Lab #3 Ripple Carry Adder Design

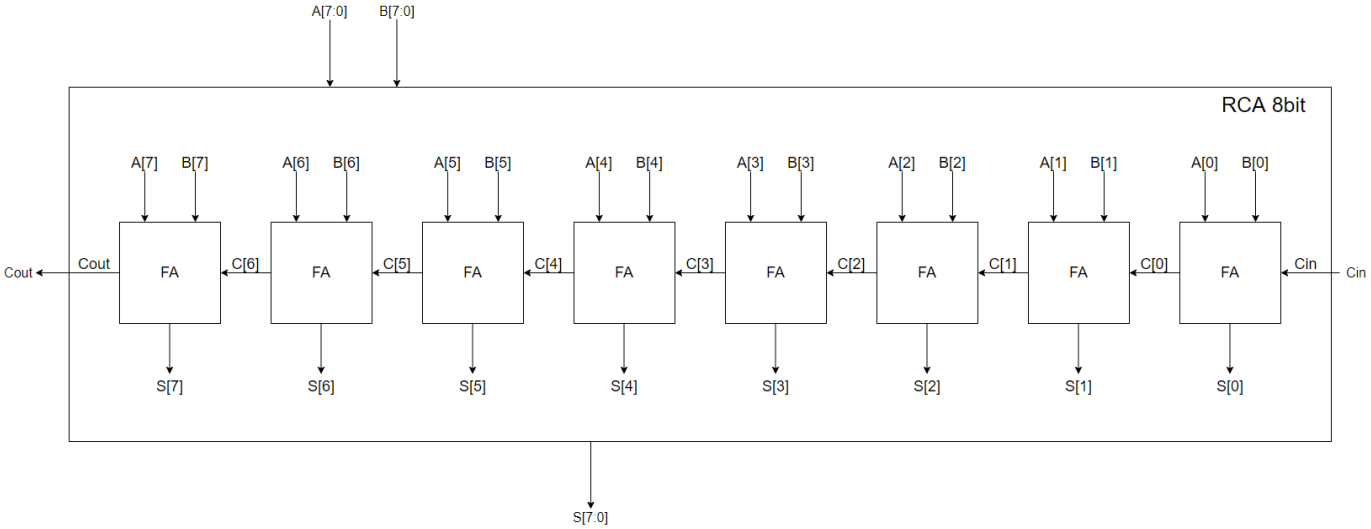
Class : 00

201602004 박태현

I. 실습 목적

8bit Ripple Carry Adder 의 설계

II. Design procedure



III. Simulation

- Full Adder

```
module FA(a, b, c, cout, sum);
    input a,b,c;
    output sum, cout;

    assign {cout, sum} = a + b + c;
    endmodule
```

전가산기를 Dataflow Style로 작성하였습니다.

- RCA 8 bit

```
module RCA8bit(x,y,cin,cout,s);
    input [7:0] x,y;
    input cin;

    output [7:0] s;
    output cout;

    wire [6:0] c;

    FA FA0 (.a(x[0]), .b(y[0]), .c(cin), .cout(c[0]), .sum(s[0]));
    FA FA1 (.a(x[1]), .b(y[1]), .c(c[0]), .cout(c[1]), .sum(s[1]));
    FA FA2 (.a(x[2]), .b(y[2]), .c(c[1]), .cout(c[2]), .sum(s[2]));
    FA FA3 (.a(x[3]), .b(y[3]), .c(c[2]), .cout(c[3]), .sum(s[3]));
    FA FA4 (.a(x[4]), .b(y[4]), .c(c[3]), .cout(c[4]), .sum(s[4]));
    FA FA5 (.a(x[5]), .b(y[5]), .c(c[4]), .cout(c[5]), .sum(s[5]));
    FA FA6 (.a(x[6]), .b(y[6]), .c(c[5]), .cout(c[6]), .sum(s[6]));
    FA FA7 (.a(x[7]), .b(y[7]), .c(c[6]), .cout(cout), .sum(s[7]));

endmodule
```

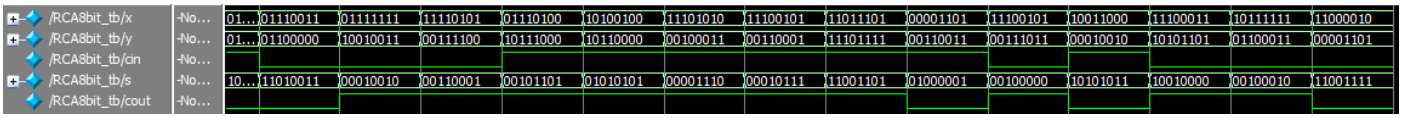
위에 그린 블록 다이어그램에 맞게 Structural Style로 전가산기를 연결시켜 주었습니다

- Test bench

```
module RCA8bit_tb;
   reg [7:0] x,y;
   reg cin;

   wire [7:0] s;
   wire cout;

   integer i;
   parameter iter = 1000;

   RCA8bit RCA8bit_0(.x(x), .y(y), .cin(cin), .cout(cout), .s(s));

   initial begin
      for(i = 0; i < iter; i = i + 1) begin
         x=$random;
         y=$random;
         cin=$random;
         #100;
      end
   end
endmodule
```

8bit 가산기에 입력 값을 랜덤으로 주어서 결과를 확인하도록 했습니다

- Waveform



임의 값에 대해 적절한 결과를 내놓는지 확인을 해보았습니다.

IV. Evaluation

엑셀을 통해서 계산이 맞는지 확인해보았습니다

A	01110011	01111111	11110101	01110100	10100100	11101010	11100101	11011101	00001101	11100101	10011000	11100011	10111111	11000010
B	01100000	10010011	00111100	10111000	10110000	00100011	00110001	11101111	00110011	00111011	00010010	10101101	01100011	00001101
CIN	0	0	0	1	1	1	1	1	1	0	1	0	0	0
SUM	11010011	00010010	00110001	00101101	01010101	00001110	00010111	11001101	01000001	00100000	10101011	10010000	00100010	11001111
COUT	0	1	1	1	1	1	1	1	0	1	0	1	1	0

해당 캡쳐 부분에 모든 결과가 일치하는 것을 확인했습니다

V. Discussions

FA를 통해서 여러 비트의 가산기를 만들 수 있다는 것을 알았습니다.

Waveform의 경우의 수가 256*256*2라서 모두 확인하기 어려운 점이 있었습니다

Lookahead 모듈을 장착한다면 계산이 O(1)에 끝날 수 있을 것 같습니다.