REPORT

텀프로젝트 최종 보고서

과목명: 논리 회로 설계 및 실험

담당 교수: 권동현 교수

제출일: 2024-12-23

학번 및 이름: 202170116 윤민석, 202355576 장대규(8조)

목 차

제출일 : 2024-12-23; 프로젝트 최종 보고서
이름 : 윤민석(202170116) 장대규(202355576) (8조)

✔ 제안서 발표 시의 주제 및 구현 계획

● 주제 및 기능

주제

이 프로젝트의 주제는 기존에 연필과 펜으로 진행하던 숫자 게임을 FPGA보드를 통해 기계적으로 구현하는 것. 이 게임은 난수를 생성하는 회로를 만들고, 그 회로를 응용하여 3자리 또는 4자리 숫자를 생성한 뒤 사용자가 해당 숫자를 맞추도록 하는 구조임.

● 구현 내용 및 방법

난수생성 회로를 구성하여 FPGA 상에서 무작위 숫자를 만드는 기능을 구현. 버튼을 통해 사용자가 입력한 숫자를 처리하고, 음악 재생과 7-세그먼트·LED 표시는 기존 과제 회로를 응용하여 연동.

8 array 7-segment에서 왼쪽 4비트는 S0B0, 오른쪽 4비트는 0000으로 초기화.

버튼으로 3자리 또는 4자리 숫자를 입력.

입력 후 엔터(예: 0번 버튼)를 누르면 왼쪽 4비트에 결과가 표시.

9회 안에 맞추지 못하면 Full Color LED에 빨간 불이 들어옴.

숫자를 맞추면 초록 불이 들어옴.

결과에 맞는 텍스트를 16×2 Text LCD에 출력.

결과에 알맞은 음악이 재생.

✔ 주제 및 구현 계획의 변경된 부분

1. 왼쪽 4비트에 'S0B0'을 표시하는 대신 정답 난수 4자리를 표시하도록 변경(스트라이크, 볼이 제대로 계산되는지 확인하기 위함).

2. 스트라이크와 볼은 'STRIKE 0 BALL 0' 형식으로 Text LCD에 표시.

3. 9회가 아닌 무제한 시도 가능(9회가 종료되었을 때, 불이 들어오는 대신, 매 회마다 LED로 정답 또는 오답을 알림).

4. 결과에 따른 음악 재생 구현하지 않음.

구현된 기능에 대한 상세한 설명

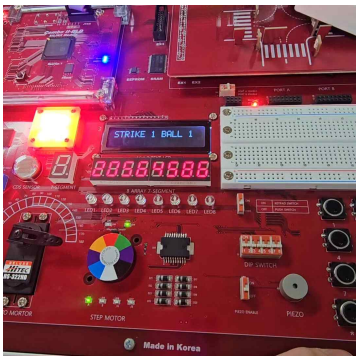
1. 처음 프로그램을 보드에 올렸을 때, 8 array 7-segment는 모두 0, Text LCD에는 'STRIKE 4 BALL 0'이 표시되고 LED에는 초록 불이 들어옴.

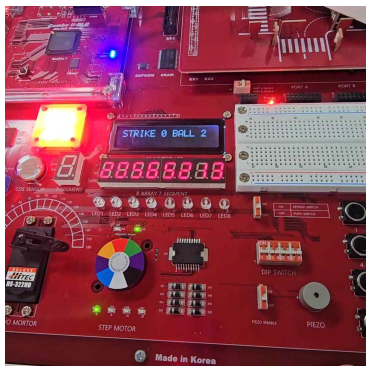
2. *(게임 시작) 버튼을 누르면 8 array 7-segment 상위 4비트에 정답 난수 값이 표시됨.

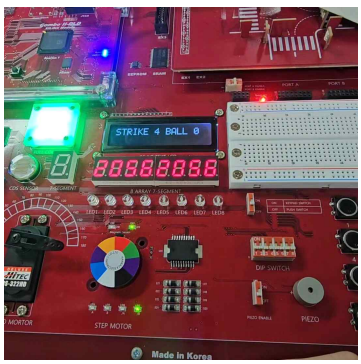
3. 사용자가 4자리 숫자를 입력하면 8 array 7-segment 하위 4비트에 표시되며, #(제출) 버튼을 누르면, 정답 값과 입력 값을 비교 후 스트라이크와 볼의 개수가 Text LCD에 표시됨.

4. 만약, 정답 값과 일치할 경우 LED에 초록 불이, 그렇지 않을 경우 빨간 불이 들어옴.

5. 새로운 게임을 시도하고 싶다면 다시 * 버튼을 누르면 됨. 이후 진행 방식은 동일함.

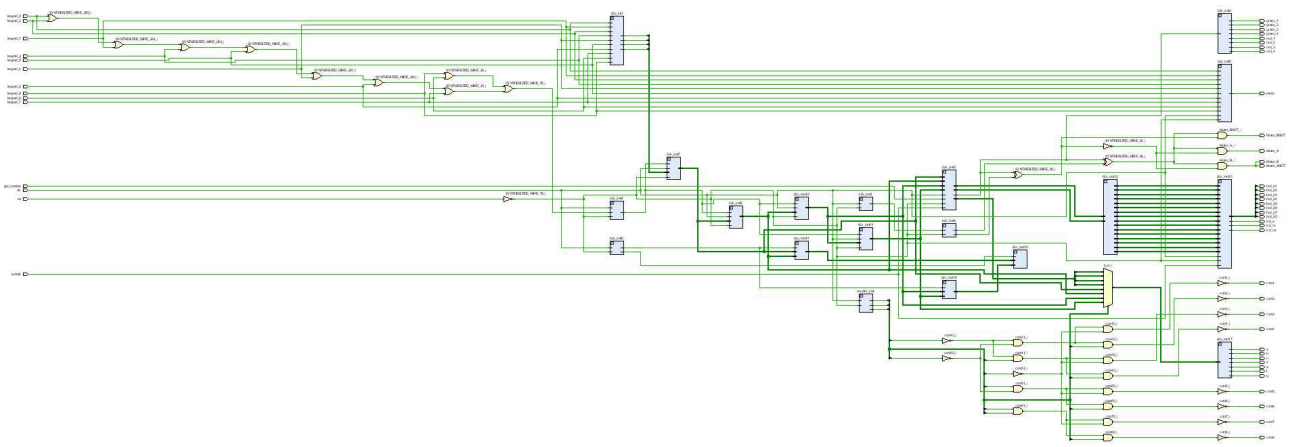






회로도 첨부 및 모듈 별 기능 설명

● 전체 회로도



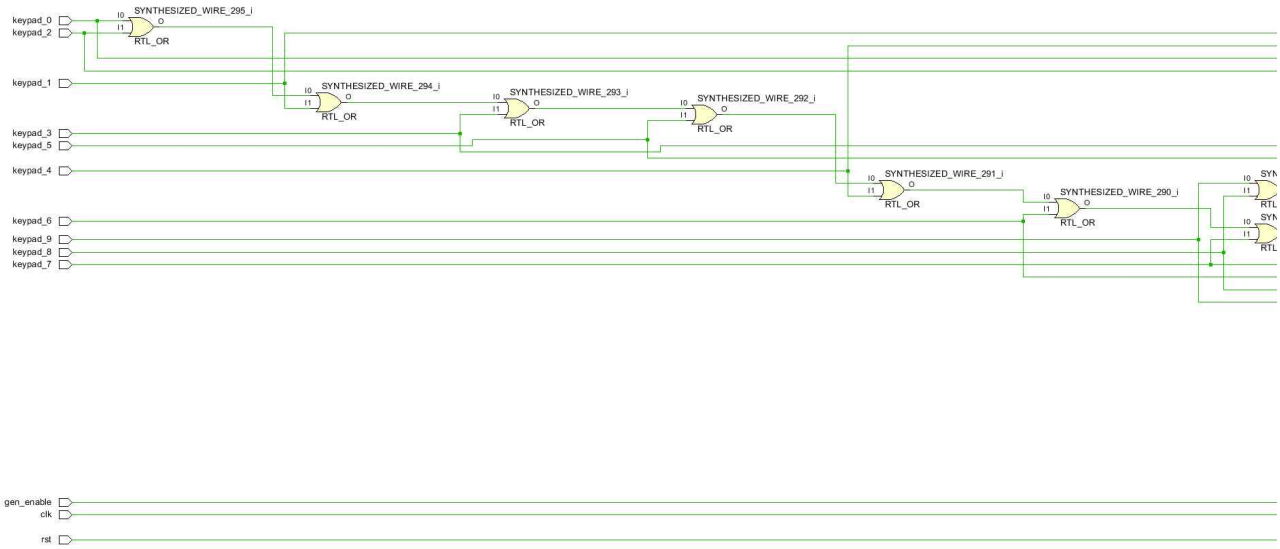
● 부분 회로도(6분할 확대)

<1> | <2> | <3>

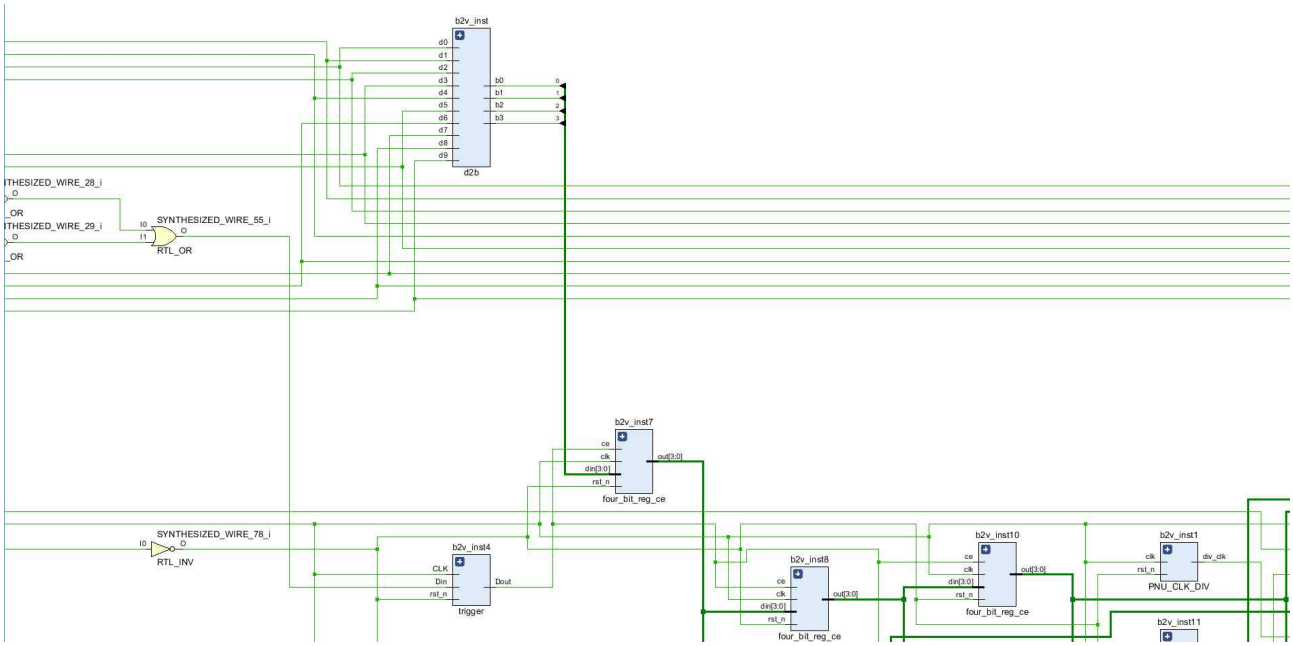
<4> | <5> | <6>

위와 같이 나누어 자름.

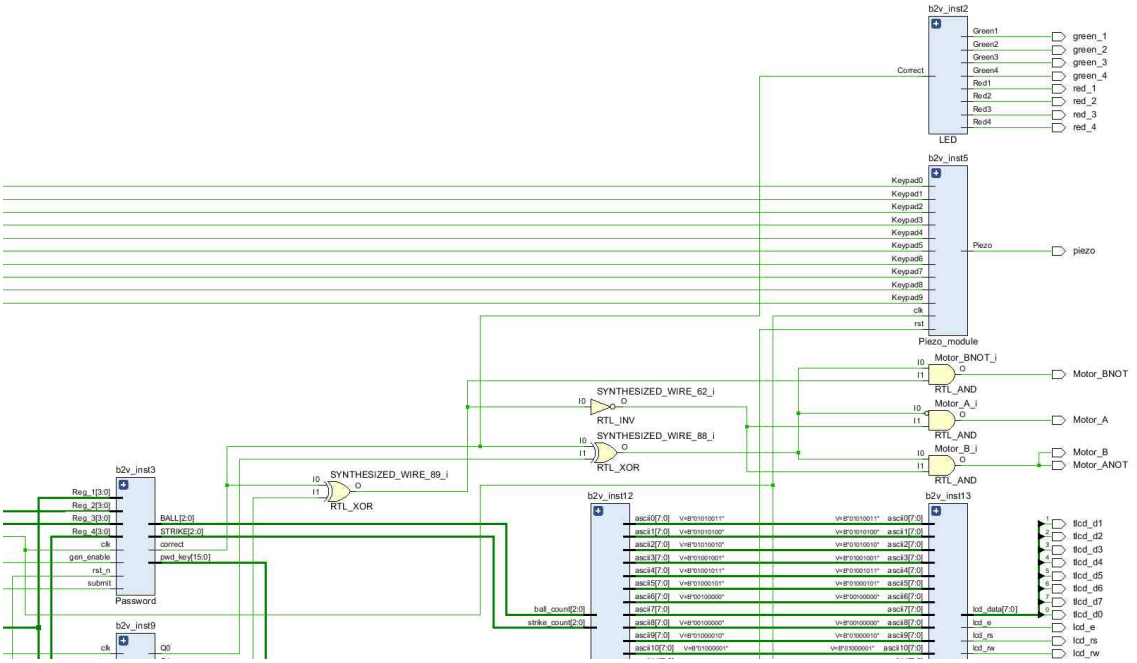
<1>


keypad_0
keypad_2
keypad_1
keypad_3
keypad_5
keypad_4
keypad_6
keypad_9
keypad_8
keypad_7
SYNTHESIZED_WIRE_295_i
SYNTHESIZED_WIRE_294_i
SYNTHESIZED_WIRE_293_i
SYNTHESIZED_WIRE_292_i
SYNTHESIZED_WIRE_291_i
SYNTHESIZED_WIRE_290_i
RTL_OR
gen_enable
clk
rst

<2>


b2v_inst
d2b
SYNTHESIZED_WIRE_55_i
SYNTHESIZED_WIRE_78_i
RTL_OR
RTL_INV
b2v_inst7
b2v_inst8
b2v_inst10
b2v_inst1
b2v_inst4
b2v_inst11
four_bit_reg_ce
PNU_CLK_DIV
trigger

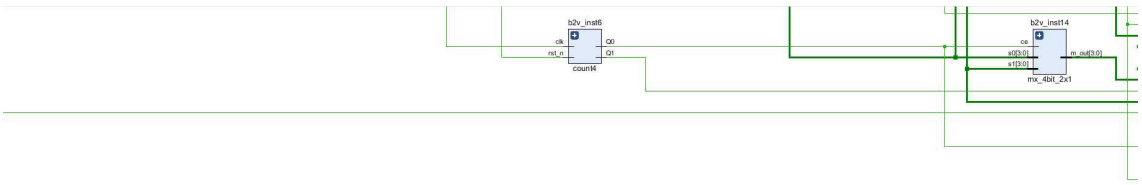
<3>


b2v_inst2
Green1
Green2
Green3
Green4
Red1
Red2
Red3
Red4
Correct
green_1
green_2
green_3
green_4
red_1
red_2
red_3
red_4
LED
b2v_inst5
Keypad0
Keypad1
Keypad2
Keypad3
Keypad4
Keypad5
Keypad6
Keypad7
Keypad8
Keypad9
clk
rst
Piezo
piezo
Piezo_module
Motor_BNOT_i
RTL_AND
Motor_BNOT
Motor_A_i
RTL_AND
Motor_A
Motor_B_i
RTL_AND
Motor_B
Motor_ANOT
SYNTHESIZED_WIRE_62_i
RTL_INV
SYNTHESIZED_WIRE_88_i
RTL_XOR
SYNTHESIZED_WIRE_89_i
RTL_XOR
b2v_inst3
Reg_1[3:0]
Reg_2[3:0]
Reg_3[3:0]
Reg_4[3:0]
clk
gen_enable
rst_n
submit
BALL[2:0]
STRIKE[2:0]
correct
pwd_key[15:0]
Password
b2v_inst9
clk
Q0
b2v_inst12
ball_count[2:0]
strike_count[2:0]
b2v_inst13
lcd_data[7:0]
lcd_e
lcd_rs
lcd_rw
lcd_d1
lcd_d2
lcd_d3
lcd_d4
lcd_d5
lcd_d6
lcd_d7
lcd_d0

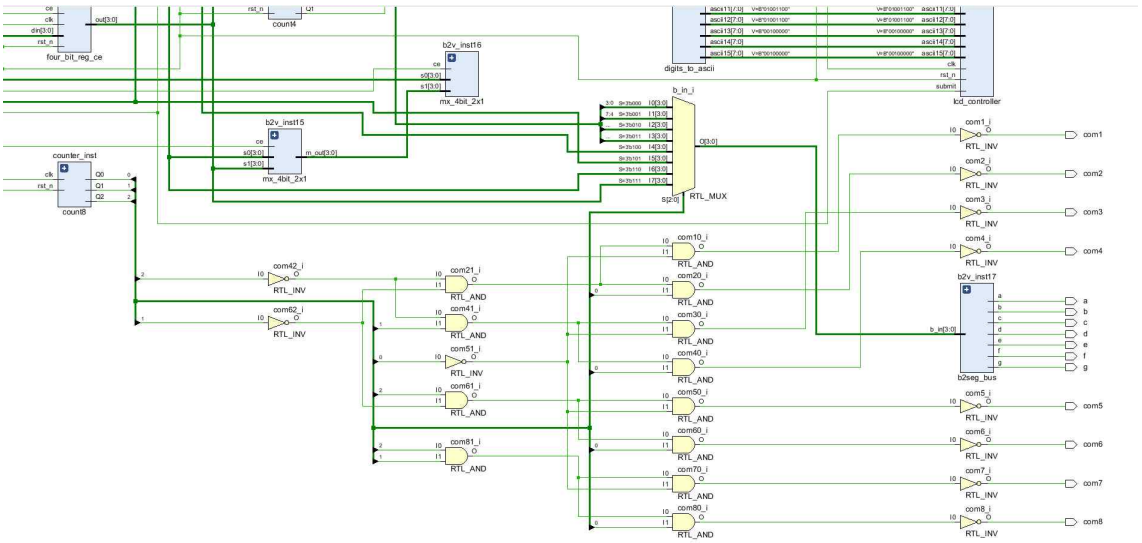
<4>

submit

<5>

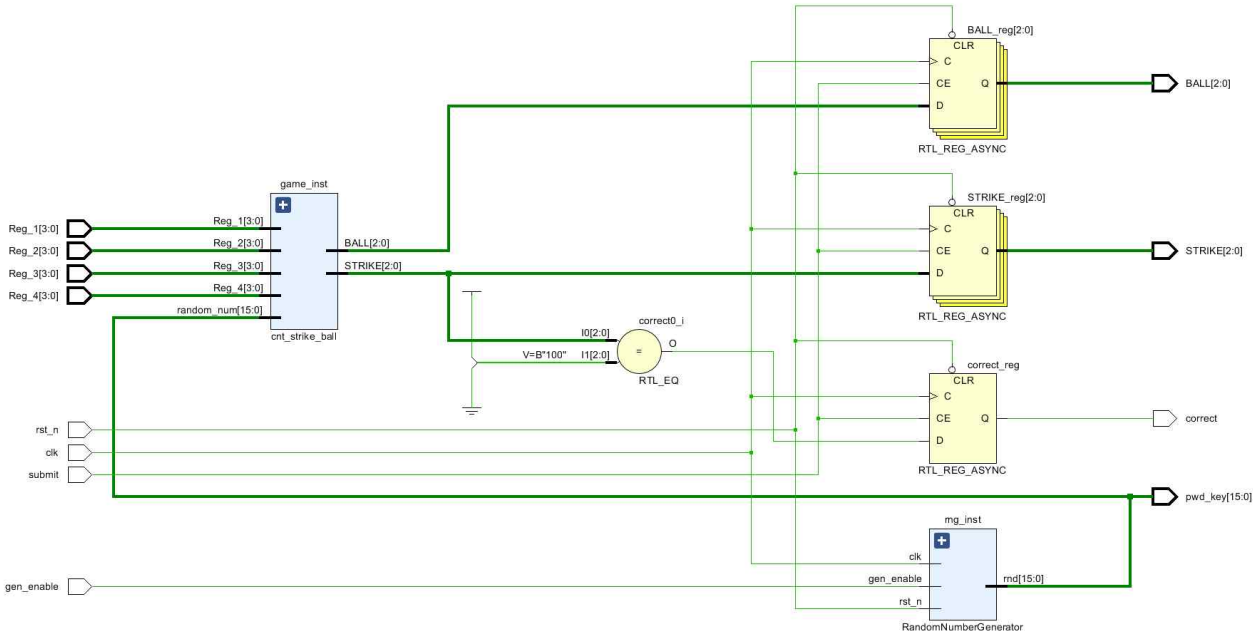

b2v_inst6
clk
rst_n
Q0
Q1
count4
b2v_inst14
ce
s0[3:0]
s1[3:0]
m_out[3:0]
mx_4bit_2x1

<6>


four_bit_reg_ce
count4
b2v_inst16
mx_4bit_2x1
digits_to_ascii
lcd_controller
b_in_i
RTL_MUX
b2v_inst15
counter_inst
count8
com1_i
com2_i
com3_i
com4_i
com5_i
com6_i
com7_i
com8_i
RTL_INV
com10_i
com20_i
com30_i
com40_i
com50_i
com60_i
com70_i
com80_i
RTL_AND
com42_i
com62_i
com21_i
com41_i
com51_i
com61_i
com81_i
b2v_inst17
b2seg_bus

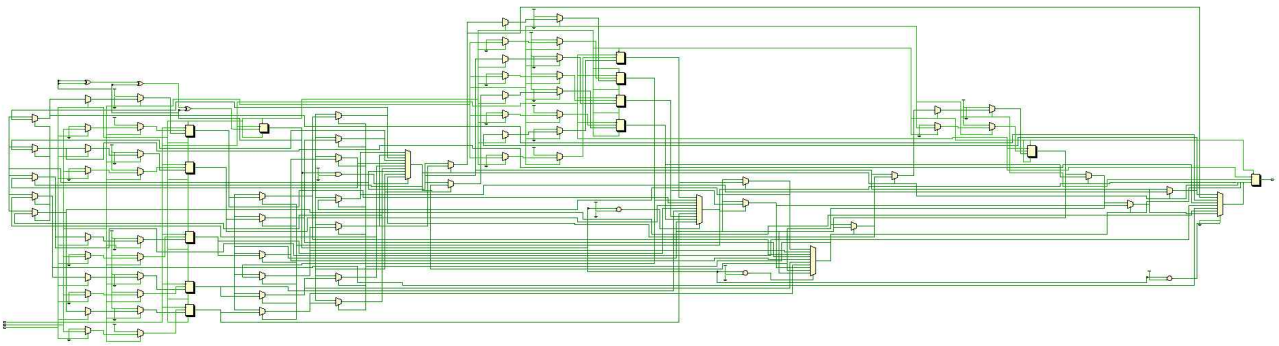
● 모듈 별 기능

● Password



이 모듈은 내부에 난수 생성기를 연결하고, 사용자가 입력한 네 자리 숫자를 STRIKE/BALL 판정 모듈과 비교하여 결과를 갱신하도록 설계되었다. 클럭이 들어올 때마다 gen_enable 신호가 활성화되어 있으면 난수를 업데이트하고, submit 신호가 1이 되는 순간 현재 난수와 입력된 숫자를 비교해 STRIKE와 BALL 값을 구한 뒤, 모든 자리가 일치할 경우 correct 신호를 1로 설정한다. 이때 STRIKE와 BALL은 계속해서 저장되며, submit이 비활성화되면 이전 결과를 그대로 유지하도록 동작한다. 또한 pwd_key 포트로는 현재 난수가 전달되어 외부(7-Segment)에서 확인이 가능하다.

● RandomNumberGenerator(Password 내부)



이 모듈은 난수를 생성하는 모듈이며, 내부에 LFSR(Linear Feedback Shift Register)을 탑재한 회로로서, 클럭 신호가 들어올 때마다 피드백 연산을 통해 무작위 비트를 계속 생성하고, gen_enable 신호가 활성화되면 이 비트들을 활용해 0부터 9까지의 숫자 중 서로 다른 숫자만을 뽑아 4자리 난수를 만들도록 동작한다. 먼저 리셋이 걸리

면 내부 레지스터와 가능한 숫자 집합을 초기화하며, 이후 LFSR가 생성한 값에 따라 각 자리에 들어갈 숫자를 선택하고 선택된 숫자를 다시 숫자 집합에서 제거하는 방식으로 중복을 제거한다. 네 자리 숫자를 하나로 결합해 출력하는 구조이므로, 매 클럭마다 LFSR 부분이 새로운 난수 비트를 만들어내고, gen_enable 신호에 맞춰 필요한 자리수만큼 추출해 중복 없는 난수를 생성할 수 있다.

- cnt_strike_ball(Password 내부)

이 모듈은 16비트 입력 중 4자리 숫자를 분리해 내고, 사용자가 입력한 4자리 숫자와 비교하여 각 자리가 동일하면 STRIKE를, 다른 위치에 있지만 같은 숫자면 BALL을 세도록 설계된 구조이다. 입력으로 들어오는 랜덤 숫자와 사용자 입력을 각각 multiplexer를 통해 한 자리씩 나누고, 먼저 자리 인덱스가 동일한 블록끼리 비교하는 비교 회로에서 STRIKE를 계산한다. 이후 STRIKE에 포함되지 않은 자리들만 다시 비교해 서로 다른 인덱스에서 숫자가 같으면 BALL을 증가시키며, 최종적으로 STRIKE와 BALL을 출력하는 형태로 동작한다.

- fourbit_comparator(Password 내부)

이 모듈은 두 4비트 입력(Reg1(), Reg2())이 완전히 일치하는지를 비교하여, 모든 비트가 같다면 out 신호를 1로, 하나라도 다르면 0으로 만드는 기능을 한다. 내부적으로는 네 개의 XOR 연산(~^)을 활용해 각 비트를 비교하고, 이 결과들을 전부 AND로 묶어 동일성 여부를 최종적으로 판별한다. 이를 통해 4비트 수준에서 빠르게 동등 비교를 수행할 수 있는 구조이다.

- lcd_controller

이 모듈은 LCD를 초기화하고, 사용자가 입력을 완료하여 submit 신호가 올라가는 타이밍에 맞춰 LCD 화면을 지우고 새로운 문자열을 출력하는 상태 기계(State Machine) 형태의 로직을 수행한다. 내부에 여러 상태(state)가 정의되어 있으며, 먼저 LCD 초기화 명령어들을 순차적으로 전송해 디스플레이를 세팅한 뒤, submit 신호가 감지되면 LCD 클리어 명령을 보내고, 그 후 digits_to_ascii에서 생성된 메시지를 한 글자씩 출력한다. 각 단계마다 주어진 DELAY를 통해 일정 시간 지연을 두면서 명령어가 정상적으로 처리되도록 하고, char_index로 현재 출력 중인 문자의 위치를 추적한다. 이 과정을 통해 “STRIKE x BALL y”와 같은 결과 문구를 LCD에 정확한 타이밍에 띄워 준다.

- digits_to_ascii

이 모듈은 STRIKE와 BALL 값(각각 3비트)을 받아 ASCII 코드로 변환하여, “STRIKE x BALL y” 형태의 문자열을 구성하는 역할을 담당한다. 내부 로직에서는 먼저 모든 출력(16바이트 분량)을 공백 문자로 초기화한 뒤, 앞쪽 8바이트에는 “STRIKE”라는 문자를, 뒤쪽 7바이트에는 “BALL”이라는 문자를 채우고, 마지막 한 자리에 STRIKE 및 BALL 값이 ASCII 숫자로 들어가도록 처리한다. 결과적으로 “STRIKE x BALL y”라는 문구를 생성하며, 각 문자를 ASCII 코드로 만들어 다른 모듈(lcd_controller)에서 받아 LCD에 표시하도록 한다.

- **Bulls_and_Cows(숫자야구게임 최상위 모듈)**

이 모듈은 전체적으로 숫자야구 게임("Bulls & Cows")을 FPGA 환경에서 동작시키기 위한 상위 구조이다. 입력으로 들어오는 여러 신호(키패드, 클록, 리셋, gen_enable, submit)를 받아 내부에서 숫자를 생성·저장하고, 이를 7세그먼트 디스플레이와 LCD에 표시하면서 LED나 피에조, 모터 등의 출력 장치를 동작시키도록 순차적으로 흐른다. 먼저 count8 모듈과 count4 모듈이 클록을 입력받아 비트 카운터 역할을 수행한다. count8은 3비트(Q0, Q1, Q2)를, count4는 2비트(Q0, Q1)를 카운트하여 다중화나 LED·모터 제어 등의 타이밍 신호로 쓰인다. 그다음 d2b 모듈은 키패드(0~9)에서 눌린 신호를 4비트 이진수로 변환한다. 변환된 4비트 값은 내부 레지스터(four_bit_reg_ce)를 거치면서 순차적으로 저장·이동하며, 이 과정에서 trigger 모듈은 키패드의 누름(하이 신호)을 감지해 레지스터 업데이트를 제어한다.

이렇게 얻어진 4비트들(Reg_1, Reg_2, Reg_3, Reg_4)은 Password 모듈로 전달되어, gen_enable이 켜져 있으면 내부에서 난수가 생성되고, submit이 올라갈 때 입력값과 난수를 비교해 STRIKE와 BALL을 계산한다. 또한 Password 모듈에서 생성된 16비트 난수(pwd_key)와 사용자 입력 레지스터 값들은 b2seg_bus 모듈을 통해 7세그먼트 디스플레이로 표시되며, LED 모듈은 STRIKE가 4일 때(정답을 맞춘 상황)를 판별하여 녹색 또는 빨간색 LED를 제어한다.

이와 함께 Piezo_module은 키패드 입력을 기반으로 피에조 부저를 울리는 기능을 담당하며, 모듈 내부에서 어떤 키가 눌렸는지 확인해 사운드를 발생시킨다. Motor 관련 신호들도 정답 여부에 따라 회전방향이 달라지도록 설계되어 있다.

마지막으로 STRIKE와 BALL 값을 넘겨받은 digits_to_ascii 모듈이 "STRIKE x BALL y" 문자열을 ASCII 코드로 변환하고, 이 데이터를 lcd_controller 모듈이 받아 LCD를 초기화·클리어한 후, submit 신호가 들어올 때마다 화면에 해당 문자열을 출력한다. 이 과정을 통해 사용자에게 결과를 시각적으로 알려준다.

✔ 계획에서 구현되지 않은 부분에 대한 향후 해결 방안

프로젝트 계획에서 축하 노래 구현을 위해서는 피에조 모듈에서 재생할 멜로디를 코드로 작성하고, 정답을 맞춘 시점에만 해당 코드를 실행되도록 제어 로직을 추가하면 된다. 예컨대 STRIKE가 4로 판정되는 순간 피에조 모듈에 전달할 음계 신호를 순차적으로 전송하며, 게임이 끝날 때까지 반복 재생하거나 일정 횟수만큼만 재생하도록 설정할 수 있다. 또한 9회 제한을 두는 기능은 내부 카운트를 활용해 도전 횟수를 추적한 뒤, 9회가 넘으면 더 이상 입력을 받지 않거나 자동으로 오답 처리를 해버리는 방식으로 구현할 수 있다. 이를 위해 시도 횟수를 체크하는 레지스터와 제어 로직을 준비해, submit이 발생할 때마다 카운트를 올리고 9회가 되면 게임을 종료하거나 재시작 옵션을 띄우도록 하면 될 것이다.