**Digital Logic Circuit Design Lab**

Combinational Logic Circuit Lab

Name/ID: 안견힐/21900416

Partner Name/ID: 진가람/21900727

Date: 2023.04.27

# **Problem 1**

## Introduction

본 문제는 1bit Comparator를 직접 Logic gate를 사용하여 설계한다. 본 회로는 A와 B 각각의 input이 들어왔을 때 이 둘의 크기를 비교하고 비교값에 대한 결과를 출력하는 논리 회로이다. 이는 어떠한 두 값을 비교해야하는 기기에 들어가며 대부분의 기기에 들어가 있음을 알 수 있다. 본 문제를 해결하기 위해 입력값과 출력값의 Truth Table을 작성하고 이에 기반하여 카르노 맵을 작성한다. 이를 기반으로 논리 회로를 설계하고 직접 BreadBoard에 회로를 만들고 결과가 정상적으로 나오는지 확인한다.

도표이(가) 표시된 사진

자동 생성된 설명

### Circuit Components

* Breadboard, Jumper wire
* Input: Arduino for 5V/0V source, 4-pin tactile switch(with Pull-down)

2\*LEDs for Input Indication

* Output: 3\*LED, for each output pins.
* Logic chip: 4-CH 2-Input NAND gate (74HC00) x 2

## Design Process

### Truth Table

A와 B를 비교하였을 때 A와 B는 1bit이므로 1과 0을 비교하여 A=B, A>B, A<B로 출력을 구분하여 Truth Table을 작성할 수 있다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | | **Output** | | |
| **A** | **B** | **A<B** | **A=B** | **A>B** |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

Table 1. Truth table of 1bit Comparator

### Karnaugh Map

|  |  |  |
| --- | --- | --- |
|  |  |  |
| A<B | A=B | A>B |

Fig 1. Karnaugh Map of 1 bit Comparator

Truth Table을 통해 위와 같은 Karnaugh Map을 설계할 수 있다.

카르노 맵을 통해 도출한 SOP는 각각

이다.

도출된 SOP를 통해 다음의 회로 설계가 가능하다.

### Simulation Circuit Design

**도표, 개략도이(가) 표시된 사진

자동 생성된 설명**

Fig 2. Comparator Logic Circuit

Comparator Circuit을 다음과 같이 Nand Gate 7만을 사용하여 만들 수 있다. NAND GATE는 AND GATE의 출력단에 버블이 붙은 GATE로 AND GATE와 반대의 출력을 가진다. NOT GATE는 NAND GATE에 같은 입력값이 들어가는 것으로 구현할 수 있다. A=B일때의 LOGIC은 A(XOR)B 였는데 우리는 XOR GATE를 입력핀과 가장 가까운 NAND GATE 4개처럼 만들 수 있다. 이를 통해 입력핀 A,B에 따른 Z0, Z1, Z2의 출력을 얻을 수 있는 Coparator를 설계할 수 있다. 또한 시뮬레이션을 진행하였을 때 A, B값에 따라 A,B의 값을 비교하는 출력값이 나오는 것을 알 수 있다.

## Results and Discussion

### Demo Video

[Video Link](https://youtu.be/OugyayHUy2E)

### Results

Quad Nand 소자는 각각 4개의 Nand Gate 역할을 한다. 이에 우리는 본 소자를 2개 사용하여 회로를 구성하였다.

버튼을 누르지 않은 상태에서는 가운데 다이오드가 켜지고, 버튼 A를 눌렀을 때는 첫 번째 다이오드가 켜지고, 버튼 B를 눌렀을 때는 세 번째 다이오드가 켜진다. 두 버튼을 동시에 눌렀을 때는 가운데 다이오드가 켜진다.

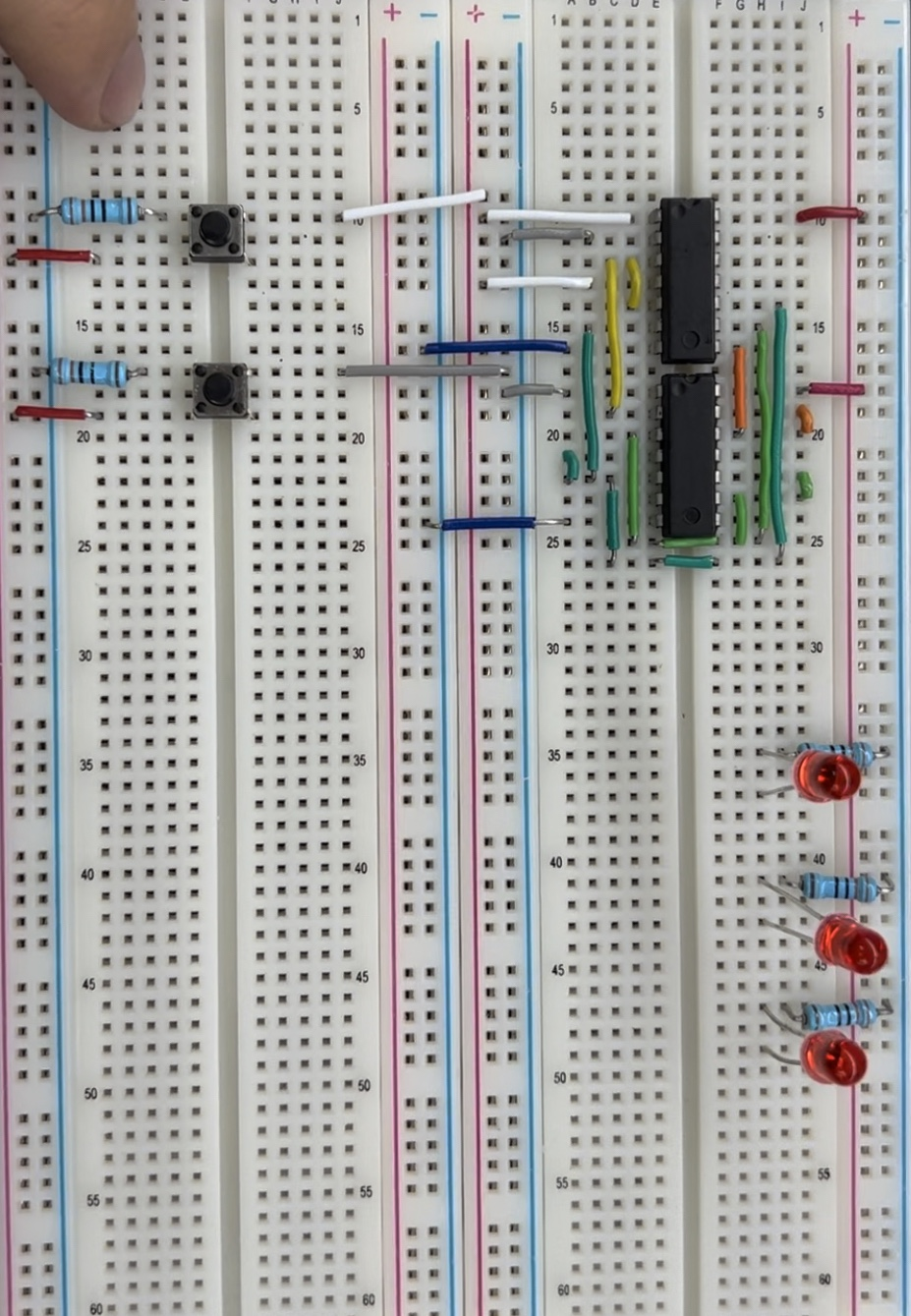


Fig 3. Actual circuit images

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

Fig 4. Circuit Acting Images

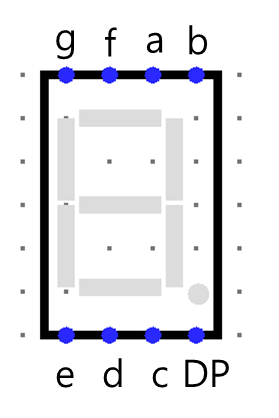
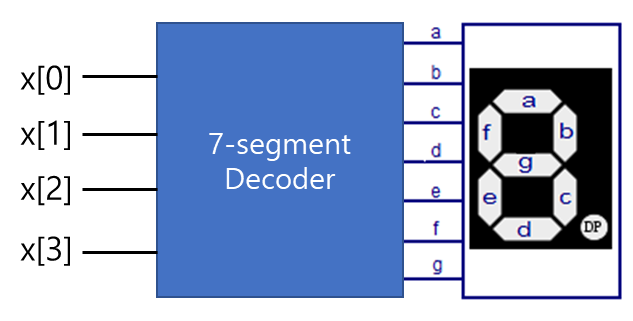
### Discussion

Logisim을 사용하여 Nand Gate로만 Comparator를 설계한다면 10개가 넘는 Nand Gate가 필요하지만 같은 역할을 수행하는 Nand Gate를 제거하고 다시 설계한다면 7개의 Nand Gate로 설계할 수 있다. 이에 2개의 Quad Nand 소자를 사용하여 위와 같은 회로를 설계할 수 있다.

# **Problem 2**

## Introduction

본 문제에서는 7개의 LED가 있는 숫자 패널을 작동시키기 위한 7-Segment Decoder를 직접 설계하고자 한다. 7-Segment Decoder는 엘리베이터, 시계, 계산기 등 많은 전자 기기에서 숫자를 표현하기 위해 사용된다. 이는 4개의 input과 7개의 output을 가진 7-Segment Decoder를 이용하여 숫자 패널을 작동시킨다. x[0],x[1],x[2],x[3] 가 0000 일 때는 0을 표시하고, 0001 일 때는 1, 0010 일때는 2를 표시한다.



## Design Process

### Truth Table

입력을 x[3], x[2], x[1], x[0], 출력은 a,b,c,d,e,f,g로 출력이 HIGH신호일 때 숫자 패널 각각의 LED를 점등한다. 또한 숫자 패널은 0부터 9까지만 유효하므로 입력이 1010 이후 부터는 Non-Care Value로 나타낸다.

**테이블이(가) 표시된 사진

자동 생성된 설명**

Table 1. Truth table of 7-Segment Decoder

### Karnaugh Map

|  |  |
| --- | --- |
| **텍스트, 시계이(가) 표시된 사진  자동 생성된 설명** | **텍스트, 시계, 클립아트이(가) 표시된 사진  자동 생성된 설명** |
| 𝑎 | 𝒃 |
| 텍스트, 시계이(가) 표시된 사진  자동 생성된 설명 |  |
| 𝑐 | 𝑑 |
|  |  |
| 𝑒 | 𝑓 |
|  |
|  |

Table 2. Karnaugh Map of 7-segment Decoder

위의 Truth Table을 통해 다음과 같은 카르노 맵을 만들 수 있다.

### Simulation Circuit Design

도표, 개략도이(가) 표시된 사진

자동 생성된 설명도표, 개략도이(가) 표시된 사진

자동 생성된 설명

Fig 1. 7-segment circuit (Logisim, Circuit Diagram)

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Table 3. 7-Segment Circuit Output (Logisim)

위와 같이 입력값에 변화를 주면 7-segment circuit에서 숫자판의 출력값이 0부터 9까지 변화함을 확인할 수 있다.

## Results and Discussion

### Results

7-segment Decoder를 위의 Circuit과 같이 AND gate, OR gate, NOT gate로 구성하였다.

### Discussion

4개의 input과 7개의 output을 가진 7-segment decoder를 직접 설계하였다.

X[0]부터 X[3]까지 input값이 변화하면서 a부터 g의 출력값이 변화하는 것을 확인하였으며 10부터 15까지의 값은 Don’t Care Value로 처리해야함을 알 수 있었다..

# **Problem 3**

## Introduction

두 개의 7-Segment Decoder를 사용하여 두 자리의 숫자를 표현할 수 있다. 이는 Problem 2에서도 설명했던거와 같이 엘리베이터, 시계 등 많은 Digital Display 기기에 들어가며 우리는 Problem2를 발전시켜 두 자리 숫자를 표현할 수 있는 Circuit을 추가로 설계 하고자 한다.

2개의 7-Segment Decoder와 문제의 Circuit을 이용하여 두 개의 숫자 패널을 작동시킨다. x[0],x[1],x[2],x[3] 가 0000 일 때는 0을 표시하고, 0001 일 때는 1, 0010 일때는 2를 표시한다. 따라서 출력 패널은 0부터 15까지 나타낼 수 있어야 한다.

이 때 우리가 설계하고자 하는 논리 회로는 4개의 입력값으로 8개의 출력값을 만들어야 한다.

도표이(가) 표시된 사진

자동 생성된 설명

## Design Process

### Truth Table

입력을 a[3], a[2], a[1], a[0], 출력은 z[7], z[6], z[5], z[4], z[3], z[2], z[1], z[0]로 z[0]~z[3]까지의 출력값을 첫번째 LED가 연결되어 있는 7-segment Decoder로 보내고, 나머지 z[4]~z[7]의 출력값은 두 번째 LED가 연결되어 있는 7-segment Decoder로 보낸다. 총 4개의 input을 다루므로 0부터 15까지의 숫자밖에 출력할 수 없다.

**테이블이(가) 표시된 사진

자동 생성된 설명**

Table 1. Truth table of Problem Circuit

### Karnaugh Map

|  |  |
| --- | --- |
|  |  |
| Z[0] | Z[1] |
|  |  |
| Z[2] | Z[3] |
|  |  |
| Z[4] | Z[5] |
|  |  |
| Z[6] | Z[7] |

Table 2. Karnaugh Map of 7-segment Decoder

위의 Truth Table을 통해 다음과 같은 카르노 맵을 만들 수 있다.

Z[5]~Z[7]의 경우 항상 입력값과 관계 없이 출력값이 0을 갖기 때문에 카르노 맵 또한 0으로 표시된다.

### Simulation Circuit Design

도표, 개략도이(가) 표시된 사진

자동 생성된 설명도표, 개략도이(가) 표시된 사진

자동 생성된 설명

Fig 1. Problem Circuit + 7-Segment Circuit Output (Logisim)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 3. 2-digit decimal number(0~15) Circuit Output (Logisim)

위의 Truth Table에서 확인했던 것처럼 z[5]~z[7]는 0이므로 input과 연결되어 있지 않은 것을 확인할 수 있다. Problem Circuit은 a[3]부터 a[0]까지의 입력과 8개의 출력이 있으며 4개씩의 출력을 Problem2의 7-segment Circuit에 각각 입력으로 주었다.

## Results and Discussion

### Results

Problem Circuit을 4개의 input과 8개의 output으로 구성하였다. Z[0]~Z[3]은 첫 번째 LED의 7-Segment Decoder의 input으로 들어가고 Z[4]~Z[7]는 두 번째 LED의 7-Segment Decoder의 input으로 들어간다.

### Discussion

위의 Truth Table을 통해 Problem Circuit을 다음과 같이 구성하였다. 2개의 숫자 패널에 불이 들어오기 위해 7+7개의 output을 최종적으로 구해야한다. 7-segement는 4개의 input과 7개의 output을 가지므로 따라서 Problem Circuit은 4개의 input과 8개의 output을 가져야한다. 따라서 입력값은 a[0]부터 a[3]까지, 출력값은 z[0]부터 z[3]까지는 첫 번째 7-segment Decoder에, z[4]부터 z[7]까지는 두 번째 7-segment Decoder의 Input으로 입력되어야 한다.

# Appendix