

# 6 장

## 세그먼트 (7-Segment) 제어기



### 1. 실험 목표

---

- 디지털 디스플레이 7-Segment 의 구조와 제어 방법에 대해 이해한다.
- 7-Segment 제어기 구동을 위한 디코더 구조를 설계한다.
- 디지털 숫자 표시 응용을 위한 디스플레이 전체 회로를 설계한다.

### 2. 실험 이론

---

#### 가) 7-Segment 제어기

디지털 시스템에서는 단순한 숫자나 영문자를 표시하기 위하여 7 개의 LED 를 세그먼트로 구분 조합하여 해당 LED 만을 ON 시켜 숫자를 표시하게 된다. 단순한 디지털 기기는 이러한 7-

세그먼트 구조의 디스플레이 구조를 이용하여 숫자나 영문자를 표시한다. 따라서 7-세그먼트 구조는 디지털 회로에서 숫자를 표시하기 위하여 가장 많이 사용하는 소자이며, 7 개의 LED 에 해당하는 각 세그먼트를 a, b, c, d, e, f, g 로 나누어 각각의 위치를 지정하여 해당 위치 LED 를 ON/OFF 하여 숫자를 표시한다. 다음의 [그림 6-1]과 같이 7-세스먼트로 구성하여 제어한다.

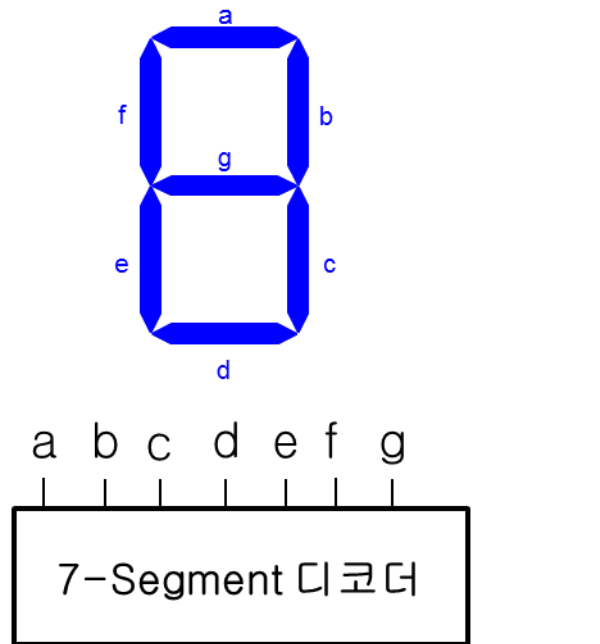


그림 6-1: 7-세그먼트 숫자 표시기의 구조.

세그먼트 상에 숫자를 나타내기 위해서는 각 부분의 LED 에 ON/OFF 신호를 제공하여야 하며, 숫자 표시의 예로 숫자 7 을 표시하기 위해서는 f, a, b, c 를 ON 시키고, 나머지는 OFF 시키며, 숫자 3 의 경우는 a, b, g, c, d 의 LED 를 ON 시키고 나머지는 OFF 시키게 된다. 8 의 경우는 모두 켜야 한다. 주어진 십진수 값에 대하여 7-세그먼트에 알맞게 배치하기 위하여 디코딩 해주는 회로를 7-세그먼트 제어기라고 한다. 0 에서부터 9 까지의 숫자를 표현하기 위한 디코더의 입력 값과 출력 값은 [그림 6-2]와 같이 정의된다.

	입력				7-Segment 출력						
	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

그림 6-2: 십진수에 대한 디코딩 결과 값.

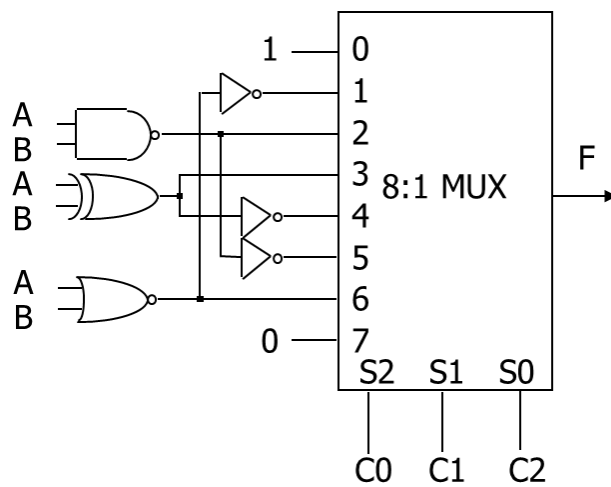
#### 나) Logical Function Unit

CPU 를 구성하는 핵심 모듈인 ALU (Arithmetic Logic Unit)은 연산을 위한 연산기와 논리 동작을 위한 논리기의 조합으로 구성된다. 일반적인 컴퓨팅 환경에서 사용되는 가장 기본적인 논리동작을 수행하는 Logical Function Unit 을 설계한다. 이 실험에서 지원하고자 하는 기본 논리동작은 [그림 6-3]과 같이 정의된다. 제어신호 C0, C1, C2 는 해당 기능을 선택하는 제어신호이며, 기능은 해당 논리 동작을 의미한다. 따라서 1-비트 논리유닛의 설계는 입력 A, B 에 대하여 주어진 제어신호에 따라 논리 연산을 수행한다. 간단한 회로 설계를 위하여 8:1

MUX 를 이용하여 각 제어신호 값에 따라 해당하는 논리의 게이트 (AND, OR, XOR 등)를 입력으로 선택하여 구동되도록 설계하며, [그림 6-4]와 같은 진리표에 따라 수행하데 된다..

C0	C1	C2	기능
0	0	0	1
0	0	1	$A + B$
0	1	0	$(A \cdot B)'$
0	1	1	$A \text{ XOR } B$
1	0	0	$A \text{ XNOR } B$
1	0	1	$A \cdot B$
1	1	0	$(A + B)'$
1	1	1	0

그림 6-3: 논리 유닛의 기능 정의표.



C0	C1	C2	A	B	F
0	0	0	0	0	1
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	0

그림 6-4: Logical Unit 동작 진리표와 회로구성.

### 3. 예비보고서 작성

■ [그림 6-2]의 7-세그먼트 디코딩 회로를 Verilog HDL 로 표현하시오.

■ 영문자(A~F)에 대한 디코딩 결과값을 [그림 6-2]의 형식으로 구하시오. (진리표를 작성하시오.)

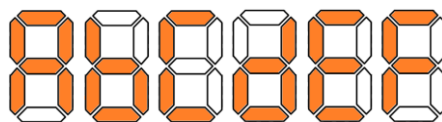


그림 6-4: 영문자 표기에 대한 예시 (A~F)

■ [그림 6-4]에 대한 회로를 Verilog HDL 로 표현하시오. (8:1 MUX 는 이미 구현되어 있다고 가정하고, 해당 모듈에 대한 인스턴스를 활용하는 측면에서 작성해도 무방함. 이전 시간에서 구현했던 8:1MUX 를 그대로 사용하는 것도 무방.)

## 4. 실험에 필요한 기기

---

■ 실험용 PC

■ MAX PLUS II 소프트웨어 / Quartus II-Modelsim 소프트웨어

## 5. 실험 내용과 과정

---

- ① Verilog HDL 로 설계한 코드를 프로젝트 상에서 구현한다.
- ② FPGA 로 program 하기 전, 미리 동작을 검증하고 싶다면 testbench 작성 및 시뮬레이션을 수행한다.
- ③ Synthesis 가 이루어진 결과물을 FPGA 로 program 하는 과정은 추가적인 자료를 통해 습득한다.
- ④ FPGA 상에서 여러 가지의 입력들에 대응해, 올바른 결과 값을 출력하는지 확인해본다.

## 6. 실험 결과 보고서

---

- ① 16 진수를 표현할 수 있도록 A 에서 F 까지의 표시 방법(대, 소문자 이용가능)을 생각해 보고, 이를 [그림 6-2]처럼 진리표를 구하시오. 그리고 이것을 Verilog HDL 을 통해 구현하시오.
- ② 위의 ① 에서 작성한 Verilog HDL 코드를 기반으로, FPGA 상에서 키 입력에 대해 7-segment 에 값을 표시해주는 장치를 구현하시오. (결과보고서에는 Verilog HDL 코드와 작동하는 과정 중 일부가 담긴 사진이 첨부되어야 합니다.)
- ③ 예비 보고서에서 구현한 [그림 6-4]의 회로를 Simulation 을 통해 검증하시오.

- ④ YONSEI 전광판 만들기: [그림 6-5]와 같이 일정 시간을 간격으로 "YONSEI"라는 글자가 오른쪽에서 왼쪽으로 이동하는 전광판을 FPGA 상에서 동작하도록 구현하시오. (전원이 꺼질 때까지 아래와 같은 패턴이 계속 반복되어야 합니다. 그리고 결과보고서에는 Verilog HDL 코드와 작동하는 과정 중 일부가 담긴 사진이 첨부되어야 합니다.)

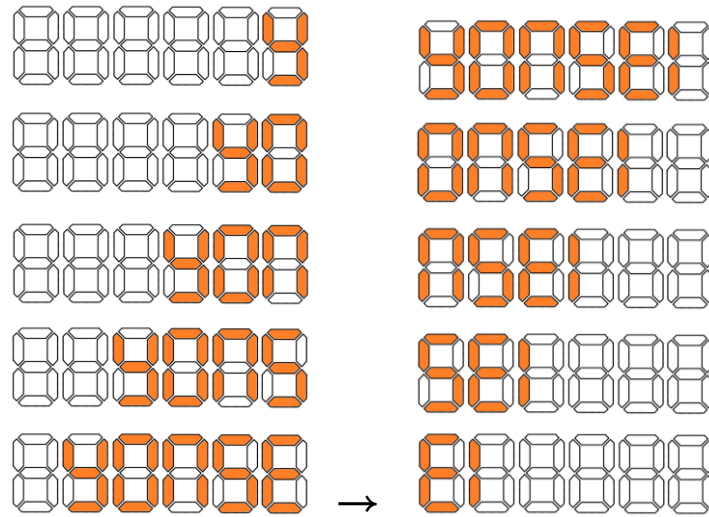


그림 6-5. YONSEI 전광판