

## [EE201] Term project: A simple calculator design

### 1. 목적

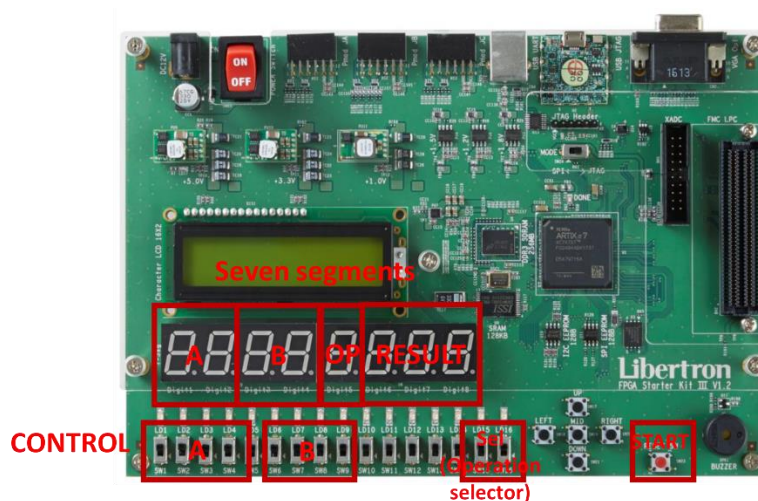
디지털 회로 설계를 하드웨어로 구현함으로써 이를 통해 Combinational Logic 과 Sequential Logic 을 설계하는 능력, Hierarchical Design, Synchronous Design 에 대한 이해, Simulator 를 이용한 검증 능력 등을 배양한다.

### 2. 설계 목표

덧셈, 뺄셈, 곱셈을 수행하는 간단한 calculator 설계한다. Combinational logic 과 Sequential Logic 을 적절하게 설계하여 각 연산을 위한 알고리즘을 구현하되, 주어진 디자인 조건을 만족하여야 한다. Top-down 방식으로 설계하되 각 블록의 설계를 검증한 후 전체 블록을 집적하여 설계를 완성한다. Test Bench 을 통해 simulation 을 이용하여 자신이 설계한 회로를 검증하고, 이를 하드웨어 상에서 실제 동작 여부를 통해 검증한다.

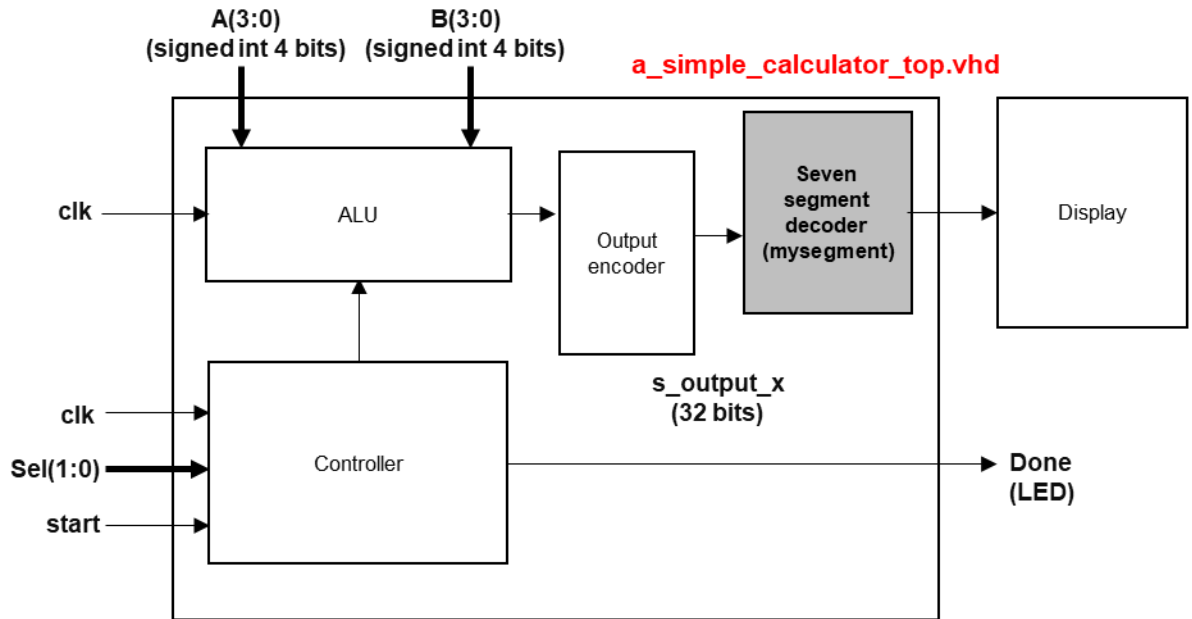
### 3. 설계

- (a) 동작: A 와 B 스위치(각 4 개)를 통해 Operand A 와 B 를 입력하고, Sel 스위치 (2 개)를 이용하여 operation 을 정의한다. 입력을 한 후 Start 버튼을 눌러 Result 가 출력되도록 한다.
- (b) Hardware 사진



FPGA: Xilinx Artix-7 xc7a75tfgg484-1

(c) Block Diagram



- Seven segment decoder 는 제공되는 파일 (myseven\_segments\_v2.vhd) 를 사용할 것.

(d) Signal Descriptions

(1) Input Signals

Name	Type	Description	Note
<b>Clk</b>	std_logic	System Clock	10 MHz
<b>Start</b>	std_logic	Start signal	Active High
<b>Sel</b>	std_logic_vector(1:0)	Operation code *See the section (c)	3 Operations (Switch 2EA)
<b>A</b>	std_logic_vector(3:0)	Operand A	Signed 4-bit: -8~7 (Switch 4EA)
<b>B</b>	std_logic_vector(3:0)	Operand B	Signed 4-bit: -8~7 (Switch 4EA)

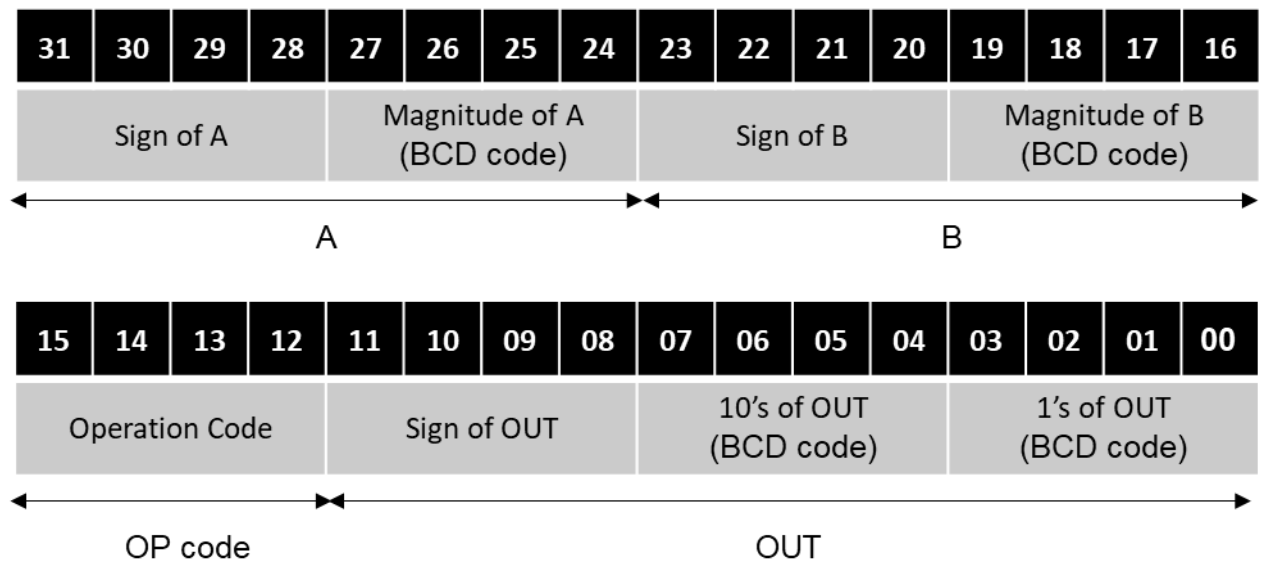
- Clk 는 10MHz 로 들어오는 마스터 클럭으로 시스템 전반을 컨트롤하는 클럭임.
- Sel 은 스위치로 조정되며, 다음과 같이 정의된다.

Function	Sel(1:0)	Note
<b>Addition</b>	00	s_output_x(15:12) = 1010
<b>Subtraction</b>	01	s_output_x(15:12) = 1011
<b>Multiplication</b>	10	s_output_x(15:12) = 1100
<b>Reserved</b>	11	s_output_x(15:12) = 1111

## (2) Internal & Output signal (Input to Seven-segment decoder)

Name	Type	Description	Note
<b>s_output_x</b>	std_logic_vector(31:0)	Result	See the table
<b>Done</b>	std_logic	Operation End	Active High

- 외부로 출력은 Done 신호 하나만 연결되며 LED (LD02)에 연결하여 동작 완료를 확인한다.
- 모든 입출력 결과는 Seven segment 를 통해 출력이 된다. 이를 위하여, 제공된 seven segment decoder module 의 입력 신호 s\_output\_x 를 생성한다.
- s\_output\_x 는 다음과 같이 정의되며, 4 비트씩 8 개의 Seven segment 를 제어한다. (A 의 부호, A 의 크기, B 의 부호 B 의 크기, OP CODE, 출력값의 부호, 출력값 크기 두자리)



- s\_output\_x 의 숫자 출력은 BCD 코드를 사용하고, 부호와 OP-CODE 는 다음과 같다.

Operation code encoding table (s_output_x(15:12))				
Bit	03 (MSB)	02	01	00 (LSB)
Addition	1	0	1	0
Subtraction	1	0	1	1
Multiplication	1	1	0	0
Reserved	1	1	1	1
Sign encoding table (s_output_x(31:28), s_output_x(23:20), and s_output_x(11:08))				
Positive	1	1	1	0
Negative	1	1	0	1

(e) Design Constraints

- ✓ ARITHMETIC LIBRARY 연산 사용하여 최종 출력 금지. (OPERATOR : +, -, \*, / , shift\_operator) 이것으로 구현한 경우 0 점 처리.
  - 단, Counter module 구현시 counter 내부에서 +'1'만 사용 가능
- ✓ VHDL 의 3 가지 표현 방식인 Structural Description, Behavioral Description, Data Flow Description 의 방법을 모두 사용하여 설계하여야 한다.
- ✓ ADDER 와 SUBTRACTOR 를 먼저 설계한 후 MULTIPLIER 는 위에서 설계한 ADDER 와 SUBTRACTOR 를 이용하여 제시된 Shift and Add Algorithm 에 의해 SM CHART 를 작성한 후 구현한다.
- ✓ 모든 operation 는 부호를 고려해야한다.
- ✓ 반드시 제공한 TOP 파일 내부에 설계하고, Bitstream 제작시에는 반드시 주어진 Pin-map (.XDC) 파일을 첨부하여 제작할 것. (추후 별도 매뉴얼 제공)

#### 4. 설계 과정

(1) 목표 및 기준설정

- 제시된 기능을 구현하기 위해 필요한 이론 정리
- 덧셈, 뺄셈, 곱셈, 나눗셈 각 연산을 하드웨어로 구현하기 위한 이론 정리
- SM Chart
- 설계 문제 및 기준(방법 또는 설계사양)을 정의하기 위한 SM Chart 를 그린다.

(2) 합성 및 분석

- VHDL 로 각 모듈들을 설계한다.
- 각 모듈 별 시뮬레이션을 통해 원하는 구조대로 만들어졌는지 확인한다.
- 각 모듈을 합성하여 전체 설계를 완성한 후 시뮬레이션을 통해 검증.

(3) 결과

- 이상의 내용을 hard copy 를 통해 제출하고 작성한 코드는 파일로 제출.
- 위의 실험 결과 및 구현 방법을 보고서로 작성한다.
- 작성된 코드를 최종 Bitstream 으로 변환 후 하드웨어 상에서 테스트 진행

## 5. 보고서 작성 (50 점)

### (1) 도입부 (8pt)

- A. 목적 (2pt)
- B. 목표 및 기준 설정 (3pt)

### C. 팀원간 역할 명시 (2pt)

### (2) 합성 및 분석 (7pt)

- A. State Diagram 을 작성하였는가? (3pt)
- B. 시뮬레이션을 통하여 만든 모듈을 검증하였는가? (4pt)

### (3) 결과 및 논의 (35pt)

- A. 결과 도출 (10pt)
- B. 토의 (25pt)
  - i. 제시된 주요 설계요소 및 제한요소를 만족하였는가? (10pt)(설계 요소 및 제한 요소는 위의 Design Constraints 항목 참조)
    - 1. VHDL 의 3 가지 표현 방식인 Structural Description, Behavioral Description, Data Flow Description 의 방법을 모두 사용하여 설계하였는가? (3pt)
    - 2. Shift and add algorithm 에 의해 SM chart 를 작성한후 multiplier 를 설계하였는가? (3pt)
    - 3. 설계한 Adder, subtractor, multiplier 가 부호를 고려하는가? (4pt)
  - ii. Top-down 방식의 설계가 적용되었는가? (3pt)
  - iii. Hierarchical Design 방식의 설계가 적용되었는가? (3pt)
  - iv. Synchronous Design 방식의 설계가 적용되었는가? (3pt)
  - v. Testbench 의 작성 요령 및 임의의 입력에 대해서 안정적으로 동작하는가? (5pt)
  - vi. 기타 논의사항 (4pt)

## 6. 평가방법 (총 100 점)

- (1) 동작여부 - (50 점)
  - A. Hardware 동작 여부 30 점
  - B. Testbench 를 이용하여 시뮬레이션 검증 20 점.
- (2) 보고서 내용 - (50 점)
- (3) 카피 적발 시 모두 0 점 처리
- (4) 참여하지 않은 팀원은 개별 0 점 처리 가능

## 7. 보고서 제출방법 및 데모 검증

**(1) 보고서 제출기한: 2022/12/9 (Fri) 11:59PM (예외 없음)**

- (2) 보고서제출방법: 프로젝트 파일과 보고서를 묶어서 이메일 (dongku713@dgist.ac.kr)로 제출. 제출시 제목을 [EE201\_TermProject] 조번호\_[학번\_이름]\_[학번\_이름]으로 보낼 것.

**(단, 보고서는 .pdf 파일로 변환하여 묶을 것.)**

**(3) 데모검증 기간: 11/28(월) ~ 12/9(금) 까지 기간내에 2 회까지 기회가 주어짐.**

- i. **가급적 11/28(월) ~ 12/2(금) 기간에 1 회이상 데모를 마칠 것을 권장.**
- ii. **2 회안에 성공할 경우 동작점수는 모두 만점. (1 회 실패라고 감점은 없음.)**
- iii. **데모 스케줄은 별도 안내가 나갈 예정임.**

- (4) 조교 Office hours 운영

- i. 화: 2:30 ~ 6:00PM, 목: 3:30 ~ 6:00PM
- ii. 해당 시간이 안될 경우 조교와 메일로 약속 후 질문 수행.
- iii. Nizar Guezzi 조교에게 메일 혹은 방문(E5-322)을 통해 질의 응답진행.

## 8. Demonstration examples



$A = (0111)_2$

$B = (1011)_2$

$Sel = (00)_2$

Addition



$A = (0111)_2$

$B = (1011)_2$

$Sel = (01)_2$

Subtraction



$A = (0111)_2$

$B = (1011)_2$

$Sel = (10)_2$

Multiplication