

Transparent latch:

A transparent latch is a storage element. It has an input, an output, and an enable or gate pin. When the enable is active, the output transparently follows the input (with some small delay). When the enable becomes inactive, the output freezes.

علت اصلی به وجود آمدن لچ های ناخواسته در مدار، وجود شرط های ناقص است.

if مثلاً

else بدون

چون شرط های ناقص باعث به وجود آمدن یک سری شرایط اضافی در مدار میشوند

که اگر بررسی نشوند باعث بالاترین سیگنال ها و وضعیت میشود

Transparent latches will appear if you write a **combinational process** or always block where **an output is not assigned under all possible input conditions**. In other words, it is possible for one of the inputs to change without affecting the output. In synthesis jargon, this is known as *incomplete assignment*.

یکی از عناصر حافظه لچ است. تفاوت آن با فلیپ فلاپ در نحوه فعال شدن آن ها و انتقال ورودی داده به خروجی است. فلیپ فلاپ ها با لبه ی سیگنال کلاک فعال میشوند و ورودی را در خروجی ظاهر میکنند و فوراً خروجی دریافت شده را حفظ میکنند و تا لبه ی بعدی گلاک آنرا تغییر نمیدهند

اما لچ حساس به سطح سیگنال است و مادامی که سیگنال خاص فعال باشد، مقدار ورودی در خروجی منعکس میشود برخی اوقات در توصیف ما یک سری لچ های ناخواسته تولید میشود مانند وقتی که یک شرط بررسی نشود و حالت های اضافی بی سر و سامان باقی بمانند. (وجود شرط های ناقص در توصیف)

```
process (sel, sel_2, sel_3, a,b)
begin
  if sel = "1" then
    f<=a;

    if sel_2 = "1" then
      g <= not a ;
    else
      g <= not b ;
    if sel_3 = "1" then
      g <= a xor b ; ==> شرط ناقص
    end if ;
  end if ; //
  else
    if sel_2 = '1' then
      g <= a and b ;
    else
      if sel_3 = '1' then
        g <= a nand b; ==> شرط ناقص
      end if ;
    end if;
    f <= b ; ==> بدون شرط
  end if ;
end process;
```

کد تغییر یافته:

```
process (sel, sel_2, sel_3, a,b)
begin
if sel = '1' then
f<= a ;
if sel_2 = '1' then
g<=not a ;
else
g <=not b ;
if sel_3 = '1' then
g <= a xor b ;
else
g<='U';
end if ;
end if ;
else
if sel_2 = '1' then
g <= a and b ;
else
if sel_3 = '1' then
g <= a nand b;
else
g<='x';
end if ;
end if;
f <= b ;
end if ;
end process;
```

Subject: _____
 Year: _____ Month: _____ Date: _____

~~Assignment 1~~

(1)

a

Process(binp)

begin

goutp(N-1) \leftarrow binp(N-1) after 10ns;

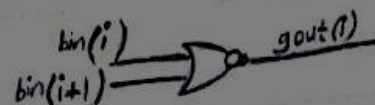
for i in N-2 down to 0 loop

goutp(i) \leftarrow binp(i+1) xor binp(i) after 10ns;

end loop;

end process.

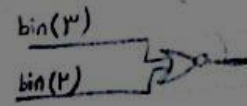
لا تأخیر
در هر level



binp(N-1) goutp(N-1)



!- N-2



(b)

port(A,B,C : in

Q1,Q2 : out

signal V,R

begin

process(V,C)

begin

if (V='1') then

Q2 \leftarrow C;

end if;

end process

R \leftarrow B xor C;

process(A)

begin

if (rising-edge(A)) then

Q1 \leftarrow C;

V \leftarrow R;

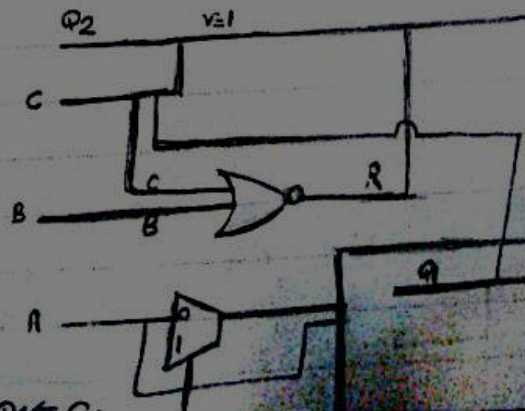
end if;
 end process

V=1 \rightarrow Q2 \leftarrow C;

R \leftarrow B xor C

if (A: 0 \rightarrow 1) \rightarrow Q1 \leftarrow C;

V \leftarrow R;



Subject, _____
 Year, _____ Month, _____ Date, _____ ()

(1)

states : s0, s1, s2

~~state~~, next-state, state signal, state, (state), next-state, (state), .C

begin

sync-proc

process (clk)

begin

if (rising-edge (clk)) then

if (reset = '1') then

state <= s0;

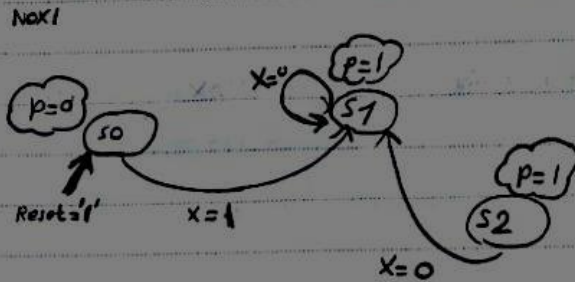
else

state <= next-state;

end if;

end if;

end process;



output-decode

process (state)

begin

case (state) is

when s0 => parity <= '0'; p = 0

when s1 => parity <= '1'; p = 1

when s2 => parity <= '1'; p = 1

when others => parity <= '0'; p = 0

end case;

end process;

next-state-decode: process (state, x)

begin

next-state <= s0;



دانشگاه

 Subject: _____
 Year: _____ Month: _____ Date: _____

case (state) is
 when $s_0 \Rightarrow$ if ($x='1'$) then
 next-state $\leftarrow s_1$;
 end if

when $s_1 \Rightarrow$ if ($x='0'$) then
 next-state $\leftarrow s_1$;
 end if;

when $s_2 \Rightarrow$ if ($x='0'$) then
 next-state $\leftarrow s_1$;
 end if;

when others \Rightarrow next-state $\leftarrow s_0$;

end case;

end process;

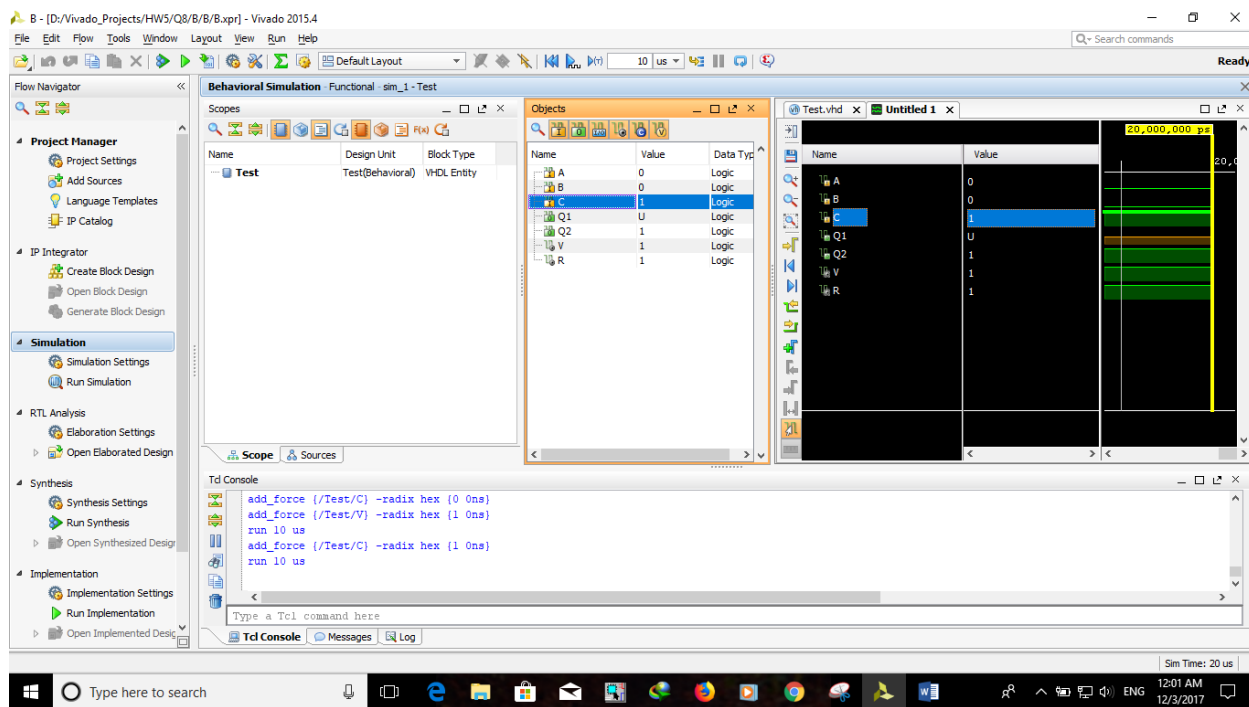
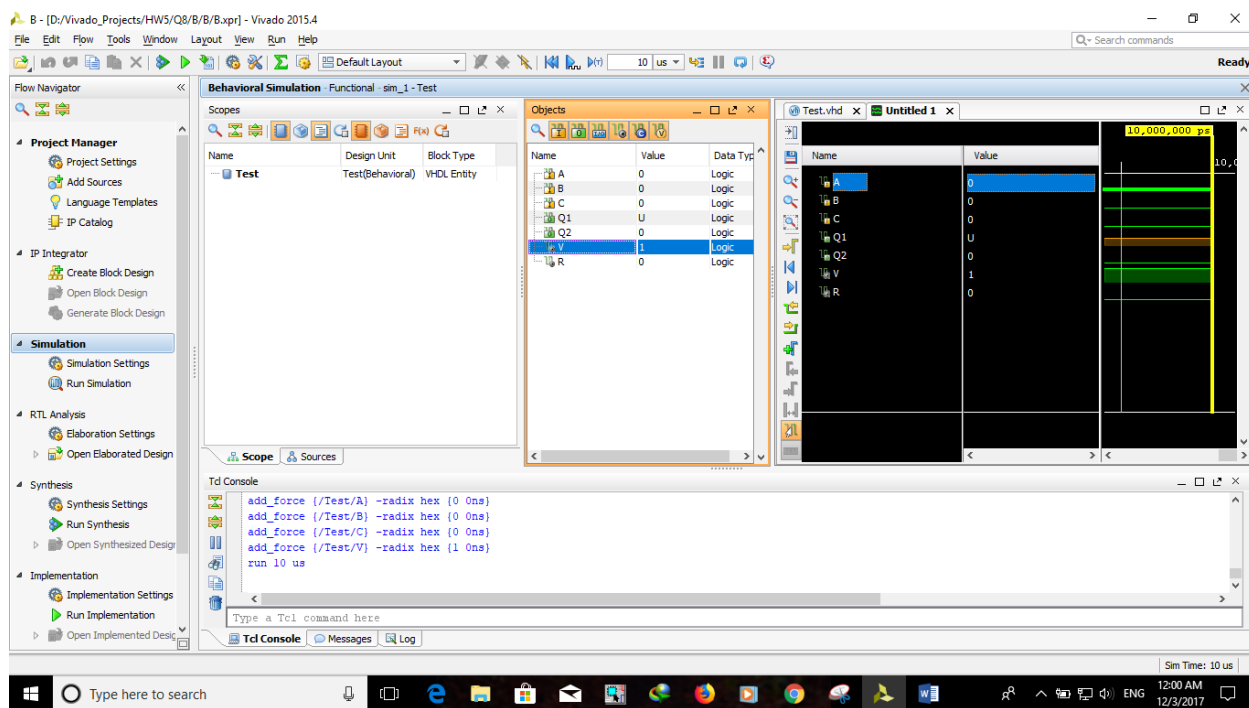
 ل های اخت
 شده اند

 طراحی
 جالش
 ج را ا
 سطوح
 انواع
 مثال

اصلی

بر ا حل

 در کد
 جلو



B - [D:/Vivado_Projects/HW5/Q8/B/B/B.xpr] - Vivado 2015.4

File Edit Flow Tools Window Layout View Run Help

Search commands

Ready

Flow Navigator

- Project Manager
 - Project Settings
 - Add Sources
 - Language Templates
 - IP Catalog
- IP Integrator
 - Create Block Design
 - Open Block Design
 - Generate Block Design
- Simulation
 - Simulation Settings
 - Run Simulation
- RTL Analysis
 - Elaboration Settings
 - Open Elaborated Design
- Synthesis
 - Synthesis Settings
 - Run Synthesis
 - Open Synthesized Design
- Implementation
 - Implementation Settings
 - Run Implementation
 - Open Implemented Design

Behavioral Simulation - Functional - sim_1 - Test

Scopes

Name	Design Unit	Block Type
Test	Test(Behavioral)	VHDL Entity

Objects

Name	Value	Data Type
A	0	Logic
B	0	Logic
C	0	Logic
Q1	U	Logic
Q2	0	Logic
V	1	Logic
R	0	Logic

Test.vhd

Name	Value
A	0
B	0
C	0
Q1	U
Q2	0
V	1
R	0

Tcl Console

```
run 10 us
add_force [/Test/A] -radix hex [1 0ns]
run 10 us
add_force [/Test/A] -radix hex [0 0ns]
run 10 us
```

Type a Tcl command here

Tcl Console Messages Log

Sim Time: 100 us

12:03 AM 12/3/2017

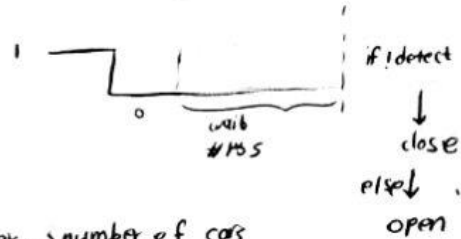
Subject: _____
 Year: _____ Month: _____ Date: _____

(سؤال 9)

state machine for Q9

 $\begin{cases} a \\ b \end{cases}$

detect-a
 detect-b



clock: 10MHz

counter \Rightarrow number of cars

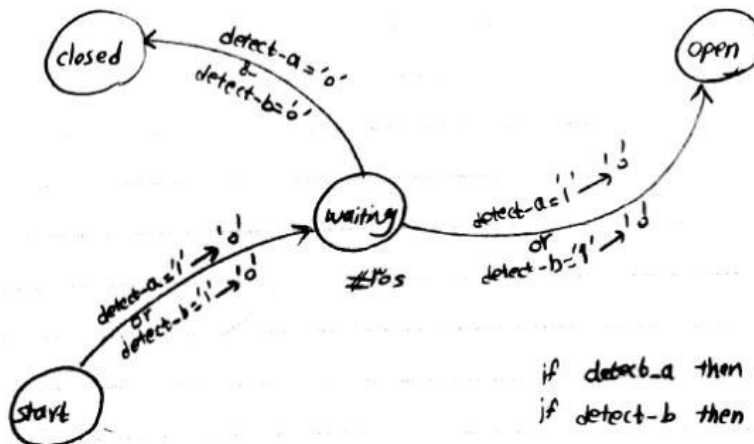
acknowledge for enter or exit

if (detect-a then detect-b) \rightarrow counter++

ack: a signal

if (detect-b then detect-a) \rightarrow counter--

states: open, close, waiting, start



if detect-a then detect-b counter++

if detect-b then detect-a counter--

signal: exit;

نتایج شبیه سازی مطلوب نبود و به درستی سنتز نشد