

# **LAB #05 REPORT**

**Name: 윤효정**

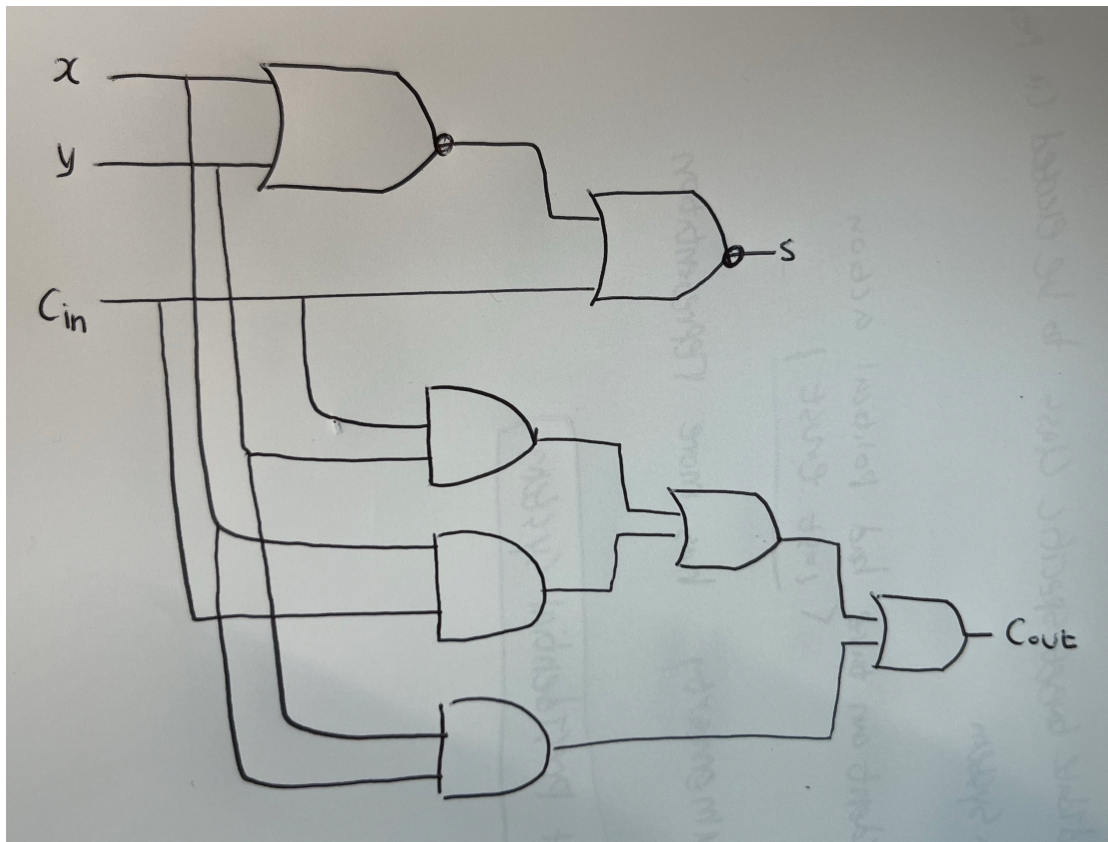
**Student ID: 2076281**

### 1. Lab. Objective (10%)

Half-adder에서 출발하여 ripple-carry-adder의 원리를 이해한다. full-adder 4개를 이용하여 4-bit ripple-carry-adder 를 FPGA 보드 상에 구현해본다. 0~15 사이의 두 숫자를 합하여 30 이하의 숫자를 보드 상에 나타내는 계산기를 구현한다. 하나의 module을 다른 module에 참조하여 복잡한 회로를 만드는 준비 과정을 배운다.

### 2. Implementation & Result (40%)

Full-adder circuit



### 3. Discussion & Conclusion (10%)

한 모듈을 다른 모듈에 참조하여 복잡한 회로를 만드는 과정을 처음 해 보았는데, 와이어를 통해 내부 모듈의 결과값을 다음 모듈로 넘기는 부분이 헷갈렸다. 벡터 값에서 인덱스를 매기는 순서에 주의를 해야했다. 보드에 맵핑을 할 때도 Max, Min 이 바뀌지 않도록 해야 결과를 잘 확인할 수 있었다. 이론 수업에서 full adder와 ripple carry adder의 Cin과 Cout이 실제로 어떻게 구현이 되는 건지 궁금했는데 실제로 실험을 통해 구체적으로 결과를 확인할 수 있었다.

### 4. Reference(s)

없음

## 5. Code (40%)

### lab05.v

```
Module lab05( x, y, Cin, Cout, s);  
    input x;  
    input y;  
    input Cin;  
    output Cout;  
    output s;  
  
    xor(s, x, y, Cin);  
  
    and(out1,y, Cin);  
    and(out2, x, Cin);  
    and(out3, x, y);  
    or(Cout, out1, out2, out3);  
  
endmodule;
```

### ripple\_adder.v

```
Module ripple_adder.v( x, y, s);  
    input[3:0] x;  
    input[3:0] y;  
    output[4:0] s;  
  
    wire[2:0] Cout;  
  
    lab05 m1(x[0], y[0], 0, Cout[0], s[0]);  
    lab05 m2(x[1], y[1], Cout[0], Cout[1] , s[1]);  
    lab05 m3(x[2], y[2], Cout[1], Cout[2], s[2]);  
    lab05 m4(x[3], y[3], Cout[2], s[4], s[3]);  
  
endmodule;
```