FPGA를 이용한 디지털 시스템 설계

Final Project – Microprocessor

Professor : 강진구
Assistant : 고현준

개요

- Microprocessor

- Final Project - Simple Microprocessor

1. I/O
2. Register
3. Control
4. ALU
5. 세부 동작 설명

- 설계 요령 & 평가 기준

Microprocessor

- Concept of Microprocessor

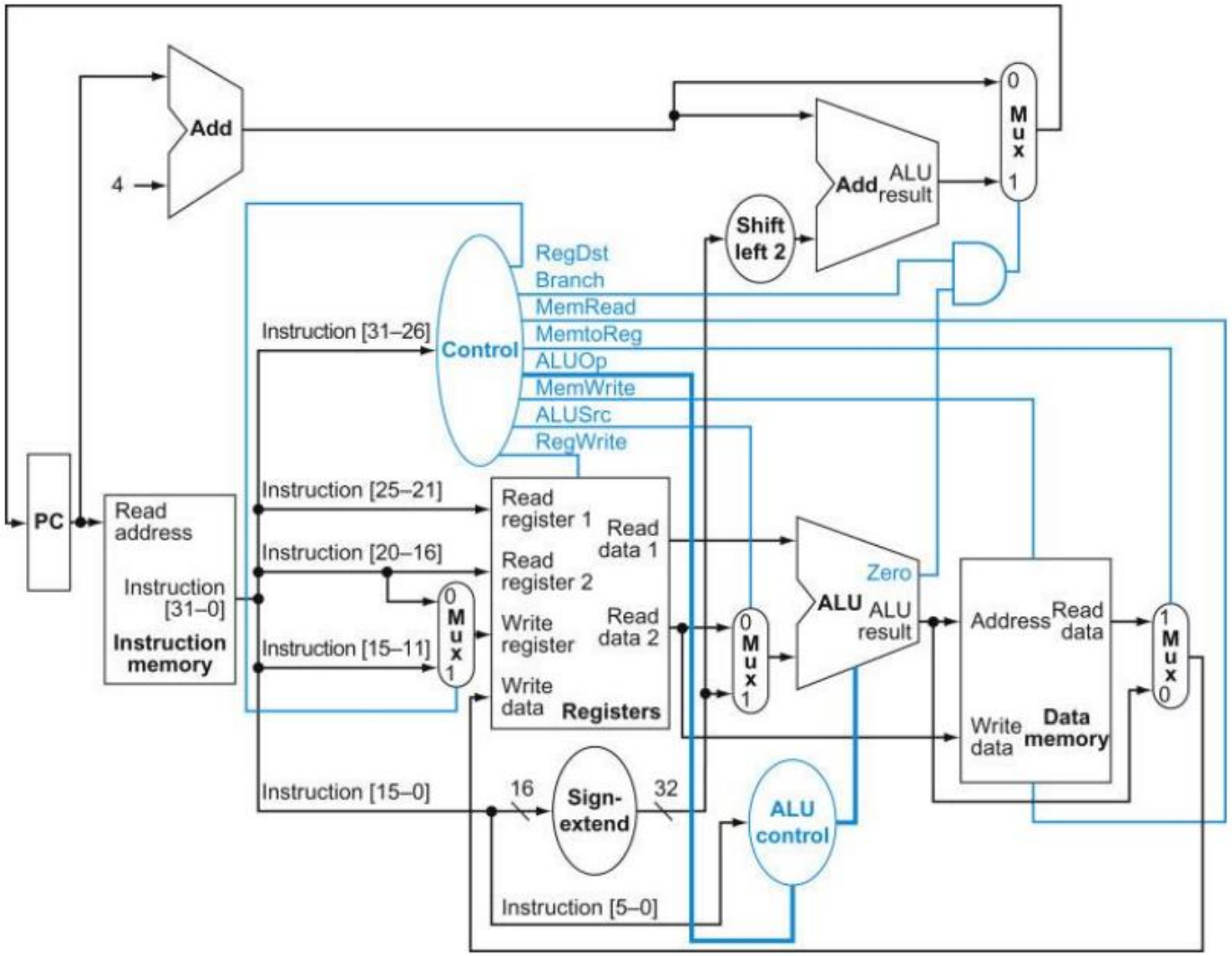
- 컴퓨터의 가장 핵심적인 부분으로 Memory로부터 명령어와 data를 읽어 들이고 (fetch), 이를 해독하여(decode), 특정한 일을 수행(execute)한다.

- 정해진 명령에 따라 레지스터 연산, 산술 연산, 논리 연산 등을 수행한다.

- 명령어를 조합하여 특정 Algorithm을 프로그래밍함으로써 원하는 계산 결과를 얻는다.

Microprocessor

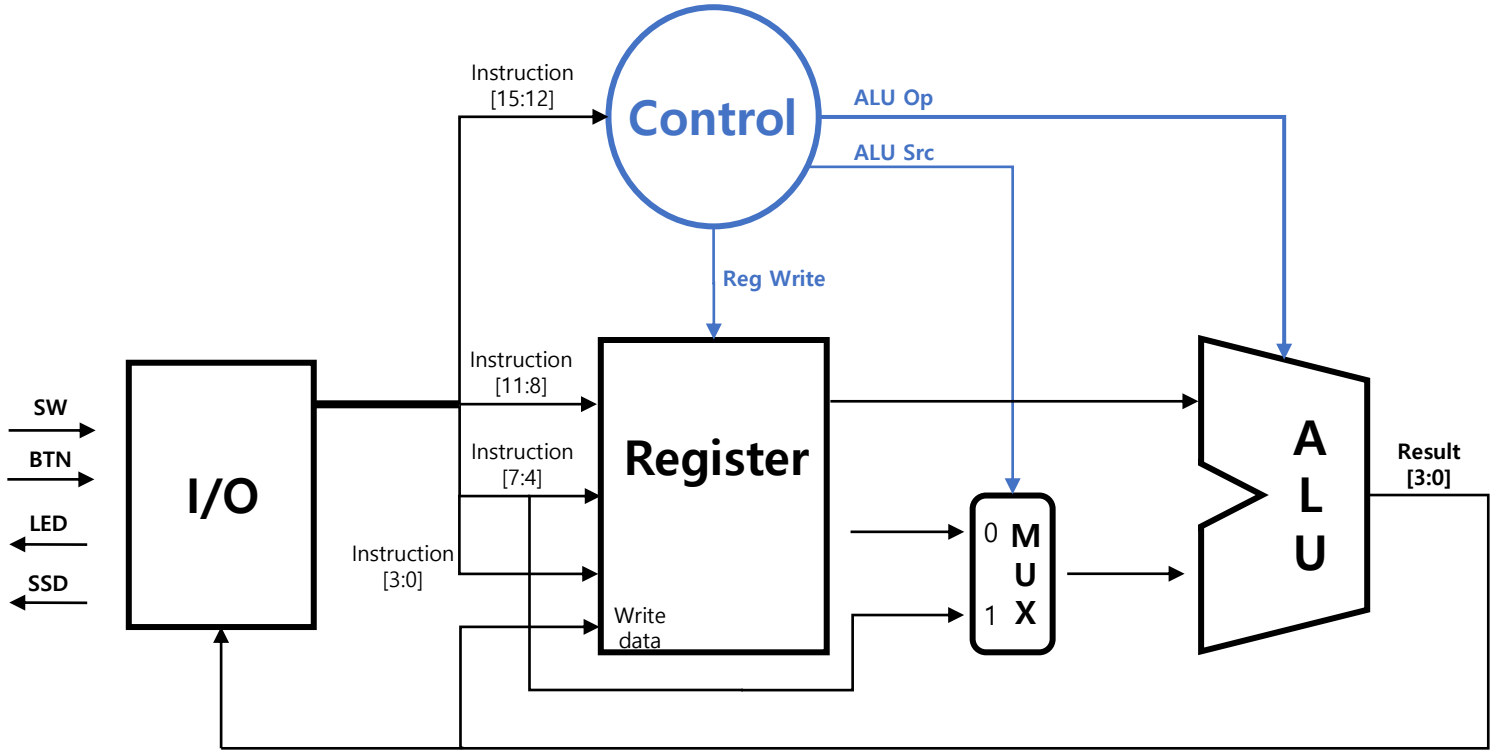
- 32bit MIPS Microprocessor



Simple Microprocessor

- Final Project - Simple Microprocessor

- 사용자 입력으로 명령어를 구성하고 동작하는 Simple Microprocessor 설계

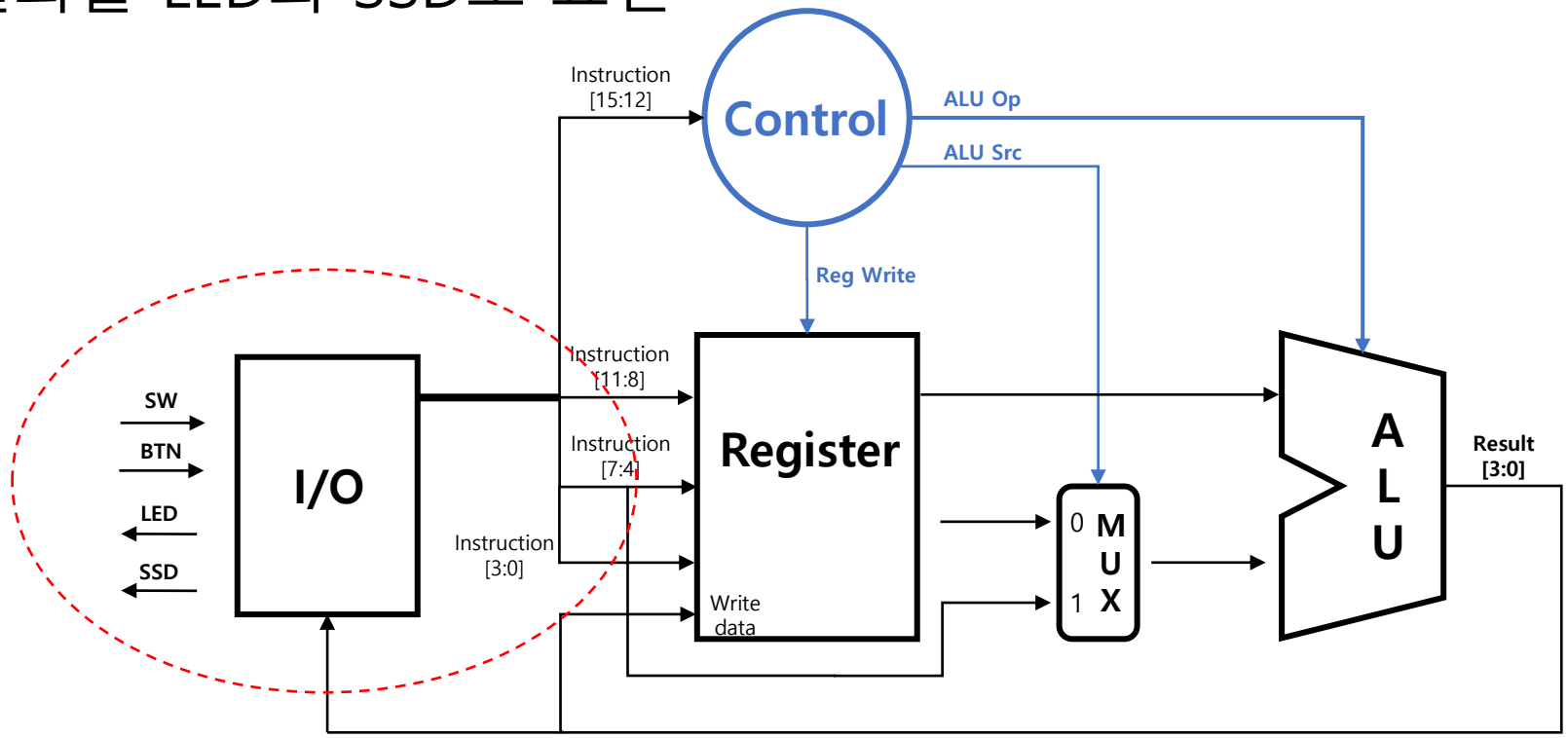


- Microprocessor Block Diagram -

Simple Microprocessor

- 1. I/O Block

- 사용자 입력으로 명령어를 구성 및 전달
- 상태 및 결과를 LED와 SSD로 표현



- Microprocessor Block Diagram -

Simple Microprocessor

- 1.1 I/O Block Input/output(1)

- 각 4개의 Slide Switch, Button을 통해 값을 입력 받아 명령어를 생성해 전달한다.
- Slide Switch는 명령어 구성에 사용된다.
- Button[0]는 누르는 순간만 High가 되어 오동작을 막아야 한다.(Pulse)

- Slide Switch 구성 -

Slide Switch	동작
Slide Switch[0]	명령어 구성
Slide Switch[1]	
Slide Switch[2]	
Slide Switch[3]	

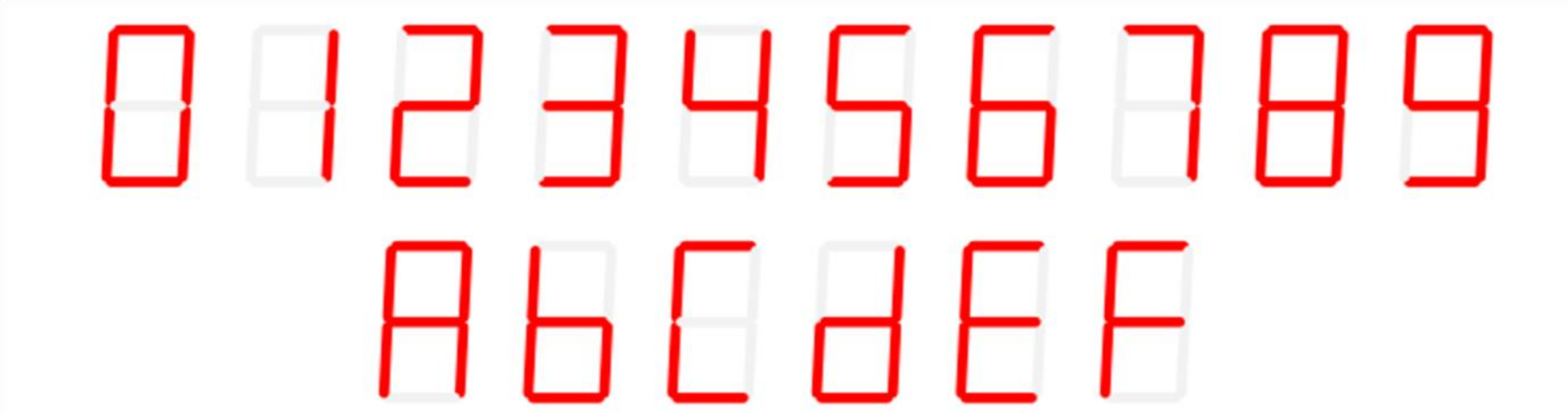
- Button 구성 -

Button	동작
Button[0]	Next State 이동
Button[1]	Done State에서 명령어 표기
Button[2]	필요에 따라 사용
Button[3]	Idle State 복귀

Simple Microprocessor

- 1.2 I/O Block Input/output(2)

- 현재 상태 및 결과를 LED와 SSD를 통해 표기한다.
- 4개의 LED는 현재의 상태를 나타내는데 사용된다.
- 4개의 SSD(7-Segment)는 현재 상태 및 결과를 나타내는데 사용된다.
- 4개의 SSD(7-Segment)는 아래와 같이 0~F(hex)를 표현해야 한다.



- 7-Segment 표현 기준 -

Simple Microprocessor

- 1.3 I/O Block 명령어 구성

- Instruction은 16bit이며, 각 4bit의 Op, Rd1, Rd2, Wr으로 구성된다.

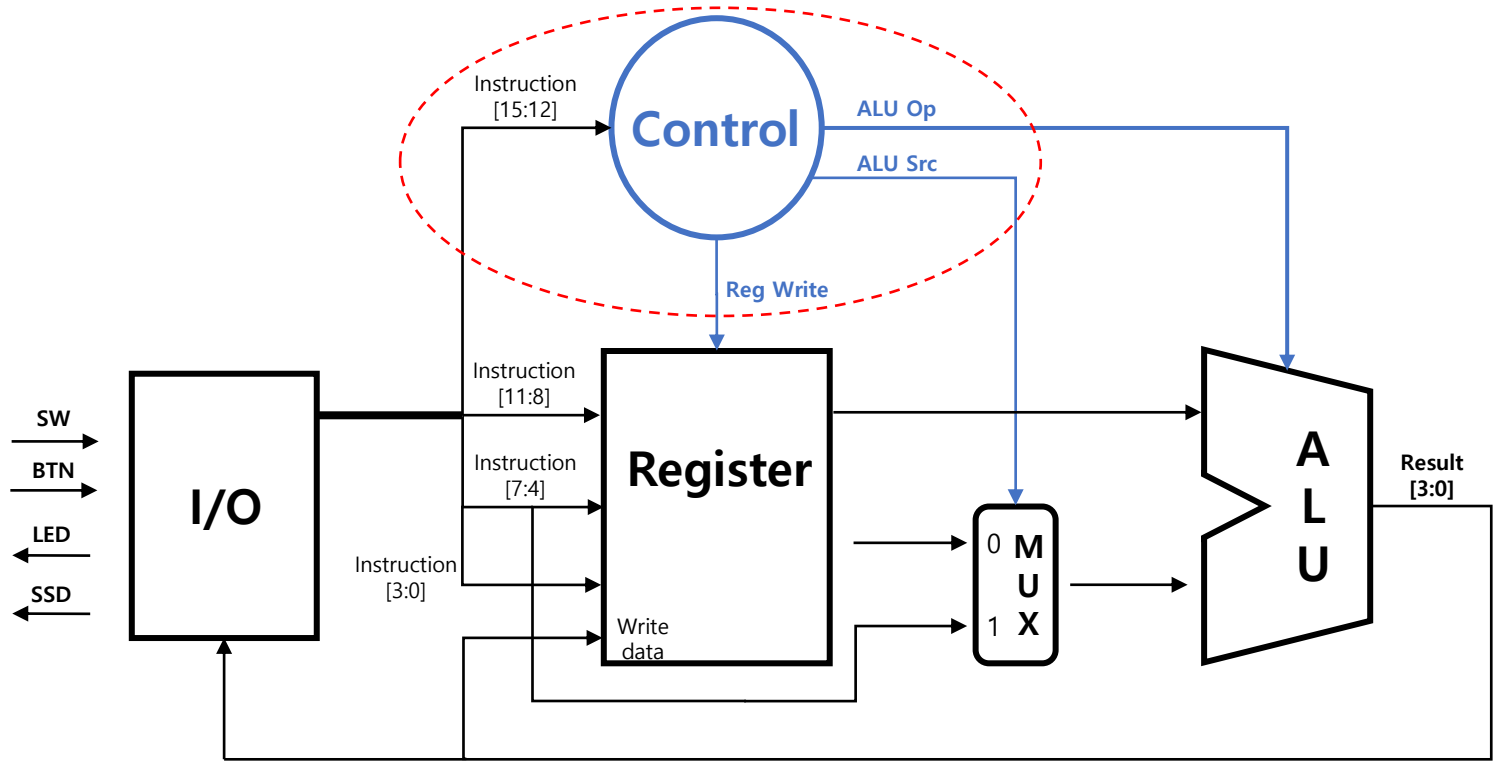
- Instruction 구성 -

Instruction[15:12]	Instruction[11:8]	Instruction[7:4]	Instruction[3:0]
Op	Rd1	Rd2	Wr
연산 코드 (ex. Or, Add)	Read Address	Read Address Or Data	Write Address

Simple Microprocessor

- 2. Control

- 명령어의 'Instruction[15:12]'(Opcode)를 디코딩해 Control 신호를 출력한다.



- Microprocessor Block Diagram -

Simple Microprocessor

- 2. Control Block의 Control 신호

- Opcode를 디코딩해 아래 3개의 Control 신호를 출력한다.

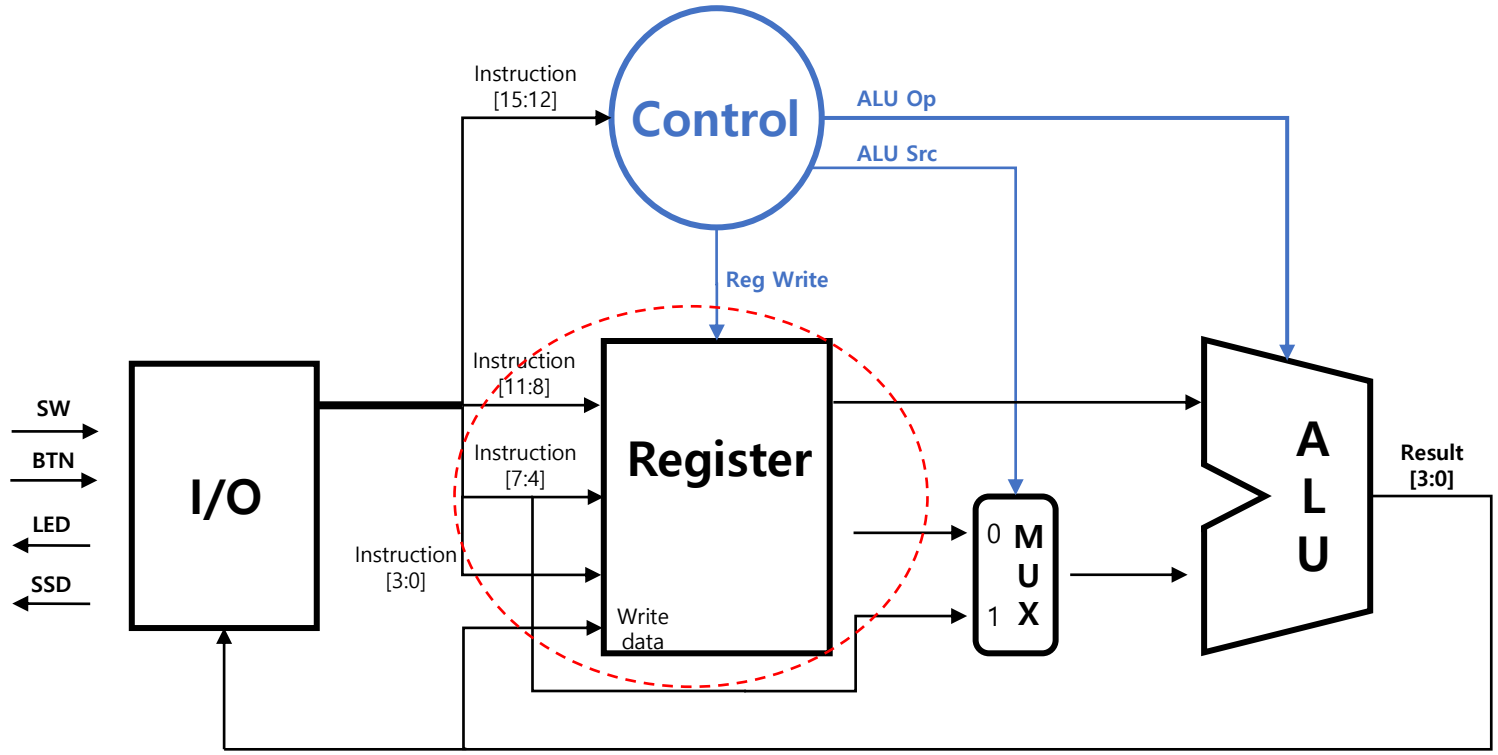
- Control Signal 구성 -

Control Signal	동작
ALU Op	ALU 연산 동작 제어
ALU Src	ALU Input2의 소스 선택 제어
Reg Write	Register에 들어온 Data Write 제어

Simple Microprocessor

- 3. Register

- Data를 저장, Read, Write 동작 수행



- Microprocessor Block Diagram -

Simple Microprocessor

- 3. Register

- Register는 16개의 4bit Array로 구성된다.

- **Rd1**, **Rd2**의 주소에 해당하는 값을 반환한다.

- 'Reg Write'가 High일 때, **Wr**('Instruction[3:0]')의 주소에 Write data값을 저장한다.

- 주소 '0'의 Data는 항상 0으로 고정되며, 각종 연산의 기준이 된다.

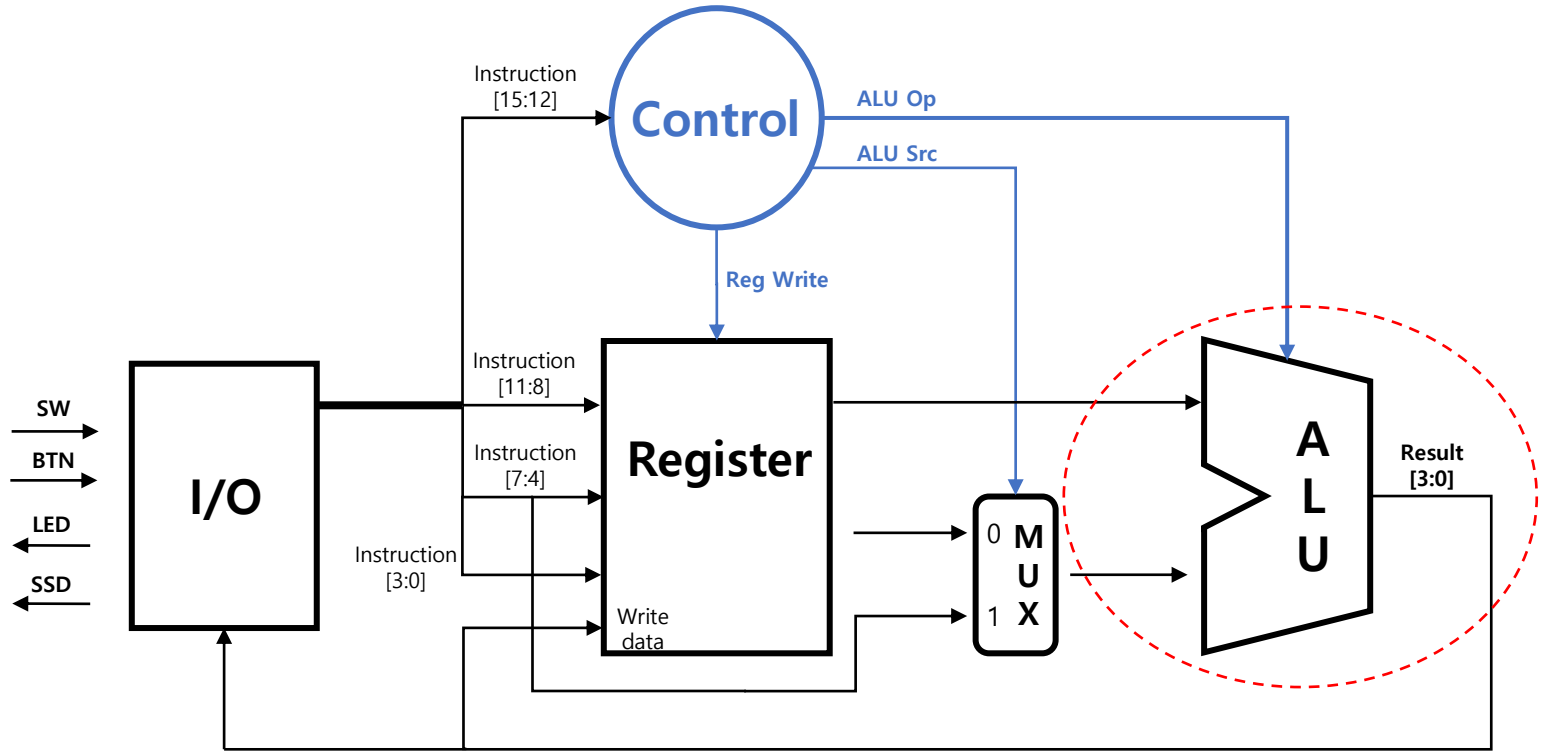
- Register 구성 -

ADDR	4bit Data
15	4'b????
14	4'b????
…	…
…	…
1	4'b????
0	**(고정)4'b0000**

Simple Microprocessor

- 4. ALU

- ALU Op를 기준으로 두 입력간 연산 후 결과를 출력한다.



- Microprocessor Block Diagram -

Simple Microprocessor

- 4. ALU 동작

- Microprocessor는 Instruction에 따라 아래 16가지 동작을 한다.
- ADD, SUB 연산시에는 4bit Data를 2의 보수로 취급한다.

- Instruction 종류 및 동작 -

Instruction[15:12]	Opcode	동작
0000(0)	NOP	Work Nothing
0001(1)	Write	Rd2 Data **Write** to Wr addr
0010(2)	Read	Rd1 addr's Data **Read**
0011(3)	Copy	Rd1 addr's Data **Copy** to Wr addr
0100(4)	NOT	Rd1 addr's Data **Not** Operation
0101(5)	AND	Rd1 addr's Data, Rd2 addr's Data **AND** Operation
0110(6)	OR	Rd1 addr's Data, Rd2 addr's Data **OR** Operation
0111(7)	XOR	Rd1 addr's Data, Rd2 addr's Data **XOR** Operation
1000(8)	NAND	Rd1 addr's Data, Rd2 addr's Data **NAND** Operation
1001(9)	NOR	Rd1 addr's Data, Rd2 addr's Data **NOR** Operation
1010(10)	ADD	Rd1 addr's Data, Rd2 addr's Data **ADD** Operation
1011(11)	SUB	Rd1 addr's Data, Rd2 addr's Data **SUB** Operation
1100(12)	ADDI	Rd1 addr's Data, Rd2 Data **ADD** Operation
1101(13)	SUBI	Rd1 addr's Data, Rd2 Data **SUB** Operation
1110(14)	Left Shift	Rd1 addr's Data << 'Rd2 Data' **Left Shift** Operation
1111(15)	Right Shift	Rd1 addr's Data >> 'Rd2 Data' **Right Shift** Operation

Write to Wr addr

Simple Microprocessor

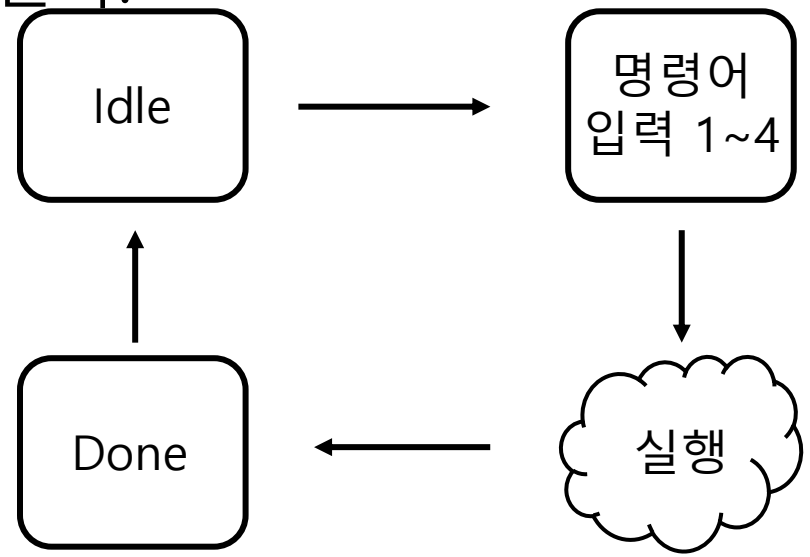
- 5. 세부 동작 설명(1) – 전체 동작

- Microprocessor의 동작은 'Idle', '명령어입력 1~4', '실행', 'Done'으로 구성된다.

- '실행'을 제외한, 각 상태는 Button[0]에 의해 변경되어야 한다.
- '실행'을 제외한, 각 상태는 LED에 의해 구분되어야 된다.

- '실행'에서의 Button, LED 사용은 자율

- **100MHz로 동작하도록 구성한다.**



- Microprocessor 동작 상태 -

Simple Microprocessor

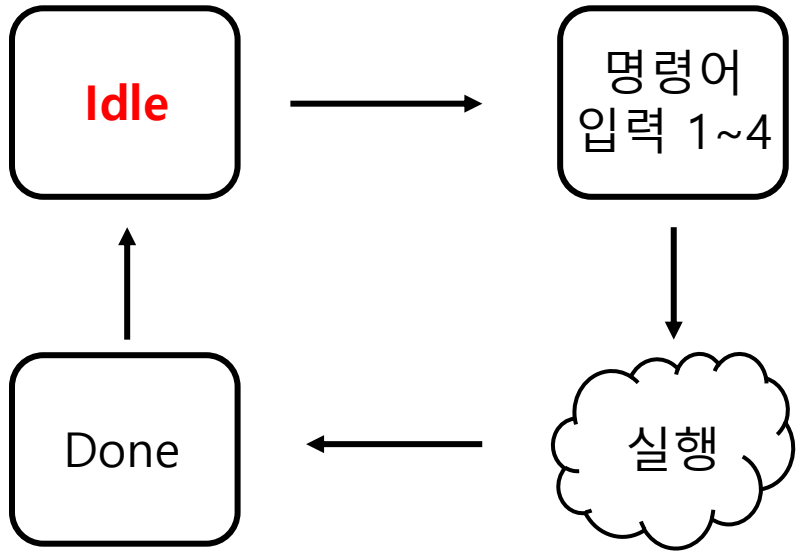
- 5. 세부 동작 설명(2) – Idle

- 'Idle'상태는 어떠한 동작도 하지 않는 대기 상태이다.
- 다른 상태일 경우 Button[3]에 의해 'Idle'상태로 복귀한다.
- Button[0]에 의해 '명령어 입력 1' 상태로 변경된다.
- LED는 모두 off, SSD는 0를 출력한다.

- Example

LED ○ ○ ○ ○

SSD 0 0 0 0



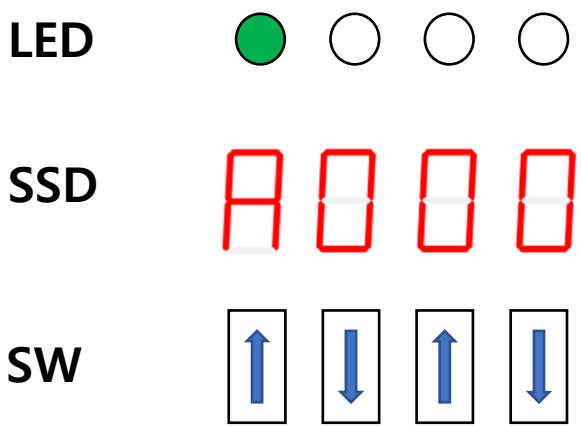
- Microprocessor 동작 상태 -

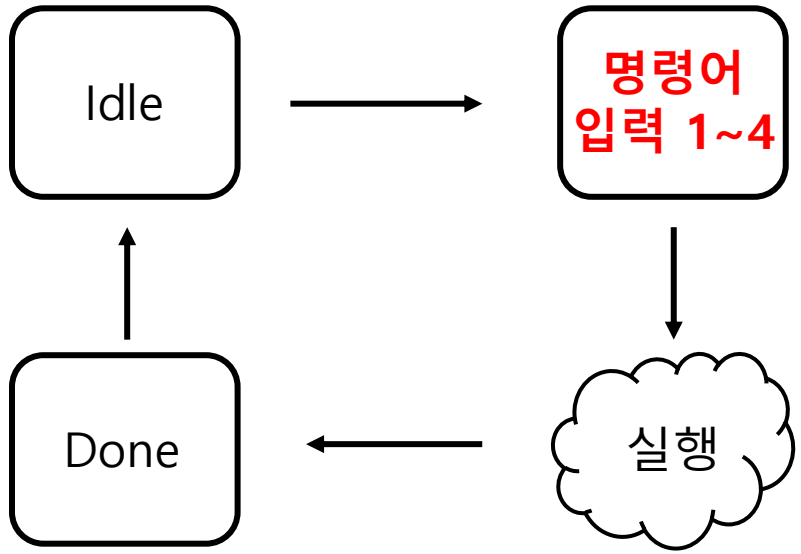
Simple Microprocessor

- 5. 세부 동작 설명(3) – 명령어 입력 1

- '명령어 입력'상태는 Microprocessor를 동작 시킬 명령어를 구성하는 상태이다.
- '명령어 입력 1'상태는 'Instruction[15:12]' Opcode를 구성한다.
- Button[0]에 의해 '명령어 입력 2' 상태로 변경된다.
- LED[3] ON, 1번째 SSD는 SW를 따라 출력한다.

- Example





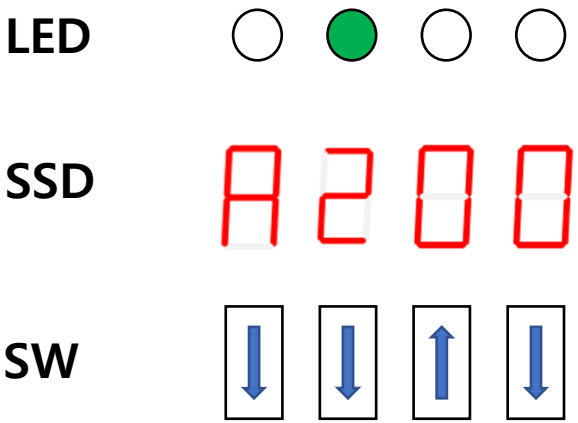
- Microprocessor 동작 상태 -

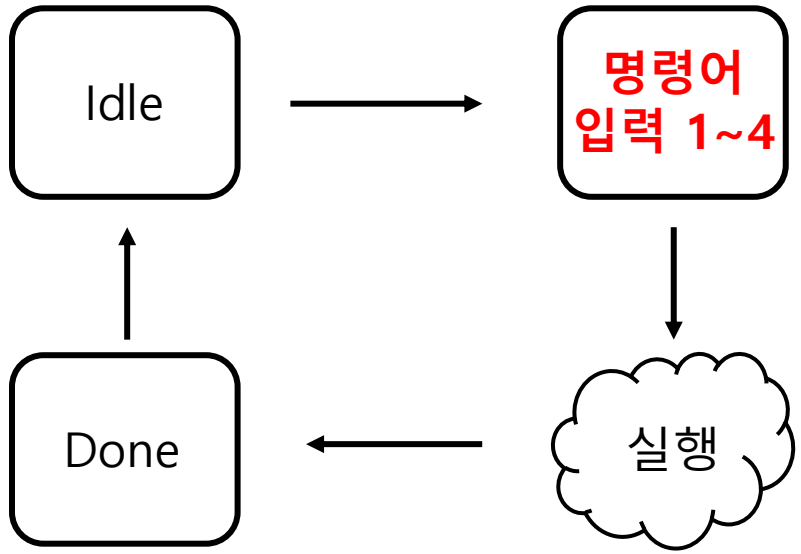
Simple Microprocessor

- 5. 세부 동작 설명(4) – 명령어 입력 2

- '명령어 입력 2'상태는 'Instruction[11:8]' **Rd1**를 구성한다.
- Button[0]에 의해 '명령어 입력 3' 상태로 변경된다.
- LED[2] ON, 2번째 SSD는 SW를 따라 출력한다.
- 1번째 SSD는 '명령어 입력 1'에서의 값을 출력한다.

- Example





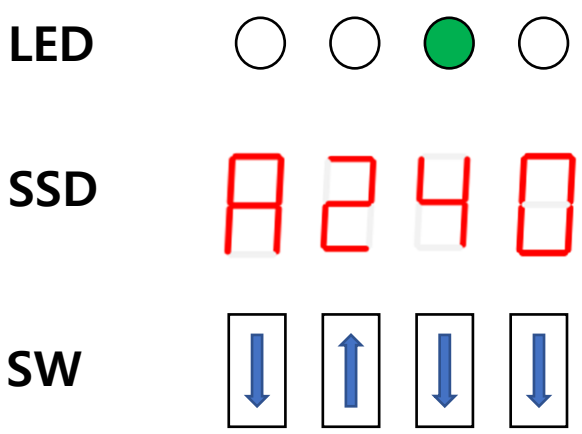
- Microprocessor 동작 상태 -

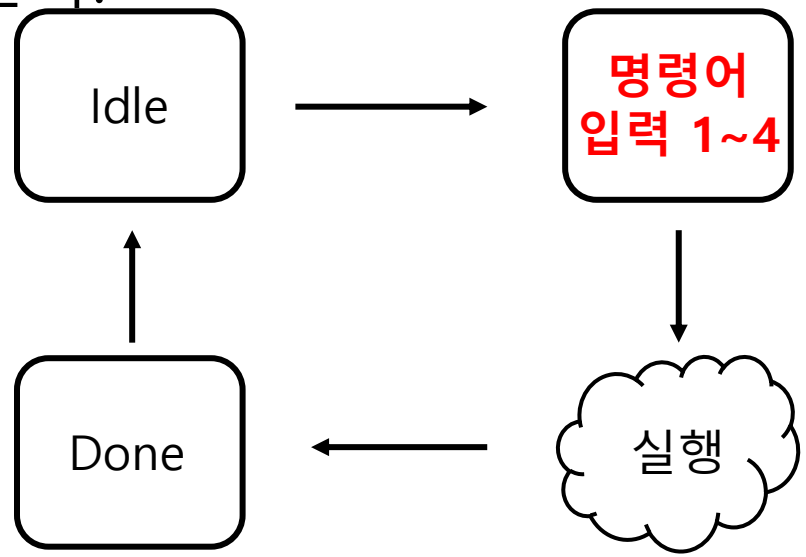
Simple Microprocessor

- 5. 세부 동작 설명(5) – 명령어 입력 3

- '명령어 입력 3'상태는 'Instruction[7:4]' **Rd2**를 구성한다.
- Button[0]에 의해 '명령어 입력 4' 상태로 변경된다.
- LED[1] ON, 3번째 SSD는 SW를 따라 출력한다.
- 1~2번째 SSD는 '명령어 입력 1,2'에서의 값을 출력한다.

- Example





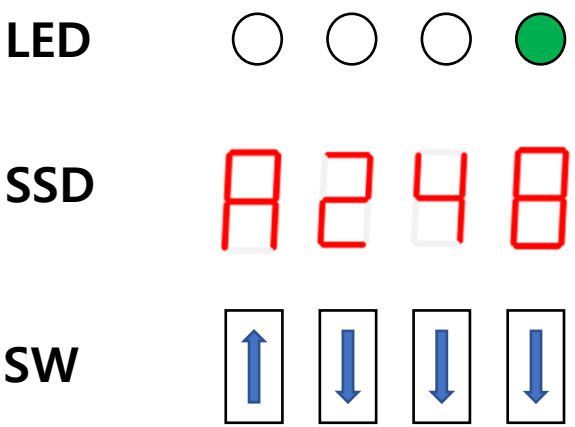
- Microprocessor 동작 상태 -

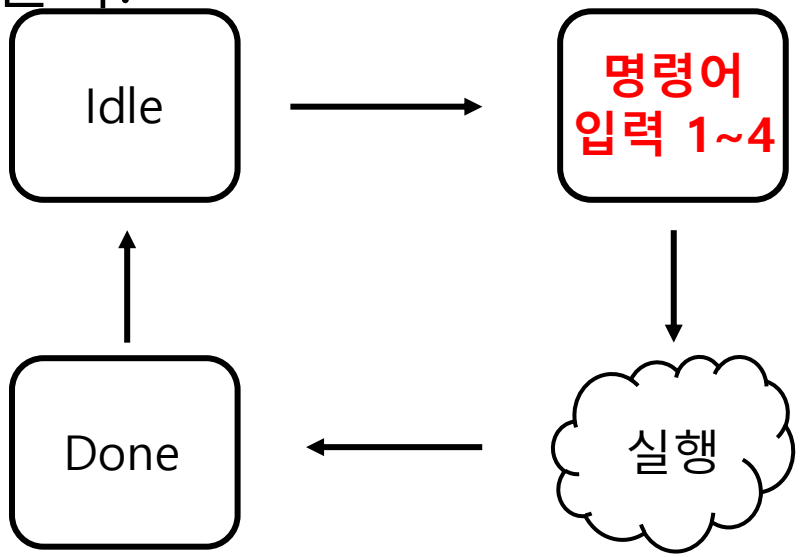
Simple Microprocessor

- 5. 세부 동작 설명(6) – 명령어 입력 4

- '명령어 입력 4'상태는 'Instruction[3:0]' **Wr**를 구성한다.
- Button[0]에 의해 '실행' 상태로 변경된다.
- LED[0] ON, 4번째 SSD는 SW를 따라 출력한다.
- 1~3번째 SSD는 '명령어 입력 1~3'에서의 값을 출력한다.

- Example





- Microprocessor 동작 상태 -

Simple Microprocessor

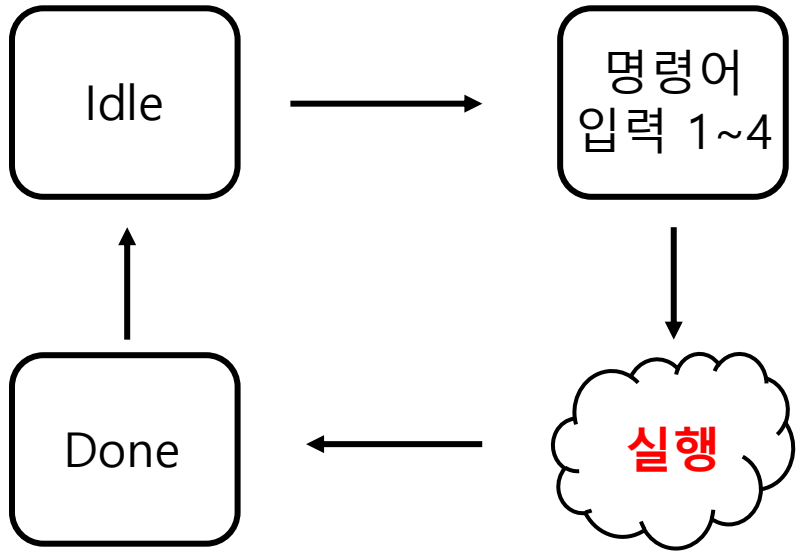
- 5. 세부 동작 설명(7) – 실행

- '실행'상태는 '명령어 입력 1~4'에서 구성한 명령어에 따라 실행되는 상태이다.
- 명령어에 따라 Read, Operation, Write등의 동작이 실행된다.
- 적절한 동작 후 'Done' 상태로 변경된다.
- LED 및 SSD의 출력이 필요할 경우 적절히 구성한다.

- Example

Instruction = 16'h**A248**
Opcode : A(ADD 연산), **Rd1** : 2, **Rd2** : 4 ,**Wr** : 8

=> 레지스터의 2번, 4번 addr의 data **ADD 연산**
연산 후 결과 8번 addr에 Write



- Microprocessor 동작 상태 -

Simple Microprocessor

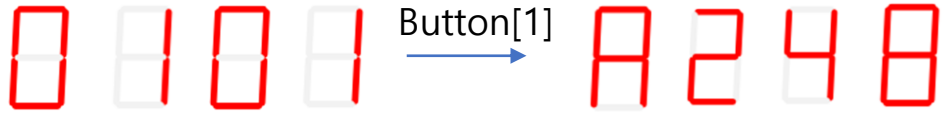
- 5. 세부 동작 설명(8) – Done

- 'Done'상태는 명령어에 대한 Microprocessor 동작 종료 후 결과를 출력한다.
- Button[0]에 의해 'Idle' 상태로 변경된다.
- LED는 모두 ON, SSD는 실행 결과를 2진수로 출력한다.
- Button[1]을 누르고 있으면 실행한 명령어를 SSD에 표기한다.

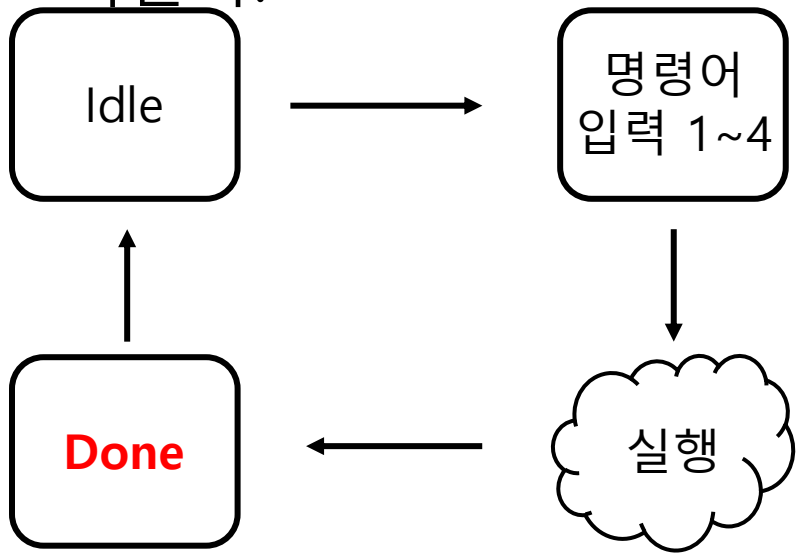
- Example

LED

SSD



*2번 addr의 data = 3, 4번 addr의 data = 2로 가정
ADD 연산 결과인 5(0101)을 SSD에 출력



- Microprocessor 동작 상태 -

설계 요령 & 평가 기준

- 설계 참고 사항

1. 현 자료에서 설명, 요구하는 블록 및 동작의 구현을 최우선으로 한다.
2. 명시되지 않은 예외사항에 대해서는 적절히 처리하되, 그 이유가 합리적이야 한다. (ex. Overflow)
3. 설계 시 추가, 변경되는 부분이 있다면 변경된 Block Diagram과 동작에 대한 충분한 설명이 있어야 한다.

- Final Project 평가 방법

1. FPGA 보드 시연 (50%)
2. Final Project 보고서 제출 (50%)

설계 요령 & 평가 기준

- FPGA 보드 시연

1. 동작의 정확도
2. 모듈 설계의 완성도

- **평가 일정 추후 공지 (15~16주차 예정)**
- 시연 후 보드 반납

- 보고서 작성 필수 사항

1. Microprocessor 및 설계하는 Microprocessor 대한 소개
2. Project 진행 내용 : 각 Block 별 Code 및 Design 방안
3. Project 결과 및 분석 : RTL결과 및 합성 결과, Simulation 파형, H/W 디버깅
4. 고찰

- 16주차 목요일(12/14) 17:00 보고서 마감

END

Thank you ! :D