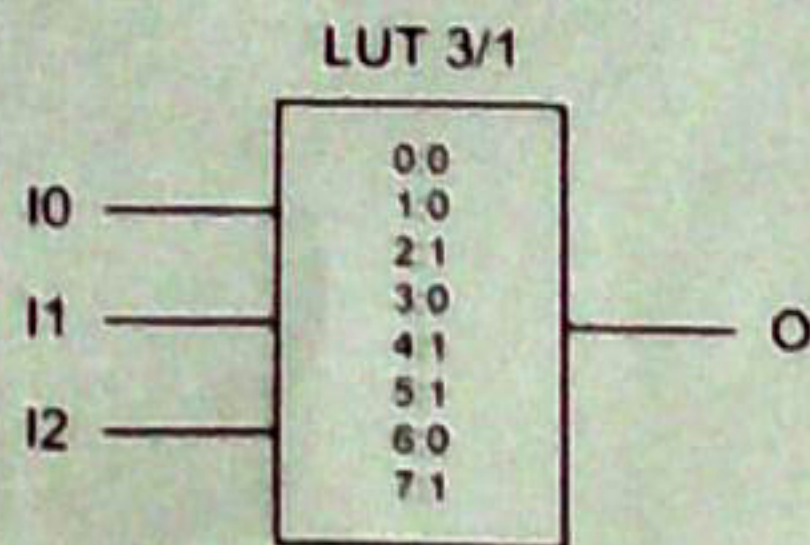
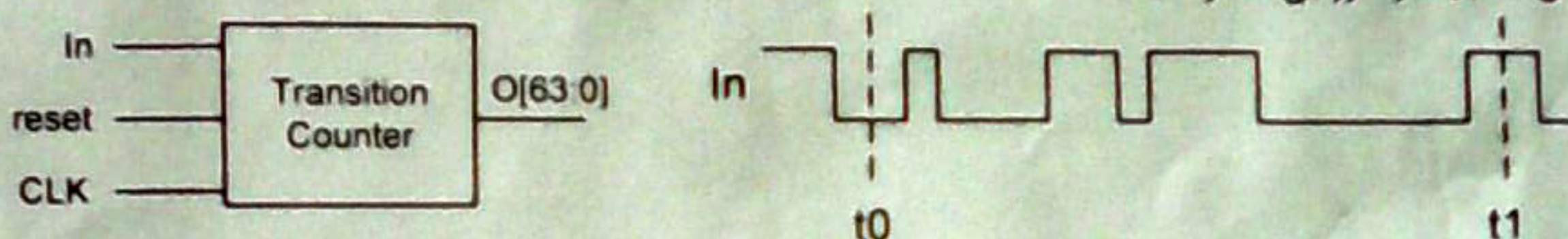


آزمون پایان ترم درس طراحی سیستم‌های دیجیتال
دانشگاه صنعتی شریف - دانشکده مهندسی کامپیوتر مدت آزمون: ۹۰ دقیقه
توجه: در این آزمون هر جا از شما توصیفی خواسته شده است فقط زبان Verilog قابل قبول است و توصیف به هر زبان دیگر غیر قابل قبول است.

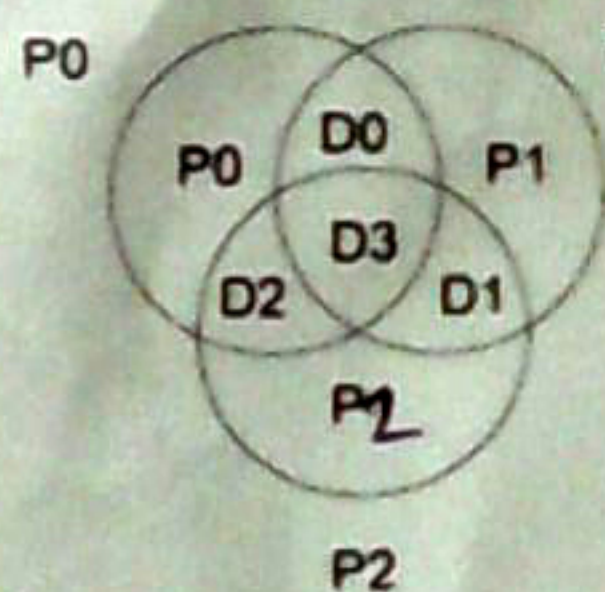


۱) فقط با استفاده از سه عدد LUT با سه ورودی (و بدون استفاده از هیچ واحد دیگر) یک Encoder اولویت دار چهار به دو PENC 4/2 را ایجاد کنید. مانند شکل مقابل (که تابع $f(I_2, I_1, I_0) = \sum m(2, 4, 5, 7)$ را پیاده‌سازی می‌کند) دقیقاً مشخص کنید که هشت بیت محتوی هر LUT باید چه باشد و نحوه اتصال LUTها به یکدیگر را نیز مشخص کنید (۴ نمره).
توجه فرمایید که اگر از واحد اضافه استفاده کنید هیچ نمره‌ای در نظر گرفته نمی‌شود.

۲) شکل زیر یک مدار شمارنده تعداد تغییرات برای ورودی In است. مثلاً با فرض اینکه برای شکل موج زیر در زمان t_0 شمارنده reset شده باشد در زمان t_1 باید در خروجی مقدار ۷ بدهد.



الف- ورودی In این مدار باید دارای چه شرطی باشد تا این مدار بتواند تعداد تغییرات را به درستی بشمارد؟ (۱ نمره)
ب- یک توصیف رفتاری قابل سنتر برای این مدار ترتیبی بنویسید. در این توصیف باید قسمتی را قرار دهید که هنگام شیماری وقتی شرط به الف رعایت نشود روی صفحه شیمار خطا چاپ کند ولی این قسمت هنگام سنتر می‌اثر نبوده و سنتر نگردد. (۲ نمره)



۳) کد همبگ یکی از کدهای مشهور کشف و تصحیح خطا است که در آن به هر چهار بیت داده سه بیت Parity اضافه می‌گردد. در کد همبگ فرض بر این است که یا خرابی رخ نمی‌دهد و یا فقط یک بیت خراب می‌شود. عملکرد این کد معمولاً با شکل مقابل نشان داده می‌شود. مثلاً اگر بیت داده D1 خراب شود این مطلب با خراب بودن Parityهای P1 و P2 معلوم می‌شود یا اگر بیت D3 خراب شود خود را با خراب بودن هر سه Parity نشان می‌دهد. تصحیح هم به این صورت است که وقتی خرابی بی‌شی مشخص شد با نقیض کردنش آن را تصحیح می‌کنند. ابتدا یک تابع بنویسید که Parity عادی را محاسبه کند سپس با استفاده از این تابع یک توصیف جریان داده (data flow) برای مدار کد کننده و همچنین تصحیح کننده کد همبگ بنویسید. (۴ نمره)

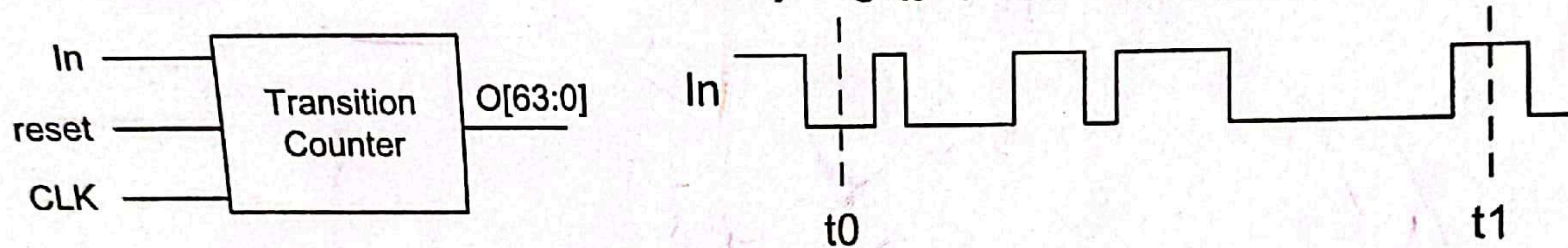
۴) یک مدل تأخیر Inertial به این صورت است که مقدار تأخیر یا مقدار پارامتر rejection فرق دارد. مثلاً اگر مقدار rejection برابر $r=3$ باشد و مقدار تأخیر برابر با $d=4$ در این صورت هر پالسی که پهنایش کمتر از 3 باشد reject می‌شود ولی اگر بیش از 3 باشد reject نشده ولی به اندازه 4 تأخیر پیدا می‌کند. اولاً ثابت کنید که مقدار تأخیر هرگز نمی‌تواند از مقدار rejection کمتر باشد. ثانیاً توصیف رفتاری یک گیت not را بنویسید که مقدار rejection آن برابر با 3 و مقدار تأخیر آن برابر با 4 باشد. (۴ نمره)

۵) این موارد را بیان کنید: سه تفاوت میان FPGA و CPLD (۱ نمره)، مفهوم Reconfigurable Computing (۱ نمره)، فرق Logical Verification با Functional Verification (۱ نمره)، مزایا و معایب FPGAهای MUX-Based نسبت به LUT-Based (۱ نمره).

موفق باشید
اجلایی

(۱) فقط با استفاده از توصیف **dataflow** و بدون استفاده از توصیف رفتاری یا ساختاری یک واحد **Register** با پهنای n بیت را با استفاده از زبان **Verilog** توصیف کنید که n در این توصیف یک پارامتر است. این ثبات باید دارای پایه‌های کنترلی **Load** و **Increment** باشد (۳ نمره).

(۲) شکل زیر یک مدار شمارنده تعداد تغییرات (**Transition Counter**) برای ورودی **In** است. مثلاً با فرض اینکه برای شکل موج زیر در زمان t_0 شمارنده **reset** شده باشد در زمان t_1 باید در خروجی مقدار ۷ بدهد.



الف- فرکانس ورودی **In** نسبت به فرکانس **CLK** باید دارای چه شرطی باشد تا این مدار بتواند تعداد تغییرات را به درستی بشمرد؟ (۱ نمره)
 ب- یک توصیف رفتاری قابل سنتز برای این مدار ترتیبی بنویسید. (۴ نمره)

(۳) با فرض اینکه فرکانس **CLK** در یک سیستم برابر 1KHz باشد یک شمارنده **BCD** که از عدد 00 تا 99 را در مبنای ۱۰ در مدت یک ثانیه می‌شمارد را با استفاده از زبان **ASM** توصیف کنید. فرض بر این است که شمارنده دارای یک ورودی **reset** سنکرون است. همچنین پس از اینکه شمارنده به مقدار 99 می‌رسد مجدداً از مقدار 00 شروع به شمارش می‌کند. یادآوری: در نمایش **BCD** در واقع اعداد در مبنای ده هستند و فقط هر رقم مبنای ده به شکل یک کد دودویی نشان داده می‌شود. مثلاً نمایش 427 برابر است با 0100 0010 0111. نیازی به ارائه شبیه‌سازی یا سنتز از توصیف **ASM** نیست. ولی توصیه می‌شود برای آنکه خودتان از مدل **ASM** خود مطمئن باشید پیش خود برای چند پالس ساعت متوالی شبیه‌سازی **cycle accurate** انجام دهید. (۷ نمره)

(۴) یک مدل تأخیر **Inertial** به این صورت است که مقدار تأخیر با مقدار پارامتر **rejection** فرق دارد. مثلاً اگر مقدار **rejection** برابر $r=3$ باشد و مقدار تأخیر برابر با $d=4$ در این صورت هر پالسی که پهنایش کمتر از 3 باشد **reject** می‌شود ولی اگر بیش از 3 باشد **reject** نشده ولی به اندازه 4 تأخیر پیدا می‌کند. اولاً ثابت کنید که مقدار تأخیر هرگز نمی‌تواند از مقدار **rejection** کمتر باشد (راهنمایی: اثبات این مورد ارتباطی با بحث مدارهای الکتریکی یا بحث‌های مشابه ندارد). ثانیاً توصیف رفتاری یک گیت **not** را بنویسید که مقدار **rejection** آن برابر با 3 و مقدار تأخیر آن برابر با 4 باشد. (۴ نمره)

(۵) مفهوم **Hardwired** کردن یک واحد حافظه را شرح دهید و برای آن یک مثال ارائه کنید (۱ نمره).

موفق باشید
 اجلالی

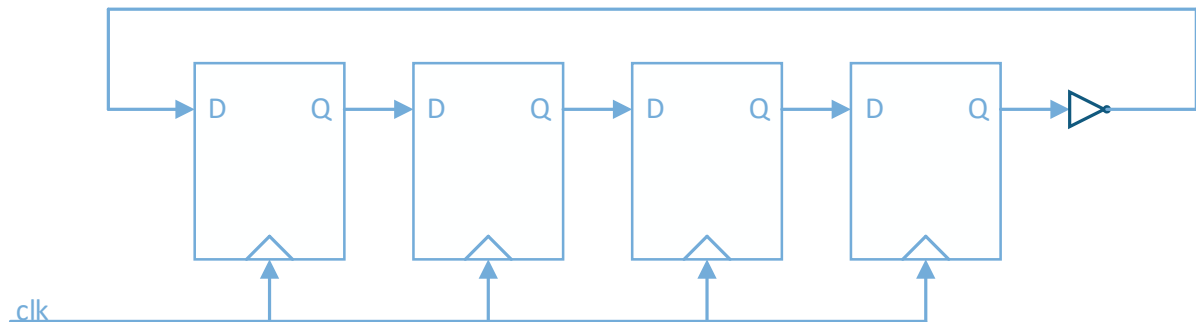
سوال اول

۱) یک شمارنده‌ی walking ring counter (Johnson) چهار بیتی را فقط با استفاده از توصیف dataflow در زبان Verilog توصیف کنید. راهنمایی: ترتیب شمارش در چنین شمارنده‌ای از چپ به راست عبارت است از:

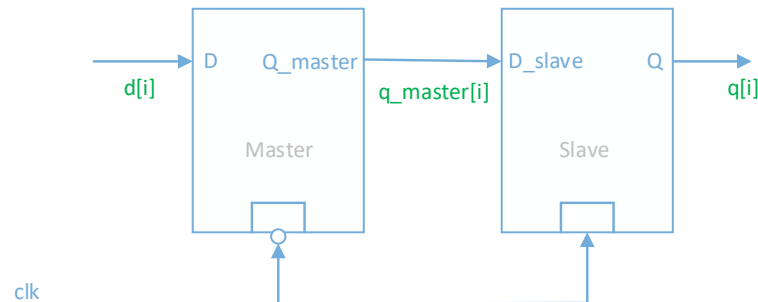
0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001, and so on.

پاسخ

مدار Johnson Counter در واقع متشکل از تعدادی Master-Slave D-FlipFlop است که به هم متصل شده‌اند. در شکل زیر مدار آن را مشاهده می‌شود:



مدار هر کدام از FlipFlop‌ها نیز به صورت زیر است. این FlipFlop‌ها متشکل از دو واحد D-Latch حساس به سطح هستند که با قرارگیری کنار هم یک FlipFlop حساس به لبه را تشکیل می‌دهند.



کد این شمارنده با فرض اینکه FlipFlop‌ها حساس به لبه ی مثبت clk باشند به صورت زیر است:

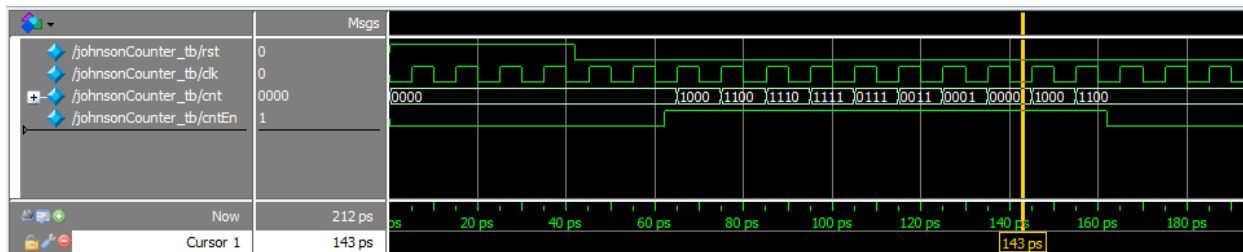
```
1 module johnsonCounter (clk, q);
2     input clk;
3     output [3:0] q;
4     wire [3:0] d, q_master;
5
6     assign d = {~q[0], q[3:1]}; //flipflops' input
7     assign q_master = clk? q_master : d; //masters
8     assign q = clk? q_master : q; //slaves
9
10    endmodule
```

برای آشنایی بیشتر دانشجویان (جزو خواسته ی سوال نمی باشد)

با توجه به اینکه تقریباً اکثر سیستم های دیجیتال ترتیبی سنکرون دارای سیگنال ریست هستند، می‌توان این سیگنال را نیز در مدار لحاظ نمود. همچنین معمولاً شمارنده ها دارای یک سیگنالی هستند که می‌تواند عملیات شمارش را فعال یا غیرفعال کند. این سیگنال را با نام countEnable به مدار اضافه نمودیم. کد کامل این مدار به صورت زیر است:

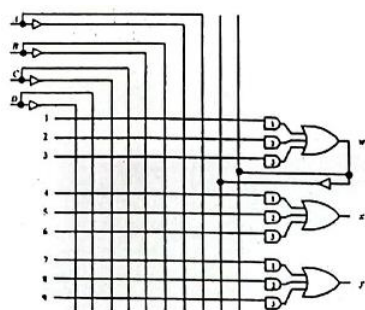
```
1 module johnsonCounterNew (clk, rst, conuntEnable, q);
2     input clk, rst, conuntEnable;
3     output [3:0] q;
4
5     wire [3:0] d, q_master;
6
7     assign d[0] = conuntEnable? q[1] : q[0]; //functionality
8     assign q_master[0] = rst? 0 : (clk? q_master[0] : d[0]); //master
9     assign q[0] = rst? 0 : (clk? q_master[0] : q[0]); //slave
10
11    assign d[1] = conuntEnable? q[2] : q[1]; //functionality
12    assign q_master[1] = rst? 0 : (clk? q_master[1] : d[1]); //master
13    assign q[1] = rst? 0 : (clk? q_master[1] : q[1]); //slave
14
15    assign d[2] = conuntEnable? q[3] : q[2]; //functionality
16    assign q_master[2] = rst? 0 : (clk? q_master[2] : d[2]); //master
17    assign q[2] = rst? 0 : (clk? q_master[2] : q[2]); //slave
18
19    assign d[3] = conuntEnable? ~q[0] : q[3]; //functionality
20    assign q_master[3] = rst? 0 : (clk? q_master[3] : d[3]); //master
21    assign q[3] = rst? 0 : (clk? q_master[3] : q[3]); //slave
22
23    endmodule
```

شکل موج و کد تست بنچ مدار فوق به این صورت است:



```
26 module johnsonCounter_tb;
27     reg clk, rst, cntEn;
28     wire [3:0] cnt;
29     johnsonCounterNew counter (clk, rst, cntEn, cnt);
30     always #5 clk = ~clk;
31
32     initial begin
33         cntEn=0; clk=0;
34         rst=1; #42
35         rst=0;
36         #20 cntEn=1;
37         #100 cntEn=0;
38         #50 rst=1; $stop;
39     end
40 endmodule
```

سوال دوم



۲) شکل مقابل ساختار کدام نوع SPLD است؟ همچنین با پیکربندی اتصالات آن (گذاشتن نقطه‌ی سیاه‌رنگ در محل اتصال قابل برنامه‌ریزی که در آنجا باید اتصال برقرار باشد) یک واحد تمام جمع‌کننده (FA) را با آن پیاده‌سازی کنید. راهنمایی: مدار FA دارای سه ورودی و دو خروجی است و لذا یکی از ورودی‌ها در اینجا بدون استفاده باقی می‌ماند.

پاسخ

این ساختار PAL است زیرا گیت‌های AND آن programmable هستند اما گیت‌های OR آن fixed هستند.

رابطه بین ورودی و خروجی‌های یک FA به صورت زیر است:

$$s = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$$

$$c_{out} = ab + ac + bc$$

اگر سیگنال کمکی w را به این صورت تعریف کنیم:

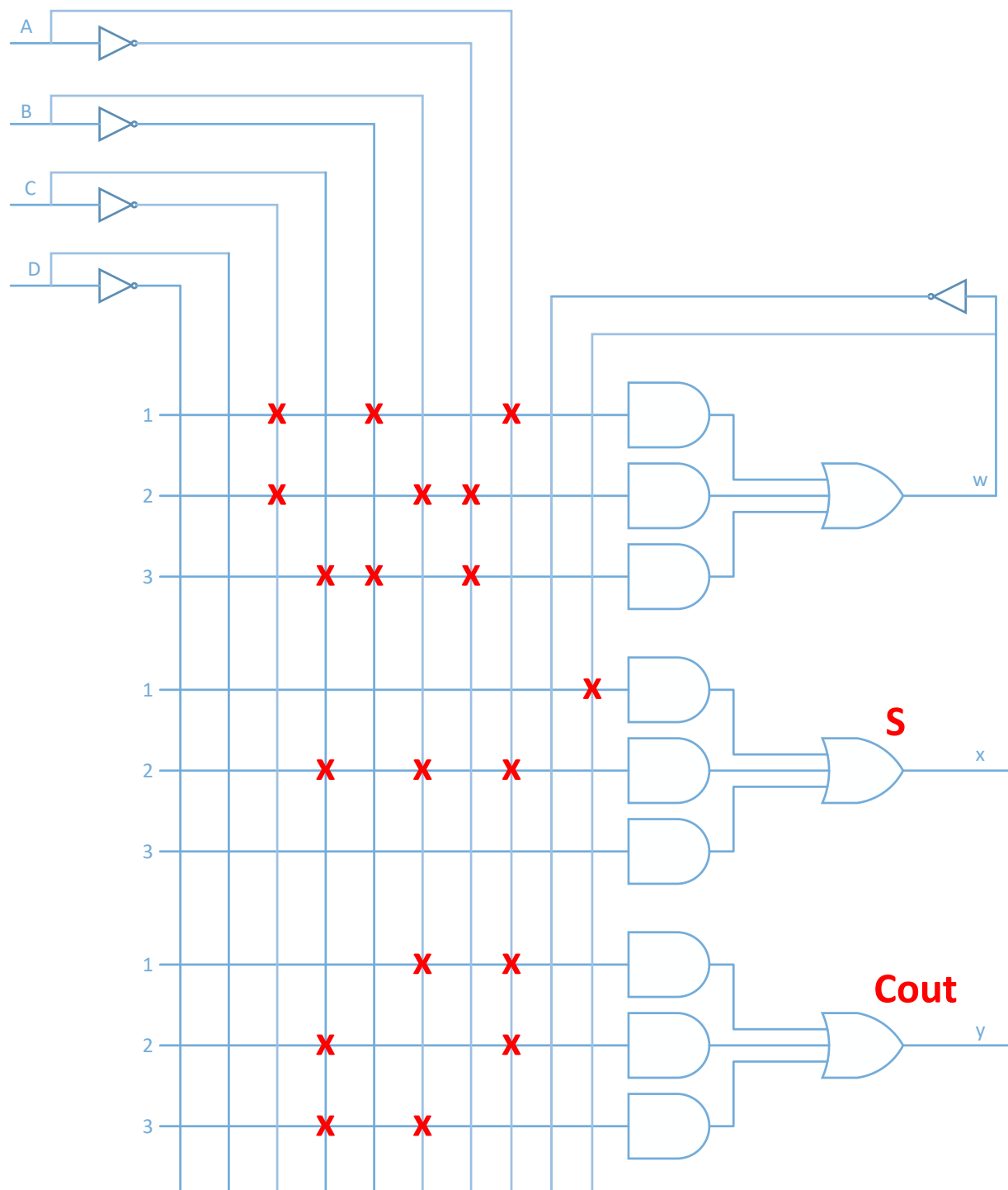
$$w = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c}$$

می‌توان سیگنال‌های x و y را بدین صورت به عنوان خروجی‌های مدار تعیین نمود:

$$x = s = w + abc$$

$$y = c_{out} = ab + ac + bc$$

بنابراین پیکربندی مدار صورت سوال به صورت زیر خواهد بود:



سوال سوم

۳) یک شمارنده دودویی-پنج‌پنجی (bi-quinary) به شکلی که در زیر آمده است عمل شمارش را انجام می‌دهد.

0100001, 0100010, 0100100, 0101000, 0110000, 1000001, 1000010, 1000100, 1001000, 1010000, and so on.

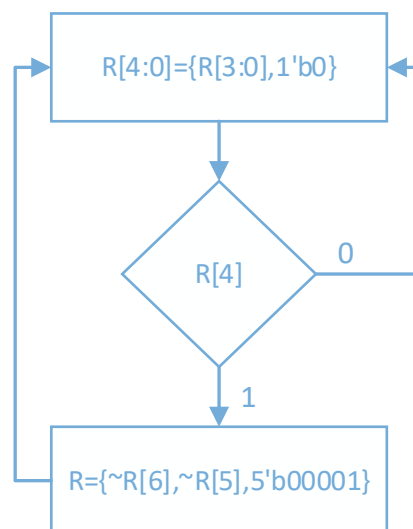
توصیف ASM این شمارنده را بنویسید به نحوی که مجاز هستید در توصیف خود حداکثر سه ASM Block داشته باشید.

پاسخ

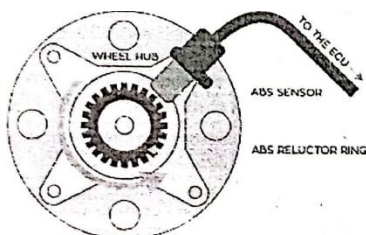
فرض بر این است که در شمارنده، یک رجیستر ۷ بیتی به اسم R داریم.

مقدار اولیه رجیستر R برابر $7'b0100001$ است.

توصیف ASM آن به شرح زیر است:



سوال چهارم



۴) سنسوری که در شکل مقابل مشاهده می‌کنید برای اندازه‌گیری سرعت دوران چرخ خودرو است. این سنسور به شکل دودویی عمل می‌کند به نحوی که هر خار در مقابل سنسور مقدار 0 ایجاد کرده و نبودن خار مقابل سنسور مقدار 1 ایجاد می‌کند. ۲۴ خار در اینجا وجود دارد (پهنای خارها با فاصله‌ی بین آن‌ها برابر است). چرخ امکان ندارد بتواند با سرعتی بالاتر از ۱۵۰ دور کامل در ثانیه بچرخد. خروجی سنسور به یک مدار دیجیتال ترتیبی می‌رود که فرکانس پالس ساعت آن حداقل 200KHz است.

این مدار دیجیتال در خروجی خود سرعت زاویه‌ای را (با واحد تعداد خار عبوری در ثانیه) ارائه می‌کند. توصیف رفتاری این مدار دیجیتال را با زبان Verilog بنویسید.

پاسخ

نحوه ی کار مدار بدین صورت است که تعداد خارهای دیده شده در یک ثانیه را می‌شمارد و به عنوان سرعت اعلام می‌کند. همچنین از یک شمارنده (clockCounter) به منظور تشخیص اینکه یک ثانیه سپری شده است، استفاده می‌شود.

- با توجه به اینکه حداکثر سرعت چرخ ۱۵۰ دور در ثانیه است و ۲۴ خار داریم، حداکثر سرعت زاویه‌ای ۳۶۰۰ خار در ثانیه است؛ بنابراین طبق رابطه ی زیر رجیستر ۱۲ بیتی برای نمایش سرعت کافی است:

$$\lceil \log_2(150 \times 24) \rceil = 12$$

- فرکانس کاری مدار به عنوان پارامتر در این ماژول دریافت می‌شود. اگر شمارنده کلاک به تعداد آن بشمارد به معنای سپری شدن یک ثانیه است.
- تعداد بیت های لازم برای شمارنده تعداد کلاک نیز به عنوان پارامتر در این ماژول دریافت می‌گردد. رابطه ی زیر بایستی بین پارامترهای عرض بیت شمارنده تعداد کلاک و فرکانس کاری مدار برقرار باشد:

$$CLOCK_COUNTER_WIDTH \geq \lceil \log_2(CLOCK_FREQUENCY) \rceil$$

کد این مدار به صورت زیر است:


```

1  module sensor (clk, rst, sensorValue, velocity);
2      parameter CLOCK_FREQUENCY=200_000;
3      parameter CLOCK_COUNTER_WIDTH=18;
4      input clk, rst, sensorValue;
5      output reg [11:0] velocity;
6
7      reg [CLOCK_COUNTER_WIDTH-1:0] clockCounter;
8      reg [11:0] spurCounter;
9      reg lastSensorValue;
10     always @(posedge clk, posedge rst) begin
11         if(rst) begin
12             velocity = 0;
13             clockCounter = 0;
14             lastSensorValue = 1;
15             spurCounter = 0;
16         end
17         else begin
18             clockCounter = clockCounter+1;
19             if(lastSensorValue==1 && sensorValue==0)
20                 spurCounter = spurCounter + 1;
21             if( clockCounter == CLOCK_FREQUENCY) begin
22                 velocity = spurCounter;
23                 spurCounter = 0;
24                 clockCounter = 0;
25             end
26             lastSensorValue = sensorValue;
27         end
28     end
29 endmodule

```

برای آشنایی بیشتر دانشجویان (جزو خواسته ی سوال نمی باشد)

به منظور شبیه سازی مدار فوق در ابتدا باید دانست که مقیاس زمانی پیش فرض Modelsim برحسب پیکوثانیه است.

فرض می شود مدار با فرکانس ۲۵۰ کیلوهرتز کار می کند.

در ابتدا فرض می شود فاصله دیدن دو خار متوالی ۱۲۰۰×۲ میکروثانیه است (سرعت ۴۱۶/۶ خار بر ثانیه). این کار به مدت ۳ ثانیه انجام شده (۱۲۰۰۰۰۰۰۰×۲۵۰۰).

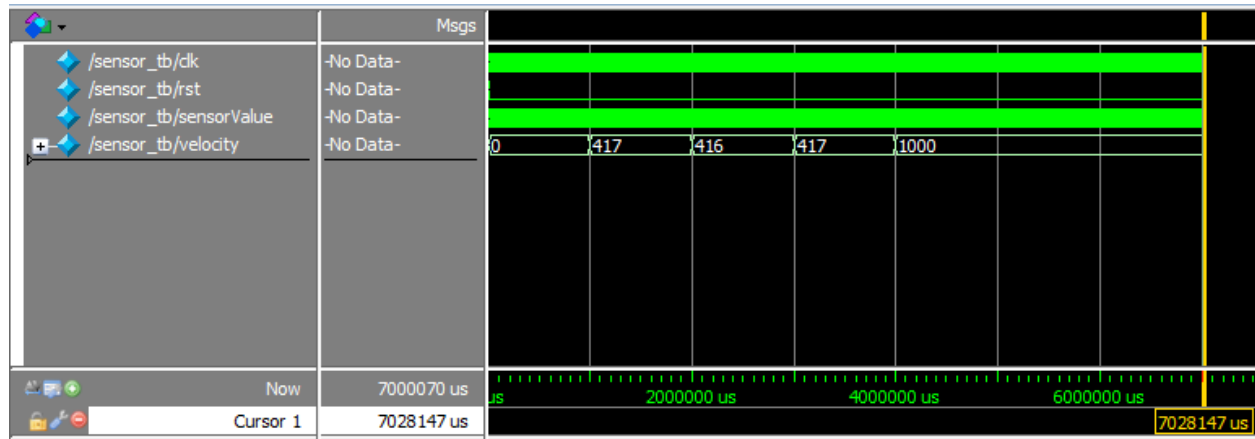
سپس به مدت ۴ ثانیه (۵۰۰۰۰۰۰۰۰×۸۰۰۰) فاصله دیدن دو خار متوالی ۵۰۰×۲ میکرو ثانیه می شود (سرعت ۱۰۰۰ خار بر ثانیه).

کد تست بنچ و شکل موج مدار فوق به صورت زیر است:


```

31 module sensor_tb;
32     reg clk, rst, sensorValue;
33     wire [11:0] velocity;
34     sensor #(.CLOCK_FREQUENCY(250000),
35             .CLOCK_COUNTER_WIDTH(18)
36             ) DUT (clk, rst, sensorValue, velocity);
37     always #2000000 clk = ~clk;
38     initial begin
39         clk = 0; rst = 0; sensorValue = 1;
40         #2000 rst=1;
41         #5000 rst=0;
42         repeat(2500) #1200000000 sensorValue = ~sensorValue;
43         repeat(8000) #500000000 sensorValue = ~sensorValue;
44         $stop;
45     end
46 endmodule

```



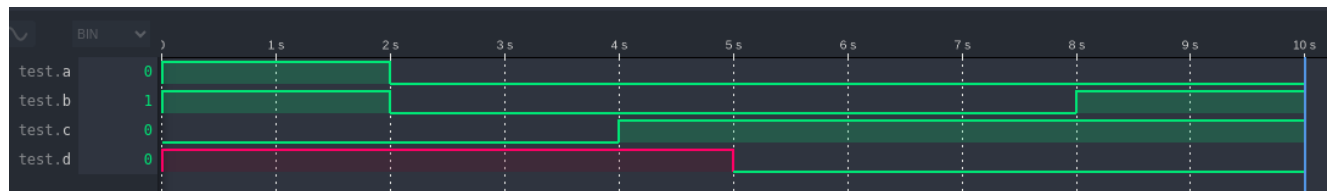
سوال پنجم

(۵) شکل موج سیگنال‌های a، b، c و d ناشی از اجرای کد مقابل را ترسیم کنید.

```
module test;
  reg a,b,c;
  wire d;
  initial
  begin
    a=1'b0;
    @(a) b=1'b1; c<=#4 ~(a^b);
  end
  initial
  begin
    #0 b=1'b0; c=1'b0; a=1'b1;
    #2 b=1'b0; a=1'b0;
    @(c) #4 b=~b; c=#2~c;
    @(d) a=#2~a;
  end
  assign #3 d=a&b;
endmodule
```

پاسخ

شکل موج خروجی کد فوق بدین صورت است:



سوال ششم

۶) به چه دلیل در FPGAها استفاده از تعداد زیاد LUTهای کوچک (fine-grained) بهتر از آن است که تعداد کمی LUTهای بزرگ (coarse-grained) داشته باشیم؟ با ذکر یک مثال از پیاده‌سازی یک منطق این برتری را نشان دهید.

پاسخ

استفاده از تعداد زیاد LUT کوچک سبب می‌شود حافظه‌ی کمتری مورد استفاده قرار بگیرد، حجم مدار و هزینه ساخت آن کمتر شده و توان مصرفی نیز کاهش یابد، اگرچه تأخیر مدار افزایش می‌یابد.

برای مثال، بیایید تابع زیر را یک‌بار با LUTهای دو ورودی و بار دیگر با LUTهای چهار ورودی پیاده‌سازی و مقایسه کنیم:

$$F(A,B,C,D) = (A \wedge B) \vee (C \wedge D)$$

- پیاده‌سازی آن با LUTهای دو ورودی به این صورت است:

$$LUT1 = A \wedge B$$

$$LUT2 = C \wedge D$$

$$LUT3 = LUT1 \vee LUT2$$

که در مجموع از سه LUT دو ورودی یعنی دوازده خانه‌ی حافظه استفاده می‌کند.

- پیاده‌سازی با LUTهای چهار ورودی:

$$LUT1 = (A \wedge B) \vee (C \wedge D)$$

که در مجموع از یک LUT چهار ورودی یعنی شانزده خانه‌ی حافظه استفاده می‌کند.

سوال هفتم

(۷) این موارد را شرح دهید: الف) تعریف واحد DP در مقابل واحد CU، ب) تعریف سنتز در مقابل سنتز منطقی، پ) تفاوت توصیف ساختاری (Structural) با توصیف هندسی (Geometrical Layout)، ت) مفهوم decomposition and mapping، ث) دلیل استفاده از آنتی فیوز به جای فیوز در FPGAها، ج) تعریف تأخیر Inertial در مقابل تأخیر Transport

پاسخ

الف) تعریف واحد DP در مقابل واحد CU

برخلاف Datapath که وظیفه‌ی انجام عملیات‌های جبری و منطقی روی داده‌ها را برعهده دارد و شامل رجیسترها، ALU و MUXها و Bus است، Control Unit سیگنال‌های کنترلی‌ای تولید می‌کند که وظیفه‌ی مدیریت عملیات‌های گوناگون Datapath را برعهده دارند. واحد CU مشخص می‌کند که در هر پالس ساعت باید چه عملیاتی (چه محاسباتی) صورت گیرد و ترتیب اعمال (محاسبات) را مشخص می‌کند ولی خود عملیات (محاسبات) توسط DP انجام می‌شود.

ب) تعریف سنتز در مقابل سنتز منطقی

سنتز به معنای تبدیل توصیف رفتاری به توصیف ساختاری است. سنتز منطقی به معنای تبدیل هر نوع مدل (رفتاری، ساختاری یا جریان داده) به توصیف ساختاری سطح گیت (gate-level netlist) است.

پ) تفاوت توصیف ساختاری با توصیف هندسی

توصیف ساختاری توصیف یک سیستم دیجیتال به کمک اجزای تشکیل‌دهنده‌ی آن و اتصال بین آن‌ها مانند moduleها، gateها و سیم‌هاست. توصیف هندسی محل قرارگیری و مسیریابی بین اجزای یک سیستم دیجیتال را بیان می‌کند. این توصیف شامل جزئیات محل قرارگیری و لایه‌های لازم برای فرآیند fabrication است.

ت) مفهوم decomposition and mapping

فرآیندی است که در آن طراحی به واحدهای کوچکتری تقسیم می‌شود تا بتوان هر واحد را توسط یک Cell از کتابخانه پیاده سازی کرد.

ث) دلیل استفاده از آنتی فیوز به جای فیوز در FPGAها

آنتی فیوزها در حالتی که وصل باشند مقاومت کمتری نسبت به فیوزها دارند، لذا توان مصرفی آنها کمتر بوده و سرعت مدار نیز در این حالت بیشتر است.

ج) تعریف تأخیر Interial در مقابل تأخیر Transport

تأخیر Inertial ناشی از شارژ و دشارژ خازن‌های موجود در مدار است. اگر طول پالس از مقدار rejection threshold بیشتر باشد، با تأخیر عبور داده می‌شود و در صورتی که طول پالس کمتر از آن میزان باشد حذف می‌گردد؛ بنابراین تغییرات کوتاه‌مدت در خروجی قابل مشاهده نیستند.

تأخیر Transport ناشی از محدودیت سرعت نور است. تأخیر انتقال سیگنال‌ها را مدل‌سازی می‌کند که در آن، هر نوع پالسی با تأخیر در خروجی نشان داده می‌شود و هیچ پالسی حذف نمی‌شود.