

گزارش کار چهارم آزمایشگاه معماری کامپیوتر

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

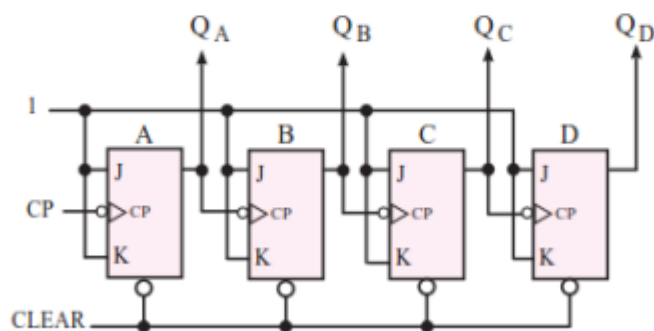
استاد راهنما: دکتر حاجی زاده

چکیده:

در این جلسه ابتدا اقدامات انجام شده در جلسه‌ی گذشته به طور کلی مرور و سپس با انواع مختلف دیکودرها و شمارنده‌ها و اجزای تشکیل دهنده‌ی آن‌ها آشنایی به عمل آمد. بعد از پایان این بخش، دانشجویان طراحی و پیاده‌سازی قطعه‌ی Latch و مقدمات لازم برای طراحی یک شمارنده به کمک زبان سطح بالای Verilog را در قالب گروه‌ها آغاز کردند. در ادامه ابتدا به معرفی کلی شمارنده‌ها و دیکودرها پرداخته و به کمک چند تصویر و جدول، با ویژگی‌های مختلف آن‌ها آشنا می‌شویم.

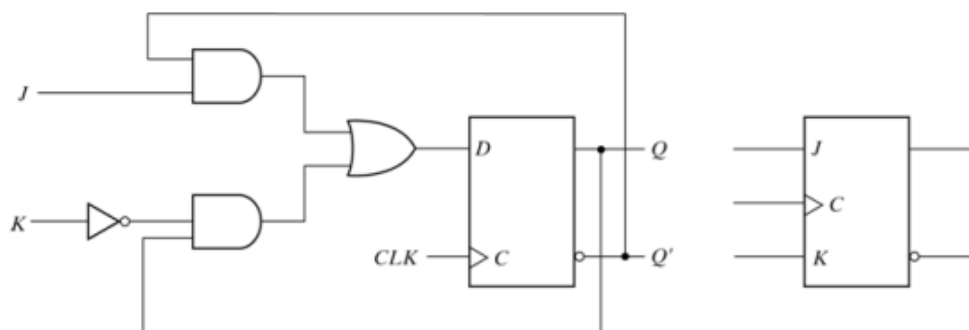
شکل زیر ساختار کلی یک نوع شمارنده را نشان می‌دهد:

شمارنده ها (Counters)



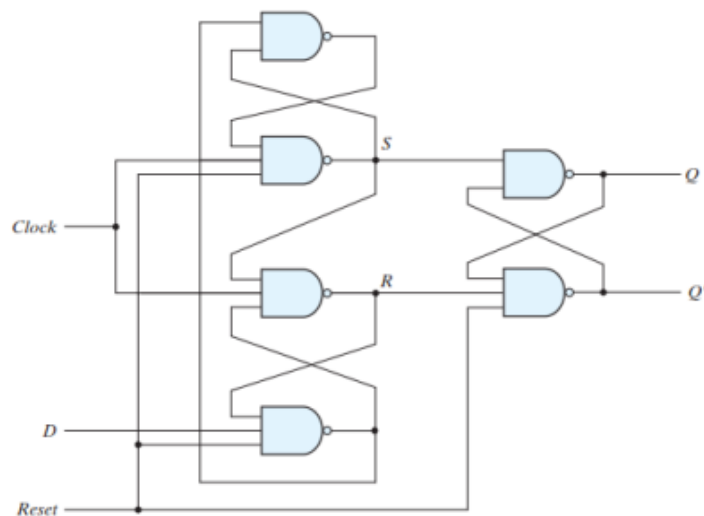
شکل زیر مدار داخلی هر کدام از بلوک‌های تشکیل‌دهنده این شمارنده را نشان می‌دهد:

مدار داخلی هر بلوک



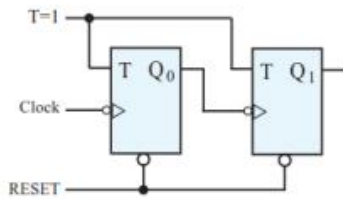
فلیپ‌فلاپ نیز، یکی دیگر از قطعاتی است که در آینده با آن سروکار خواهیم داشت. شکل زیر ساختار یک فلیپ‌فلاپ از نوع D را نشان می‌دهد:

فلیپ‌فلاپ D دارای پایه Reset

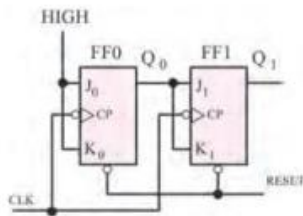


به‌طور کلی شمارنده‌ها را می‌توان به دو نوع تقسیم کرد:

• شمارنده های آسنکرون (Asynchronous)



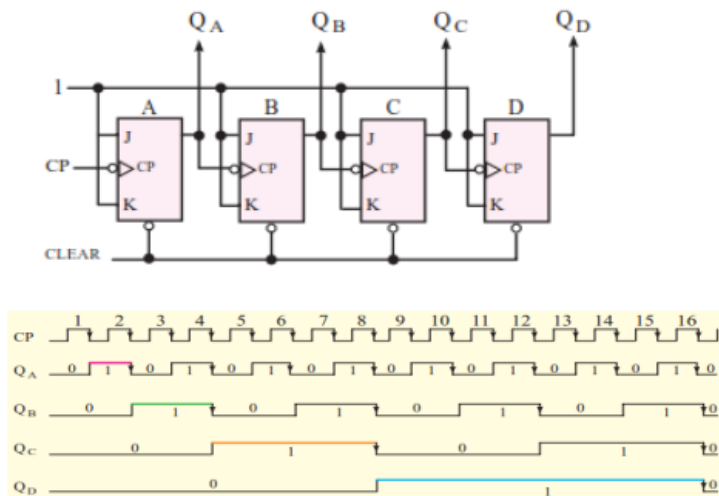
• شمارنده های سنکرون (synchronous)



5

شکل زیر اطلاعاتی را درباره‌ی شمارنده‌های آسنکرون صعودی نشان می‌دهد:

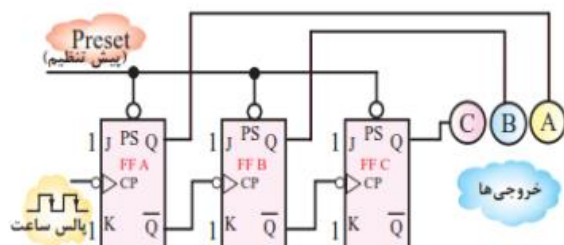
شمارنده های آسنکرون صعودی (Up Counter) :



پالس ساعت	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

همچنین اطلاعاتی درباره‌ی شمارنده‌های آسنکرون نزولی در شکل زیر به نمایش گذاشته شده:

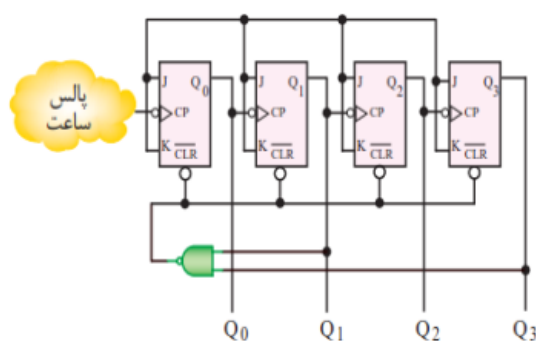
شمارنده های آسنکرون نزولی (Down Counter):



تعداد پالس های ساعت ورودی	خروجی ها			شمارش ده دهی خروجی
	C	B	A	
0	1	1	1	7
1	1	1	0	6
2	1	0	1	5
3	1	0	0	4
4	0	1	1	3
5	0	1	0	2
6	0	0	1	1
7	0	0	0	0
8	1	1	1	7

بازهی شمارش اعداد یک شمارنده را می‌توانیم به کمک اعداد BCD معین کنیم:

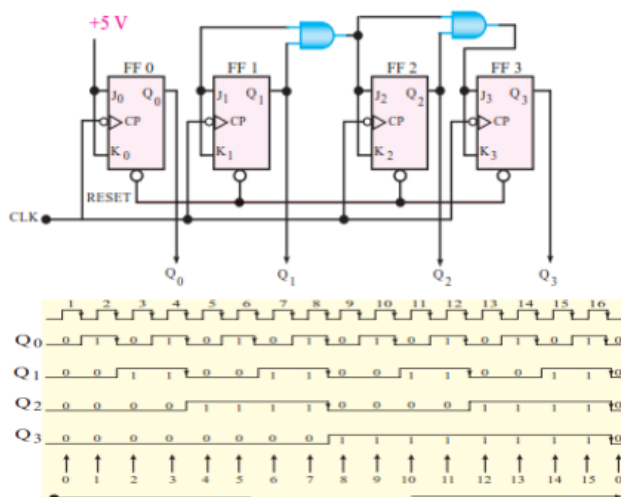
شمارنده های آسنکرون ده دهی BCD (Binary Coded Decimal):



Q ₃	Q ₂	Q ₁	Q ₀	Decimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

در تصویر بعد، مشخصات نوع دیگری از شمارنده‌ها با عنوان شمارنده‌های سنکرون صعودی را مشاهده می‌کنید:

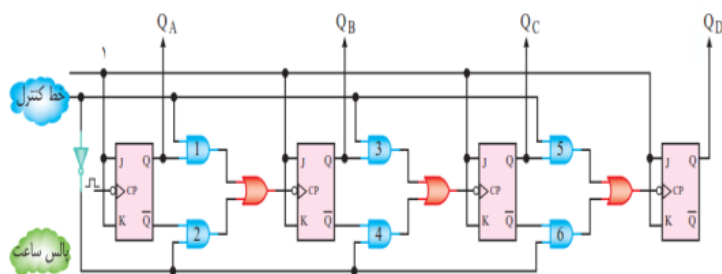
شمارنده های سنکرون صعودی :



معادل ده‌دهی	Q ₃	Q ₂	Q ₁	Q ₀
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

با ایجاد تغییرات کوچکی می‌توانیم قابلیت صعود و نزول را برای یک شمارنده تعبیه کنیم:

• شمارنده های صعودی و نزولی (Up/Down Counter) :

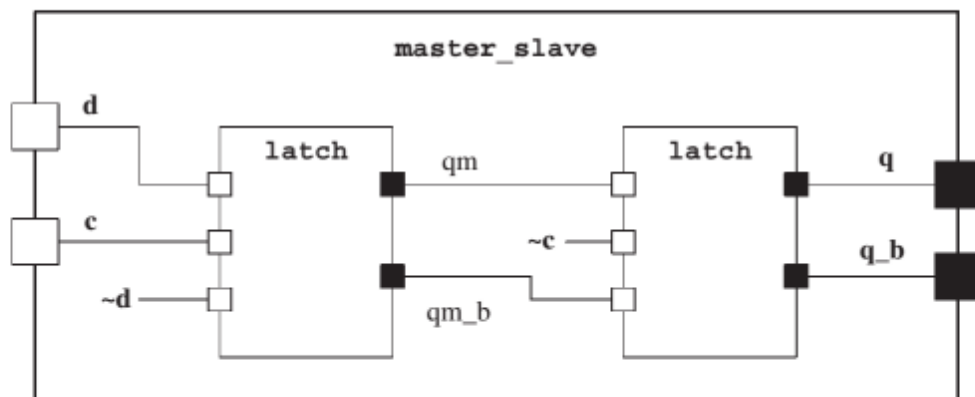


\Rightarrow خط کنترل = 1 \Rightarrow فعال $\{AND_1, AND_3, AND_5$
 غیر فعال $\{AND_2, AND_4, AND_6$
 شمارنده صعودی
 \Rightarrow خط کنترل = 0 \Rightarrow غیر فعال $\{AND_1, AND_3, AND_5$
 فعال $\{AND_2, AND_4, AND_6$
 شمارنده نزولی

در ادامه قصد داریم ساختار یک Master – Slave D Flip-Flop را شبیه‌سازی کنیم:

Basic Memory Components

Master-Slave D Flip-Flop

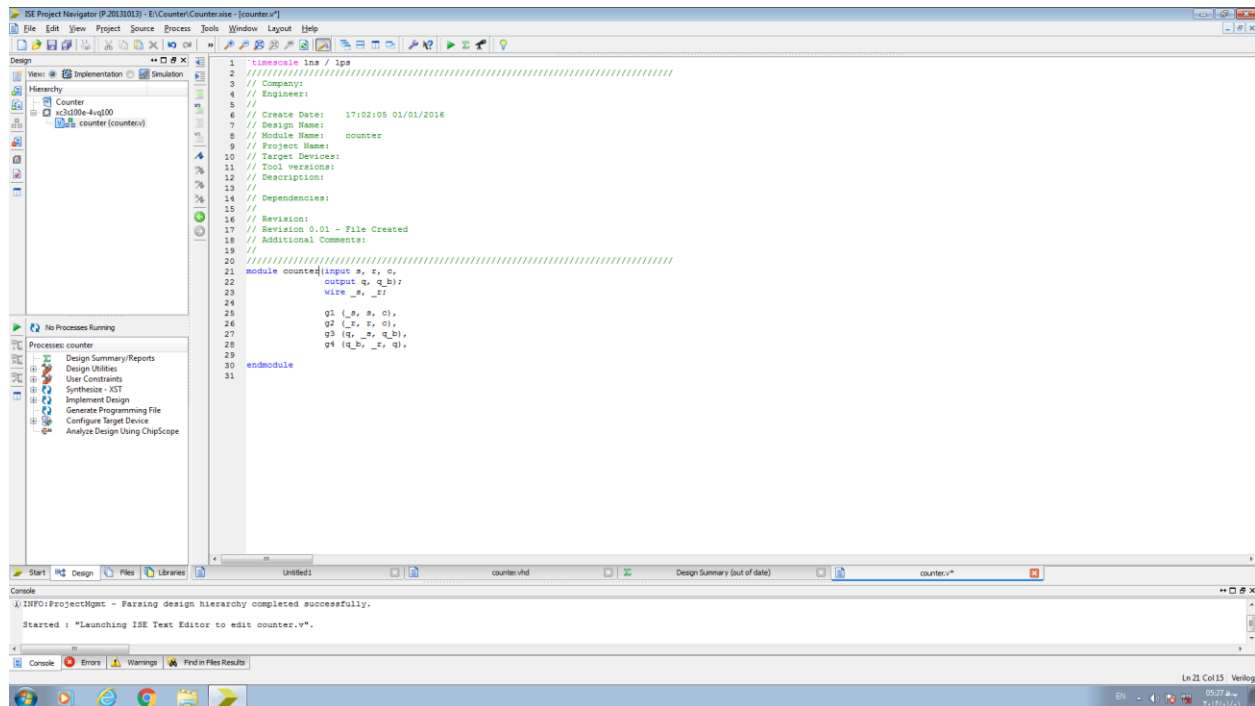


برای این منظور، لازم است که ابتدا قطعه‌کد زیر را در محیط نرم‌افزار ISE برنامه‌نویسی کنیم:

```
`timescale 1ns/100ps

module latch_p #(parameter tplh=3, tphl=5) (input s, r, c,
      output q, q_b );
    wire _s, _r;
    nand #(tplh,tphl)
        g1 ( _s, s, c ),
        g2 ( _r, r, c ),
        g3 ( q, _s, q_b ),
        g4 ( q_b, _r, q );
endmodule
```

نتیجه‌ی نوشتن این برنامه در تصویر زیر آورده شده است:

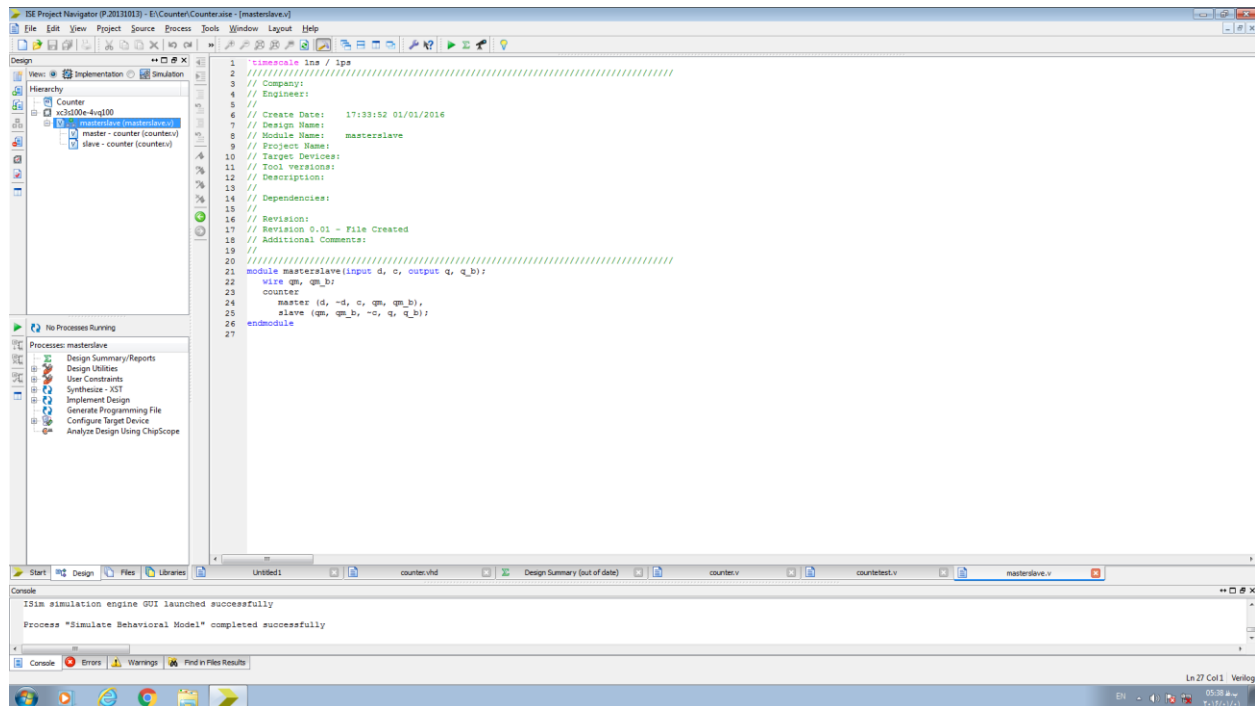


حال لازم است در یک ماژول مجزا به طراحی بخش‌های master و slave بپردازیم:

```
`timescale 1ns/100ps

module master_slave (input d, c, output q, q_b );
    wire qm, qm_b;
    defparam master.tplh=4, master.tphl=4, slave.tplh=4,
               slave.tphl=4;
    latch_p
        master ( d, ~d, c, qm, qm_b ),
        slave ( qm, qm_b, ~c, q, q_b );
endmodule
```

حاصل این کار در شکی زیر آورده شده:



در نهایت لازم است که با نوشتن یک Test Module عملکرد برنامه‌ی طراحی شده را ارزیابی کنیم:

Basic Memory Components

Master-Slave D Flip-Flop

```

module tb_master;

    // Inputs
    reg d;
    reg c;

    // Outputs
    wire q;
    wire q_b;

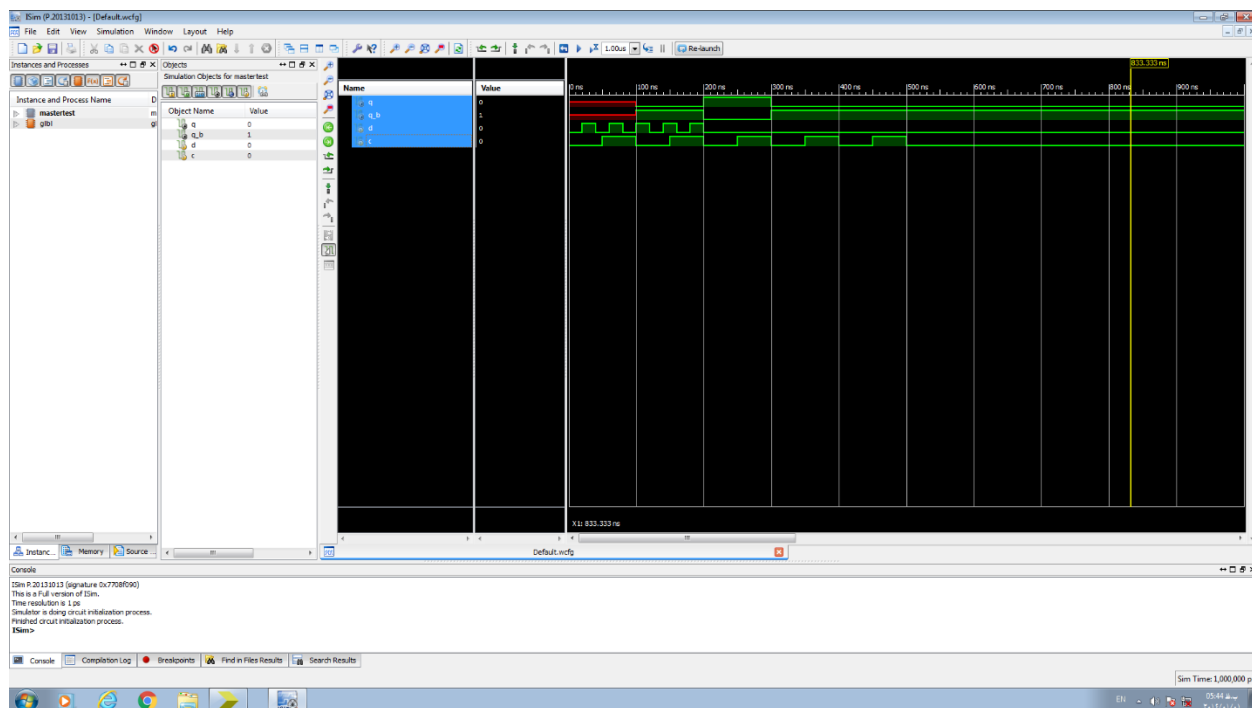
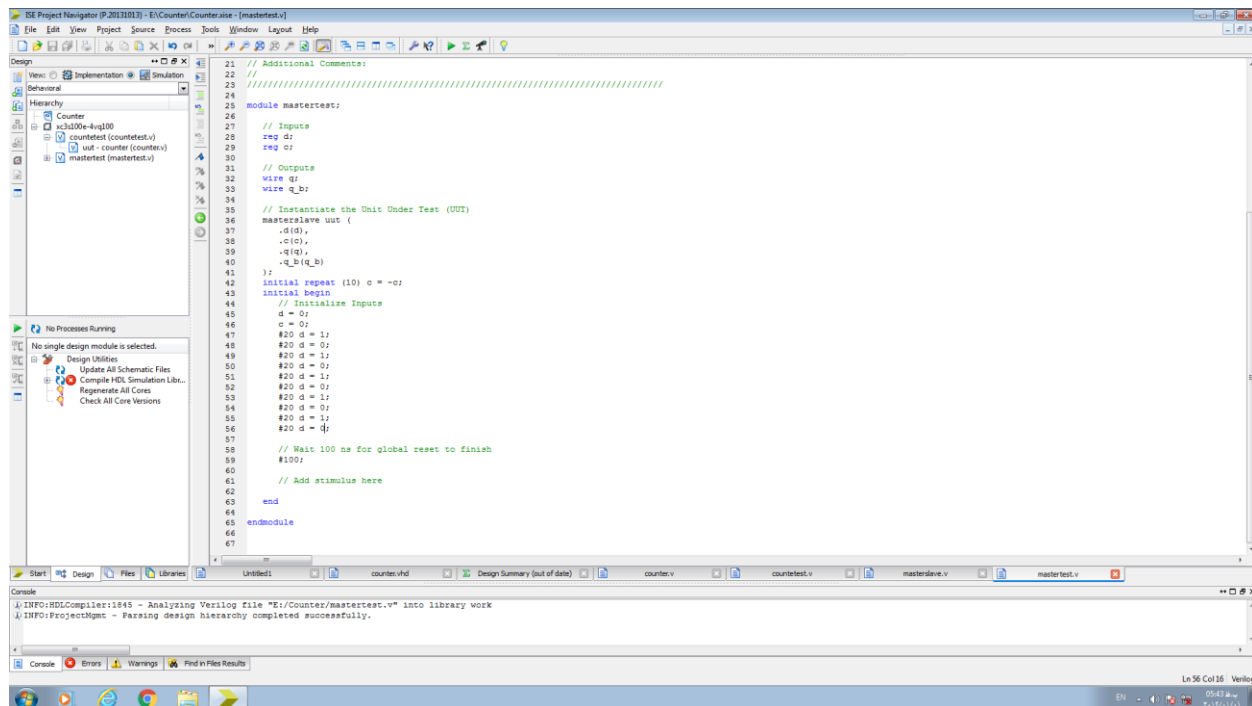
    // Instantiate the Unit Under Test
    master_slave uut (
        .d(d),
        .c(c),
        .q(q),
        .q_b(q_b)
    );

    initial repeat (10) #50 c = ~c;

```

[illegible]

حاصل نوشتن ماژول‌های مربوط به قسمت Test و شبیه‌سازی برنامه در محیط Simulation در
تصاویر زیر آورده شده‌اند:



پایان.