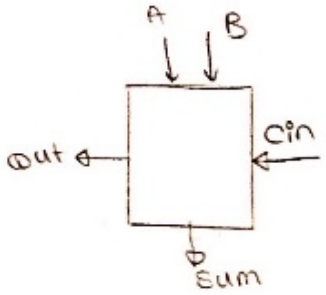
جلسه ۱۷ درس آزمایشگاه

طراحی full adder ←



جدول درستی xor

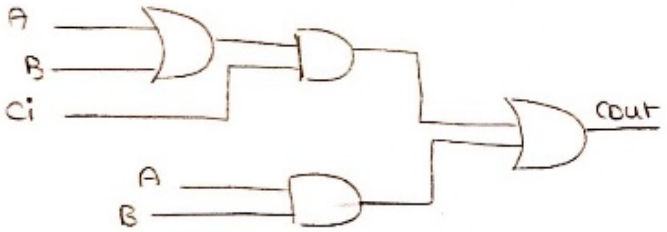
a	b	Cin	Sum	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

رسم مدار با توجه به جدول درستی

Sum = Ci xor A xor B

Ci \ AB	00	01	11	10
0	0	(1)	0	(1)
1	(1)	0	(1)	0

Ci \ AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1



Cout = (A+B) Ci + AB

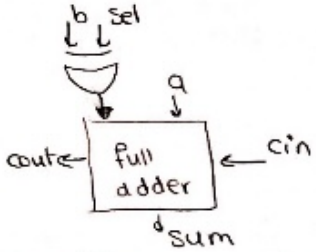
Cout = ACi + BCi + AB

$\{ X \oplus 0 = X \quad X \oplus 1 = X' \}$

$A + (-B) = A + B' + 1$

با استفاده از xor

با استفاده از Cin



دستکاری جمع‌کننده تفریق‌کننده

full adder :

```
module full_adder(
    input a,
    input b,
    input cin,
    output sum,
    output cout
);
wire x1, x2, x3;
xor (sum, a, b, cin);
and (x1, a, b);
and (x2, a, cin);
and (x3, b, cin);
or (cout, x1, x2, x3);
end module
```

```
module add_sub (
    input a,
    input b,
    input cin,
    input sel,
    output sum,
    output cout
);
  xor (y, sel, b)
  full_adder (a, y, cin, sum, cout)
  endmodule
```

اول وورودی

```
module adder_subtractor_4bit (
    input [3:0] A,
    input [3:0] B,
    input sel,
    output [3:0] S,
    output co
);
    wire [2:0] cout;
    add-sub a0 (A[0], B[0], sel, sel, S[0], cout[0]);
    add-sub a1 (A[1], B[1], cout[0], sel, S[1], cout[1]);
    add-sub a2 (A[2], B[2], cout[1], sel, S[2], cout[2]);
    add-sub a3 (A[3], B[3], cout[2], sel, S[3], co);

endmodule
```

دقت: sel و cin برابر بودند

with delay:

{ AND ۵ نانو / or ۵ نانو } { xor ۱۰ نانو / Not ۲ نانو }

Full adder delay:

```
module full_adder_delay (
    input a,
    input b,
    input cin,
    output sum,
    output cout
);
    wire x1, x2, x3;
    xor #10 (sum, a, b, cin);
    and #5
        and1 (x1, a, b),
        and2 (x۲, a, cin),
        and3 (x۳, cin, b);
    or #5 or1 (cout, x1, x۲, x۳);
endmodule
```

add_sub_delay

```
module add_sub_delay (
    input a,
    input b,
    input cin,
    input sel,
    output sum,
    output cout
);
    wire y;
    xor #10 x1 (y, b, sel);
    full_adder_delay f1 (a, y, cin, sum, cout)
```

```
module adder-subtractor_4bit_delay(
        input [3:0]A,
        input [3:0]B.
        input sel,
        output [3:0]S,
);      output co

wire[2:0]cout;
add_sub_delay  a0 (A[0], B[0], sel, sel, S[0], cout[0]);
add_sub_delay  a1 (A[1], B[1], cout[0], sel, S[1], cout[1]);
add_sub_delay  a2 (A[2], B[2], cout[1], sel, S[2], cout[2]);
add_sub_delay  a3 (A[3], B[3], cout[2], sel, S[3], co);

endmodule
```