LED 7 đoạn.
- *pass_decoder()* dùng để hiển thị password 3 chữ số ra LED 7 đoạn.
- *LED_blinker()* dùng để chớp tắt LED theo tần số được cung cấp qua parameter.
- *state_decoder()* dùng để hiển thị trạng thái hiện tại ra LED 7 đoạn (gồm các trạng thái S, V, F).
- *password_getter_sync()* là mạch đồng bộ, dùng để nhận input từng digit một từ người dùng thông qua các SWs và KEYS.
- *door_lock_FSM()* là một máy trạng thái Mealy đồng bộ, dùng để xử lý tất cả logic của thiết kế.
- *door_lock_top()* là module top dùng để kết nối các module nhỏ với nhau.

2.2 Mô hình cây các thành phần

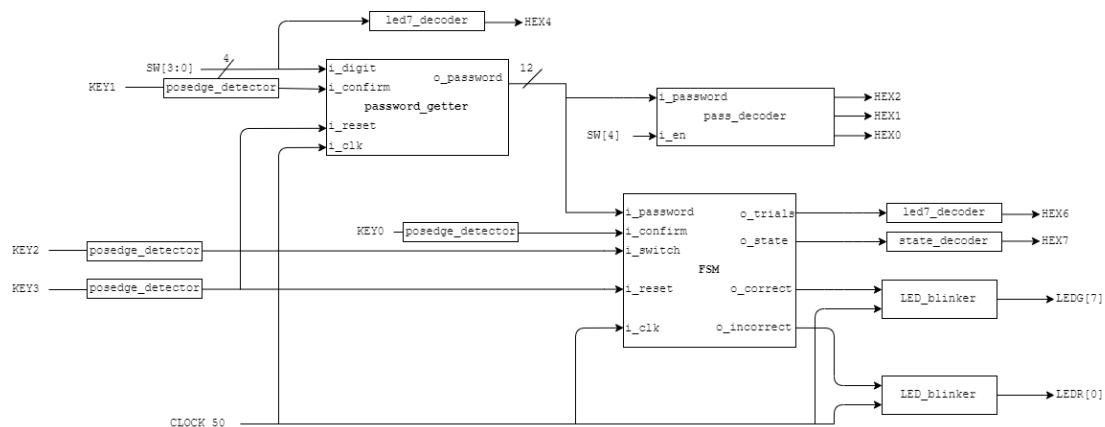
Các thành phần của thiết kế có thể được vẽ lại thành sơ đồ cây như sau:



Hình 1: Door lock hierarchy

2.3 Sơ đồ khái niệm kết nối

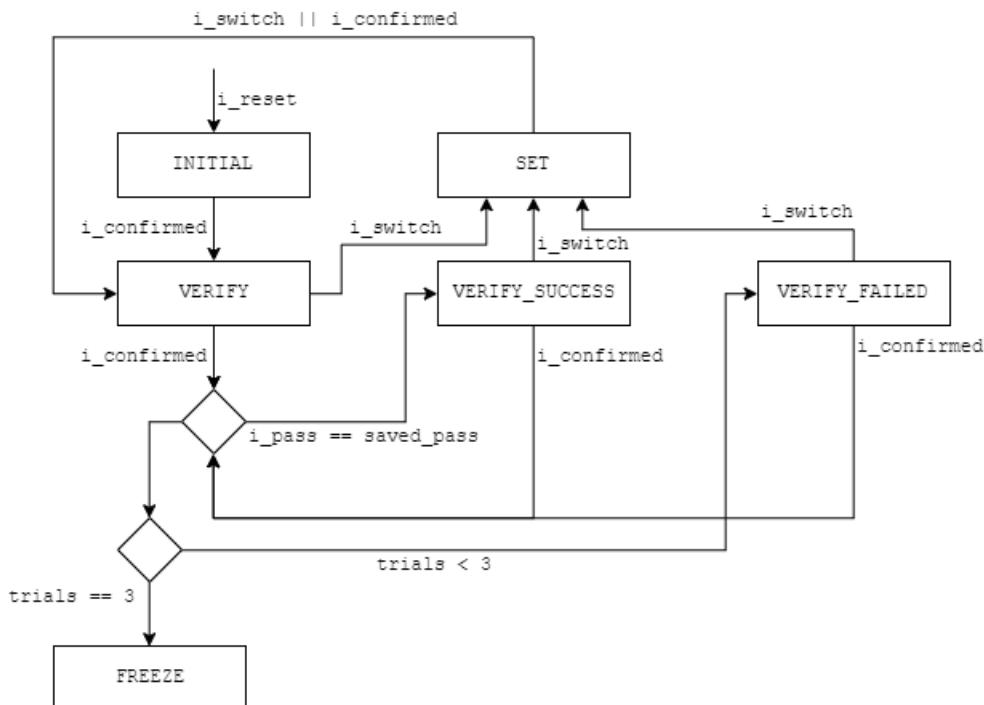
Module *door_lock_top()* có sơ đồ khái niệm kết nối như sau:



Hình 2: Block diagram của *door_lock_top()* và các input từ mạch DE2i-150

2.4 Máy trạng thái

Máy trạng thái được sử dụng trong thiết kế này là một máy trạng thái Mealy đồng bộ. Máy trạng thái hoạt động theo flow graph như sau:



Hình 3: FSM flow graph

3 Giả lập và kiểm thử

3.1 Giả lập

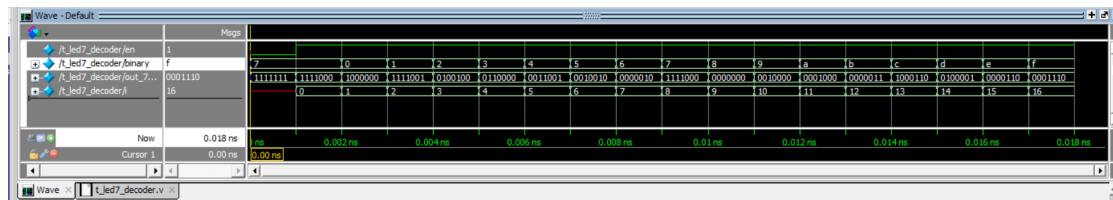
Phần mềm được sử dụng để test các modules là *ModelSim*. Đối với mỗi modules các testcases như sau (có 7 modules được test):

- *led7_decoder()*: đầu tiên, tín hiệu enable = 0, sau đó enable = 1 và tất cả các giá trị số 4 bit từ 0 tới F đều được test.
- *pass_decoder()*: đầu tiên, tín hiệu enable = 0, sau đó enable = 1 và 16 giá trị ngẫu nhiên của password được test.
- *LED_blinker()*: với tần số 5Hz, đầu tiên, tín hiệu enable = 0, sau đó tín hiệu enable được giữ bằng 1 trong một khoảng thời gian.
- *state_decoder()*: tất cả 6 states của máy trạng thái được test.

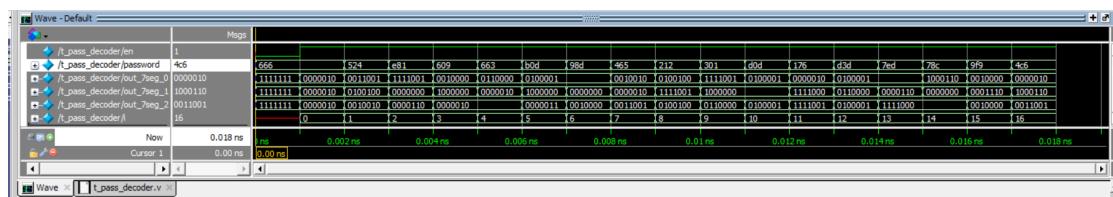


- *password_getter_sync()*: đầu tiên, mạch được reset, sau đó, 5 digits ngẫu nhiên được truyền vào.
- *door_lock_FSM()*: máy trạng thái được test lần lượt như sau: reset -> trạng thái INITIAL -> trạng thái VERIFY với password đúng -> trạng thái VERIFY với password sai 3 lần -> trạng thái FREEZE -> reset -> trạng thái SET -> trạng thái VERIFY với password mới .
- *door_lock_top()*: test tương tự như máy trạng thái.

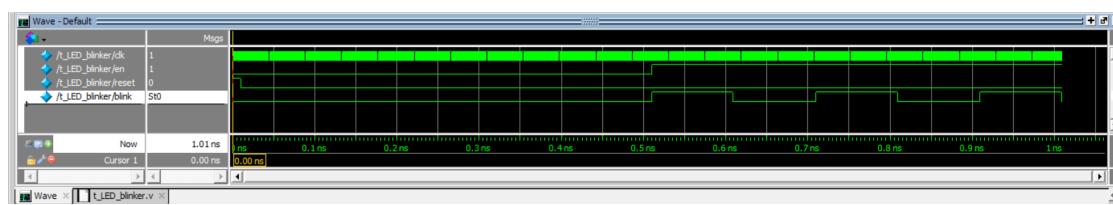
Kết quả khi chạy các testbenches thu được như sau:



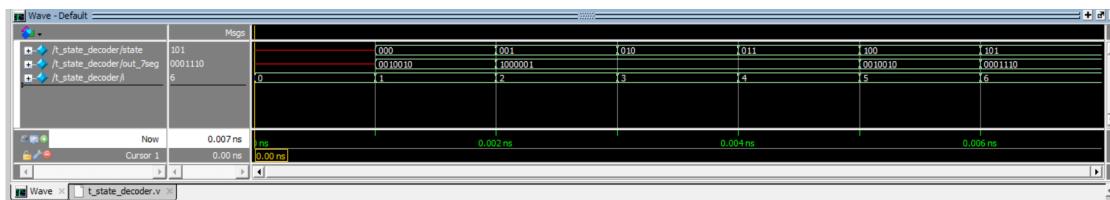
Hình 4: Kết quả test của module *led7_decoder()*



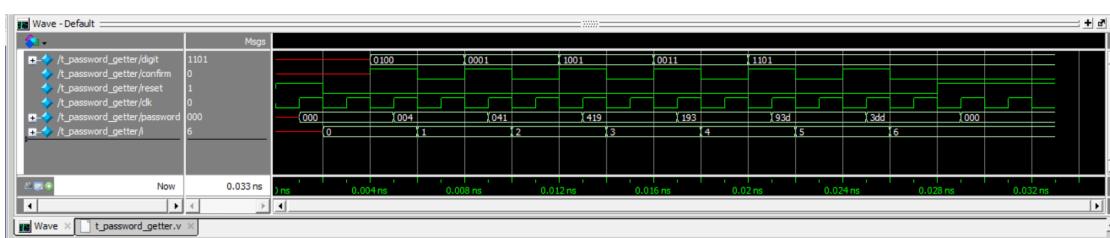
Hình 5: Kết quả test của module *pass_decoder()*



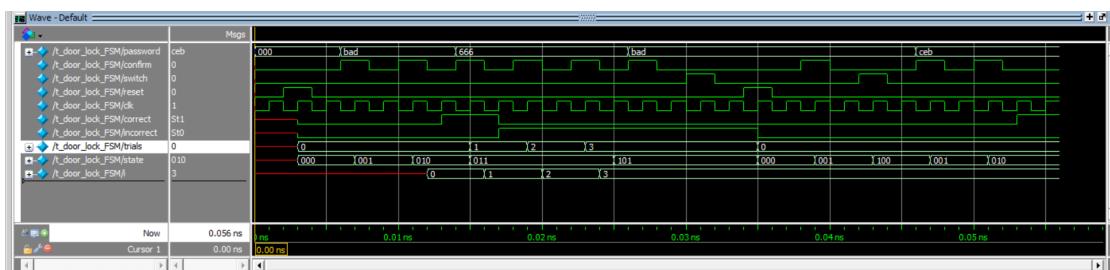
Hình 6: Kết quả test của module *LED_blinker()*



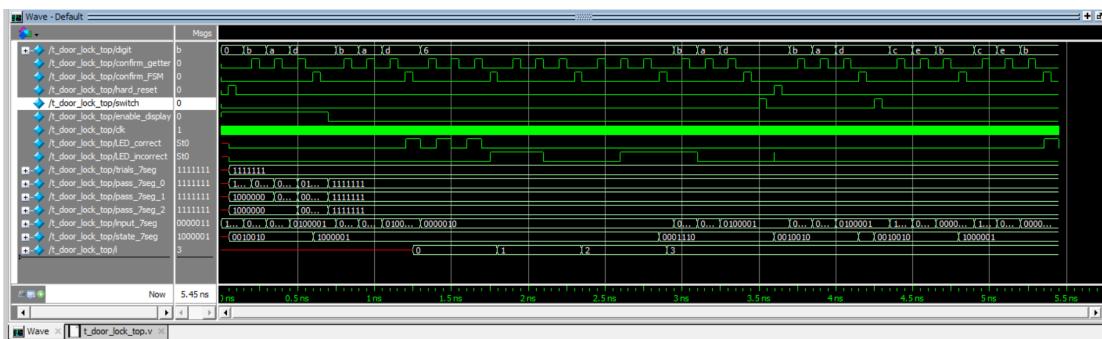
Hình 7: Kết quả test của module *state_decoder()*



Hình 8: Kết quả test của module *password_getter()*



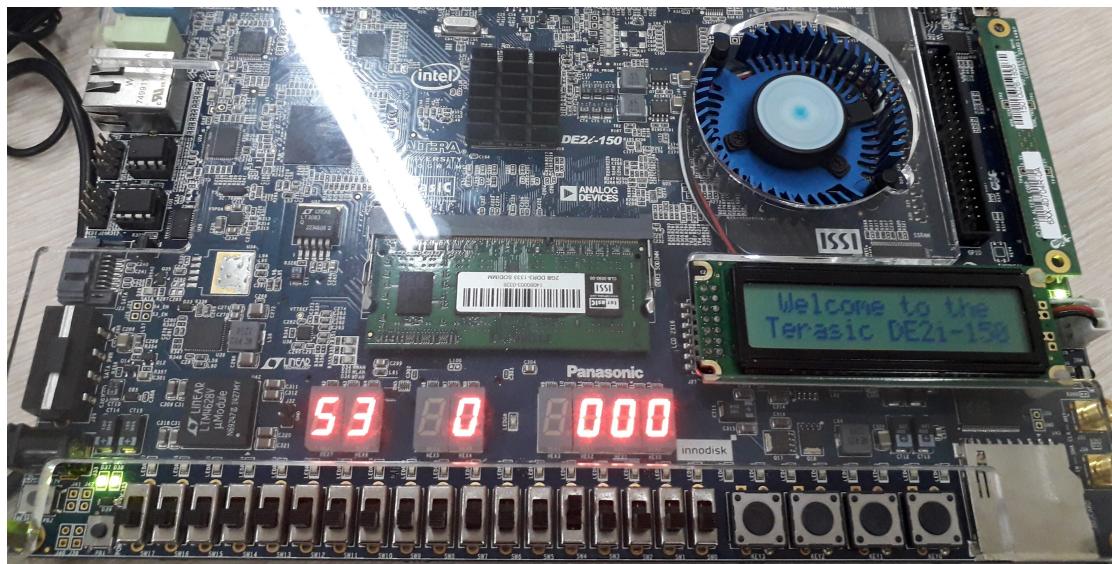
Hình 9: Kết quả test của module *door_lock_FSM()*



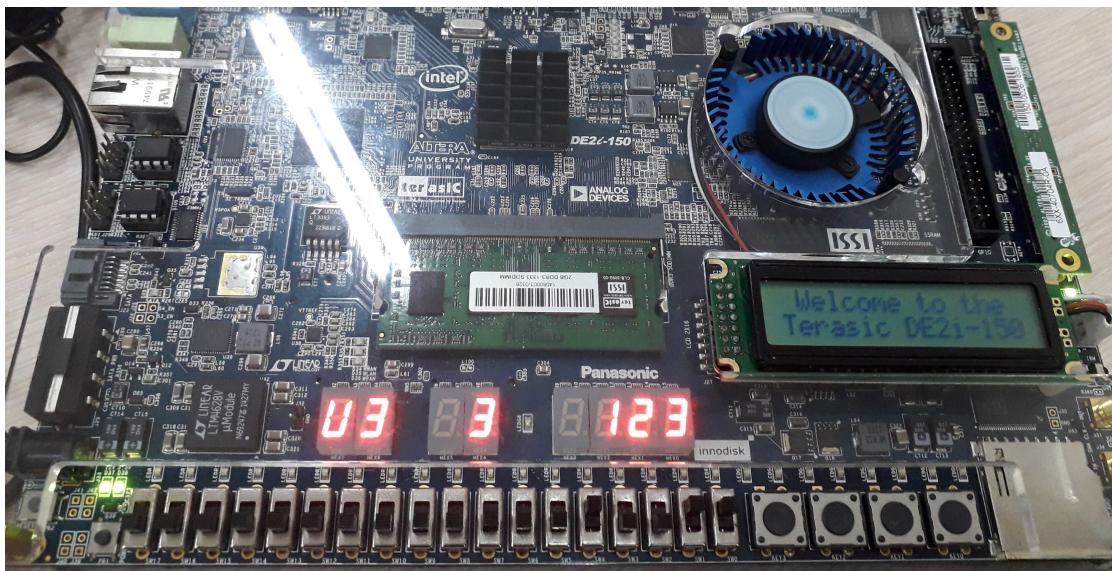
Hình 10: Kết quả test của module *door_lock_top()*

3.2 Kết quả kiểm thử trên board

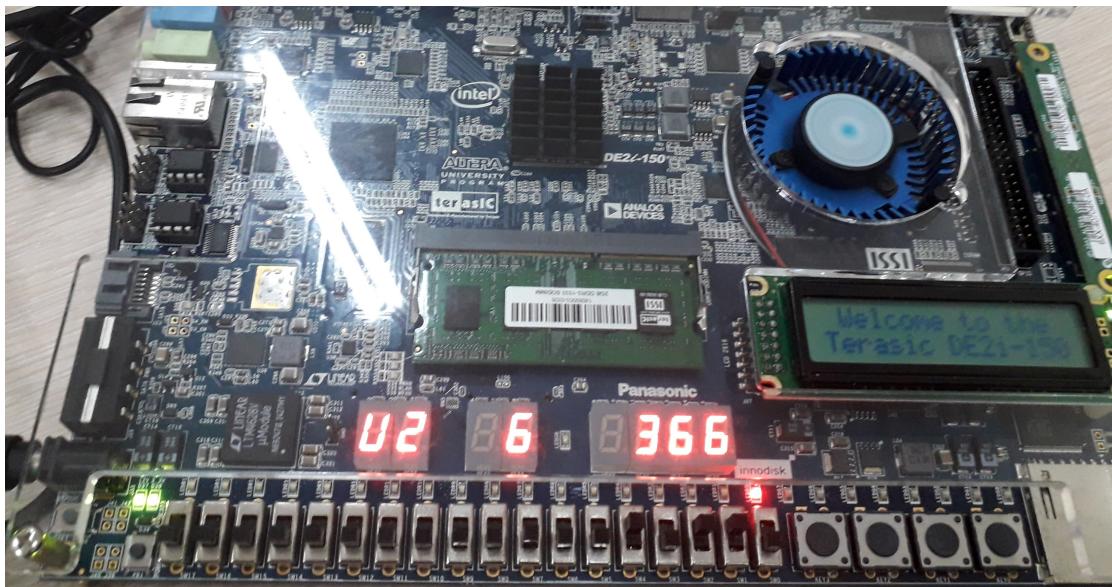
Sau đây là một số ảnh chụp kết quả chạy thử trên board DE2i-150:



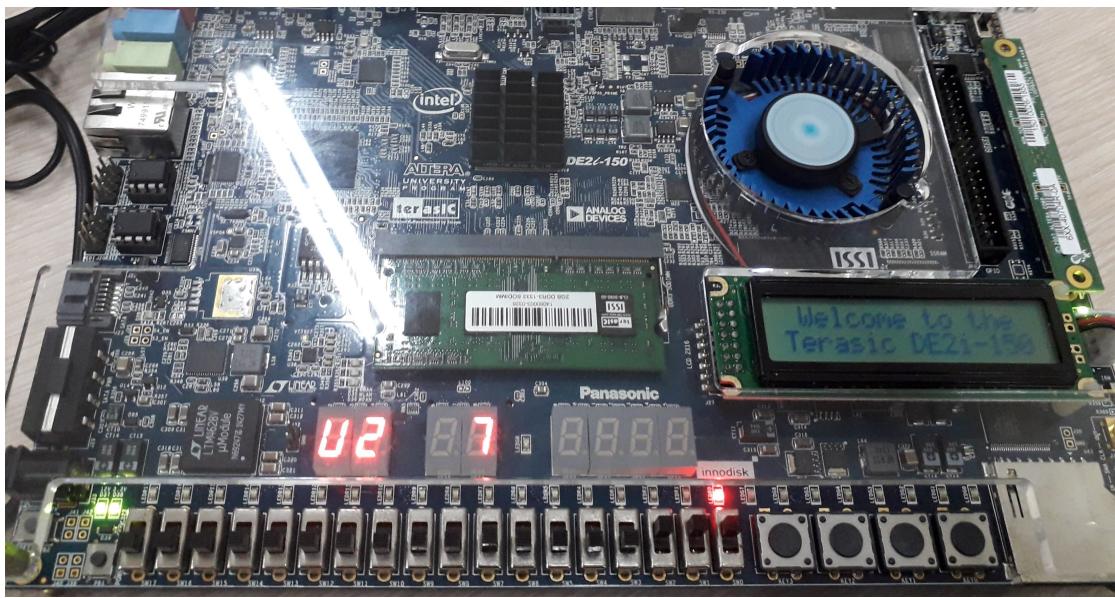
Hình 11: Mạch ở trạng thái SET, tất cả output = 0 sau khi reset



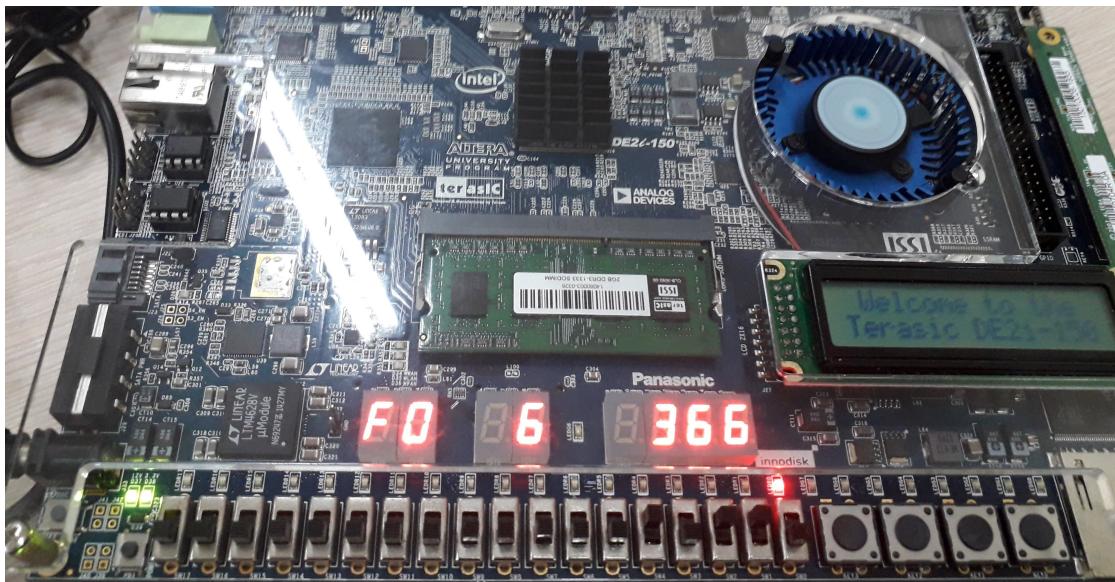
Hình 12: Mạch ở trạng thái VERIFY, password vừa được set là 123



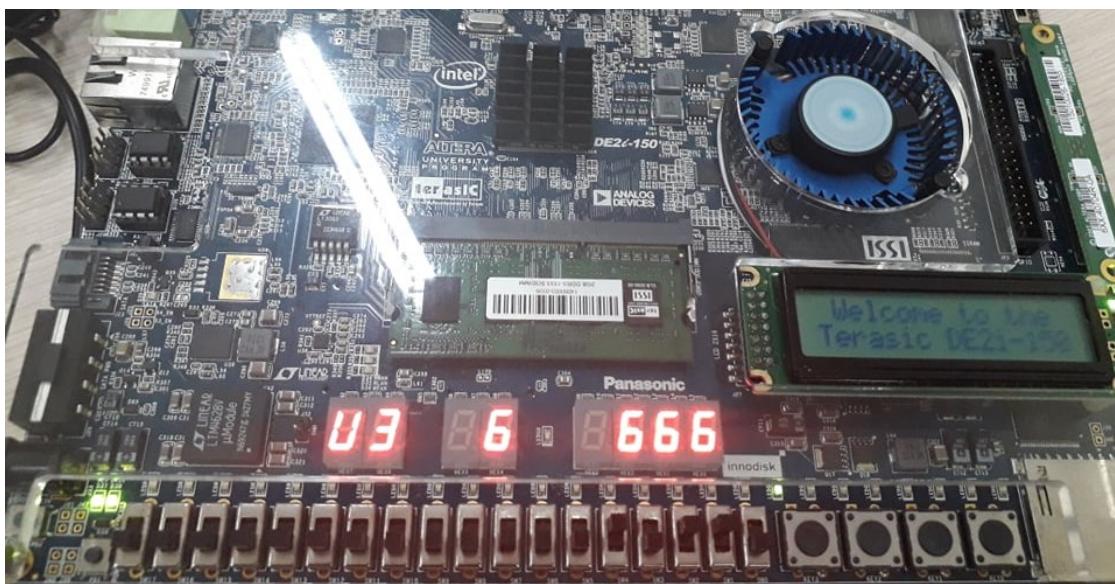
Hình 13: Mạch ở trạng thái VERIFY, 366 là password sai nên LEDR sáng



Hình 14: Mạch ở trạng thái VERIFY, SW[4] = 0, tắt tín hiệu enable của pass_decoder() nên password không được hiển thị



Hình 15: Mạch ở trạng thái FREEZE sau khi nhập password sai 3 lần



Hình 16: Mạch ở trạng thái VERIFY, 666 là password đúng nên LEDG sáng

Ngoài ra, hoạt động của mạch còn có thể được xem thông qua video sau:

<https://youtu.be/6LMBHBavZH4>

4 Kết luận

Tổng kết lại, thiết kế Door lock này có các chức năng sau:

- Nút reset KEY[3] dùng để đưa mạch về trạng thái set password ban đầu.
- Nút confirm_getter KEY[0] cùng với 4 khóa SW[3:0] dùng để đưa một chữ số vào password.
- Nút confirm_FSM KEY[1] dùng để đưa password vào máy trạng thái xử lý.
- Nút switch KEY[2] dùng để chuyển giữa 2 trạng thái SET và VERIFY.
- Khi ở trạng thái set, nếu có tín hiệu confirm_FSM thì password mới sẽ được lưu lại và trạng thái sẽ chuyển sang VERIFY.



- Ở trạng thái VERIFY, nếu có tín hiệu confirm_FSM thì password sẽ được kiểm tra. Nếu đúng, đèn LEDG sẽ chớp với tần số 5Hz, nếu sai, đèn LEDR sẽ chớp với tần số 1 Hz. Nếu sai đến 3 lần thì mạch sẽ sang trạng thái FREEZE.
- Ở trạng thái FREEZE, máy rạng thái hoàn toàn không hoạt động.
- Ngoài ra còn có khóa SW[4] để chọn có hiển thị password ra LED 7 đoạn hay không.
- Các thông số hiển thị ra LED 7 đoạn là: trạng thái hiện tại, số lần thử còn lại, giá trị hiện tại trên SW[3:0], password hiện tại.

Các ưu điểm của thiết kế:

- Dùng máy trạng thái Mealy nên tiết kiệm được số trạng thái.
- Nhận input từng chữ số một giúp giảm số lượng SW cần sử dụng.
- SW[4] có thể chọn có hiển thị password ra hay không.
- Có hiển thị trạng thái hiện tại ra LED 7 đoạn giúp dễ dàng biết được hoạt động của mạch.

Các nhược điểm của thiết kế:

- Máy trạng thái Mealy đồng bộ nên output sẽ bị trễ 1 chu kỳ clock.
- Các nút nhấn có thể bị nhiễu khi nhấn không dứt khoát.

Tài liệu

- [1] M. Morris Mano, Michael D. Ciletti, *Digital System with an Introduction to the Verilog HDL, VHDL, and SystemVerilog*, Pearson Education, Inc, 2017.
- [2] <https://www.intel.com/content/www/us/en/programmable/solutions/partners/partner-profile/terasic-inc-/board/de2i-150-fpga-development-kit.html>



- [3] <https://www.element14.com/community/docs/DOC-40098/1/altera-introduction-to-the-quartus-ii-software>
- [4] <https://www.mentor.com/products/fpga/model/>
- [5] <https://snvn.net/4rum/viewtopic.php?t=80>
- [6] Slide bài giảng Thiết kế luận lý với verilog HDL của Thầy Phạm Quốc Cường, chỉnh sửa bởi Thầy Kiều Đỗ Nguyên Bình.

Lần truy cập cuối cùng của các websites: 7/12/2019.



Phân công công việc:

Lê Khắc Minh Đăng	Thiết kế, code chính, viết báo cáo phần 2, 3, tổng hợp báo cáo
Nguyễn Trần Quang Minh	Hỗ trợ code, viết testbenches, viết báo cáo phần 1
Huỳnh Phúc Khánh	Hỗ trợ code, viết testbenches, viết báo cáo phần 4