

실험 8. 보드 주변 입출력 구현

Digital System Design and Experiment

Made by Mobile Multimedia Systems Group

Graduate School of Convergence Science and Technology

Seoul National University



Mobile Multimedia Systems Group

| 목차

- 실험 목표
- 스위치 인터페이스
- 7-segment display
- Pin locking
- Board 연결 및 bit 파일 다운로드 방법
- 실험
- 실험 결과 및 제출 방법

실험 목표

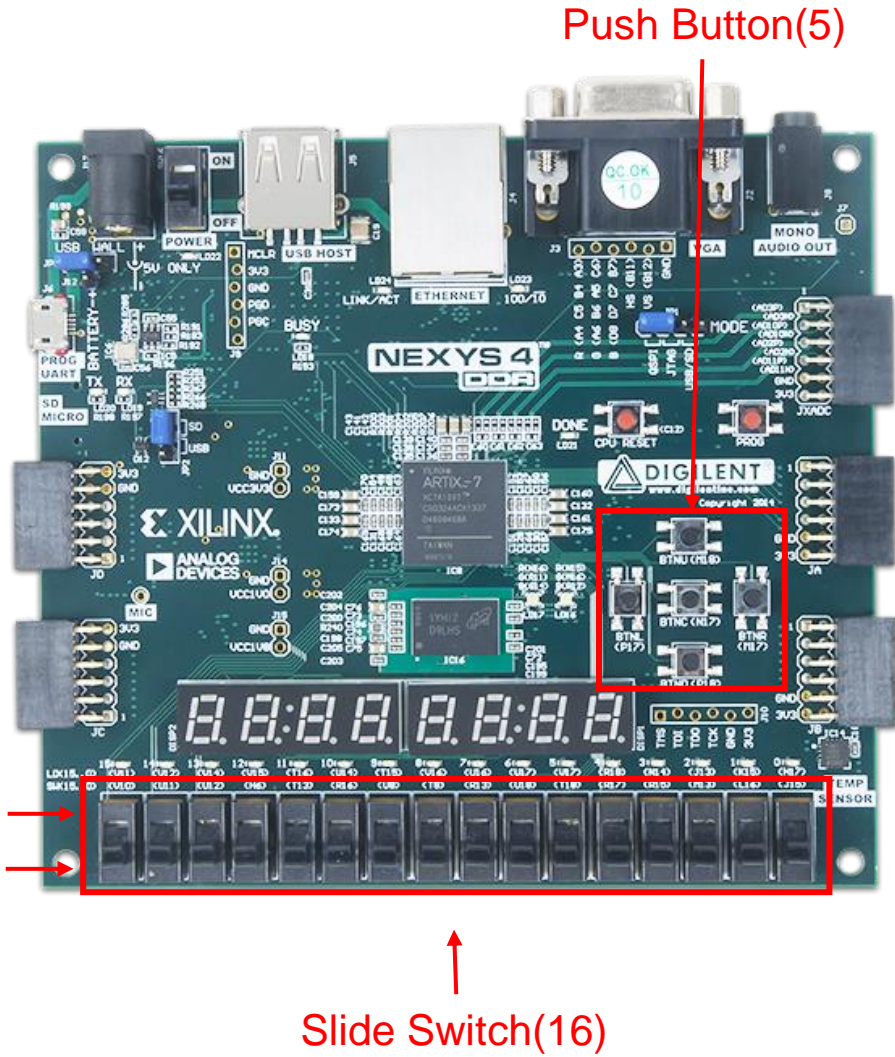
- GPIO를 통해 User가 보드의 스위치를 조작하면 처리 결과를 7-Segment Display에서 확인할 수 있는 Logic 구현
- 시뮬레이션으로 구현한 모듈 검증
- 보드에서 실제 동작을 확인



1.

Background

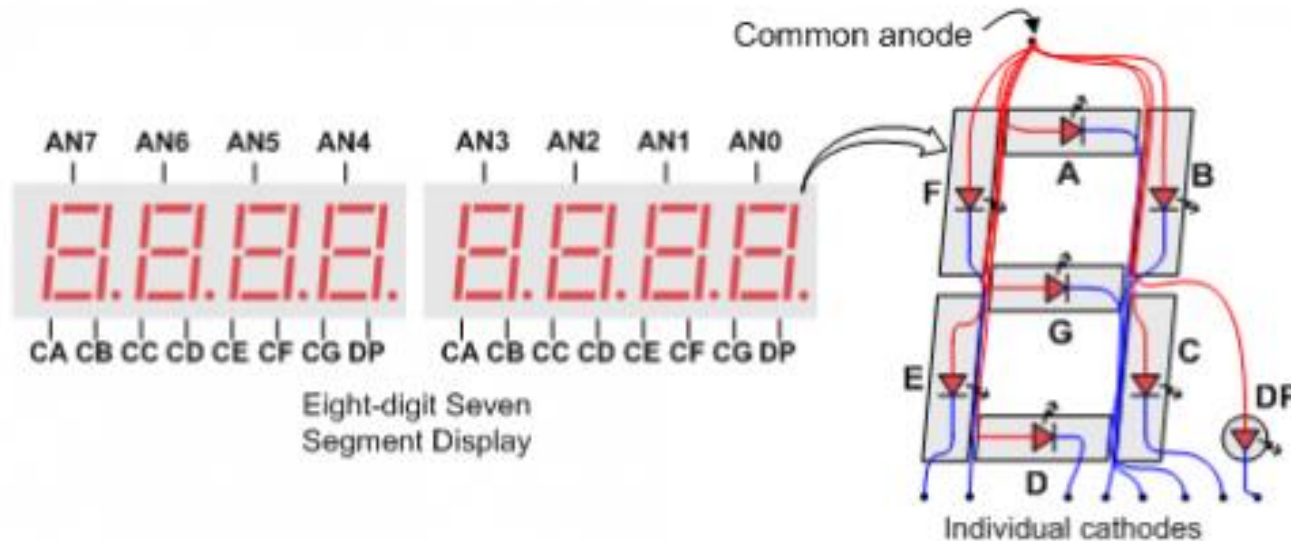
스위치 인터페이스



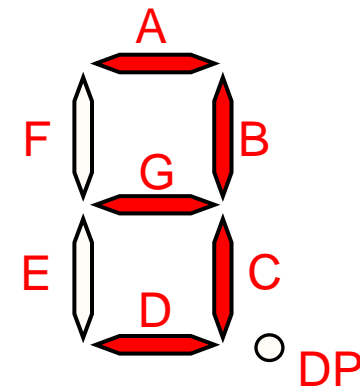
- Nexys 4 DDR 보드에서는 범용 목적의 Slide Switch 16개와 Push button 5개를 제공한다.
- 기계식 스위치를 통해 User는 직관적이고 손쉽게 Logic에 주어지는 Input 조건을 변경할 수 있다.
- 스위치를 이용하고자 하면, Verilog code에서 물리적인 스위치에 대응하는 입력 포트를 선언하고, PCB에서 스위치가 FPGA의 어떤 Pin에 연결되어 있는지를 확인하여 .xdc 파일에 반영한다.
- Logic을 Implementation 하여 bit 파일을 보드에 다운로드 하고 나면, 스위치에서 발생하는 High / Low 신호를 받을 수 있다.

7-Segment Display

- 8개의 획으로 숫자나 문자를 표시할 수 있는 장치이다. 간단하기 때문에 계산기 등에 많이 활용된다.
- Nexys 4 DDR 보드의 경우 2개의 4자리 7-Segment display가 제공되어 있다.
- 각 숫자별로 모든 획을 독립적으로 제어하기 위해서는 64(8X8)개의 Pin이 할당되어야 한다.
- Nexys 4 DDR보드에서는 필요한 Pin 개수를 줄이기 위해 8개의 LED의 Anode는 공통으로 한 Pin에 연결하고, 각 자리 별 획 (A,B,C,.,DP)의 Cathode도 공통으로 제어하여 필요한 Pin 개수를 16개로 줄였다.
- 따라서 Anode를 따로 제어하지 않는다면, 모든 Display에 같은 숫자가 나타나게 되는 구조이나, 8개 숫자 중 한 자리만 켜지도록 AN_x를 제어하고, 나머지 7개 자리의 Display는 Off 시키는 과정을 각 자리별로 순차적으로 반복하여 각 자리별로 다른 숫자가 나타나도록 한다.
- 이 과정을 빠르게 반복하여, 사람 눈에는 잔상효과로 인해 숫자가 계속 나타나고 있는 것처럼 보이게 한다.

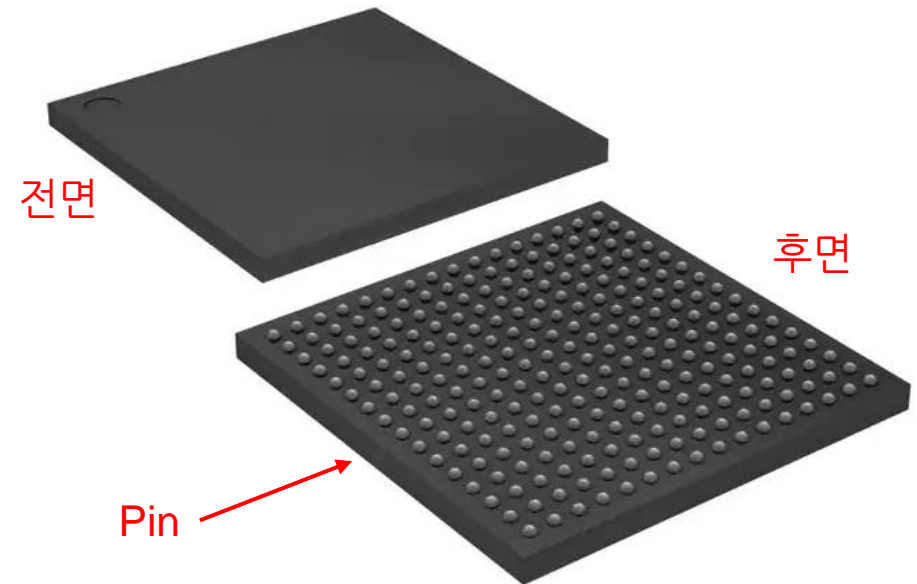


예시로 '3'을 표시 8'b00001101



Pin Locking

- Verilog Code를 작성하여 구현한 Logic이 외부 주변장치들과 연동하기 위해서는 주변장치와 FPGA의 Port 사이의 경로가 연결되어야 한다.
- 주변장치에서 FPGA의 Physical Pin 까지는 PCB의 trace가 연결되어 있고, FPGA 내부에서는 PCB의 trace가 연결된 FPGA의 Pin과, Logic의 Input port를 연결시켜 주는 작업이 필요하다. 이를 Pin locking 이라고 한다.
- Custom 보드를 설계할 경우, FPGA 내부의 routing과 PCB의 trace가 최적화 되도록 Pin location을 잡는 것을 권장한다. 그러나 우리는 이미 제작된 Board를 사용하고 있으므로 Board의 회로도를 보고 Pin location을 .xdc 파일에 반영하면 된다.



.xdc 파일 작성

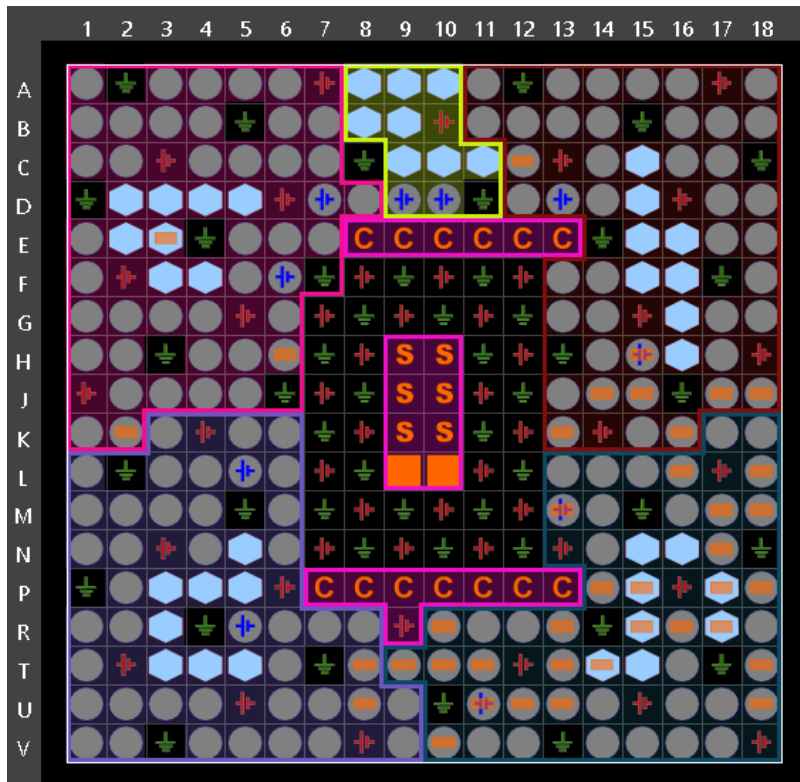
- .xdc 파일에서 Pin에 대한 기술은 다음 예시를 참고한다.

```
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { LED[0] }];  
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { LED[1] }];  
set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 } [get_ports { LED[2] }];  
...
```

Pin Location

IO Standard

Input / Output port



※ IO Standard

우리가 사용하는 Artix-7 FPGA는 LVCMOS, LVTTTL, HSTL, SSTL 등의 Standard를 지원한다. 이번 수업에서는 LVCMOS33만 사용한다. 각기 Standard는 아래와 같은 차이점들을 가진다.

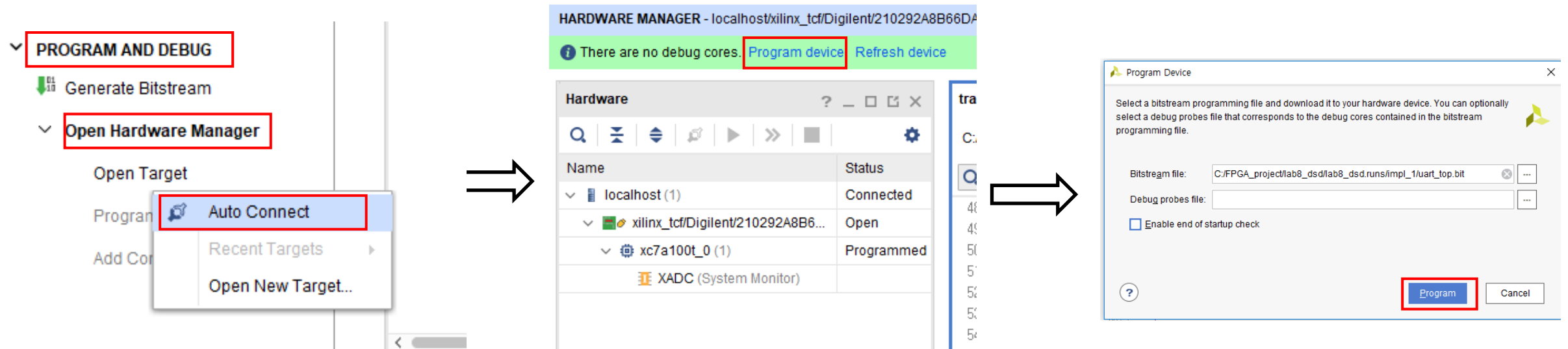
- Single ended vs. Differential
- VREF referenced input vs. standard CMOS input
- termination on inputs and/or outputs
- driver characteristics based on “SLEW RATE” and “DRIVE_STRENGTH”

자세한 내용은 Xilinx UG471 문서 참조.

Program Device (Configuration)

- Step 0: FPGA Board power on
- Step 1: Open Hardware Manager > Open Target > Auto Connect
- Step 2: Hardware Manager하단, Program device 클릭
- Step 3: Program 클릭
- Step 4: 보드의 DONE led (LD21) 점등 확인

(Note: Step 3에서 Bitstream File이 없을 경우, “[project name]폴더 / **** .runs / impl1**” 에서 **“** .bit”**파일을 찾으세요.)



2.

실험 개요

실험

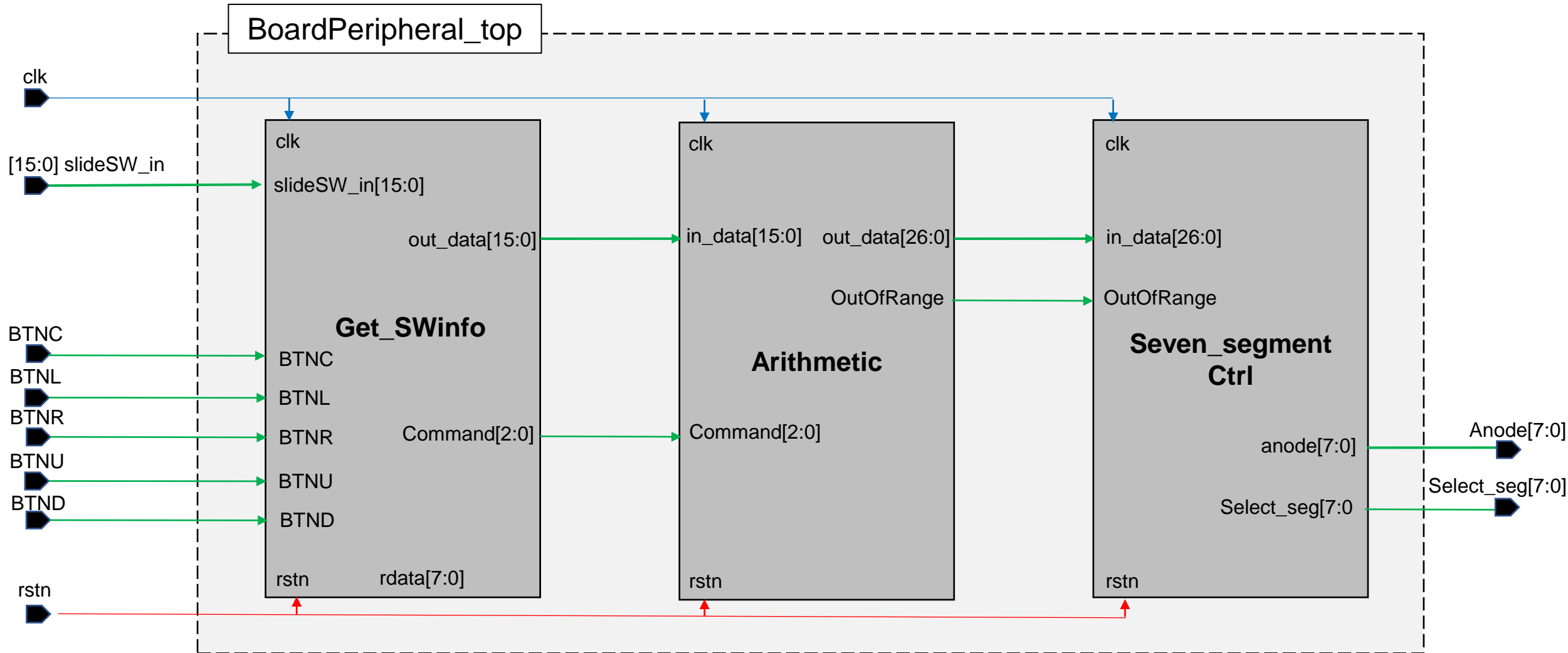
■ 실험 목표

- Slide Switch를 이용하여 처리하고자 하는 16bit 데이터를 Logic으로 보낸다
- 5개의 Push button을 활용하여 현재 가지고 있는 데이터에 연산을 수행한다
 - Center button : Logic에 저장된 값을 Slide Switch에 설정된 값으로 변경
 - Left button : Logic에 저장된 값에 +1 수행
 - Right button : Logic에 저장된 값에 -1 수행
 - Up button : Logic에 저장된 값에 x2 수행
 - Down button : Logic에 저장된 값에 ÷2 수행
- 수행된 연산결과를 7-Segment Display로 10진수 형태로 전시한다.

■ 실험 환경

- Vivado 2022.1
- Nexys 4 DDR FPGA 보드
 - System clock 100MHz

Block diagram



제공된 코드파일 설명

- Verilog 파일 (____.v)
 - BoardPeripheral_top.v
 - 전체 top module
 - Get_SWinfo.v
 - Slide Switch 와 Push button을 연동한다. 비동기 입력 신호인 스위치 신호를 Clock에 동기화 시킨다.
 - Push button의 경우, 한번 클릭에 너무 많은 명령이 생성되는 것을 막기 위해 한번 동작하면, 그 다음 100ms 동안은 button의 입력을 무시한다.
 - Slide Switch의 입력은 Arithmetic 모듈에 연산 대상 16-bit 데이터로 제공되며, push button 입력은 수행할 연산명령(3-bit)으로 변환하여 Arithmetic 모듈에 제공한다.
 - arithmetic.v
 - Get_SWinfo 모듈에서 받은 16-bit 데이터에 대해 연산명령에 따른 처리를 수행한다.
 - 7-Segment display의 자릿수(8)을 고려하여, 처리결과는 27-bit 데이터로 Seven_segment Ctrl 모듈로 전달한다.
 - 현재 Logic이 저장하고 있는 값에 연산을 수행하였을 때, 표현범위를 벗어나는 경우 연산은 수행하지 않고 현재 값을 유지하며, "OutOfRange" flag를 High 로 들어 Seven_segment Ctrl 모듈에 정보를 전달한다.

To Do

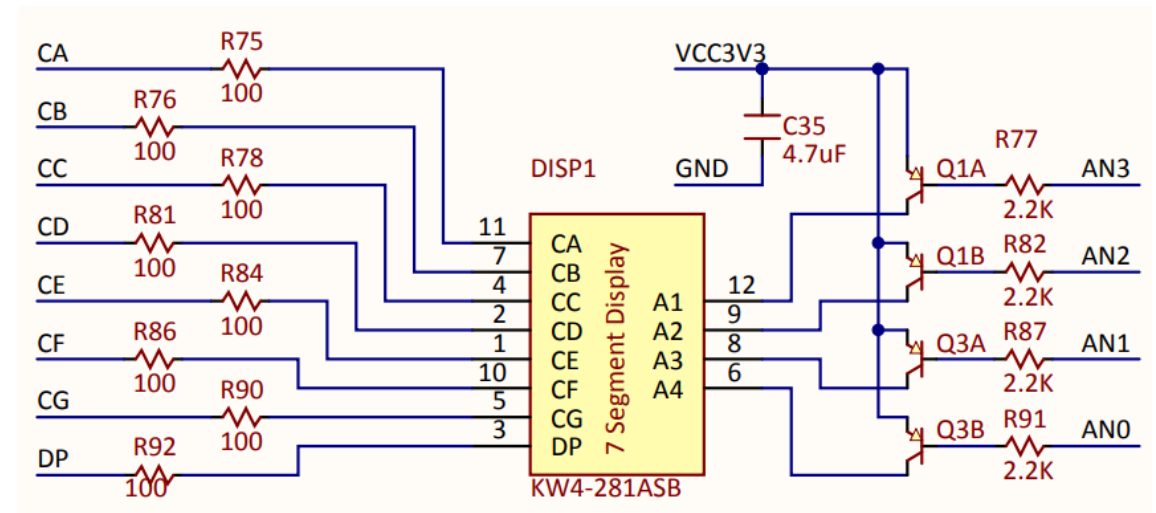
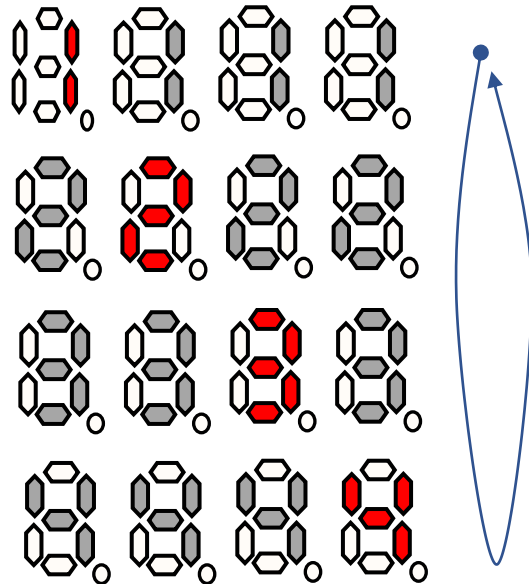
- Verilog 파일 (____.v)

- Seven_segment_Ctrl.v

- 수정 대상 파일

- Arithmetic 모듈에서 연산된 결과를 10진수로 7-segment display에 전시하는 역할을 한다.
 - Arithmetic 모듈에서 연산결과가 표현범위를 넘는다고 판단하여 “OutOfRange” flag를 High로 설정했을 경우, 7-Segment Display에는 “OutOFrAn” 이라고 표시한다.
 - 앞서 설명한 바와 같이 각 자리수의 획을 독립적으로 제어할 수 있는 것이 아니므로, 8자리의 수 중 한 자리만 나타나도록 켜고, 나머지 7자리는 끄는 것을 순차적으로 수행하며, 이 절차를 눈이 감지하지 못할 속도로 반복 수행하여야 한다.

- 예) 1234



To Do

- xdc 파일 (____.xdc)
 - Nexys4DDR_BoardPeripheral.xdc
 - 수정 대상 파일
 - 현재 파일에는 모든 Pin이 주석처리 되어 있다
 - Verilog Top 모듈을 보고, 필요한 입/출력 포트는 주석을 해제하여 사용 가능하도록 한다.
 - .xdc 파일의 포트명을 verilog 파일에 기술된 포트명과 일치시켜야 한다.

```
set_property -dict { PACKAGE_PIN H17  IOSTANDARD LVCMOS33 } [get_ports { LED[0] }];  
set_property -dict { PACKAGE_PIN K15  IOSTANDARD LVCMOS33 } [get_ports { LED[1] }];  
set_property -dict { PACKAGE_PIN J13   IOSTANDARD LVCMOS33 } [get_ports { LED[2] }];  
...
```


Simulation 검증

- “tb_BoardPeripheral_top.v” testbench파일을 추가
- Simulation 실행 후, Tcl console 창에 아래와 같은 결과가 나오면 보드에 올려서 검증

run 530ms

```
*****
Plus 1 is correct !!
*****
*****
Minus 1 is correct !!
*****
*****
Multiply by 2 is correct!!
*****
*****
Divide by 2 is correct !!
*****
```

```
$finish called at time : 524287945 ns : File "D:/DSD/Lab8/Board20/Board20.srscs/sim_1/new/tb_BoardPeripheral_top.v" Line 273
```

Simulation 검증

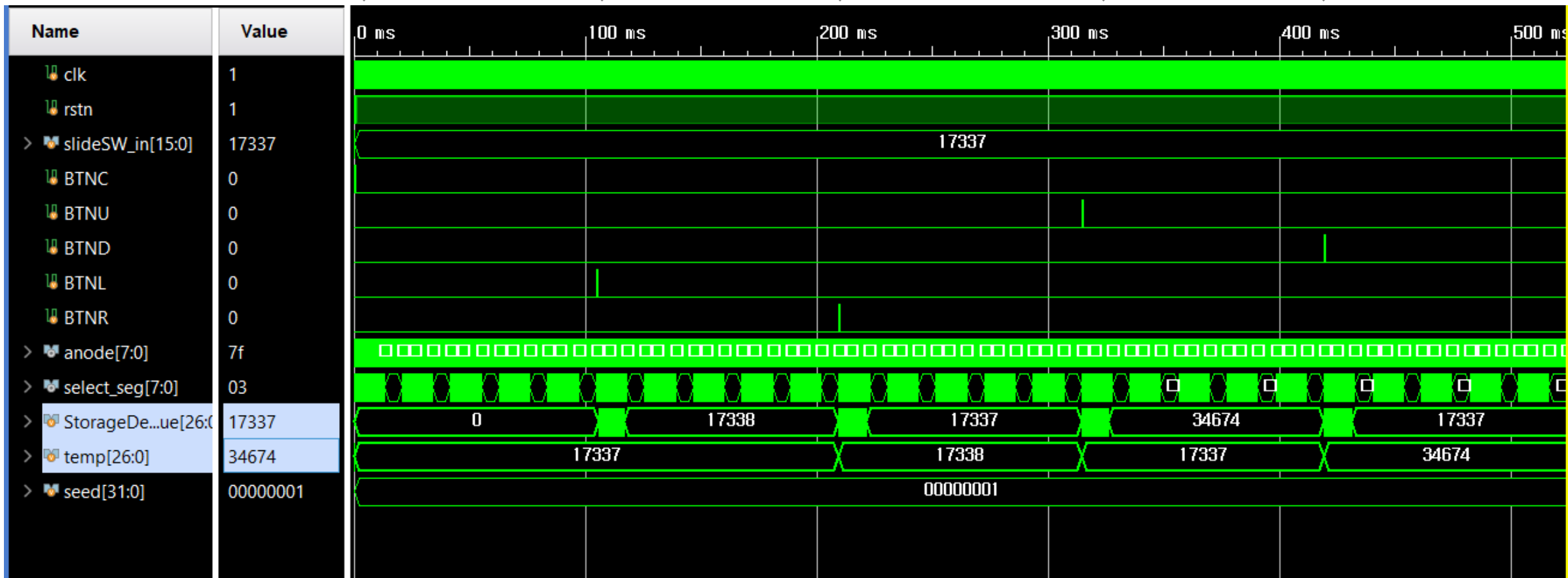
BTNC 버튼

BTNL 버튼

BTNR 버튼

BTNU 버튼

BTND 버튼



보드 검증

- Step 1: Bitstream을 생성하고 보드에 탑재하는 과정은 별도 첨부된 PPT를 참고바람
(Synthesis 이후의 과정에서는 testbench 파일을 disable 시킬 것)
- Step 2: Slide Switch를 이용하여 임의의 16-bit 데이터를 설정한다.
- Step 3: BTNC 버튼을 눌러 설정한 Bit가 10진수로 표시되는지 확인한다.
- Step 4: BTNL 버튼을 눌러서 “Step3의 값 + 1”의 값이 표시되는지 확인한다.
- Step 5: BTNR 버튼을 눌러서 “Step4의 값 - 1”의 값이 표시되는지 확인한다.
- Step 6: BTNU 버튼을 눌러서 “Step5의 값 x 2”의 값이 표시되는지 확인한다.
- Step 7: BTND 버튼을 눌러서 “Step6의 값 / 2”의 값이 표시되는지 확인한다.

| 실험 결과 및 제출 방법

■ 제출 방법

1) 아래 파일을 압축하여 제출 (파일명: “학번_이름_lab8.zip” ex) 2020-12345_홍길동_lab8.zip)

- 구현한 “**seven_segment_Ctrl.v**”, “**Nexys4ddr_BoardPeripheral.xdc**” 파일
- “**[project_name]폴더 / **.runs / impl1**” 의 “****.bit**”파일

2) 제출 기한

11월 6일(일요일) 23:59까지