논리설계실험 2주차 보고서

2012-11114

경영학과

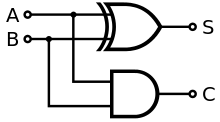
엄태휘

**Procedure**

파워, LED, 저항을 연결하여 만능기판에 전류가 흐르는 것을 확인한 뒤에, 스위치를 기판에 Pull-Down Resistors를 활용하여 연결한다. (스위치가 꺼져있을 때 floating이 생기지 않게 하기 위함이다.) 그 후, 소켓에 각각 VCC와 GND를 연결하고, 출력을 나타내는 LED에도 저항을 연결하여 GND와 연결한다. 그 후, Circuit wiring diagram에서 구현한대로 와이어링을 한다.

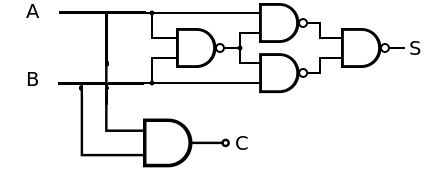
**Logic Diagram**

Half Adder using XOR & AND gates:



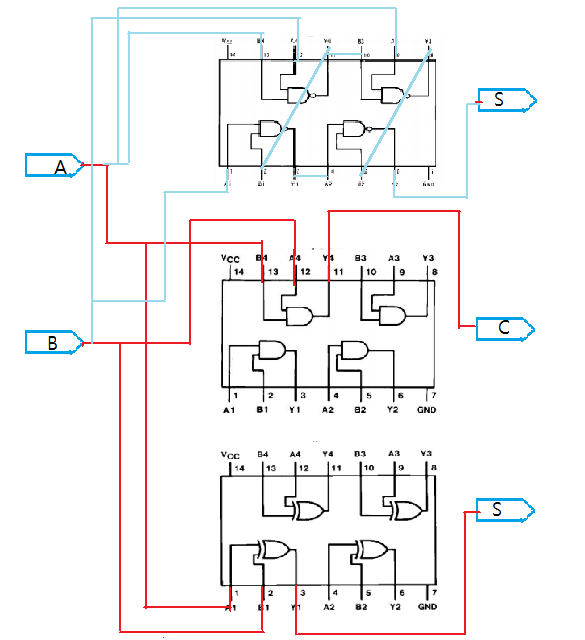
A 나 B 중 한 개의 인풋만 1이 되면 XOR게이트를 통해 S가 1이 되고, A와 B가 동시에 켜지면 S는 0, C는 1이 된다.

Half Adder using NAND & AND gates:



XOR 게이트를 NAND 게이트로 대체하여 구성하면 된다.

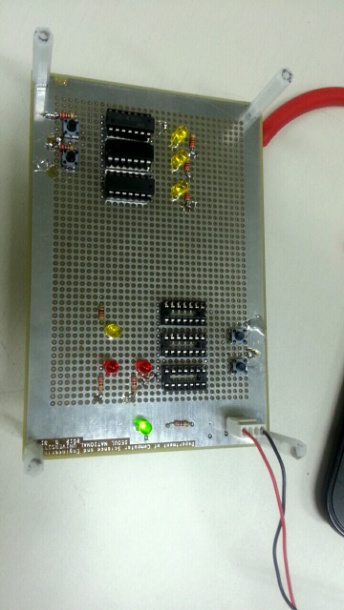
**Circuit wiring diagram**

****

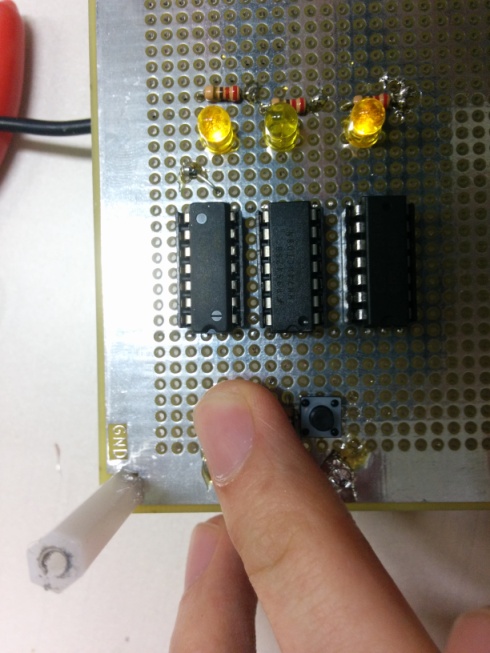
Half Adder using XOR & AND gates && using NAND & AND gates.

위에서부터 각각 NAND , AND , XOR 게이트이며, 중간의 AND게이트를 공유하도록 설계하였다.

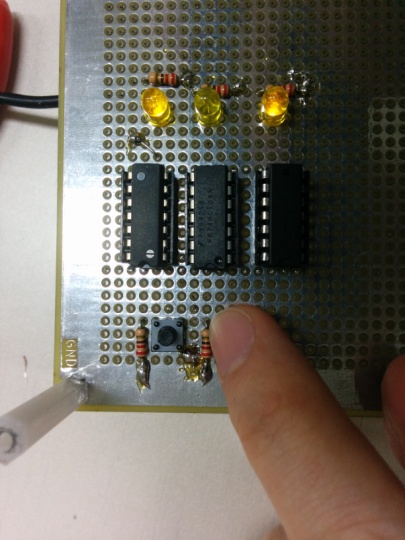
**Result**

****

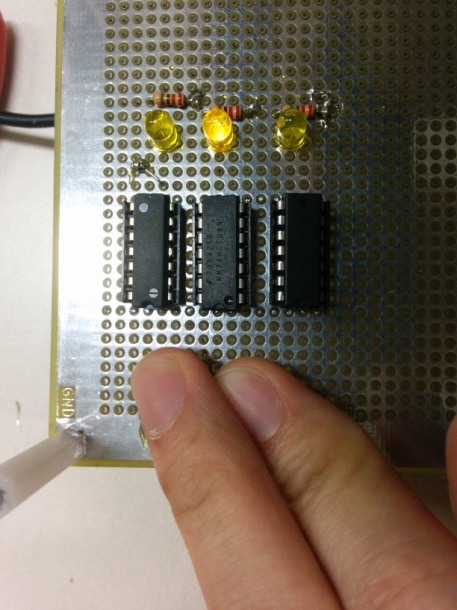
전원은 들어왔지만 아무런 인풋이 없는 상황에서는 아무 LED도 켜지지 않았다. (아래 부분은 납땜과정에서 큰 결함이 있어서 위에 새로 다시 만들었습니다.)

****

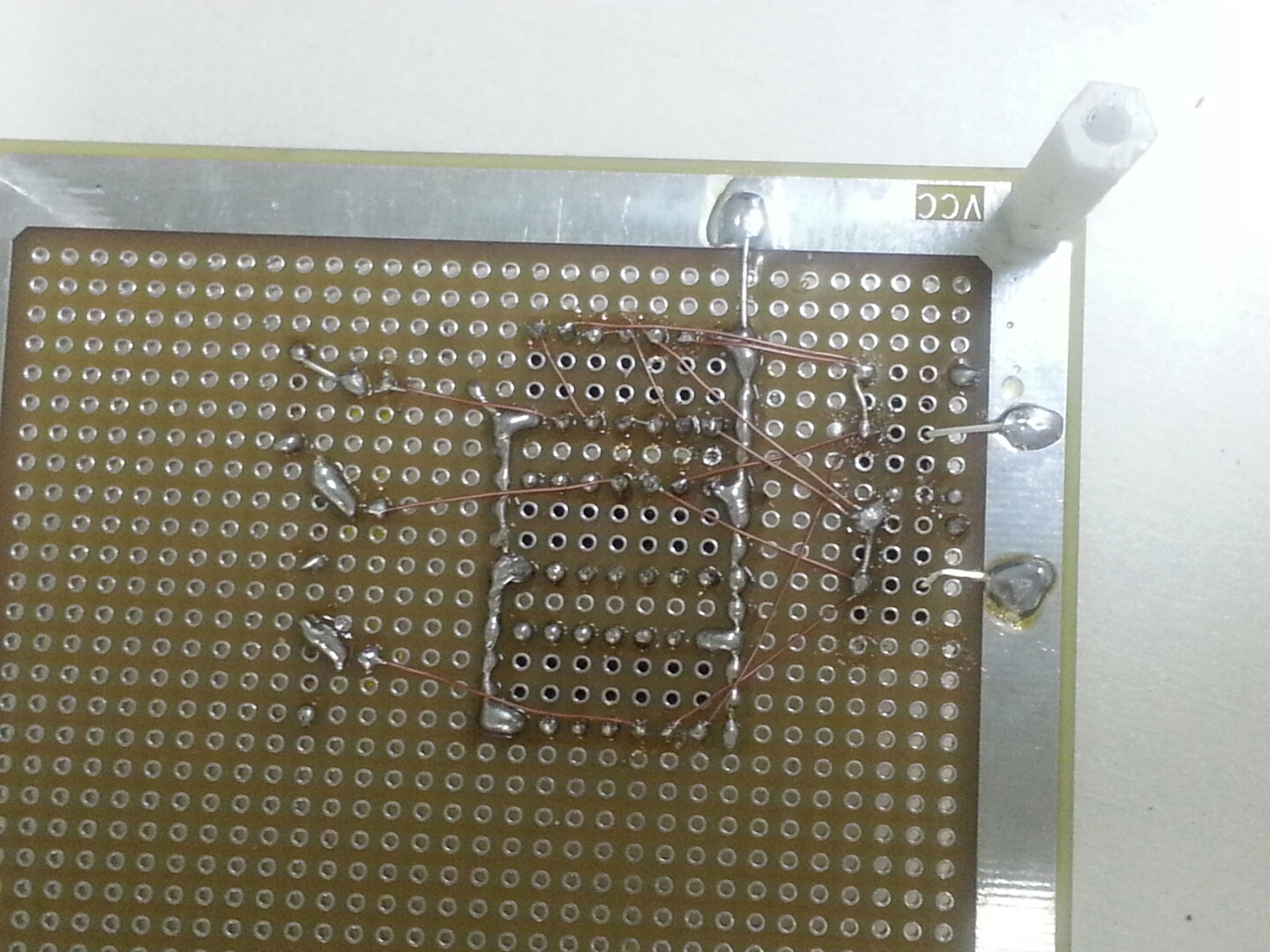
A의 스위치만 누른 상황(A의 인풋만 들어오는)에서는 각각의 S에 빛이 들어온다.

****

B의 스위치만 누른 상황(B의 인풋만 들어오는)에서는 각각의 S에 빛이 들어온다.

****

A와 B의 스위치를 누른 상황에서는 C에 빛이 들어온다.



뒷면의 와이어링 사진이다.

역할분담:

류명한: 회로 설계, 파워버스 연결, 회로 납땜, 회로에서 그라운드 연결, 스위치와 저항 각각 하나 씩, 전류 흐르는 확인용 LED연결, 소켓 연결.

엄태휘: 회로 설계, 스위치와 저항 하나 연결, 와이어링, 출력 LED와 저항 연결, 소켓 연결

**Preparation**

**2 – bit Comparator**

A = A1A0

B = B2B0

라고 할 때 (Ai, Bi는 각각 2진수로 나타낸 A B의 i 번째 자리이다.) A와 B의 값에 따라서 A > B 인지, A == B 인지, A < B 인지 알려주는 회로.

진리표:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A1 | A0 | B1 | B0 | A == B | A > B | A < B |
| **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **0** | **0** | **0** | **1** | **0** | **0** | **1** |
| **0** | **0** | **1** | **0** | **0** | **0** | **1** |
| **0** | **0** | **1** | **1** | **0** | **0** | **1** |
| **0** | **1** | **0** | **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **1** | **1** | **0** | **0** |
| **0** | **1** | **1** | **0** | **0** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** | **0** | **1** |
| **1** | **0** | **0** | **0** | **0** | **1** | **0** |
| **1** | **0** | **0** | **1** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** | **0** |
| **1** | **0** | **1** | **1** | **0** | **0** | **1** |
| **1** | **1** | **0** | **0** | **0** | **1** | **0** |
| **1** | **1** | **0** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** | **1** | **0** | **0** |

A == B : ( A0B0 + A0’B0’ )( A1B1 + A1’B1’)

각 비트의 값이 동일하면 A는 B와 같다.

각 비트에 해당하는 XNOR 게이트 두 개의 결과값을 AND 게이트를 통과시키면 된다.

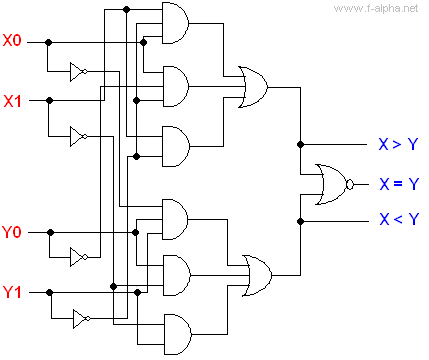
A > B : A1B1’ + (A1B1 + A1’B1’)A0B0’

A < B : A1’B1 + (A1B1 + A1’B1’)A0’B1’

A > B 의 경우, most significant bit 가 A의 경우 1이고, B의 경우 0이면 A > B이다.

만약 두 비트가 같다면, 그 다음 비트로 내려가서 A가 1이고, B가 0인지 확인하면 된다.

A < B는 B의 msb가 1이고, A의 msb가 0인지 확인하면 된다.



Circuit Diagram