

به نام خدا

نحوه اجرا کد ها به صورت جداگانه برای سادگی درک مطلب برای شما انجام شده است

در ماژل اول که یک فول اددر تک بیتی هست ۳ ورودی و ۲ خروجی تعریف کردیم و در ادامه ۲ xor و ۲ and و ۱ or نوشتیم که شامل فول اددر تک بیتی ما میشوند

```
1 module fa1bit ( input x , input y , input z , output s , output c );  
2 xor(l,x,y);  
3 xor(s,l,z);  
4 and(p,l,z);  
5 and(q,x,y);  
6 or(c,p,q);  
7  
8 endmodule
```

در ماژل دوم که در آن ماژل اول را نیز صدا زده ایم یک فول اددر ۴ بیتی است که به کمک فول اددر تک بیتی با استفاده از ۴ بار صدا زدن انرا نوشته ایم که شامل ۲ ورودی ۴ بیتی و ۱ خروجی ۵ بیتی به همراه یک وایر ۳ بیتی است تا ۲ ورودی ۴ بیتی را جمع کند و خروجی ۵ بیتی را تحویل دهد

```
10 module fa4bit ( input [3:0] x , input [3:0] y , output [4:0] s );
11 wire [2:0] c;
12 fa1bit u1 ( .x(x[0]), .y(y[0]), .z(1'b0), .s(s[0]), .c(c[0]) );
13 fa1bit u2 ( .x(x[1]), .y(y[1]), .z(c[0]), .s(s[1]), .c(c[1]) );
14 fa1bit u3 ( .x(x[2]), .y(y[2]), .z(c[1]), .s(s[2]), .c(c[2]) );
15 fa1bit u4 ( .x(x[3]), .y(y[3]), .z(c[2]), .s(s[3]), .c(s[4]) );
16
17 endmodule
```

در ماژل سوم که ماژل اصلی ما نیز هست ورودی ۴ بیتی ، ۱ ورودی ۳ بیتی و ۱ خروجی ۷ بیتی قرار می‌دهیم

این ماژل را دو طبقه در نظر می‌گیریم

در طبقه اول به ۸ گیت and نیاز داریم

در اینجا  $A0$  را and کردیم با  $B0, B1, B2, B3$  و بعد  $A1$  را and کردیم با  $B0, B1, B2, B3$  نام خروجی های گیت and را  $w$  قرار می‌دهیم و برای طبقه اول  $w1$  و طبقه دوم  $w2$  قرار می‌دهیم سپس برای خروجی ۸ گیت and به ترتیب  $w1_1$  و  $w1_2$  و ... و  $w1_7$  قرار می‌دهیم

● خروجی and اول (بیت کوچک) یعنی  $B0$  و  $A0$  اولین خروجی مدارمون هست که نام انرا  $c[0]$  قرار می‌دهیم

در مرحله بعد فول اددر ۴ بیتی را صدا می‌زنیم که ورودی های آن ۳ بیت بزرگتر گیت and دو ورودی  $A0$  و  $B[3:0]$  به همراه صفر و ۴ بیت گیت and دو ورودی  $A1$  و  $B[3:0]$  است (چهار بیت را با چهار بیت جمع می‌کنه تو پنج بیت تحویل میده)  
حاصل رو تو پنج بیت با نام های  $C, S3, S2, S1, S0$  تحویل میده  
نام خروجی های فول اددر ۴ بیتی را  $g$  قرار می‌دهیم  $g1$  و  $g2$  و  $g3$  و  $g4$

●: خروجی اول (بیت کوچیک) فول اددر دومین خروجی مدارمون هست که نام انرا  $c[1]$  قرار می‌دهیم

در مرحله بعد میرسیم به طبقه دوم که در اینجا ۴ گیت and داریم

در اینجا  $A1$  را and کردیم با  $B0, B1, B2, B3$

خروجی های انرا  $w2_1$  و  $w2_2$  و  $w2_3$  و  $w2_4$  قرار می‌دهیم

سپس فول اددر ۴ بیتی را صدا می‌زنیم

خروجی های گیت and به همراه ۳ بیت بزرگ تر خروجی فول اددر طبقه اول و  $c$  را وارد فول

اددر ۴ بیتی طبقه دوم میکنیم که خروجی های ان خروجی های مدارمان نیز هست که  $c[2]$  و  $c[3]$  و  $c[4]$  و  $c[4]$  و  $c[6]$  مینامیم

```
19
20 module zarb ( input [2:0]a , input [3:0]b , output [6:0]c );
21
22 and ( c[0] , b[0] , a[0] );
23 and ( w1_1 , b[1] , a[0] );
24 and ( w1_2 , b[2] , a[0] );
25 and ( w1_3 , b[3] , a[0] );
26
27 and ( w1_4 , b[0] , a[1] );
28 and ( w1_5 , b[1] , a[1] );
29 and ( w1_6 , b[2] , a[1] );
30 and ( w1_7 , b[3] , a[1] );
31
32 fa4bit s1 (
33 .x( { 1'b0 , w1_3 , w1_2 , w1_1 } ),
34 .y( { w1_7 , w1_6 , w1_5 , w1_4 } ),
35 .s( { g4 , g3 , g2 , g1 , c[1] } )
36 );|
37
38 and ( w2_1 , b[0] , a[2] );
39 and ( w2_2 , b[1] , a[2] );
40 and ( w2_3 , b[2] , a[2] );
41 and ( w2_4 , b[3] , a[2] );
42
43 fa4bit s2 (
44 .x( { g4 , g3 , g2 , g1 } ),
45 .y( { w2_4 , w2_3 , w2_2 , w2_1 } ),
46 .s( { c[6] , c[5] , c[4] , c[3] , c[2] } )
47 );
48
49
50 endmodule
```