به نام خدا

درس: طراحی سیستم‌های دیجیتال

استاد: دکتر فصحتی

تمرین سری اول

علی نظری
99102401

در این تمرین باید با کمک j-k flip flop نخست یک t flip flop بسازیم و سپس، با این t flip flop یک counter پنج بیتی بسازیم.
پس نخست سعی می‌کنیم که j-k flip flop را بسازیم. برای ساخت این قطعه، باید از توصیف رفتاری استفاده کنیم چون بر حسب جدول صحت آن می‌خواهیم مدار را بکشیم:

Reset	J	K	Next Q
1	Dont care	Dont care	0
0	0	0	Q
0	0	1	0
0	1	0	1
0	1	1	~Q

حال به کمک این جدول، می‌توان کد وریلاگ این بخش را زد. در واقع بیشتر این بخش را خود استاد سر کلاس زدند و برای همین این کد، بسیار شبیه به کد استاد است:

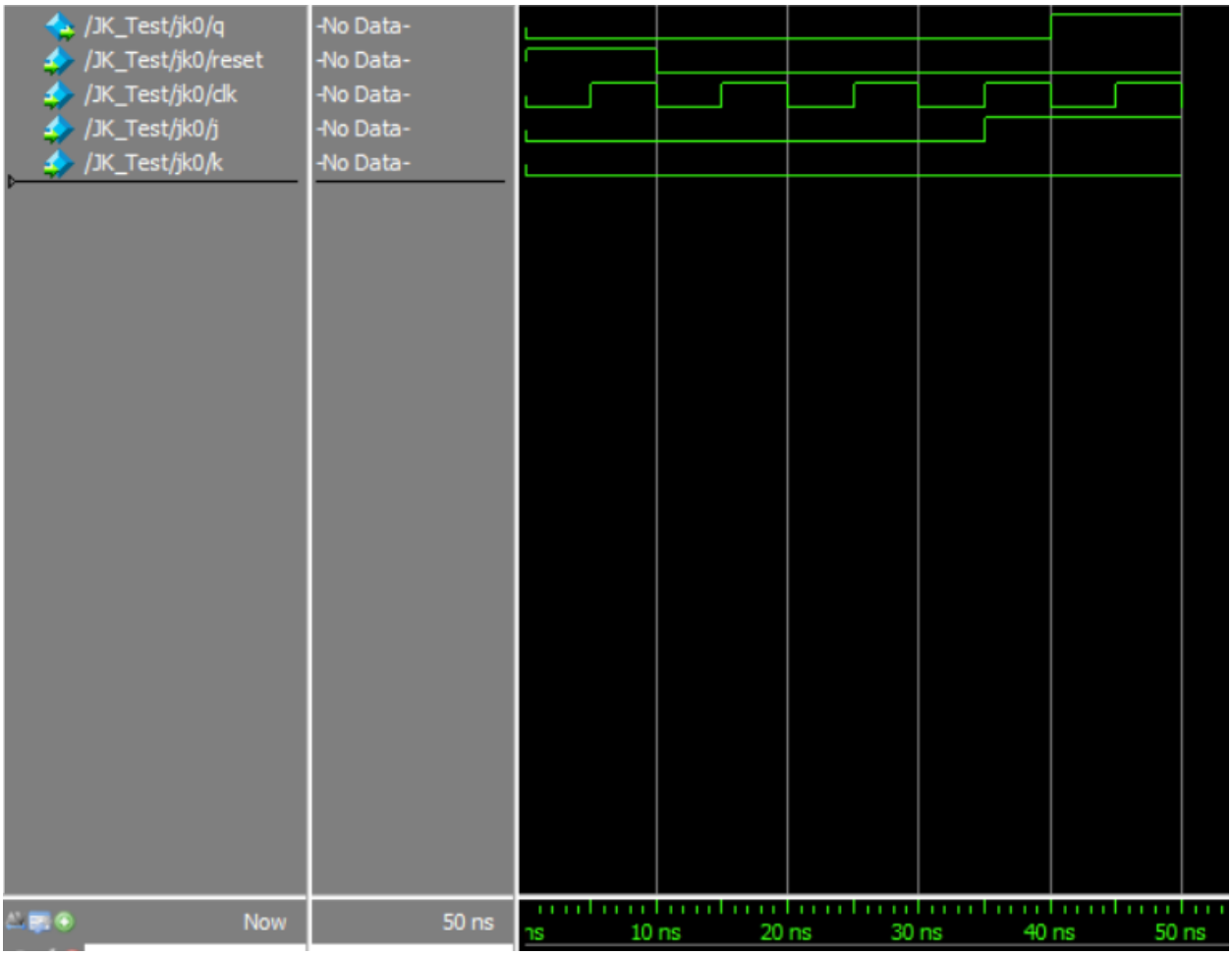
```
module JK (q, reset, clk, j, k);
        output q;
        input reset, clk, j, k;
        reg q;
        always @(posedge reset or negedge clk) begin
                if (reset == 1) begin
                        q = 0;
                end
                else begin
                        if (j == 1 && k == 0) begin
                                q = 1;
                        end
                        if (j == 1 && k == 1) begin
                                q = ~q;
                        end
                        if (j == 0 && k == 1) begin
                                q = 0;
                        end
                end
        end
endmodule
```

که در واقع حالت‌های مختلف را در نظر گرفته ایم و برای حالتی که هر دو صفر هستند و Q تغییری نمی‌کند، کار خاصی لازم نیست انجام دهیم و اصلا حالت آن را نمی‌گذاریم.

حال این بخش را تست می‌کنیم تا از صحت عملکرد آن مطمئن شویم:

```
module JK_Test;
        reg j; reg k;
        reg clk; reg reset; wire q;

        //instantiate the design block
        JK jk0(q, reset, clk, j, k);

        //control the clk signal that drives the design block
        initial begin
                reset = 1'b1;
                j = 1'b0;
                k = 1'b0;
                clk = 0'b0;
        end
        always #5 clk = ~clk;

        //control the reset signal that drives the design block
        initial
                begin
                        reset = 1'b1;
                        #10 reset = 1'b0;
                        #25 j = 1'b1;
                        #25 k = 1'b1;
                        #35 j = 1'b0;
                        #40 k = 1'b0;
                        #50 $stop;
                end

        initial //monitor the output
                $monitor($time, " Output q = %d", q);
endmodule
```

و خروجی آن آن مانند عکس زیر می‌شود که درست به نظر می‌رسد:



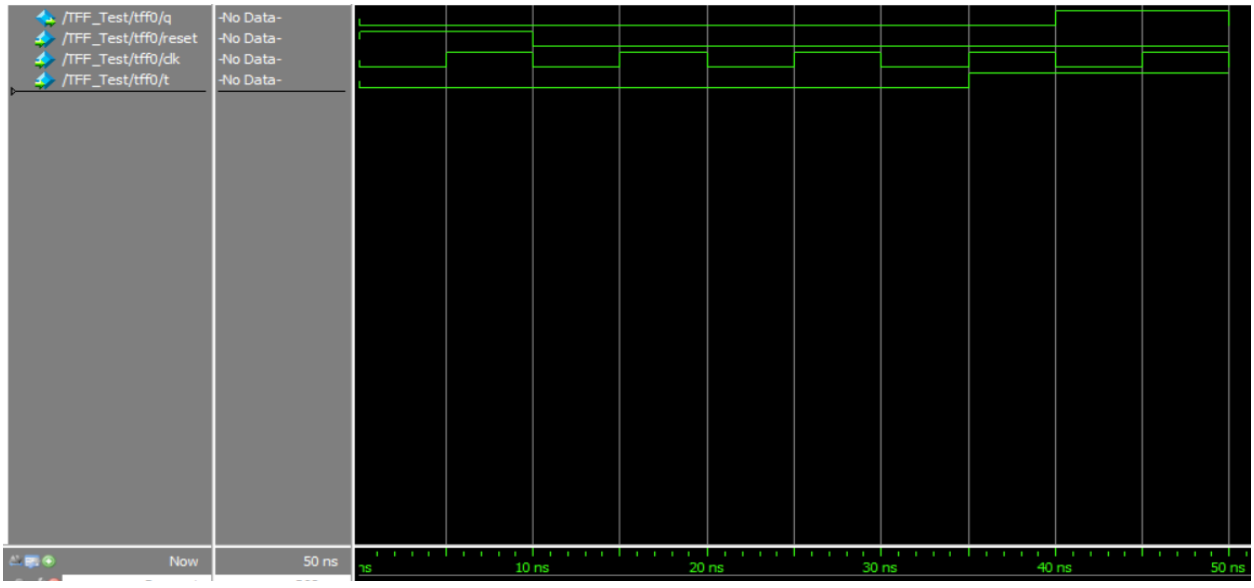
حال به کمک این J-K flip flop باید T flip flop بسازیم و همانطور که از درس مدارهای منطقی می‌دانیم، اگر ورودی T یکسان را هم به J وصل کنیم و هم به K، آنگاه T flip flop ما ساخته می‌شود. کد آن بسیار ساده و مانند عکس زیر است:

```
module TFF (q, reset, clk, t);
        output q;
        input reset, clk, t;
        JK jk0(q, reset, clk, t, t);
endmodule
```

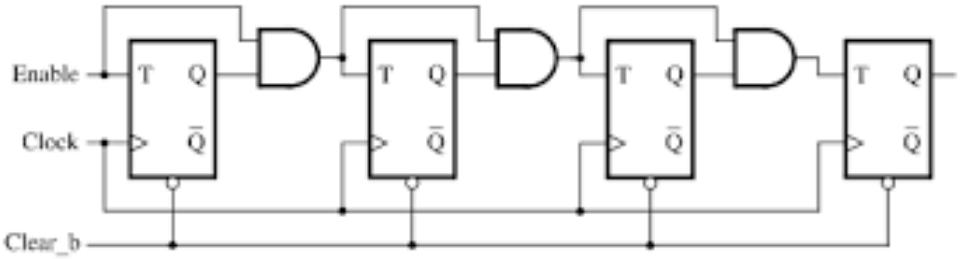
حال این کد را هم باید بررسی کنیم:

```
module TFF_Test;
        reg t;
        reg clk; reg reset; wire q;

        //instantiate the design block
        TFF tff0(q, reset, clk, t);

        //control the clk signal that drives the design block
        initial begin
                reset = 1'b1;
                t = 1'b0;
                clk = 1'b0;
        end
        always #5 clk = ~clk;

        //control the reset signal that drives the design block
        initial
                begin
                        reset = 1'b1;
                        #10 reset = 1'b0;
                        #25 t = 1'b1;
                        #35 t = 1'b0;
                        #50 $stop;
                end

        initial //monitor the output
                $monitor($time, " Output q = %d", q);
endmodule
```

و نتیجه تست هم مانند عکس زیر است که باز هم درست به نظر می‌آید:



برای ساخت شمارنده، از درس مدار منطقی و مثالی که استاد حل کردند، باید ورودی T flip flop یک باشد تا همیشه شاهد toggle باشیم و به این ترتیب، مدار کار کند و مدام بین صفر و یک جابه‌جا شود. و هر عددی که در نظر بگیریم، زمانی که رقم سمت راستش، از یک به صفر تغییر حالت می‌دهد، این رقم باید یکبار toggle شود که معادل یک کلاک حساس به لبه پایین رونده است. پس خروجی هر T flip flop باید به ورودی T flip flop بعدی وصل شود. چیزی شبیه به عکس زیر ولی بدون در نظر گرفتن آن ورودی enable:



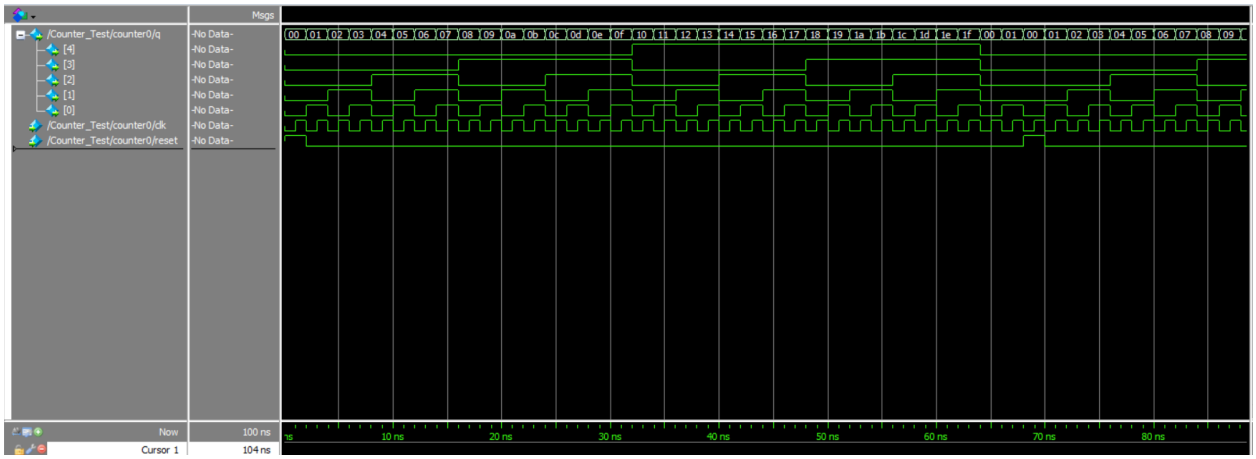
پس ما هم همین طراحی را جلو می‌بریم.
نخست کد وریلاگ همین counter را می‌نویسیم که دقیقا مانند طراحی خود استاد است:

```
module Counter (q, clk, reset);
        output[4:0] q;
        input clk, reset;
        TFF tff0(q[0], reset, clk, 1);
        TFF tff1(q[1], reset, q[0], 1);
        TFF tff2(q[2], reset, q[1], 1);
        TFF tff3(q[3], reset, q[2], 1);
        TFF tff4(q[4], reset, q[3], 1);
endmodule
```

سپس این مدار را هم طبق صورت سوال، تست می‌کنیم:

```
module Counter_Test;
        reg clk; reg reset;
        wire[4:0] q;
        Counter counter0(q, clk, reset);
        initial
                clk=0;
        always
                 #1 clk=~clk;
        initial begin
                reset = 1;
                // I assume every clock is about 1 ns
                 #2 reset = 0;
                 #66 reset = 1;
                 #2 reset = 0;
                  #30 reset = 1;
                 #2 reset = 0;
                 #20 $stop;
        end
        initial
                $monitor($time, " q = %d", q);
endmodule
```

و در نهایت تست را ران می‌کنیم:



که با توجه به 5 بیتی بودن شمارنده، به نظر درست کار می‌کند.