

# Term Project

---

**SNUCAD**

**2025-11-17**

# 프로젝트 방식

- 4인 1조로 진행
  - 조원 평가 반영
- 12/15(월) 18:30~ 프로젝트 시연
  - 실습 시간에 오프라인 채점. 지역 검사 불가
- Vivado에서 verilog로 구현 후 FPGA board에 프로그래밍
  - Vivado Project 파일은 압축하여 eTL을 통해 제출 (~12/15(월) 18:29)
- 오직 FPGA board에서 구현 조건에 맞게 작동하는지 확인
  - Testbench는 제공 되지 않음
  - Synthesis, implementation 등을 진행하는 도중 Warning, Critical Warning을 무시하면 제대로 작동하지 않을 수 있으니 주의
- 검사 당일, 조교의 특정 시나리오에 따라 잘 작동하는지 확인 할 예정
  - 특수한 시나리오의 예외처리 반영에 대해 부분점수가 있음

# 프로젝트 주제

## ■ 총 3가지 모드에 대한 게임 구현

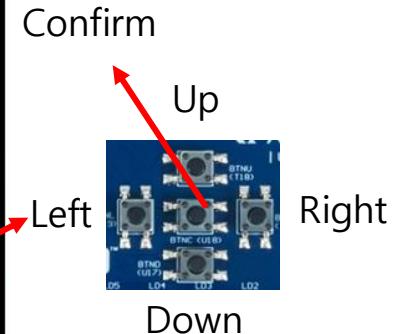
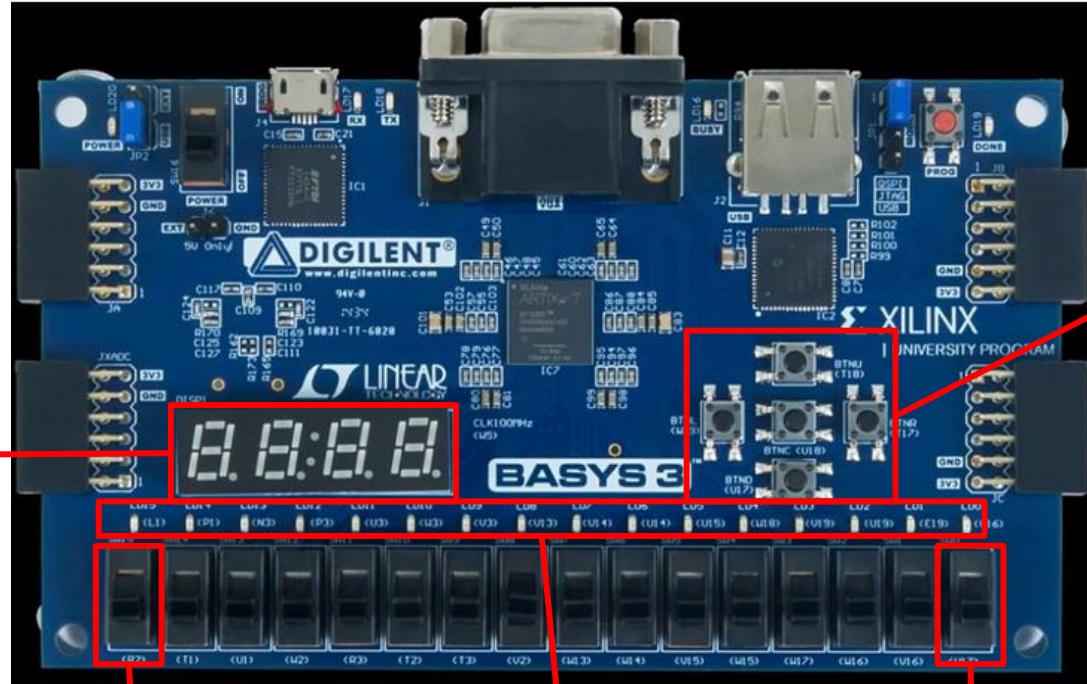
1. 숫자 야구 게임
2. LED 개수 맞추기 게임
3. Credits

# Mode1: 숫자 야구

- 사용자가 0~9 중 중복 없는 4자리 숫자를 정답으로 설정 (예: 1234)
  - 사용자가 중복된 숫자 입력 시 → "ERR" 표시
- 사용자가 다양한 숫자를 입력
- FPGA가 피드백 출력:
  - Strike (S): 숫자와 위치가 모두 일치
  - Ball (B): 숫자는 맞지만 위치가 틀림
- 사용자가 다음 숫자를 입력 → FPGA 피드백 출력 → 반복
- 사용자가 처음 입력한 숫자와 일치하는 숫자를 입력하면 7segment =[good] 출력

# Mode1: 숫자 야구 (FPGA 기능설명)

1. 정답 숫자
2. 추측값 숫자
3. S/B
4. 에러
5. 정답 표시



숫자 추측 시도 한번 할 때마다 LED 하나씩 ON

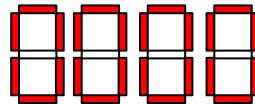
위로 올리면 MODE1(숫자야구 모드)

- Reset 동작
- 위로 올리면 Reset 유지 (Reset ON)
- 아래로 내리면 Reset 해제, 시작 (Reset OFF)

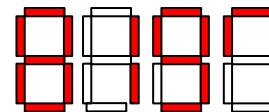
# Mode1: 숫자 야구 (1. 정답 입력)

MODE1 스위치를 올려서 숫자 야구 게임 모드로 전환

- Reset ON, OFF 하면 시작, 7Segment 0000 출력



- 버튼의 상/하/좌/우로 각 자리 수를 선택해 알맞은 입력 선택

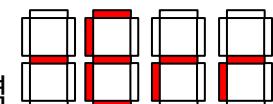


예시 정답

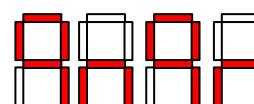
- Left/Right로 자릿수 변경, Up/Down으로 0~9까지 각 자리 숫자 선택
  - 이때, 선택중인 자릿수는 깜빡임 구현
  - Confirm 버튼을 누르기 전, 각 자리 수 숫자는 얼마든지 좌/우 버튼을 통해 변경이 가능해야 함

- 숫자를 다 선택한 후, 가운데 Confirm 버튼 누르면 정답 입력 완료

- 이때, 만약 사용자가 중복된 숫자를 입력했을 경우 7Segment =[ - E r r ] 출력

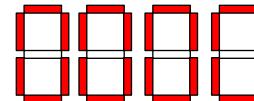


- 성공적으로 입력 됐을 경우 7Segment =[ g o g o ] 출력

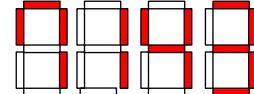


# Mode1: 숫자 야구 (2. 추측값 입력)

- 한번 더 Confirm 버튼을 누르면 Segment = [ 0 0 0 0 ] 출력



- 버튼의 상/하/좌/우로 각 자리 수를 선택해 알맞은 추측 값 선택

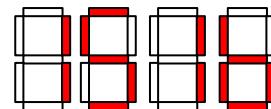


→ 예시 추측 값

- 추측 값을 선택하고 Confirm 버튼을 누를 때, 한번 시도할 때마다 LED 하나씩 ON
  - 켜지는 LED는 최대 16개 → 16번의 시도까지 LED로 표기 가능

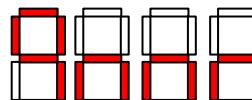
- Confirm 버튼을 누르면, 입력에 대한 Strike, Ball을 Segment에 출력

- ex) 정답 = 8197, 추측 값 = 7143 → 1S 1B

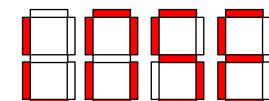


- 한번 더 Confirm 버튼을 누르면 0000 출력 → 추측 → S/B 확인 반복

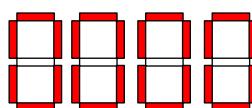
- 최종적으로 정답을 맞추면, Segment = [ g o o d ] 출력



- 만약 정답 계속 못 맞춰서 LED 가 다 켜졌다면? Segment = [ L O S E ] 출력



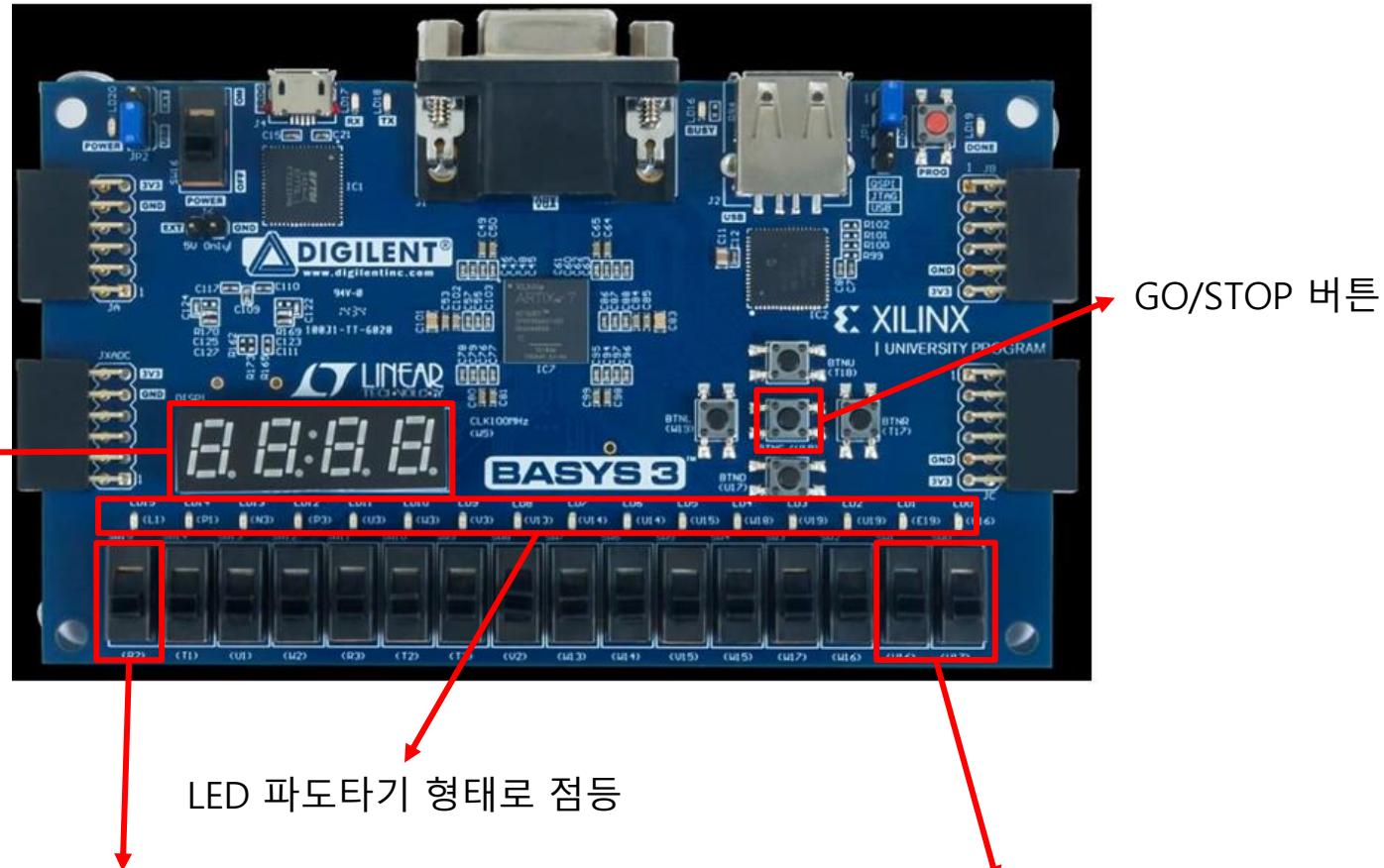
- Reset ON, OFF 하여 다음 판 시작



# Mode2: LED 개수 맞추기

- 7Segment 오른쪽 2개에 랜덤으로 1~16의 숫자가 출력됨
- 왼쪽부터 LED가 순서대로 파도타기 처럼 켜졌다 꺼짐.
- 지속적으로 점등하는 LED중에서 7segment가 출력하는 숫자 개수만큼 LED가 켜지도록 GO/STOP 버튼을 눌러서 맞추기
- 왼쪽 2개의 Segment에 제시한 숫자보다 많이 켜지면 오른쪽 segment 두개에 dn 출력, 적게 켜지면 UP 출력

# Mode2: LED 개수 맞추기 (FPGA 기능설명)



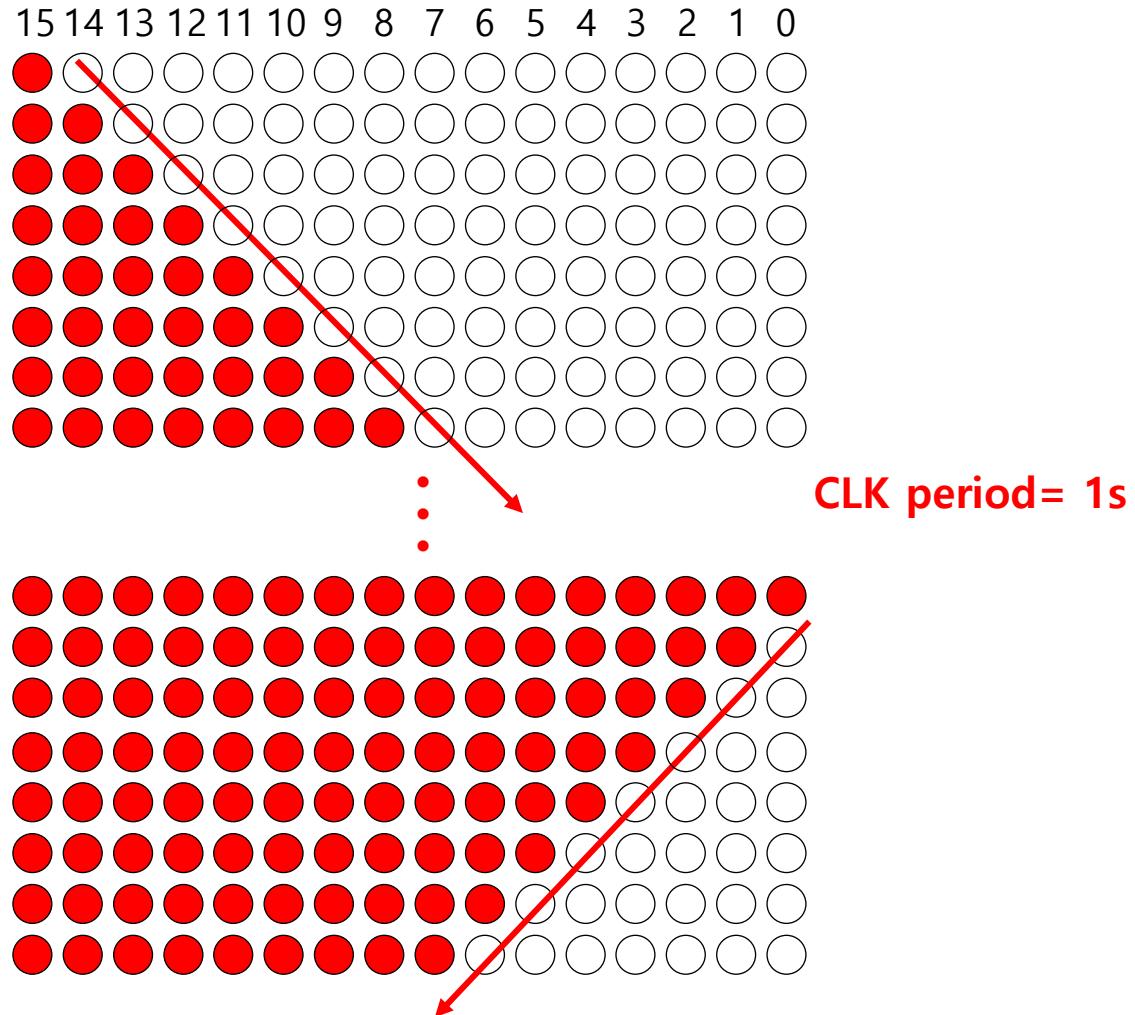
1. 켜져야하는 LED 개수
2. 현재 켜진 개수와 UP/DOWN 표시
3. 정답 표시

- Reset 동작
- 위로 올리면 Reset 유지 (Reset ON)
- 아래로 내리면 Reset 해제, 시작 (Reset OFF)

스위치 둘 다 올리면 Mode2 (LED맞추기)

# Mode2: LED 개수 맞추기 (LED Figure)

LED 켜지는 순서 Figure



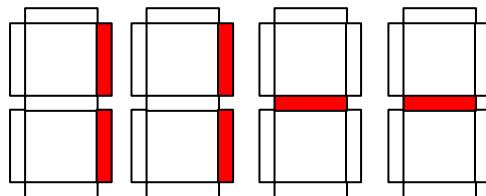
# Mode2: LED 개수 맞추기 (Step)

- Step0)**
- 오른쪽 끝 switch 2개 올려 Mode 2 진입
  - 왼쪽 끝 Reset switch ON, OFF 하여 시작

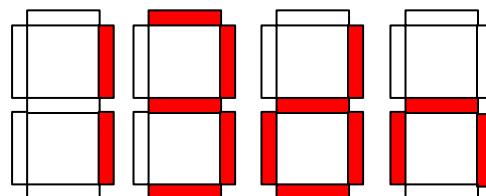
**예를 들어 요구된 켜진 LED 개수 = 11**

이 숫자는 1~16 중 FPGA에서 랜덤하게 출력하도록 구현  
ex) Segment=[ 1 1 - - ]

**Step1)** 출력된 값을 표기

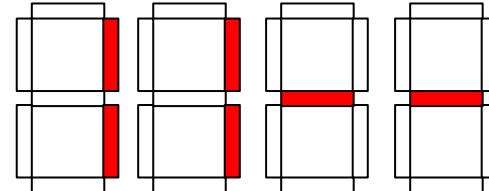


**Step2)** 만약, GO/STOP 버튼을 눌렀을 때 켜진 LED의 개수가 13개면?

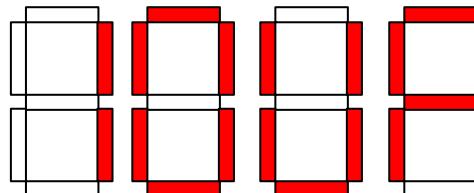
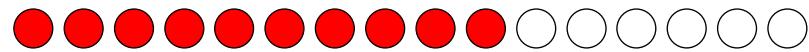


현재 켜진 LED의 개수는 13개이며, down이라는 의미

**Step3)** 다시 GO/STOP 버튼을 누르면, 원래 목표하는 LED개수로 돌아옴

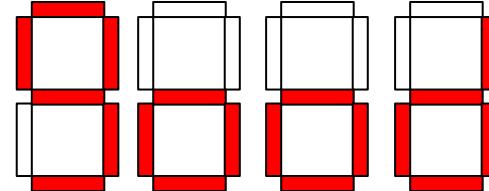
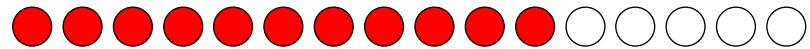


**Step4)** 다시 GO/STOP을 눌렀을 때, 켜진 LED의 개수가 10개면?



현재 켜진 LED의 개수는 10개이며, up이라는 의미

**Step5)** 11개를 정확히 켰을 경우

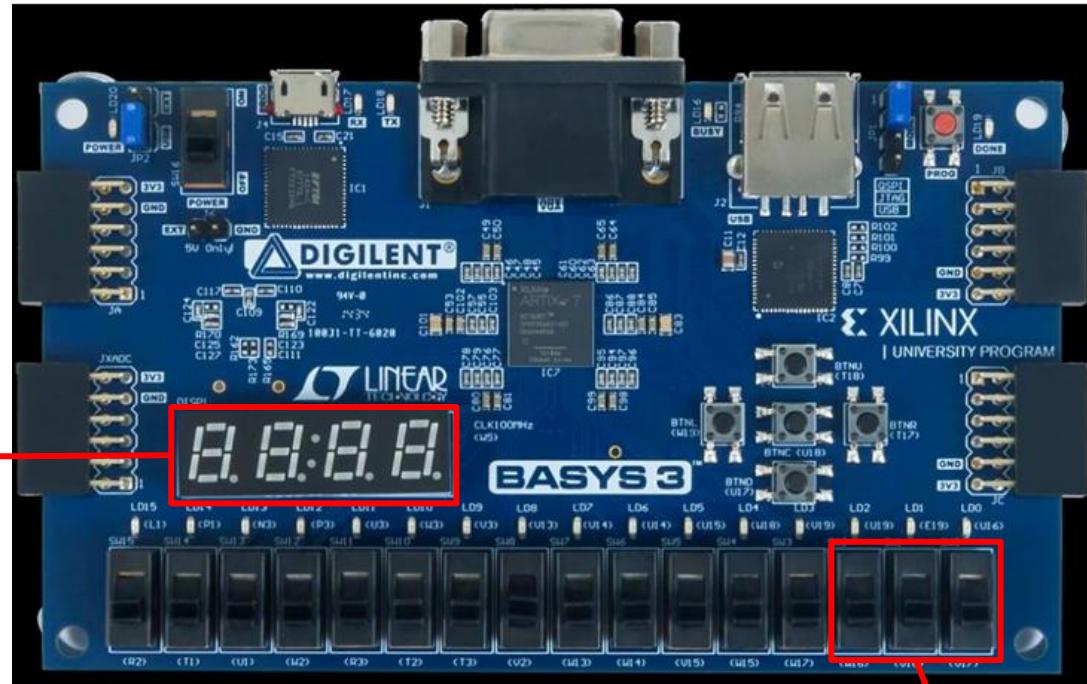


**Step6)** Reset ON, OFF 하여 다음 판 시작 (Go to Step 1) 11

# Mode3: Credits

- 함께한 조원의 성/이름을 이니셜로 Segment에 순서대로 점등

# Mode3: Credits (FPGA 기능설명)



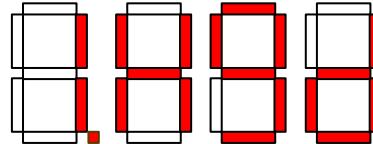
함께한 조원의 순서와  
이름을 차례대로 표기

스위치 세개 올리면 Mode3 시작(Credit)

# Mode3: Credits (Detail)

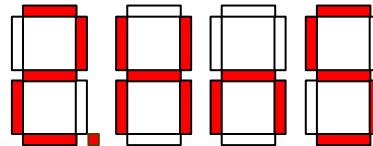
## ■ 예시 조원 이름

1. 홍길동



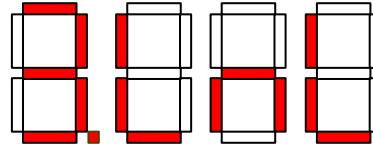
1. Hgd ( 1. Hong Gill Dong)

2. 김논설



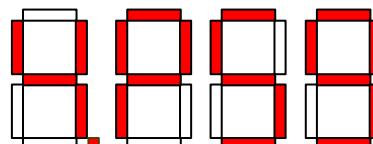
2. KnS ( 2. Kim Non Seol)

3. 이논리



3. LnL ( 3. Lee Non Li)

4. 박설계



4. PSG ( 4. Park Seol Gye)

지속적으로 돌아가면서 출력 할 것 (CLK period= 3s)

7Segment에 표기하기 애매한 알파벳은 적절하게 알아서 출력

# 표절 관련

## ■ 표절 절대 금지

- Skeleton Code 금지
- 모든 Code에 대해 엄격한 Copy Check 진행 예정
- 코드의 일부분이 표절로 판단될 경우, 학칙에 의거하여 처리 진행