

CSI-2108 Digital Logic Laboratory

Fall 2014

Assignment #2: Combinational Door lock Design

Due: December 15th, 2014 at 11:59 PM

마감기한: 2014년 12월 15일 오후 11시 59분

마감기한이 지나면, 과제를 받지 않습니다.

Submission after due date: DO NOT ALLOW

You are to do this assignment **ALONE!**

이 과제는 개인 과제입니다!

1. 과제 내용

- ❖ Quartus II 툴을 사용하여, Combinational Door lock을 FPGA 상에서 구현하기 (Verilog HDL기반)

2. 과제 목적

- ❖ 지금까지 실습/이론 수업에서 다룬 여러 Combinational Logic / Sequential Logic 설계 기법들을 활용하여 복잡한 회로를 설계한다.
- ❖ FPGA상의 여러 가지 입 출력 장치들을 활용하는 기회를 가져보며, 그 동작 특성을 이해한다.

3. 제출 기한

- ❖ 2014년 12월 15일 오후 11:59까지(YSCEC 서버 시간 기준) (마감 기한 이후부터의 과제 제출은 허용하지 않음.)

4. 제출 방법

- ❖ **보고서 파일(pdf, word, hwp)과 Verilog HDL 코드(.v) 파일이 포함된 프로젝트 폴더의 압축파일:** YSCEC 보고서 제출 게시판에 제출
(* .zip 혹은 *.7z, *.tar 계열의 압축만 허용하며, 알집을 통한 압축은 절대적으로 권장하지 않는다.
권장하지 않은 압축 프로그램의 사용 및 압축 형식으로 인해 발생하는 불이익은 본인이 모두 감수해야 한다.)

5. 제한 사항

- ❖ Quartus II 13.0 버전 사용
- ❖ FPGA 키트는 본 실습에서 제공하는 기종을 기준으로 함. (EP2C8F256C8)

6. 유의 사항

- ❖ **1인 1프로젝트이며, 각자 개인의 환경에서 작업해야 한다. 그리고 타인의 코드를 표절하는 행위는 엄중히 징계를 받는다.**
(코드 표절 및 보고서 표절 검사를 위하여, YSCEC의 자체 표절 검사 기능뿐 아니라, **별도의 검사 프로그램을** 활용할 예정)
- ❖ 표절이 적발된 학생들의 해당 프로젝트 점수는 모두 0점 처리
- ❖ 아래 명시된 Verilog HDL 모듈들의 input/output 인터페이스(port list)들은 엄격히 지켜주어야 한다.
- ❖ Unused Pins에 대해서 Reserve all unused pins를 "As input tri-stated"로 설정하시오. (이를 지키지 않을 경우, 막대한 감점이 있음.)

1. 과제에 대한 질문 사항은 가급적 YSCEC 학생 게시판을 활용하십시오!

(본인이 의문을 갖는 사항은 보통 다른 학생들도 마찬가지로 궁금해합니다.)

2. **과제의 세부적인 안내 사항이 수정될 예정이니, 수시로 YSCEC 공지를 확인하십시오!**

3. 의문 사항이 있으면, 개의치 말고 조교에게 제의하길 바랍니다. 효율적으로 과제를 수행하는 길입니다.
(단, 문법 세부사항 및 과제 해결법, 컴파일 문제 해결법 등 본인이 스스로 해결해야 하는 부분은 절대 도와줄 수 없습니다.)

7. 과제 세부 명세

❖ 프로젝트 제출물은 보고서 + 프로젝트 폴더 전체를 압축한 파일임

1) YSCEC 보고서 게시판에는 1. 프로젝트 보고서와 2. 프로젝트 폴더 전체를 한 폴더에 담아서,
zip/7z/tar 계열의 포맷으로 압축해서 내야 함. (압축 해제 오류 시 0점 처리. 위의 4. 제출 방법의 안내를 참조)

1. 프로젝트 보고서

❖ 보고서의 경우 프로젝트에서 수행한 Verilog HDL 작성 및 FPGA 프로그램 과정에 대해서 상세하게 서술한다.

❖ 다음과 같은 항목을 위주로 서술한다.

1) 프로젝트 개요 - 프로젝트 설명 및 개요

2) 블록 다이어그램 - Combinational Door lock 내부의 모듈 및 net의 연결 상태 등에 대해 묘사하는 그림
(일반적인/보편적인 블록 다이어그램의 작성법에 의거하여 작성요망. 형식이 엉망이거나 부실할 경우 감점 요인이 됨.)

3) FSM 상태 전이도 다이어그램 - 본 프로젝트를 수행하는 과정 중에 설계된 FSM에 대해서 명확히 알 수 있도록 FSM 다이어그램을 작성한다.

4) 모듈 별 동작 설명 - 본인이 작성한 Verilog HDL 코드 중 중요하다고 판단되는 부분들을 발췌하여, 코드와 함께 동작 원리 및 부가 설명 제시. (본인이 작성한 코드라는 것을 충분히 설명할 수 있을 수준으로 제시해야 함.)

5) 시뮬레이션 및 FPGA 동작 결과 - 동작 결과에 대해서 서술하거나, 시뮬레이션을 통해 산출된 Waveform을 기반으로 설계가 올바르게 동작함을 검증. 하지만 주 목적은 FPGA에 program을 성공적으로 완료하는 것에 있다.

6) 결과 분석 및 토의, 고찰할 점 등 - 5)의 결과와 연계하여, 추가적으로 고찰할 점들이나, 프로젝트 수행 과정 중의 여러 이슈들에 대해 서술한다.

7) 시연 동영상 - 해당 시연 동영상은 보고서 내부가 아닌, 보고서 외부의 별도의 파일 형태로 첨부해서 제출한다. FPGA에 program한 후 올바르게 동작하는지 확인하는 과정을 담는다.

2. 프로젝트 - Verilog HDL 작성 / Simulation (optional) / Pin Planning / FPGA Programming

❖ 컴파일 불가, FPGA 보드 업로드 불가 등과 같은 문제가 발생시 [2. 프로젝트] 항목은 0점 처리

❖ 프로젝트 폴더를 압축해서 해제했을 시 정상적으로 컴파일이 가능해야 하며, Pin 할당이 완료된 상태여야 함. (업로드 전 재검토 요망)

❖ 본 프로젝트에서는 Modeling 기법에 따른 차등 점을 두지 않으니, 본인이 원하는 Modeling 기법을 택하여 수행해도 무방하다.

❖ 단, 전체 프로젝트를 하나의 모듈로만 구성해서 설계하는 것은 치명적인 감점요인이 된다.

❖ 즉, 다음과 같이 기능에 따라 분리한 체계적인 형태의 구조로 프로젝트를 설계해야 한다.

(아래에서 볼 수 있는 권장하는 프로젝트 구성 요소는 한 예에 불과 하다. 그리고 각 모듈의 기능 및 작동 원리들에 대해서는 주석 및 보고서에 충분히 명시해야 한다. 기능에 따라 모듈의 분리를 수행하지 않고, 단순히 감점을 피하기 위해서 Verilog 코드 파일을 의미 없이 나누는 행위는 허용되지 않는다.)

<권장하는 프로젝트 구성 요소 (예시)>

[Combinational Door lock (main module)].v

[Keypad module].v

[Frequency Divider / clock generator].v

[Sound Generator with buzzer].v

[LED output controller].v

[Main FSM].v

❖ 요구되는 주요 기능

[사용자의 암호 설정 기능]

- 1) 3~6자리까지 사용자가 원하는 자릿수의 암호를 설정 가능 (기존 암호를 입력해서 인증을 받은 뒤, 설정/수정이 가능)
- 2) 사용자가 설정한 암호가 12345, 11111, 222, 321과 같이 입력 자릿수에 대해서 모두 같은 숫자가 반복되거나, 자릿수 전부가 1씩 증가/감소 하는 패턴을 보일 경우, 다시 암호를 입력하도록 강제한다.

[보안 기능]

- 3) 사용자가 1)에서 설정한 올바른 암호가 입력되었을 경우 lock을 해제하고, 일정 시간 후 문이 닫힌 것으로 감지되면 다시 잠금 기능을 수행한다. (문의 열림/닫힘 여부는 Push button을 통해 구별한다.)
- 4) 일정 시간 범위 내에서 암호가 3번 틀린 경우, 경고음과 함께 Door lock의 작동이 1분간 멈춘다. (잠금 상태 유지)
- 5) 2)의 상황이 5분 이내에 3회 이상 반복될 경우 긴 경고음과 함께, 보안 회사에 연락이 간다. (LED로 해당 기능 작동 여부 표시)
- 6) 오랜 시간 동안 사용자가 한 암호만을 사용할 경우, 암호 변경 권고를 위해 알림 음을 울린다. 알림 음은 일정 시간 간격으로 사용자가 암호를 변경할 때까지 울린다. (현실적인 환경에서는 2개월 주기로 암호 변경 여부를 확인해야 하나, 본 프로젝트에서는 채점의 현실성을 위해 해당 간격을 다소 짧게 설정한다.)

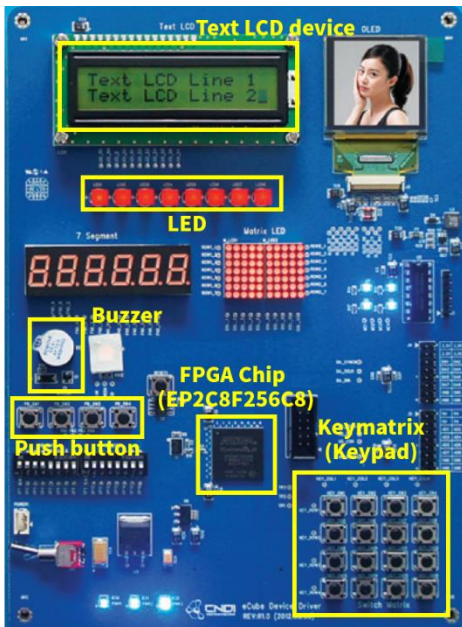
[기타 동작 관련]

- 7) Key matrix의 버튼을 누를 때 마다 buzzer 음이 울림
- 8) 잠금 및 잠금 해제 시 이 둘을 구분 할 수 있는 고유의 buzzer 음이 울림
- 9) 처음 전원을 공급할 시(reset 상태)에는 1)에서 언급된 암호 설정 모드로 작동하나, 기존 암호를 입력할 필요는 없음

[분석 기능 (추가 구현 점수 할당)]

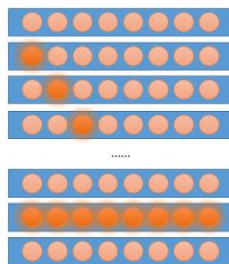
- 10) 7-Segment 장비를 통해 '문 열림', '잠금' 상태를 표시하고, 사용자가 입력한 암호를 표시
(문 열림: OPEN, 문 닫힘: CLOSE, 암호: 왼쪽부터 입력한 순서대로 표시)

❖ FPGA 상에서의 입 출력 디바이스 사용 예



- 1) Text LCD device 및 Buzzer의 작동 방식에 대해서는 추가 자료를 통해 안내 예정.

- 2) 위의 [보안 기능] 5)의 상황이 발생시 왼쪽 LED가 왼쪽에서 오른쪽 끝까지 차례로 깜빡이며, 다시 오른쪽에서 왼쪽으로 같은 동작을 반복한 뒤, 8개 전부가 한꺼번에 꺼졌다가 켜진 뒤 다시 꺼진다.



[Figure 1] LED 경고등

1	2	3	
4	5	6	
7	8	9	
*	0	#	

[Figure 2] Key matrix 할당

- 3) Push button 중 하나는 reset 상태를 결정하며, 나머지 하나는 문 열림/닫힘 상태를 알려주는 센서 역할을 한다.
- 4) Key matrix는 위의 그림과 같이 사용된다.

❖ 동작 시나리오

- 1) 전원을 넣었을 시, 혹은 Reset 버튼을 눌렀을 경우, 초기 암호 설정 모드로 들어간다. (잠금이 풀려있는 상태)
 - A. 초기 암호 설정 모드에서는, 암호를 변경(설정)하기 위해서 기존 암호를 입력하는 절차가 필요 하지 않다.
 - B. 암호가 다음의 두 경우와 같이 보안에 취약할 경우, 사용자에게 다시 암호를 설정하게 한다.
 - i. 22222, 555555와 같이 모든 암호의 자릿수가 한 숫자로만 구성된 경우

- ii. 123456, 876과 같이 암호 전체의 구성이 1씩 증가하거나 1씩 감소하는 패턴일 경우
 - C. 새로운 암호 입력의 종료는 '*' 키로 구별한다.
 - D. 정상적으로 암호가 설정된 경우, 디바이스는 '잠금' 상태로 전환된다.
- 2) 초기 암호 설정이 이루어진 이후부터는 '문 잠금' 상태가 유지되며, Key matrix의 '*'를 누르면 암호 입력 상태로 전환된다.
- A. 암호 입력 상태에서는 최대 6자리의 숫자를 입력 받는다.
 - i. '*'이 아닌, 7번째 숫자를 누르는 순간에는 잘못된 암호 입력을 했을 경우와 같이 취급된다.
 - B. 사용자는 암호를 입력한 뒤, '*'키를 누르는 것을 통해 입력이 끝났음을 알린다.
 - C. 디바이스의 판단에 따라, 사용자가 올바른 암호를 입력한 것으로 검증이 되면 '잠금 해제' 상태로 전환한다.
- 3) '잠금 해제' 상태 이후, '문 열림' 상태인지를 판단하고, '문 닫힘' 상태로 판명이 된다면 일정 시간 이후, '잠금'을 시도한다.
- A. 여기서 '문 열림' 상태를 판단하는 센서의 신호는 위 FPGA 이미지에서 Push button을 통한 입력으로 대체한다.
 - i. 여기서는 편의를 위해서, 해당 버튼을 누르고 있을 때를 '문 열림' 상태로 취급한다.
- 4) 초기 암호 설정이 이루어진 이후, 다시 암호를 설정할 경우에는 다음과 같은 과정을 따른다.
- A. 1)의 B~D의 과정과 같이 수행한다.
 - B. 다만, A에서 언급한 과정을 거치기 전에, 우선적으로 '기존 암호'가 무엇인지 사용자에게 검증을 받는 과정을 따로 거친다.
 - i. 해당 과정에서 올바른 암호를 입력하지 않았을 경우, '잠금 해제' 상태로 돌아간다.
 - C. 암호 재설정을 위한 키는 '#' 키를 사용하는 것으로 한다.
 - i. '#'키를 누르면 '암호 재설정' 모드로 전환되며, 사용자가 암호를 입력한 뒤 최종적으로 '#'을 누르면 입력이 완료된다.
 - ii. 올바른 암호 입력이 이루어진다면, Key matrix의 key를 통해 새로운 암호를 입력 받는다.

❖ Key Matrix 시뮬레이션에 대한 사항 (중요!)

- 1) Key Matrix의 작동은 해당 입력 HW에 의존적인 형태라, 시뮬레이션의 부분은 다음과 같이 해도 무방함.
- A. 해당 모듈을 비워두고, Testbench 상에서 시간 흐름에 따라 입력을 직접 변화시키면서 시뮬레이션 해도 무방함.
예) #100 output_keypad = 8; #100 output_keypad = 5'bxxxxx; // output_keypad가 어떤 키를 눌렀는지 알림. 여기서는 8번
 - i. Simulation 과정에서는 A의 방법을 적극 권장함.
 - B. Key Matrix의 실제 사용성을 테스트해보고 싶다면, 다음과 같이 구현해도 무방함 (단, 해당 방법은 난이도가 있음. 따라서 권장하는 방법은 아님.)
 - i. Key Matrix에 들어가는 output은 해당 line의 key 버튼을 사용할 수 있게 활성화 시키는 기능을 함.
 - ii. 따라서, module keymatrix_hardware()와 같은 모듈을 별개로 제작하여, line에 대한 output이 1인 input에 대해서만 버튼을 enable 시킬 필요가 있음.
 - iii. 장점: FPGA 기기가 없더라도, 최대한 실제에 가까운 동작성을 확인 가능함.
단점: 실제 FPGA 작동에 연관이 없음에도, 따로 Hardware를 모사하는 모듈을 제작해야 함. 난이도 상승
- 2) 단, 실제 FPGA 상에서 구동 시에는 정상 작동하는 방향으로 해야 함

8. 참고 사항

- ❖ Wiki 및 여타 Web 자료들을 활용 시, reference를 명확히 밝힐 것. (학생간 표절 의심 발생시, 명확히 해소할 수 있는 근거가 됨.)
- ❖ Quartus II를 사용해야 함. (시뮬레이션 수행의 경우 Modelsim-Altera 사용)
- ❖ 본 프로젝트의 수행을 과제 대행 서비스 등에 의뢰하거나, 기타 관련 분야 포럼 등에 의뢰하는 행위는 엄중한 징계가 따름.

9. 제출 관련

❖ YSCEC 제출 시점을 기준으로 제출 기한을 판단함.

❖ 프로젝트 파일 및 보고서를 업로드 할 시, 보편적으로 성능을 인정받은 압축 프로그램들을 사용할 것.
(위에서 언급된 제출 유의 사항을 참고하십시오.)

❖ Verilog HDL 코드 파일 (*.v)에는 반드시 충분한 만큼의 주석을 달 것. 본인이 작성한 코드임을 충분히, 명확히 보여주어야 함.

❖ 언어 문제로 인하여 본 과제의 명세를 명확히 파악하기 힘들 경우, 메일을 통한 질문 및 YSCEC 게시판을 적극 활용할 것.

10. 참고 자료

❖ 이론 교재 (Contemporary Logic Design) 및 실습 교재의 FSM관련 항목, 그리고 실습 슬라이드들을 적극 활용 하시오.