**디지털회로실험 설명서**

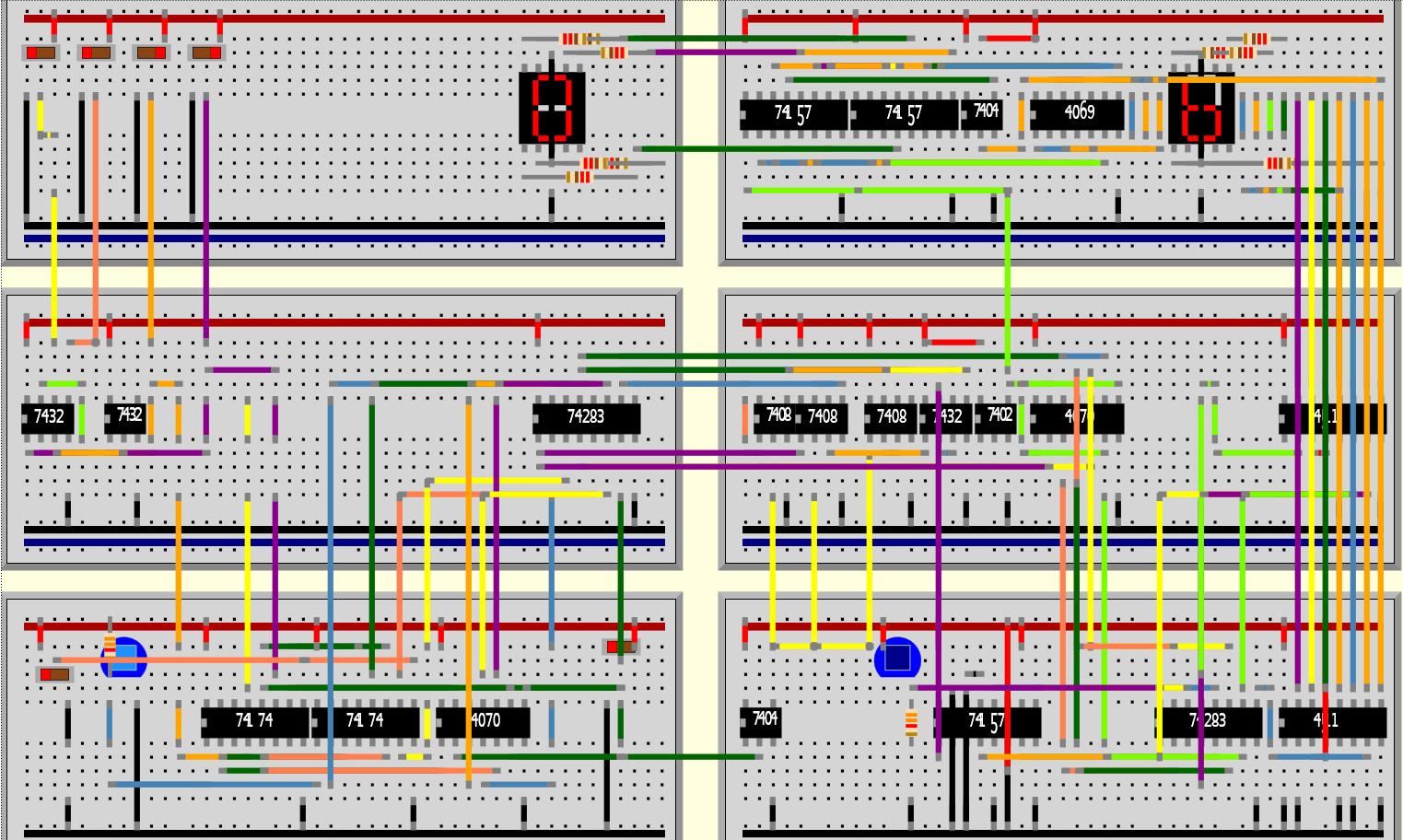
-중간 프로젝트-

전자공학과

2015104027

박정진

Bread Board Simulation



equal

Addn\_Sub

clk start

A[3] A[2] A[1] A[0]

seg\_10

seg\_1

사용 방법 :

0. Add 는 스위치를 왼쪽에 Subtract는 스위치를 오른 쪽에 둡니다.

0. equal 버튼은 누른 상태로 지속해도 되고 두 개의 입력 을 마치고 누른 상태에 있어도 됩니다.

0. 실행 후에도 segment에 불이 들어오지 않는다면 start 버튼을 여러 번 누르시다 보면 segment에 불이 들어옵니다. 혹 결과가 들어와있는 상태라면 start를 누르시면 리셋이 되어 처음부터 시작할 수 있습니다.

1. A를 컨트롤하여 비트를 조절합니다. (2진수, 사진 상 스위치 왼쪽 - 1'b0, 스위치 오른쪽 - 1'b1) (계산 상 A)

2. clk 스위치를 두 번 클릭하여 펄스를 만든다. (첫 시작 스위치 위치가 사진 상 처럼 왼 쪽에 있다면)

3. 다시 A를 컨트롤하여 두 번 째 입력에 들어갈 비트를 조절합니다. (계산상 B)

4. clk 스위치를 두 번 클릭하여 펄스를 만든다. (첫 시작 스위치 위치가 사진 상 처럼 왼 쪽에 있다면)

5. 0번의 설명처럼 equal 버튼을 누른 상태로 두었으면 4번을 실행 했을 때 바로 결과가 나오고 만약 누르지 않은 상태라면 누른 상태로 바뀌면 결과가 나옵니다. (이 때 결과가 나옴)

6. start 를 눌러 리셋을 함. (segment 상태가 00이됨)

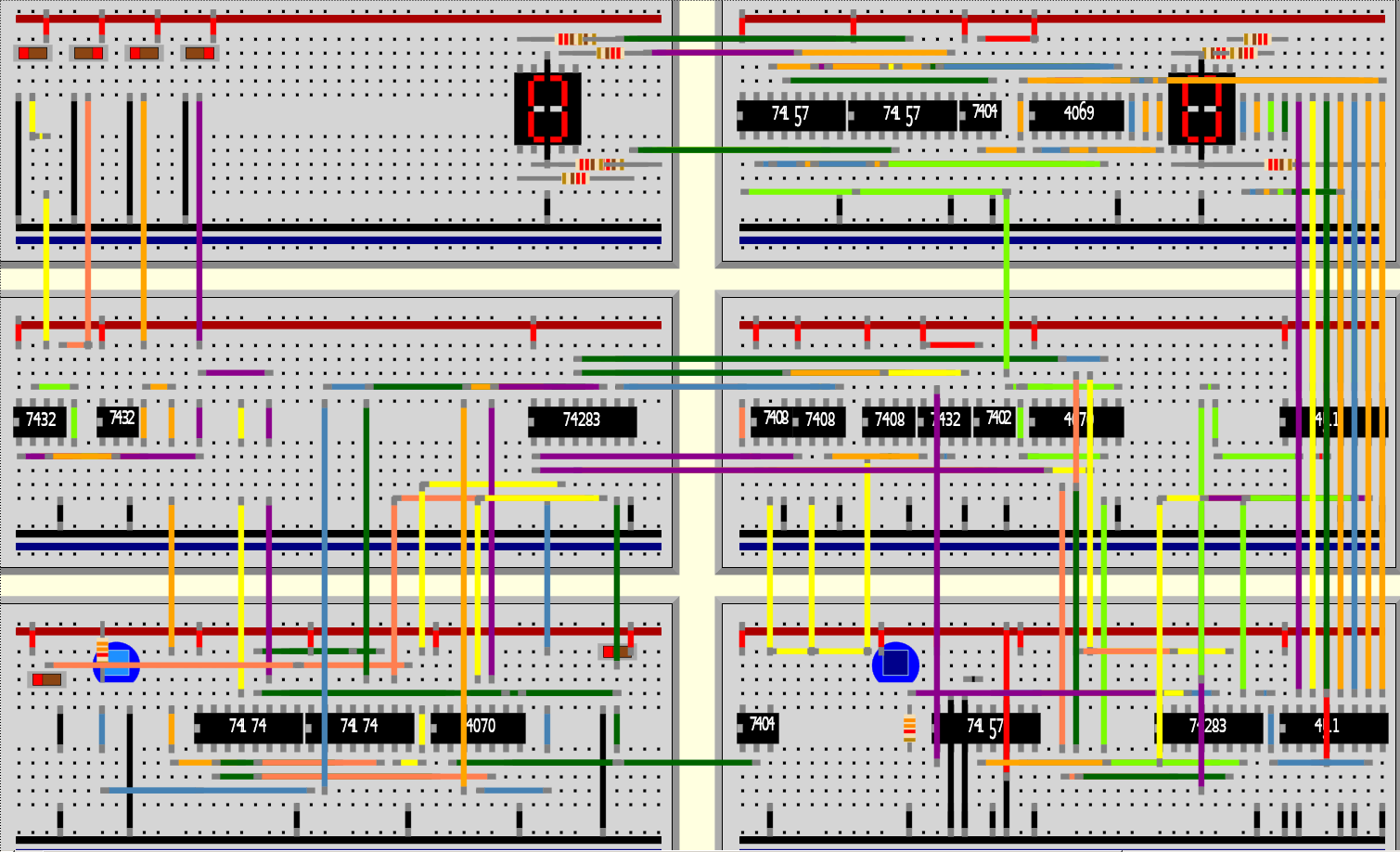
7. 1 ~ 6 반복

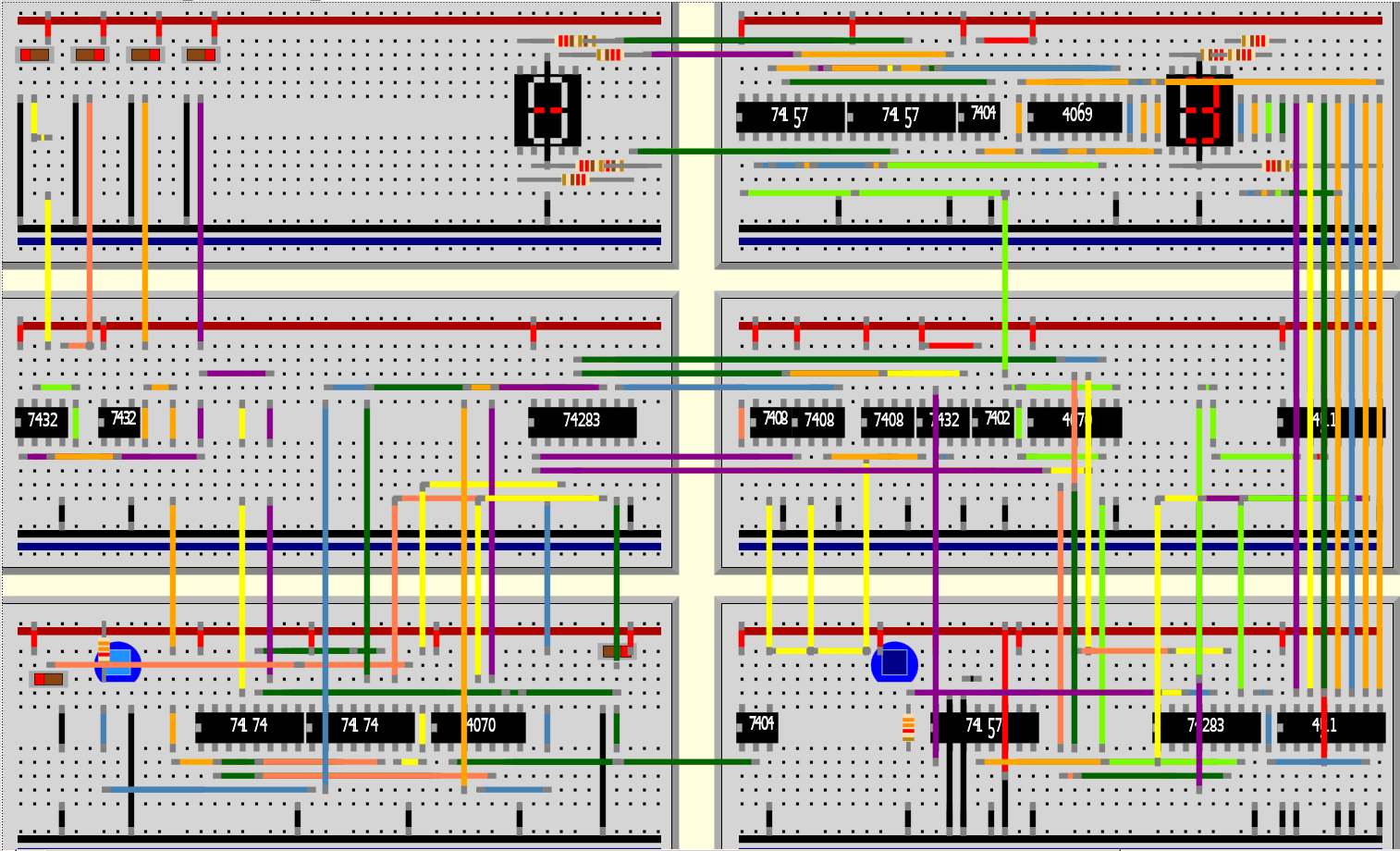
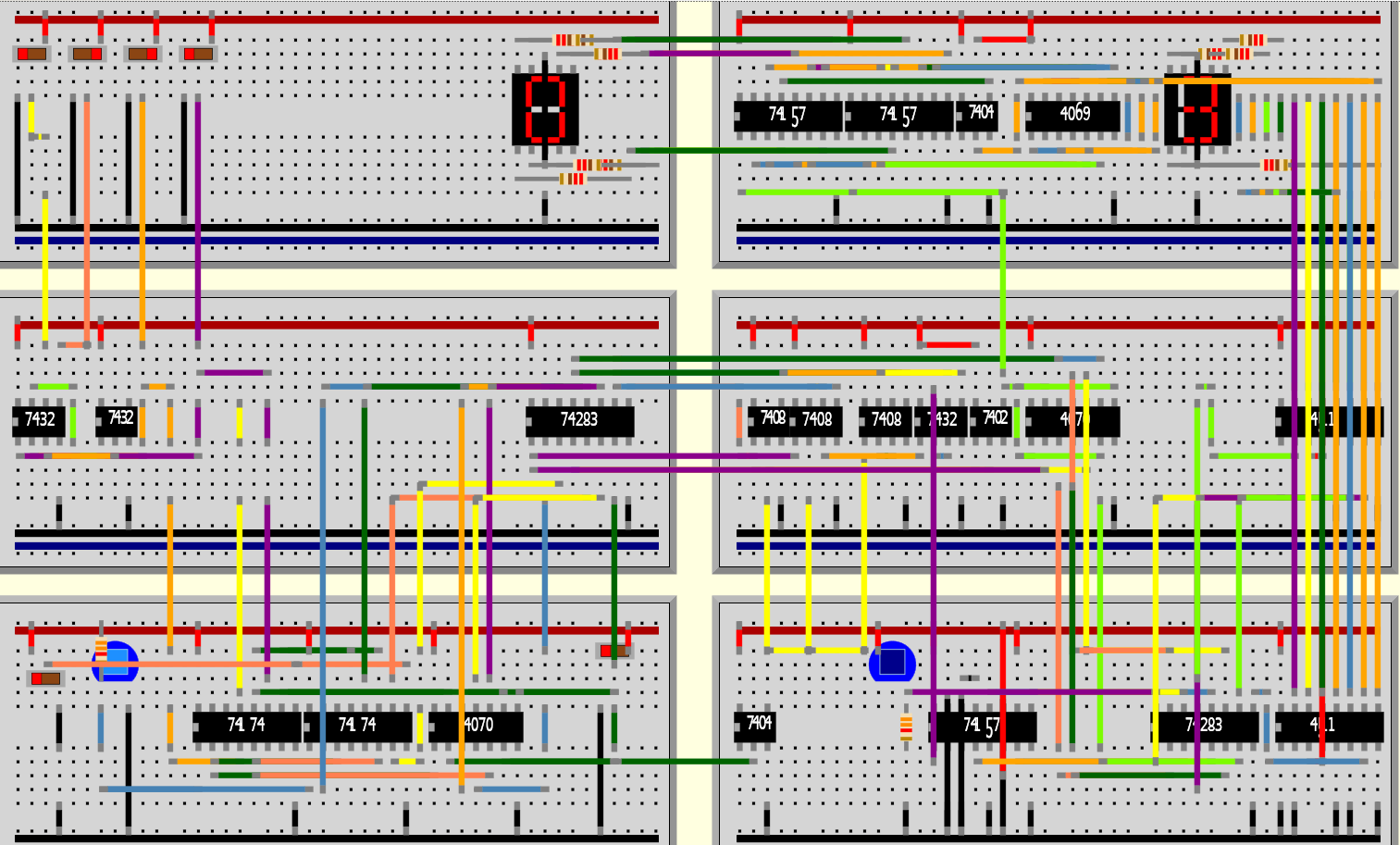
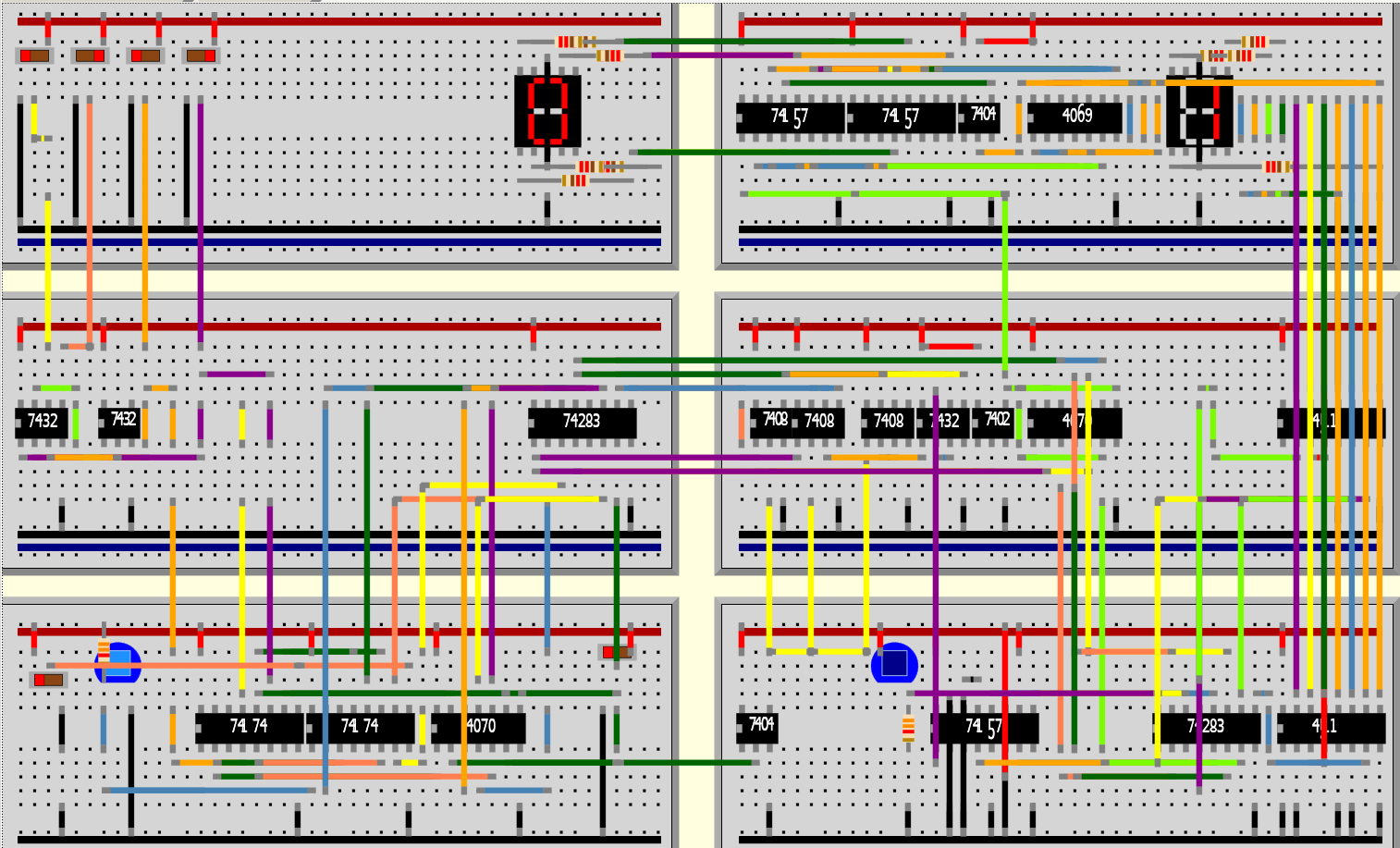
**파일 결과 A[4], A[3] 쪽 선이 74174에 연결되어있지 않아 작동하지 않습니다.**

**즉, 들어갈 입력의 제한이 0~3 까지 밖에 안되어 결과가 -3 ~ 6 까지의 결과만 나옵니다.**

하지만 선이 연결이 안 되어있다 뿐이지, 기존 중간 프로젝트의 목표였던 더하기와 빼기, 그리고 그 결과를 segment에 디스플레이 하는 것은 모두 작동이 됩니다.

저는 리눅스 우분투 16.04 에서 프로그램을 돌려 타 os에서 와인 프로그램이 충돌이 나 안될 것을 염려하여 직접 해봐서 나온 결과를 첨부해서 넣습니다.





동작 원리 :

버튼(input A[3:0])을High, Low로 컨트롤 하여 동작하게 하였습니다. 초기에 아무것도 건들지 않았을 때 A는 4'b0 상태이다. 이 상태에서 인풋을 조작하며 ttl74174의 DFF가 인코딩 된 BCD 4bit를 저장한다. ttl74174는 Series로 연결되어 있어 인 풋을 두 번 조작하고 클락을 컨트롤 하면 차례로 저장이 된다. (이하 저장된 두 개의 값을 A, B라 칭함) 들어온 A와 B는 xor와 4bit adder ttl74283로 만들어진 adder and subtractor는 input Addn\_Sub 의 값이 0일땐 add, 1 일땐 subtract를 수행한다. 계산 결과 상 add 일 땐 경우가 0~18, subtract일 땐 -9 ~ 9 의 경우가 나올 수 있다. 계산 결과를7segment decoder에 표시하기 위해서는 ttl7448에 들어갈 두 자리의 BCD코드에 의해 총 8bit가 필요한데 10의 자리는 add일 땐 10의 자리를, subtract일 땐 - 를 표시해야 한다. 이를 위해 K map 을 이용해 {Addn\_Sub, Cout, S} 총 6비트로 위의 경우를 알려 줄 수 있는 flag bit를 만들었다. 더해서 10의 자리를 넘어가면 멀티플렉서를 이용해 4'b0110 을 두 번째 4bit adder에서 첫 번째 4bit adder의 결과와 더해 일의 자리 BCD 4bit를 나타냈고, -를 표시해야 하는 상황에선 4'b0000 - (첫 번째 4bit adder의 결과) = 첫 번째 4bit adder의 결과 를 이용해 일의 자리 BCD 를 표현했다. 이런 계산을 통해 일의 자리 segment는 따로 컨트롤 하지 않아도 계산 결과를 표현하게 된다. 하지만 10의 자리 segment는 위에 언급한 두 가지의 경우를 구분해서 표현해야한다. add일 때 십의 자리가 나오는 경우는 특별히 따로 처리하지 않고 그대로 보여주면 맞는 경우가 되지만 subtract 에서 - 가 나오는 경우는 컨트롤을 필요로한다. 만약 그대로 내보내면 -9 ~ -1 의 경우 밖에 없으므로 segment는 0을 표시할 것이다. 하지만 위에서 언급 한 flag bit를 멀티플렉서의 select bit로 사용하면 - 경우 일때 -를 표시할 수 있게 된다. 0의 not 을 취하면 - 가 되므로, 따라서 - 경우인 flag bit를 받으면 멀티플렉서에서 not을 연결한 decoder값을 출력하게 하였다. 마지막으로 input EQUAL 신호를 넣어 EQUAL (DFF의 CLK)을 눌러야 계산 결과가 segment 에 나타나게 하였다. input START신호는 activate low로 설정하여 START를 누르면 리셋이 되고 계산을 새로 시작할 수 있게 하였다.

빵판이 제대로 돌아가는 것을 확인하여 쿼터스 파일은 제출을 하지 않았습니다.

Quartus Simulation

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generated

A screen shot of a computer

Description automatically generatedA close up of a map

Description automatically generated