

گزارش کار پنجم آزمایشگاه معماری کامپیوتر

تهیه و تنظیم: مبین خیبری

شماره دانشجویی: 994421017

استاد راهنما: دکتر حاجی زاده

چکیده:

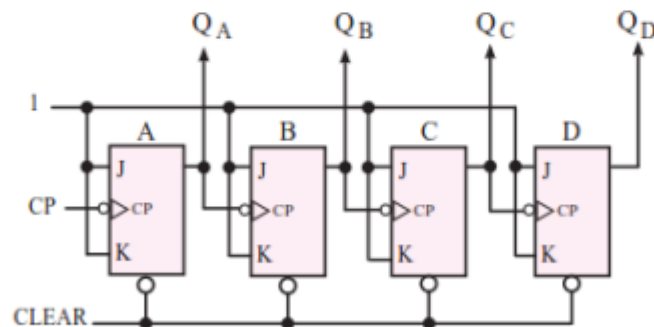
در این جلسه ابتدا اقدامات انجام شده در جلسه‌ی گذشته به طور کلی مرور و سپس با ویژگی‌ها و اجزای مختلف شمارنده‌ها آشنایی به عمل آمد.

بعد از پایان این بخش، دانشجویان ادامه‌ی فرآیند طراحی و پیاده‌سازی شمارنده را به کمک قطعات لچ و فلیپ‌فلاپ و نیز محیط شبیه‌سازی نرم‌افزار ISE پی گرفتند.

در انتهای جلسه نیز به طور مختصر، مشخصات انواع مختلف شیفت رجیستر مورد بررسی قرار گرفتند.

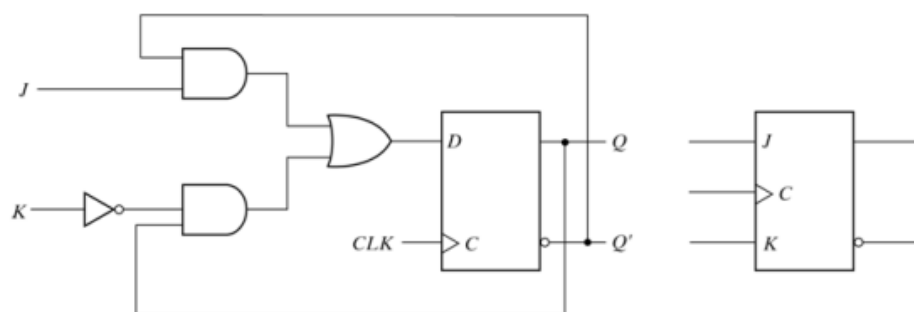
شکل زیر ساختار کلی یک شمارنده را نشان می‌دهد:

شمارنده ها (Counters)



مدار داخلی هر کدام از بلوک‌های تشکیل‌دهنده‌ی این شمارنده در شکل زیر آورده شده است:

مدار داخلی هر بلوک



برای طراحی چنین مداری به کمک زبان Verilog، نیاز داریم که ابتدا یک قطعه SR-Latch را مطابق برنامه‌ی زیر شبیه‌سازی کنیم:

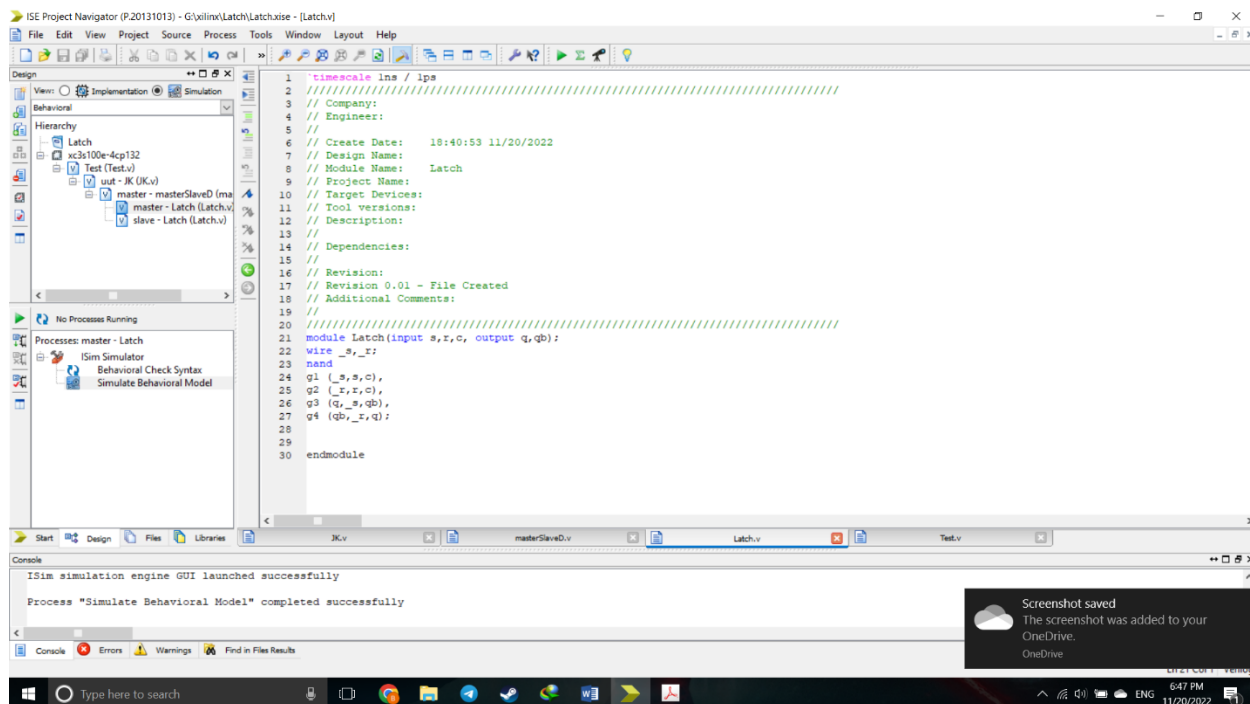
Basic Memory Components

All NAND Clocked SR-Latch

```
`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
```

تصویر زیر حاصلِ نوشتنِ کد بالا در محیط نرم‌افزار ISE را نشان می‌دهد:



یکی دیگر از قطعاتی که برای طراحی هر کدام از بلوک‌های یک شمارنده به آن احتیاج داریم، D Flip-Flop نام دارد. کد زیر به منظور طراحی چنین قطعه‌ای نوشته شده است:

Basic Memory Components

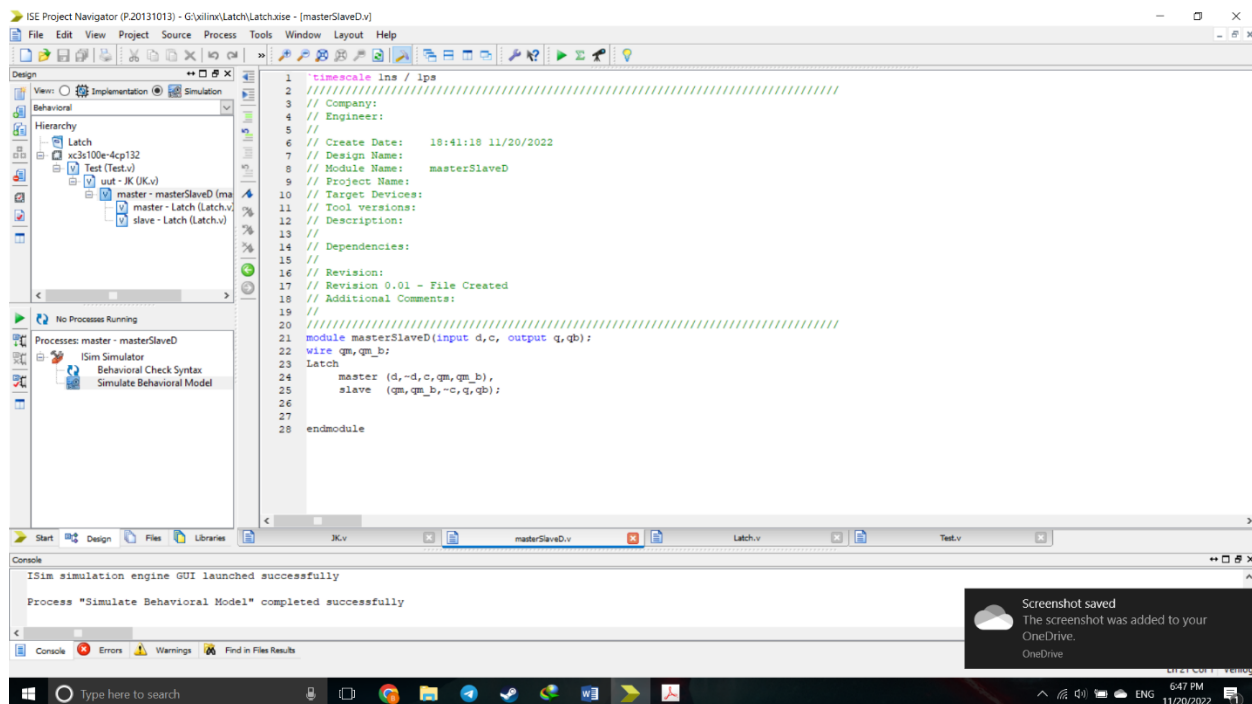
Master-Slave D Flip-Flop

```
`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
```

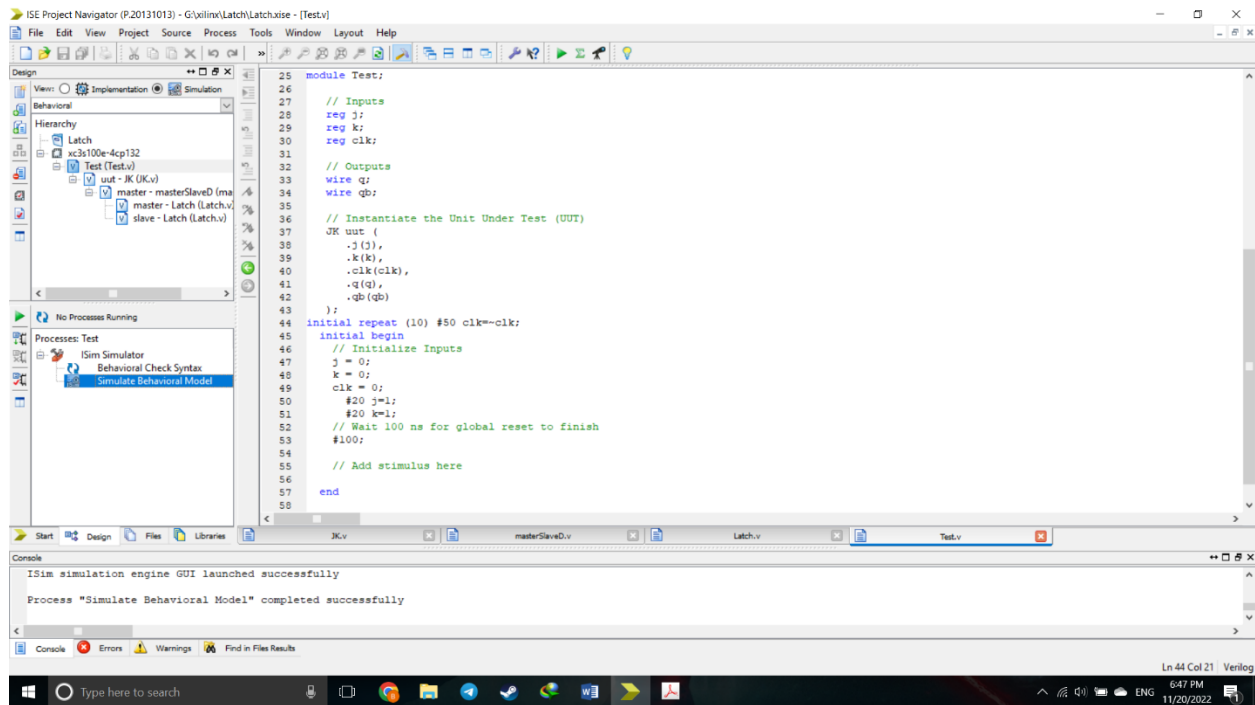
تصویر زیر حاصلِ نوشتنِ کد بالا در محیط نرم‌افزار ISE را نشان می‌دهد:



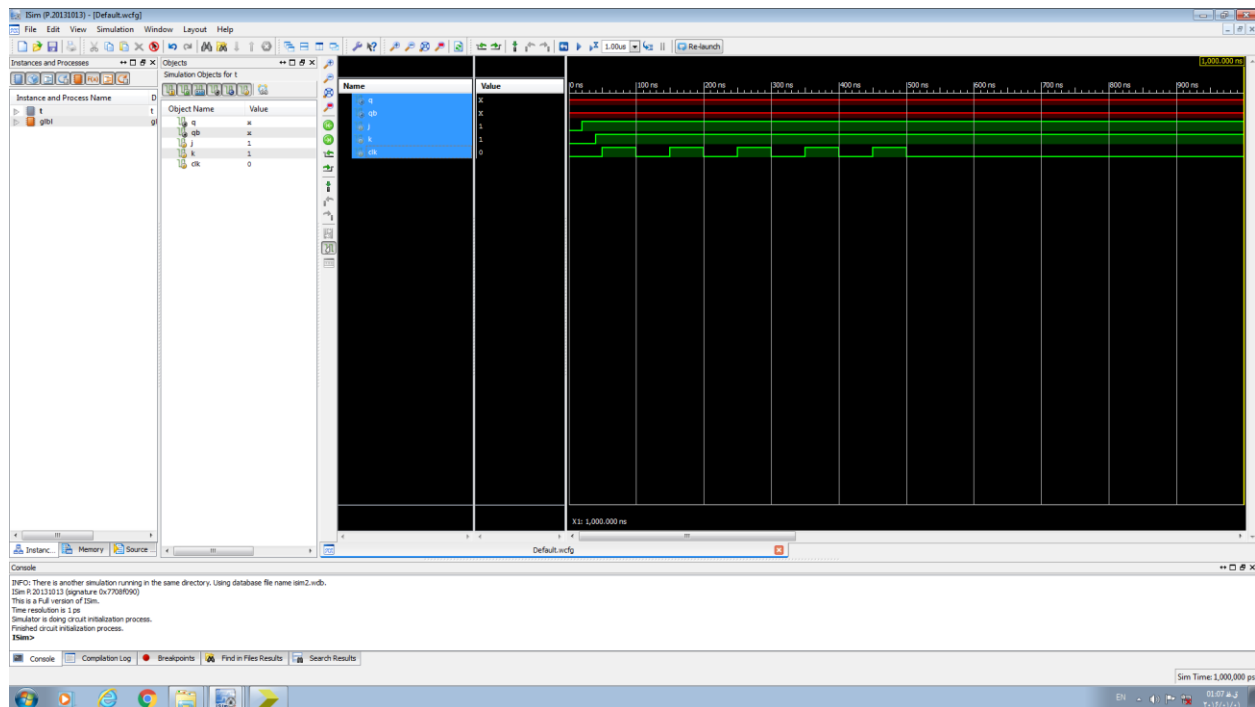
در نهایت لازم است که کدهای اجرایی مربوط به قطعه‌ی JK Flip-Flop را مطابق تصویر زیر بنویسیم:

```
1 `timescale 1ns / 1ps
2 ///////////////////////////////////////////////////
3 // Company:
4 // Engineer:
5 //
6 // Create Date:    18:41:42 11/20/2022
7 // Design Name:
8 // Module Name:    JK
9 // Project Name:
10 // Target Devices:
11 // Tool versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////
21 module JK(input j,k,clk, output q,qb);
22 wire and1,and2,or1;
23
24 and(and1,j,qb);
25 and(and2,~k,q);
26 or (or1,and1,and2);
27
28 masterSlaveD
29 master(or1,and1,and2);
30
31
32 endmodule
33
```

دقت کنید که در این کد، توابع نوشته شده برای D Flip-Flop یا بخش های master فرخوانی شده اند. در انتها برای ارزیابی عملکرد کدهای نوشته شده، قطعه کد تست زیر را نوشته و کل برنامه را در محیط Simulation شبیه سازی می کنیم:

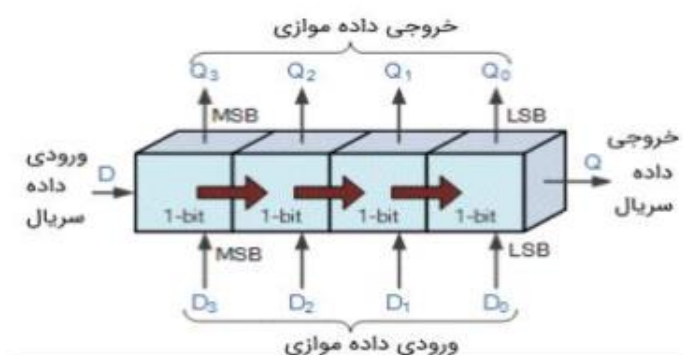


حاصل نوشتن ماژول های مربوط به قسمت Test و اجرای شبیه سازی، در تصویر زیر آورده شده:



در انتها، به طور کوتاه به معرفی انواع مختلف شیفت رجیسترها می‌پردازیم. در جلسه‌ی بعدی، به طور مفصل درباره‌ی این قطعات بحث و گفتگو خواهیم کرد. شکل زیر ساختار کلی این قطعات را نشان می‌دهد:

شیفت رجیستر (Shift Register)



این قطعات، چهار مد کاری مختلف دارند که در تصویر زیر به آن‌ها پرداخته شده است:

مدهای کاری شیفت رجیسترها :

- ورودی سریال-خروجی موازی (Serial-in to Parallel-out) یا SIPO
- ورودی سریال-خروجی سریال (Serial-in to Serial-out) یا SISO
- ورودی موازی-خروجی سریال (Parallel-in to Serial-out) یا PISO
- ورودی موازی-خروجی موازی (Parallel-in to Parallel-out) یا PIPO

پایان.