

## رجسٹر ac

Ln#	
1	module ac(d,clk,load,inr,clr,q);
2	input [15:0] d;
3	input clk,load,inr,clr;
4	output reg [15:0] q;
5	
6	always@(posedge clk,d)
7	begin
8	if (clr) q = 16'b0;
9	else if (inr) q = q + 1'b1;
10	else if (load) q = d;
11	end
12	
13	
14	endmodule
15	

در این کد، کارکرد رجسٹر AC توصیف می شود. در خط ۶ گفته شده است در صورت رخداد لبه ی بالا رونده ی CLK و یا تغییر d، خطوط زیر always اجرا می شود. اگر clr یک باشد خروجی q صفر می شود. اگر inr یک باشد یک واحد به q اضافه می شود. اگر load یک باشد ورودی d درون خروجی q قرار میگیرد.

با توجه به ترتیب توصیف if، clr نسبت به inr و inr نسبت به load اولویت دارد. یعنی اگر هر سه یک باشند clr انجام می شود.

ورودی d و خروجی q ۱۶ بیتی هستند. ورودی d از مدار جمع کننده و منطق می آید و خروجی q به گذرگاه ریخته می شود.

ورودی inr,load,clr,clk تک بیتی هستند.

حال می توانیم از این مدار استفاده کنیم در ساختن مدار اصلی.

## مدار اصلی

Ln#	
1	<code>module main(d0,d1,d2,p,r,t5,b11,b4,b7,b6,b5,d_ac,clk,q_ac);</code>
2	<code>input d0,d1,d2,p,r,t5,b11,b4,b7,b6,b5;</code>
3	<code>input [15:0] d_ac;</code>
4	<code>input clk;</code>
5	<code>output [15:0] q_ac;</code>
6	
7	<code>and(q_and,d0,t5);</code>
8	<code>and(q_add,d1,t5);</code>
9	<code>and(q_dr,d2,t5);</code>
10	<code>and(q_inpr,p,b11);</code>
11	<code>and(q_com,r,b4);</code>
12	<code>and(q_shr,r,b7);</code>
13	<code>and(q_shl,r,b6);</code>
14	<code>and(inc,r,b5);</code>
15	<code>and(clr,r,b11);</code>
16	
17	<code>or(ld,q_and,q_add,q_dr,q_inpr,q_com,q_shr,q_shl);</code>
18	
19	<code>ac ul(d_ac,clk,ld,inc,clr,q_ac);</code>
20	
21	
22	<code>endmodule</code>
23	

توصیف گیت ها در خطوط ۷ تا ۱۵ انجام شده است. مثلاً حاصل `and` ورودی های `t5` و `d0` درون `q_and` ریخته می شود. حاصل `and` `b11` و `p` درون `q_inpr` ریخته می شود. حاصل `and` دو ورودی `b6` و `r` درون `q_shl` ریخته می شود.

در خط ۱۷، حاصل گیت های `q_and` و `q_add` و `q_dr` و `q_inpr` و `q_com` و `q_shr` و `q_shl` با یکدیگر `or` می شوند و درون `ld` ریخته می شوند. این یعنی که هر زمان حاصل یکی از این گیت ها یک باشد ورودی از مدار جمع کننده و منطق درون خروجی `ac` ریخته می شود. البته با شرط صفر بودن `inc`, `clr`.

در خط ۱۹، رجستر `ac` را که قبلاً نوشتیم فراخوانی می کنیم. ورودی `d_ac` را به ورودی `d` آن وصل می کنیم. `clk` را به `clk` آن وصل می کنیم. `ld` را به `load` آن وصل می کنیم. `inc` را به `incr` آن وصل می کنیم. `clr` را به `clr` آن وصل می کنیم. `q_ac` را نیز به `q` آن متصل می کنیم.

## کد تست پنج

```

Ln#
1 module main_tb;
2
3     reg d0,d1,d2,p,r,t5,b11,b4,b7,b6,b5;
4     reg [15:0] d_ac;
5     reg clk;
6     wire [15:0] q_ac;
7
8     main uut (.d0(d0), .d1(d1), .d2(d2), .p(p), .r(r), .t5(t5), .b11(b11), .b4(b4),
9             .b7(b7), .b6(b6), .b5(b5), .d_ac(d_ac), .clk(clk), .q_ac(q_ac));
10
11     always #10 clk = ~clk;
12
13     initial begin
14         d0 = 0;
15         d1 = 1;
16         d2 = 0;
17         p = 0;
18         r = 0;
19         t5 = 1;
20         b11 = 0;
21         b4 = 0;
22         b7 = 0;
23         b6 = 0;
24         b5 = 0;
25         d_ac = 16'b1100110011100001;
26         clk = 0;
27
28         #20;
29         d1 = 1'b0;
30         t5 = 1'b0;
31         r = 1'b1;
32         b11 = 1'b1;
33
34         #40;
35         r = 1'b0;
36         b11 = 1'b0;
37         d2 = 1'b1;
38         t5 = 1'b1;
39         d_ac = 16'b0001110111111111;
40
41         #20;
42         d2 = 1'b0;
43         t5 = 1'b0;
44         r = 1'b1;
45         b5 = 1'b1;
46
47         #40;
48         b11 = 1'b1;
49         b5 = 1'b0;
50
51     end

```

در تست پنج، ورودی‌های مدار اصلی بعنوان reg و خروجی‌ها بعنوان wire تعریف می‌شوند. در خط ۸ و ۹ مدار اصلی به رجیسترها و wire های این مدار متصل می‌شوند. در خط ۱۱ نوسان clk انجام می‌شود با دوره تناوب ۱۰ نانوثانیه. در خطوط ۱۴ به بعد به ورودی‌ها مقدار دهی می‌شود. ابتدا t5 و d1 یک می‌شوند تا عملیات add انجام شود. با یک شدن add، ld یک می‌شود و مقدار d درون q باید قرار بگیرد.

سپس در خط ۲۸، r و b11 یک می‌شوند تا عملیات clr انجام شود با اینکار خروجی q صفر باید بشود.

در خط ۳۴، با تغییر ورودی‌ها، باید دوباره عملیات ld انجام شود و d درون q قرار بگیرد. چون t5، d2 هر دو یک هستند.

در خط ۴۱، r و b5 یک می‌شود تا عملیات inr انجام شود. با این کار در هر لبه ی بالا رونده ی کلاک یکی به مقدار درون رجیستر ac اضافه می‌شود.

در خط ۴۷، دوباره عملیات clr انجام می شود و محتوای درون رجیستر AC صفر می شود.

## نتیجه ی شبیه سازی

