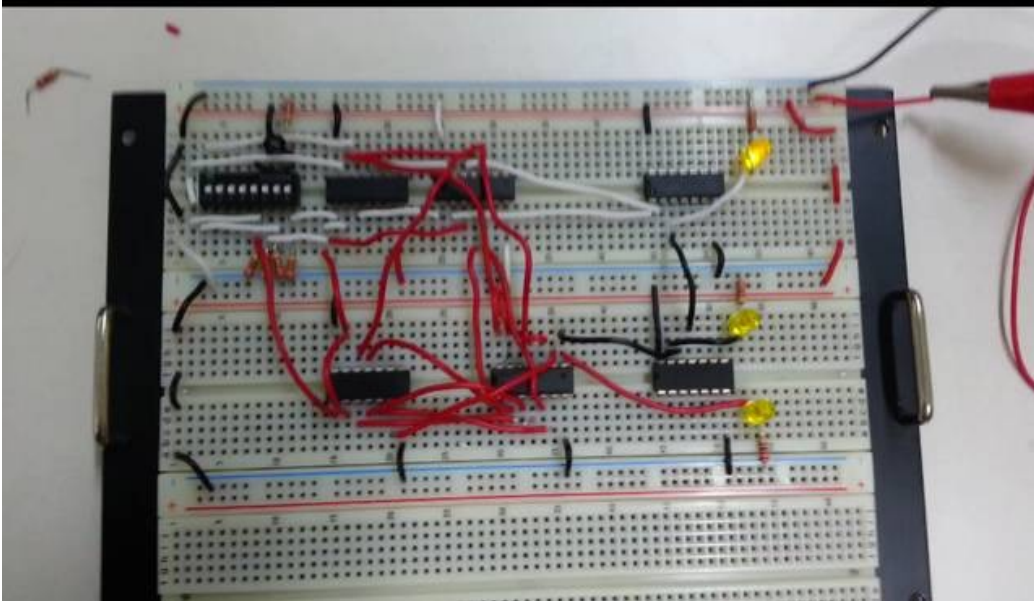


Procedure

logic diagram 의 3 input NAND 게이트를 2 input의 AND게이트와 2 input의 NAND게이트로 대체한 뒤에, 그에 따라서 circuit diagram을 구성한다. 그 후, 빵판에 각각의 게이트와 3개의 아웃풋 ($A=B, A>B, A<B$)을 의미하는 LED와 8개의 DIP스위치를 장착한 뒤, 자세한 설명은 생략한 방식으로 와이어링을 한다.

Result

인풋으로는 DIP 스위치의 5678번을 사용하였다. (5,6은 각각 숫자 B의 두 비트를 의미하며, 7,8은 숫자 A의 두 비트를 의미한다.) 위에서부터 첫 번째 LED는 $A = B$ 이며, 두 번째 LED는 $A > B$ 이며 세 번째 LED는 $A < B$ 를 뜻한다. GND, VCC와 관련된 선을 제외하고, 하얀색 선은 $A=B$ 와 관련 된 선이며, 붉은 색 선은 $A < B$ 이며 검은 색 선은 $A > B$ 와 관련된 선이다.

사진 상에서는 전원만 연결하였고, 어떠한 스위치도 조작하지 않았으며, 그 결과 A,B가 각각 0이 되어 $A = B$ 에 해당하는 LED가 점등되었다. 나머지 15개의 경우에 대해서는 파일과 함께 첨부된 동영상으로 대체하였다.

Role Allocation

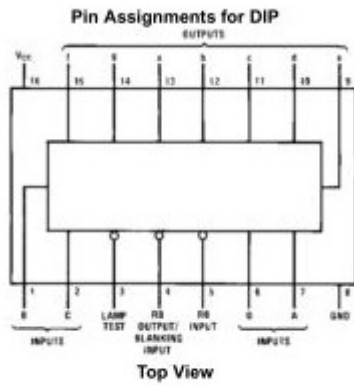
류명한씨께서 VCC 및 GND 설치, 게이트와 스위치설치, 그리고 $A > B$ 의 구현을 맡으셨다. $A = B$ 의 경우, 실습 당일에 두 명이서 같이 선 정리 없이 작동하는 지만 확인하였으며, 저는 추후에 입력 쪽 선정리 및 $A = B$ 의 재구현과, $A < B$ 의 구현을 담당하였습니다.

PREPARATION**BCD**

이진화 십진법, (Binary-coded decimal, BCD)는 이진수 네 자리를 묶어서(세 자리로는 7까지 표기할 수 있지만, 8, 9를 표기하기 위해 이진수 한 자리가 더 필요하다.) 십진수 한 자리로 사용하는 기수법이다.

7 Segment Display

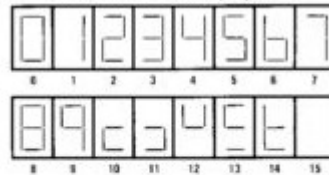
datasheet:



Segment Identification



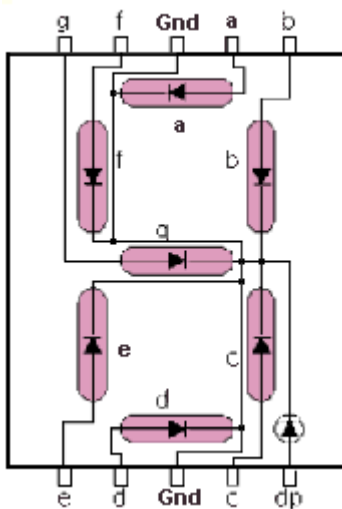
Numerical Designations and Resultant Displays



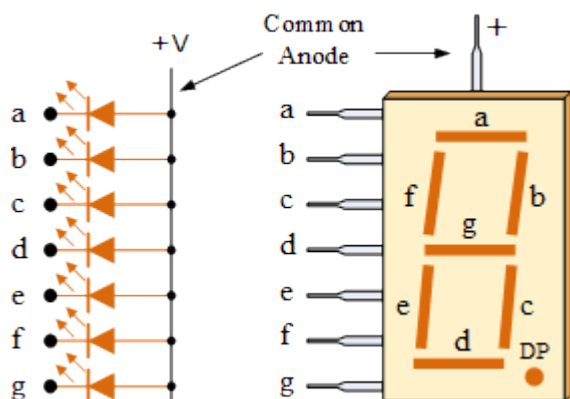
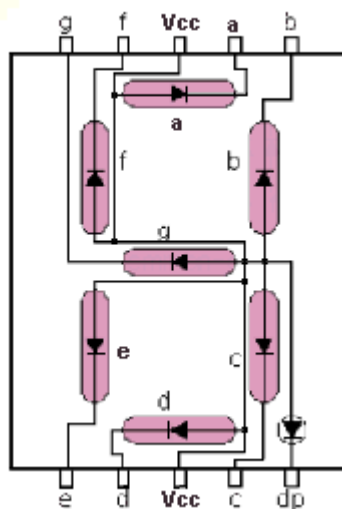
Anode & Cathode

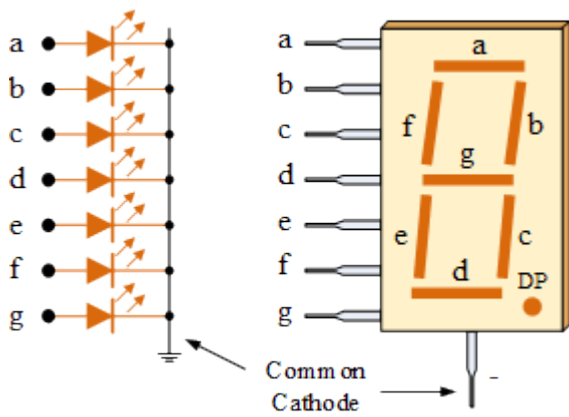
Common Cathode는 각 7 segment의 LED의 cathodes들이 같이 연결되어있으며, common anode의 경우 각 7 segment의 LED의 anode들이 같이 연결되어있다. Common Cathode(CC)의 경우, 각 LED들이 ground(logic 0)에 연결되어 있기 때문에 HIGH, (Logic 1)신호가 전달되면 불이 켜지게 되고, Common Anode의 경우 모든 전구가 logic 1에 연결되어 있기 때문에 각 전구에 LOW (logic 0)을 전달하면 불이 켜진다. 실습 때는 Common Cathode를 사용하지만, 일반적으로는 Common Anode가 선호된다고 한다. 많은 회로들이 흐르게 할 수 있는 전류의 양보다 GND로 내보낼 수 있는 전류의 양이 많기 때문이다.

Common Cathode



Common Anode





Decimal Digit	a	b	c	d	e	f	g
0	x	x	x	x	x	x	
1		x	x				
2	x	x		x	x		x
3	x	x	x	x			x
4		x	x			x	x
5	x		x	x		x	x
6	x		x	x	x	x	x
7	x	x	x				
8	x	x	x	x	x	x	x
9	x	x	x			x	x

Truth Table of Seven Segment

(6, 7, 9 의 경우에는 tail부분을 표기하냐 안하느냐에 따라서, 1의 경우 7 segment의 왼쪽이나 오른쪽이냐에 따라서 다른 방식으로 표기할 수 있지만 요구사항에서 제시된 대로 하였다.)

Decoder

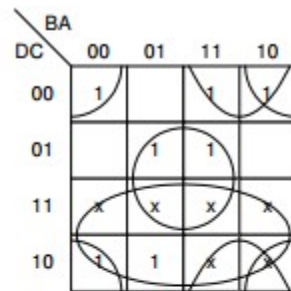
인코더의 반대 작업을 하는 장치로서, 인코딩된 정보를 해석하여 원래의 정보를 반환하는 장치를 말한다. 회로에서 n 개의 이진수로 된 정보를 받아 최대 2^n 개의 결과값을 반환하는 것이다. (BCD의 경우, 4개의 비트를 받아서 10개의 결과값을 출력한다.) 원래의 10진수가 2진수로 “인코딩”된 것을 디코딩하여 다시 10진수로 반환한다고 볼 수 있다.

BCD to 7-Segment decoder

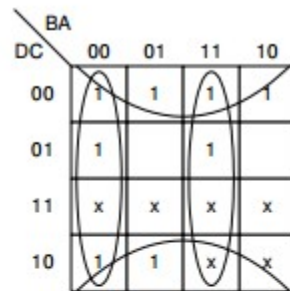
앞서 언급한 것처럼, 4개의 비트값을 받아서 7-segment가 a~g의 값을 출력할 수 있도록 디코더를 설계해야 할 것이다. 0000~ 1001 까지의 이진수를 적절히 해석하여, 7-segment의 a~g에 1의 값을 출력하면 (common cathode를 사용하기 때문에) 된다. 1001~1111까지의 값은 don't care로서, 적절히 활용하여 카르노맵을 통한 최적화에 사용하면 된다.

진리표는 다음과 같다.

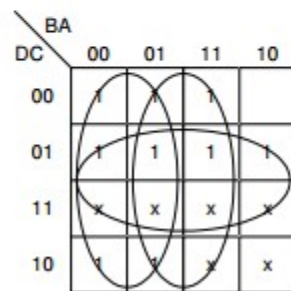
숫자	A	B	C	D	Seven Segments (cathode)						
					a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	0	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
D.C.	1	0	1	0	x	x	x	x	x	x	x
D.C.	1	0	1	1	x	x	x	x	x	x	x
D.C.	1	1	0	0	x	x	x	x	x	x	x
D.C.	1	1	0	1	x	x	x	x	x	x	x
D.C.	1	1	1	0	x	x	x	x	x	x	x
D.C.	1	1	1	1	x	x	x	x	x	x	x



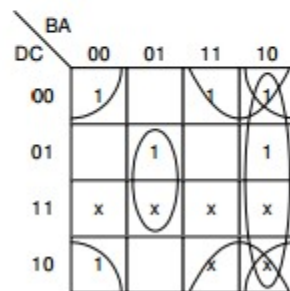
$$a = C'A' + CA + D + C'B$$



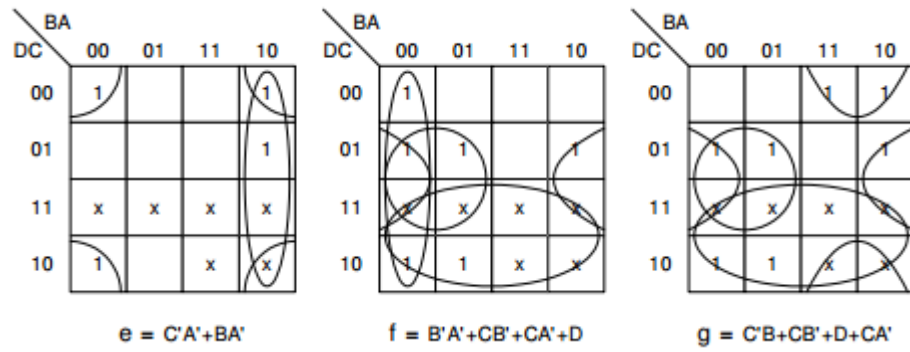
$$b = B'A' + BA + C'$$



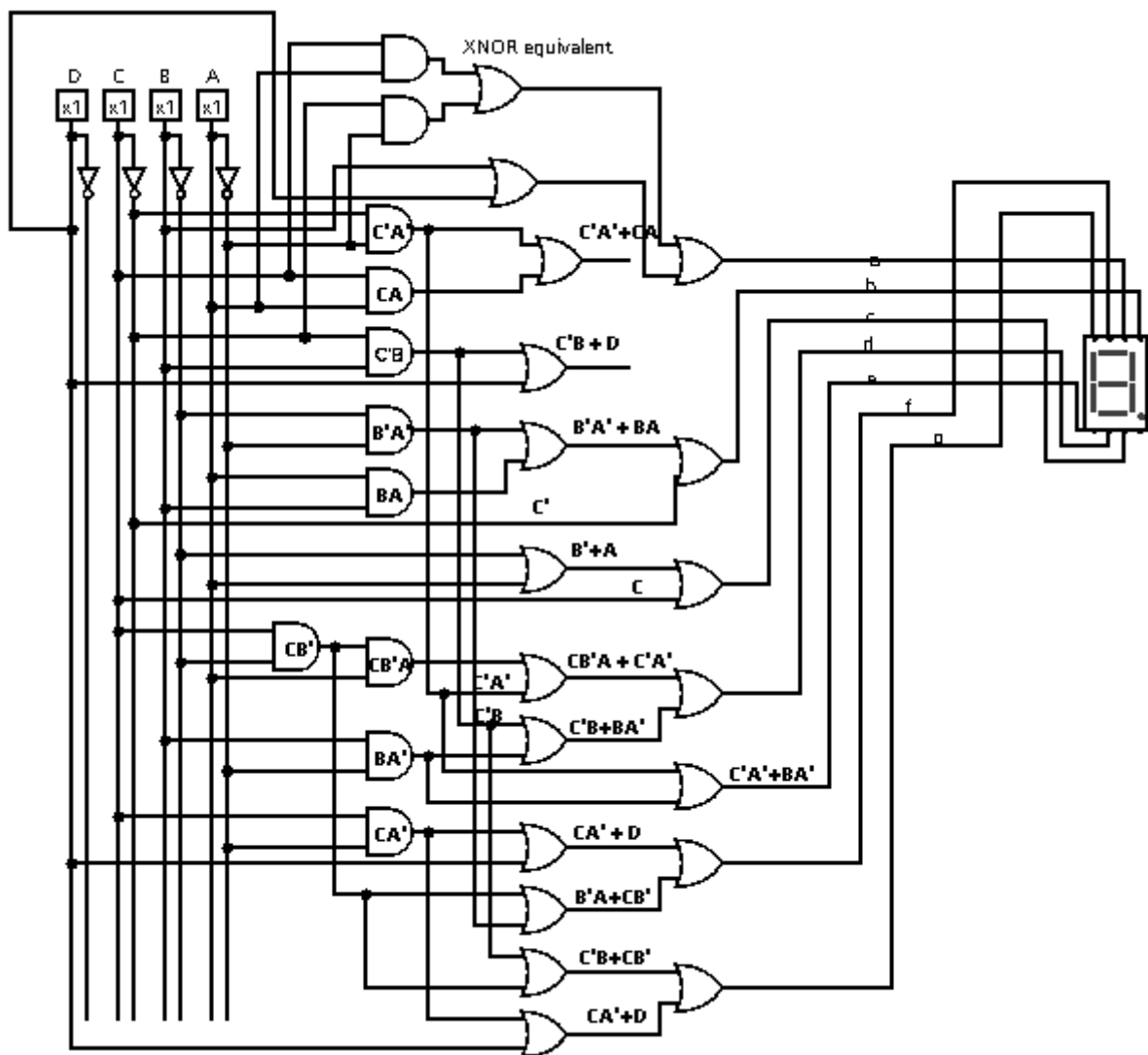
$$c = B' + A + C$$



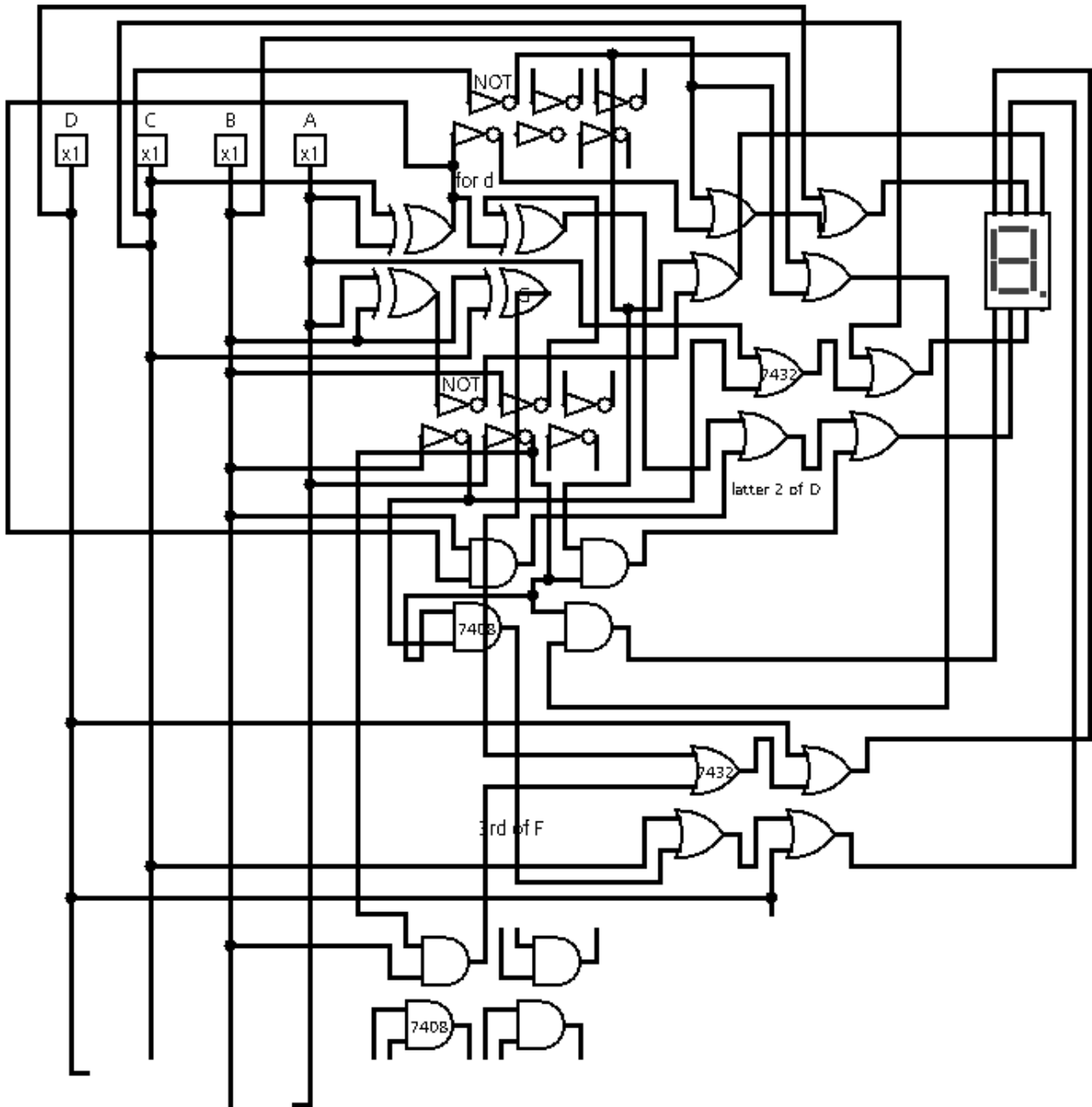
$$d = CB'A + C'A' + C'B + BA'$$



LOGIC DESIGN



CIRCUIT DESIGN



연속적인 게이트들은 하나의 TTL chip을 의미한다. (7486, 7408, 7432, 7404).

참조: <http://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>