Se 8 Lc Dr.Shaabany

← مثال FA را برای بیت i ام بررسی می‌کنیم.

$$c_{n-1} \cdots c_i \cdots c_1 \, c_0$$
$$x_{n-1} \cdots x_i \cdots x_1 \, x_0$$
$$y_{n-1} \cdots y_i \cdots y_1 \, y_0$$
$$c_n \quad s_{n-1} \cdots s_i \cdots s_1 \, s_0$$

$$c_0 = 0 \qquad c_i \in \{0,1\}$$

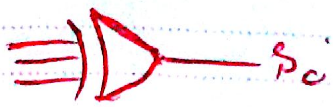
اگر $c_i=0$ ، s_i می‌شود $x_i \oplus y_i$.

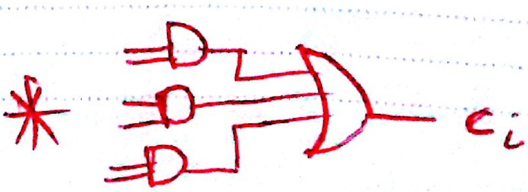
اگر $c_i=0$ ، s_1 می‌شود $xnor(x_i, y_i)$.

c_i	x_i	y_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S_i = x_i \oplus y_i \oplus c_i$$

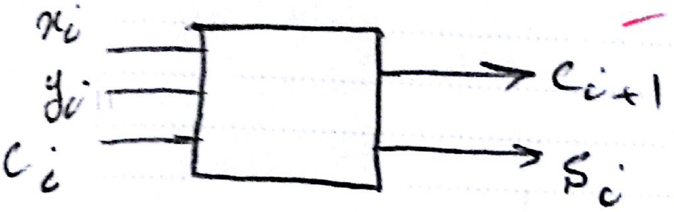
$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i$$





verilog code: → combination ⟹ ترکیبی
→ ~~seq → ترتیبی~~

verilog ترتیبی نیست.




```
module FA (xi, yi, ci, ci+1, si);
  input xi, yi, ci;
  output reg ci+1, si;
  always @ (xi, yi, ci)
  begin
  si = xi ^ yi ^ ci;
  ci+1 = xi*yi + xi*ci + yi ci;
  end
end module
```

می‌توان به جای این کد طولانی نوشت:

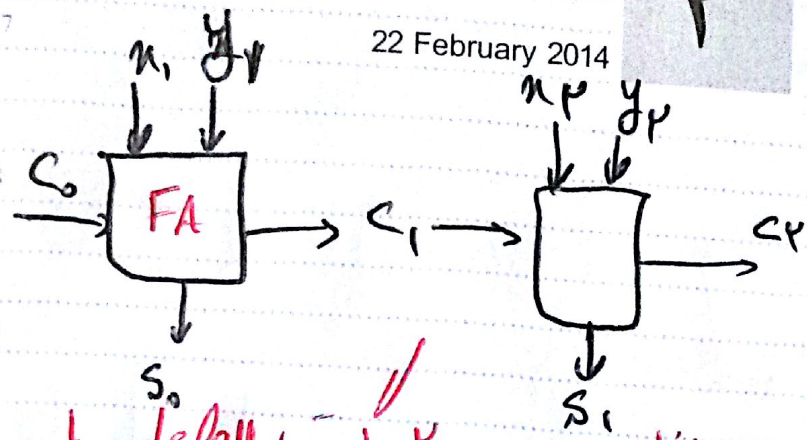
```
assign{ci+1, si} = xi + yi + ci;
```

جمع معمولی دو

(معادل این است که در s_i ، c_{i+1} ریخته می‌شود.)

22 February 2014

شنبه ۲۲ ربیع‌الثانی ۱۴۳۵



ripple carry adder

هر FA طبق شکل * به اندازه‌ی ۲ گیت delay دارد ←

که در کل ۲۸ در نظر می‌گیریم.

← در این صورت کل delay مدار $2n\Delta$ می‌شود.

← اگر از clock استفاده کنیم فاصله‌ی دو تغییر clock اگر t_{clk} باشد باید $t_{clk} > 2n\Delta$ باشد.

```
module ADD4 (ci, x, y, ...);
input [3:0] x, y;
output [3:0] S;
output cout;
wire [3:1] c; zero
FA stage0 (c0, x[0], y[0], S[0], c[1]);
FA stage1 (c1, x[1], y[1], S[1], c[2]);
end module;
```

این کد هم معادل کد هم دارد (صفحه‌ی بعد)

```
assign {cout, s} = {1'b0, x} + {1'b0, y} + cin;
```

← با وجود اینکه این کد جدید کوتاه‌تر است ولی flexibility در کد اول بیشتر است.

جمع و تفریق اعداد علامت‌دار signed :

SM : signed magnitude

1's compl.

2's compl.

← در SM اگر اعداد هم علامت باشند علامت را کنار گذاشته و جمع می‌کنیم سپس علامت را می‌گذاریم.

```
  0101
  0010
------
  0111
```

← علامت تحلیل:

← قدر مطلق عدد هایشان را از هم کم می کنیم و علامت عدد بزرگتر را از نظر اندازه را به آن می دهیم:

$$\begin{array}{r} 0010 \rightarrow +2 \\ +\ 1111 \rightarrow -7 \\ \hline \end{array}$$

$$\begin{array}{r} 1111 \\ 0010 \\ \hline 1101 \end{array}$$

جمع در 1's compl. :

- در بعضی حالات ابهام

$$\begin{array}{r} 0101 \\ 0010 \\ \hline 0111 \end{array}$$

- وقتی carry-out بیرون دهد (یعنی یک خروجی به C_4 بیت چهارم بدهد) داریم:

$$\begin{array}{rr} +5 & 0101 \\ -2 & 1101 \\ \hline & (1)0010 \\ & +1 \\ \hline & 0011 \end{array}$$

← این روش!

1 March 2014 ۲۶ ربیع‌الثانی ۱۴۳۵

← این دو نوع اعداد جهت مثال مناسب hardware نیست.