تا انتها کردم ✓
۹۸/۱۲/۱۸

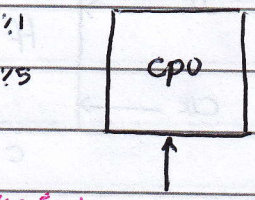
در بلوک fork-join ، همه دستورات همزمان اجرای شوند.

در طراحی از begin-end و ⇐ استفاده کنید.

جلسه دوازدهم:

موضوع تحقیق:

پروتکل هایی که در شبکه استفاده می شود TCP/IP است. غالب packet هایی که در شبکه ارسال می کنیم IP هستند. سرعت شبکه ها روز به روز در حال افزایش است. ما سخت افزاری می خواهیم که بتواند packet ها را با سرعت بسیار بالا گیرد و بفرستد. در حال حاضر این کارها توسط CPU انجام می شود.



علت می شود کامپیوتر TCP IP گیرد و بفرستد.

با افزایش سرعت، توان همه رفته توسط CPU نیز افزایش می یابد. اگر من از یک سخت افزار برای پیاده سازی TCP/IP استفاده کنم هم توان همه رفته توسط CPU افزایش می یابد و هم به سرعتهای بالاتری می توانیم دست یابیم.

پیاده سازی سخت افزاری TCP/IP

TCP/IP Hardware Implementation

وزارت بازرگانی شرکت بازرگانی دولتی ایران

packet ها چه شکلند ؟ node های مختلف شبکه چه شکلند ؟

TCP/IP stack چی میشود ؟

حال آنها را به صورت نرم افزاری اجرا کنید .

می خواهیم کد های TCP/IP را به صورت نرم افزاری پیاده کنیم .

تمرین ها :

FIR Filter

Average

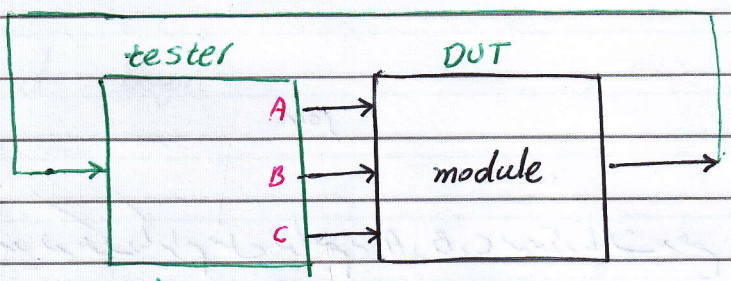
Counting

SRAM

توکیک

دور سوتور

سعی کنید ماژول ها را جدا جدا طراحی کنید و بعد یک ماژول کلی .



با استفاده از stimulator شکل موج دلخواه را انتخاب می کنیم

بزرگ شدن ماژول ها و افزایش تعداد حالات دوردی دیگر نمی توان در شبیه سازی از این تکنیک استفاده کرد

وزارت بازرگانی شرکت بازرگانی دولتی ایران

در چنین شرایطی یک ماژول دیگر طراحی می کنیم که شکل موجهای مورد نظر ما را ایجاد کند.

کل این مجموعه (دو ماژول) را test bench می گویند.

test bench (test fixture)

DUT باید synthesizable باشد یعنی باید قابل پیاده سازی روی fpga و تبدیل به گیت باشد.
ولی tester لازم نیست ← باشد به همین دلیل در tester ما امکانات بیشتری خواهیم داشت که به کار بندازد.

(1) همواره ما مقدار دهی اولیه مقیب فلاپها را با reset انجام می دادیم. ولی در اینجا می توان از initial استفاده کرد.

کدی که initial اول اول در برنامه اجرا می شود.

```
initial                    initial
    begin                      fork
    a=0;
end                        join
```

فرض کنید در یک دیاگرام متل بخواهیم به A و B و C سه مقدار ثابت بدهیم.

```
module Tester (A,B,C,out);
output A,B,C;
input out;
reg A,B,C;
initial
```

وزارت بازرگانی شرکت بازرگانی دولتی ایران

```
begin
A=1; B=0; C=1;
end
endmodule
```

❗ هرجا بخواهیم در بلوک begin-end یا fork-join استفاده کنیم متغیرهای داخل بلوک باید reg باشند.

❗ از initial هرگز در بلوک اصلی استفاده نمی‌کنیم زیرا قابلیت تبدیل به گیت را ندارد.

فرض کنید در کدفوق در زمانی معادل 10ns بعد مقادیر را عوض کنیم.

اگر بخواهیم در verilog تأخیر داشته باشیم:

3 میزان تأخیر

در مثال قبل

```
initial begin
A=1;
B=0;
C=1;
#100
A=0;
B=1;
C=0;
end
```

وزارت بازرگانی شرکت بازرگانی دولتی ایران
وزارت بازرگانی شرکت بازرگانی دولتی ایران

وزارت بازرگانی شرکت بازرگانی دولتی ایران

از آن جایی که نمی توان در بلوک اصلی استفاده کرد زیرا هیچ تعادلی ندارد.

Exp:

```
initial   begin
A = 0;
#50
A=1;
#100.01
A = 0;
#20
A = 1;        ← بعد از 170ns اجرا می شود.
end
```

باید واحد آخرین زمانی را در ابتدای برنامه مشخص کنیم.

قبل از بلوک module ها می آید.

```
`timescale  1ns/1ps      دقت
`timescale  100ns/1ns
`time scale  1ps/1fs
```

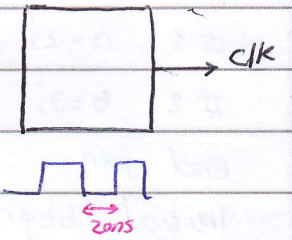
حرکت سازنده time scale default در کد برای AHDL مرسوم است.

اگر در بلوک testbench ، قبل از فراخوانی module ها، time scale را مشخص

کنیم برای تمامی module ها تعریف شود.

فرض کنید بخواهیم یک clk تولید کنیم.

```
`timescale 1ns/1ps
module clkGen (clk);
output clk;
reg clk;
initial begin
clk = 0;
forever #20 clk = ~clk;
end
endmodule
```



تا بی‌نهایت اجرا می‌شود.

روش دوم:

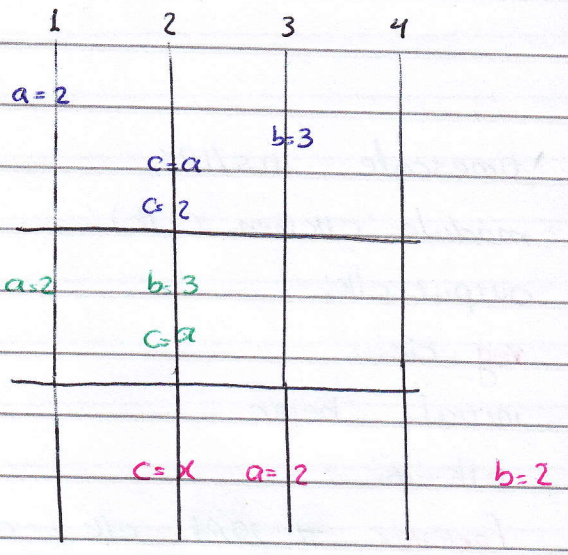
```
initial begin
clk = 0;
end
always begin
#20 clk = ~clk
end
endmodule
```

در always عباراتی که بعد از آن می‌آید را تا بی‌نهایت انجام می‌دهد.

وزارت بازرگانی شرکت بازرگانی دولتی ایران

اگر تأخیر نبود زمان شبیه سازی چطوری رفت.

```
module A;
reg a, b, c;
initial begin fork
#1  a = 2;
#2  b = 3;
end join
initial begin
#2  c = a;
end
end module
```



الآن عدد گذاری کن و یک نانوثانیه بعد تخصیص بده

```
intra_assignment delays
a = #1 2;
```

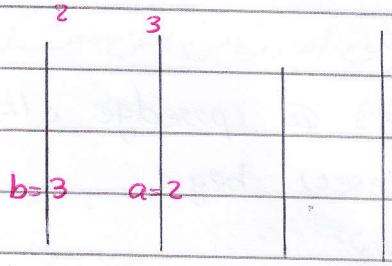
اجرای این دستور 1ns طول می‌کشد.

```
* a = #3 2;  ⟶  c = x
#3  a = 2;
a = #3 2;
```

در لحظه 3، عدد گذاری می‌شود.

وزارت بازرگانی شرکت بازرگانی دولتی ایران

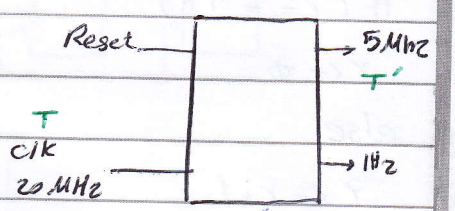
```
initial begin
    a <= #3 2;
    #2 b <= 3;
end
```

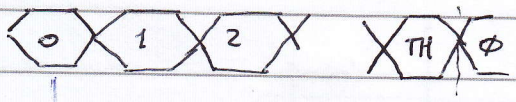


حل تمارین سری اول:

```
reg [31:0] r;        counter
reg a;
```

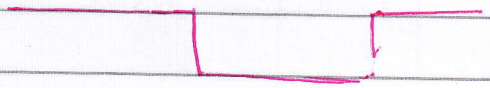
-5





x x x x → a یک سیکل

باید 20 را تقسیم کنیم



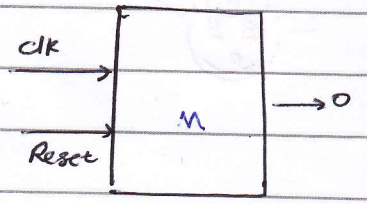
چون از صفر می‌شمارد

$\frac{20}{5} = 4$ → { دو سیکل روشن و خاموش پایین آمد

→ $\frac{4}{2} = 2$ → TH = 2-1 = 1

$\frac{20 \times 10^6}{1} = 20 \times 10^6$ → $\frac{20 \times 10^6}{2} = 10 \times 10^6 - 1 = 9999999$

```
module M (O, Reset, clk);
output O;
input clk, Reset;
Parameter TH = 1;
reg [31:0] r;
```



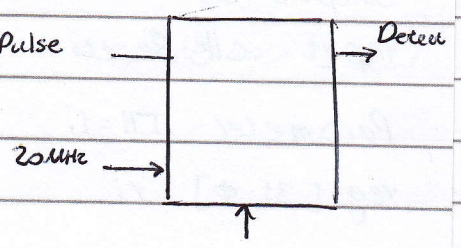
وزارت بازرگانی شرکت بازرگانی دولتی ایران

```
reg a;
always @ (posedge clk)
if (Reset) begin
r <= 0;
a <= 0;
end
else begin
if (r == TH)
  r <= 0
else
  r <= r+1;
if ( r == TH)
  a <= ~a;
end
assign O = a;
end module
```

در ادامه کافی است یک کد دیگر بنویسیم که در هر بار تاچ را با دریافت پالس کمکی برای TH تغییری دهیم

TH می تواند دوروی باشد. ! (به دست آوردن فرکانس های مختلف)

۶. فوق العاده مهم

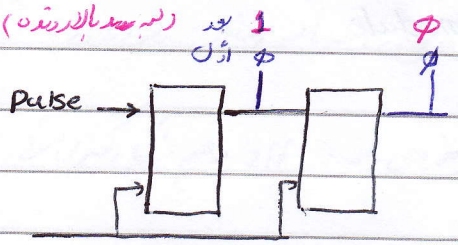


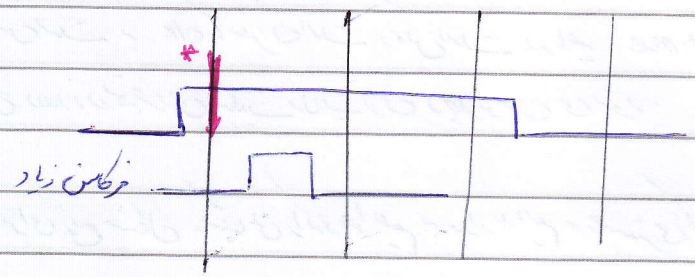
وزارت بازرگانی شرکت بازرگانی دولتی ایران

max فرکانس باید از فرکانس پالس ساعت کمتر باشد زیرا باید پالس‌های پالس ساعت این اتفاق را تغییم.

ایده طراحی:

دو reg می‌گذاریم، clk را به آن متصل می‌کنیم و در ورودی پالس را به آنها می‌دهیم






```
module D (Detect, clk, Reset, Pulse);
input Pulse, clk, Reset;
output Detect;
reg r1, r2;
always @(posedge clk)
if (Reset) begin   r1 <= 0;
                   r2 <= 0;
        end
else begin
```

وزارت بازرگانی شرکت بازرگانی دولتی ایران

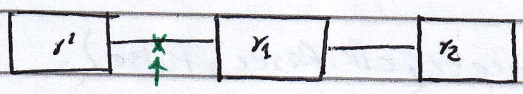
```
r1 <= Pulse;
r2 <= r1;
end
assign Detect =
    r1 & (~r2);
endmodule
```

بعضی وقتها تأخیر برای ما مهم نیست

امکان ایجاد اشکال:

زیرا پالس نسبت به clk آشکارساز لبه دیگر لبه در ناحیه set up time فلیپ فلاپ اول رخ دهد و در نتیجه خروجی فلیپ فلاپ اول دچار نوسان می شود.

معمولاً برای رفع مشکل، سیگنالی را که می خواهیم به مدار بدهیم، رجیستری می کنیم.

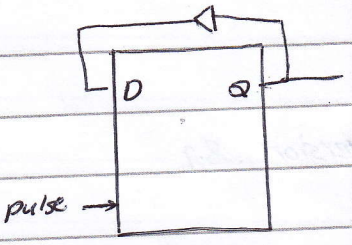


تداخل بازه باید دوری ثبت شده باشد.

مدت زمان رسیدن به حالت باید از کمتر از یک سیکل ساعت الک.

وزارت بازرگانی شرکت بازرگانی دولتی ایران

چرا از خود پالس استفاده نکردیم ؟



هیچ هزینه‌ای با لبه پالس ساعت وجود ندارد!

اگر هزینه‌ای باشد T_{ff} در هیم مشکل‌گشایی شود

وزارت بازرگانی شرکت بازرگانی دولتی ایران

به نام خدا

جلسه سیزدهم :

پروسه نصب AHDL :

version : 8.2

main installation : click & next

سپس نصب AHDL برای crack :

license : node locked

ibfs: 32.dll ⟶ copy : system 32 : paste

run :

license.dat ⟶ Dat (AHDL محل نصب کردن)

در این مرحله باید AHDL را بیابید اگر نیابید :

environment variable

my computer ⟶ properties ⟶ advanced ⟶ continue

new ⟶ LM-LICENSE-FILE : Address : مقدار

مقدار : آدرسی که license قرار دارد .

پروسه نصب synplify :

license : node locked

synpatch ⟶ copy ⟶ bin (synplify محل نصب)

روی synpatch کلیک می کنیم تا اجرا شود .

وزارت بازرگانی شرکت بازرگانی دولتی ایران

باید ID کارت شبکه را داخل license بگذاریم . (hose Id را عوض می کنیم)

license

run : cmd : ip configure

find & replace : save

expiration date : 2010 → 2012 : save

environment var در سطح سیستم و user اعم فرق می کند چون یک user داریم

نرم افزار VMware وای توان برای کارت شبکه استفاده کرد زیرا کارت شبکه مجازی تولید می کند .

در ادامه ترم سنتیز کارهای تکنیکی را خواهیم آموخت .

آشنایی با نرم افزار model sim :

مزایا : قوی ، efficient برای پروژه های بزرگتر

نرم افزار استاندارد شرکت xilinx کرا ISE کاملاً سازگار است .

model sim win 32 6.5

mg keygen 2002

file save as

وزارت بازرگانی شرکت بازرگانی دولتی ایران

کار با model sim:

در ورودی - خروجی ها را هنگام تعریف اندازه‌های ماژول تعریف می‌کنیم.

```
`define WIDTH 8
output reg [`WIDTH-1:0] out;
```

البته می‌توان پورت‌ها را جداگانه تعریف کرد.

اسم فایل testbench را همان اسم module می‌گذاریم.

یک ماژول برای تست کردن این module می‌نویسیم که در ماژول جدید جای ورودی‌ها و خروجی‌ها کاملاً عوض می‌شود.

بهتر است یک فایل جداگانه بسازیم و در آن ثابت‌ها را فراخوانی می‌کنیم و بعد

```
`include "defines.v"
```

محل قرارگیری ثابت‌ها

time scale در اول برنامه فراموش نشود

```
`timescale 1ns/1ps

module Counter-tester (input [`WIDTH-1:0] out,
  output reg clk
             Reset
             Load
             [`WIDTH-1:0] Load value,
             up down,
```

وزارت بازرگانی شرکت بازرگانی دولتی ایران

```
// clk
initial begin
  Clk = 0;
  forever  # 10  Clk = ~Clk;
end

// Reset
initial begin
  ResetL = 0;
  # 50  ResetL = 1;
end

// Load
initial begin
  Load = 0;
  Load value = 0;
#1000
  Load = 1;
  Load Value = $random();
#30
  Load = 0;
#1000
  Load = 1;
  Load value = $random ($time);
#30
  Load = 0;
end
```

وزارت بازرگانی شرکت بازرگانی دولتی ایران

وزارت بازرگانی شرکت بازرگانی دولتی ایران

// updown

initial begin

updown = 1;

#5000

updown = 0;

end

endmodule

برای حل کردن در مانول،

`time scale 1ns/1ps

`include "defines.v"

module top;

wire [`WIDTH-1:0] out, load value;

wire clk, resetL, Load, updown;

counter_tester Tester1ns (.out(out),

.clk(clk),

.ResetL( )

Add و define می‌کنیم چون ترتیب آنرا compile کنیم.

compile → compile properties →

file که در آن constant را تعریف کرده‌ایم در همان جایی که پوشه model sim را تعریف کرده‌ایم کپی می‌کنیم تا در error نگیرد.

```
> vlog hdl /*
```

همه فایلهای داخل folder hdl را کامپایل می‌کند.

```
> vsim
```

برای شبیه‌سازی استفاده می‌شود.

start simulation → unmark enable optimization

```
>run 100ns
```

شبیه‌سازی run می‌شود.

با رکه کنترل در محیط شبیه‌سازی می‌توان zoom کرد.

```
>quit -sim
```

پایان شبیه‌سازی را end simulation

اگر بخواهیم جایی که در حین شبیه‌سازی توکیدی شود را ذخیره کنیم:

وزارت بازرگانی شرکت بازرگانی دولتی ایران

```
integer fileH;
//make a new file to write data to
Initial begin
fileH = $fopen("data.", "w"
```

پس از تعریف و تعداد دهی update

```
always @ (posedge clk)
$fwrite(fileH, "value : %d \n", out);
```

out را در هر لبه بالارونده ساعت جدید در فایل .

```
>vlog hdl/*
>vsim top
>add wave [sim:/top/*]
```

در همان folder ای که modelsim را ذخیره کرده ایم یک folder با نام data ساخته شده است .

```
vlog hdl/*
vsim -novopt top
add wave [sim:/top/*]
run 5200 ns
```

این ها را با نام run.do ذخیره می کنیم و بعد با اجرای دستور زیر تمامی دستورات فوق اجرا می شود

```
do run.do
```

وزارت بازرگانی شرکت بازرگانی دولتی ایران

اگر بخواهیم مقادیر value را از فایل خوانده شود در یک فایل در هر خط یک عدد می نویسیم و آنرا با پسوند dat ذخیره می کنیم.

```
initial begin
integer readH;
read = $ fopen ("input.", "r")
$fscanf (readH, "%d \n", Load value)
```

جلسه چهاردهم:

موضوع تحقیق:

وقتی یک آزمایش شیمیایی می خواهیم انجام دهیم کاری که معمولاً انجام می دهیم این است که در یک فضای آزمایشگاهی که scale آن با توجه به پدیده ای که می خواهد آزمایش را انجام می دهیم.

عامل پدیده را می توان در VLSI انجام داد.

MEM: Micro Electrical Mechanical Device

چیپ VLSI محتوی تعدادی گیت هستند که هم مقل هستند.

چگونه می توان در scale، VLSI، 1) مدوان ایجاد کرد

2) گیرهای قابل کنترل

3) شتاب لحظه ای

موضوع MEM موارد فوق است.

وزارت بازرگانی شرکت بازرگانی دولتی ایران

موضوع Lab on a chip ، انجام پروسه های بزرگ در فضای کوچک است.

حجم خون مورد نیاز بسیار کاهش می یابد و هم چنین می توان محاسبات بسیار زیادی روی آن انجام داد.

حجم پردازش روی چیپ های VLSI و مکینالهای کنترلی می تواند بالا باشد.

Lab on a chip & FPGA

استفاده های FPGA در Lab on a chip

در مرحله بعد ، طراحی کنترل lab on a chip می پردازیم.

موضوع این جلسه :

CORE :

خیلی وقتها بخشی از مازول مورد نظر را توسط دیگران نوشته شده است. بعضی وقتها این مازولها رایگان و بعضی وقتها هزینه بر است.

بعضی وقتها یک تکه از Design خالی است زیرا هر بخش از مازول را یکبار طراحی می کند و ما فقط بخشی از Design را به عنوان یکی از طراحان شبیه سازی می کنیم.

طراحی up_down :

از بالا شروع به طراحی می کنیم تا به پایین می رسیم.

وزارت بازرگانی شرکت بازرگانی دولتی ایران

پالس‌های A و B سیگنال‌های حساب دهم فرکانس

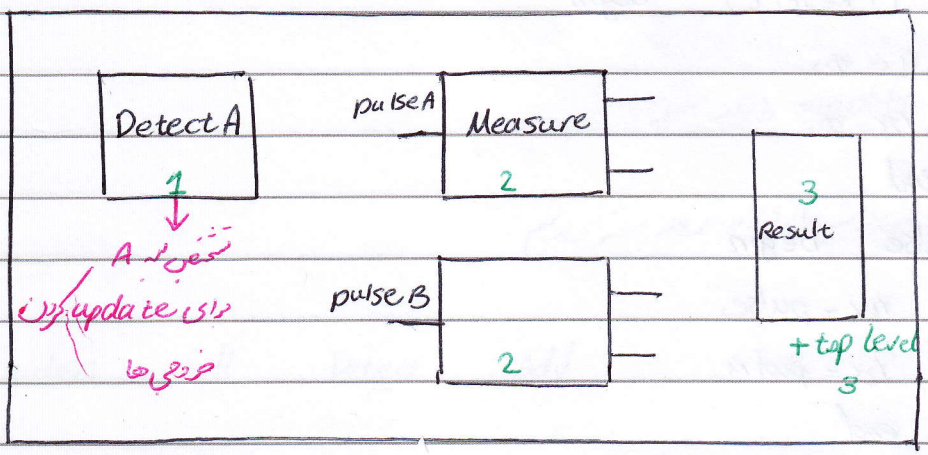
خروجی Duty 1 باشد اگر Duty Cycle پالس A، 50٪ باشد.

خروجی Compare ← 1 اگر از B بیشتر باشد.

2 اگر A از B بیشتر بود

3 اگر مساوی بودند

فرض کنید 3 developer داریم و طرح را به 3 قسمت تقسیم می‌کنیم.



تشخیص لبه A
برای update کردن
خروجی‌ها

folder‌های زیر را می‌سازیم.

dev_1 & dev_2 & dev_3 & total

ایجاد پوشه‌های زیر

hdl

sim

syn

وزارت بازرگانی شرکت بازرگانی دولتی ایران

```
module Detect (clk, ResetL, Pulse, Edge);
input clk;
input ResetL;
input Pulse;
output Edge;

reg r1, r2;

always @(posedge clk or negedge ResetL)
if (!ResetL) begin
  r1 <= 0;
  r2 <= 0;
end
else begin
  r1 <= pulse;
  r2 <= pod r1;
  end
assign Edge = (r1 && !r2) ? 1 : 0;
end module
```

حال model-sim را اجرا می‌کنیم.

File → change directory → dev1 → sim

وزارت بازرگانی شرکت بازرگانی دولتی ایران

برای ریختن حاصل فایل‌های کامپایل شده یک library می‌سازیم.

الم‌لخوان‌ورای Ubrary

vlib Detect

vlog -work Detect ../hdl/*.v

↓
یک folder بالاتر (چون ما داریم توی sim هستیم)

vsim Detect.Detect

quit -sim

ابزار سنتز synplify

پس از شبیه‌سازی به مرحله سنتز می‌رسیم.

Add file → dev1 → hdl → Detect → Add

→ Run

RTL View → ترکیب ff و گیت‌ها (بدون توجه به fpga)

Technology view → map به fpga بسته به fpga انتخابی

لازم است dev کدهای وریلاگ را بدهیم و فقط نتیجه syn

syn نیوند .edf

وزارت بازرگانی شرکت بازرگانی دولتی ایران

Disabe I/O insertion;

چین نرم‌افزاری های fpga قرار می‌گیرد

Save as Measure.v ⟶ hdl

```
module Measure (clk, ResetL, Pulse, up, down, Edge );
input clk;
input ResetL;
output [31:0] up, down;
input Edge
reg [31:0] up, down;

always @ (posedge clk or negedge ResetL)
 if (! ResetL) begin
    up <= 0 ;
    down <= 0;
end
else begin
    if(Edge) begin
    up <= 0;
    down <=0;
  end
else begin
  if (pulse)
  up <= up+1;
```

⟶ اندازه‌گیری صحیح

وزارت بازرگانی شرکت بازرگانی دولتی ایران

new → project file ⟶ syn

Add files ⟶ Measure . . v (hdl)

ctrl + n ⟶ new file

```
module Result( clk, ResetL, Edge, Duty, Compare, WAU , WAD,
               WBU , WBD);

input [31:0] WAU , WBA;


// duty
always @( posedge clk or negedge ResetL)

if (!ResetL)
Duty <= 0;
else begin
if (Edge)  begin
    if (WAU == WBD)
        Duty <= 1;
        else
        Duty <= 0;
    end
end
```

وزارت بازرگانی شرکت بازرگانی دولتی ایران

// compare

always @ (posedge clk or

else begin

if (edge) begin

if (WAU > WBU)

compare <= 1;

else if (WAU < WBU)

compare <= 2;

else

compare <=3;

end

وزارت بازرگانی شرکت بازرگانی دولتی ایران

اصلاحیه مرحله: اتصال 3 ماژول

module top(clk, ResetL, Pulse A, Pulse B, Duty, Compare);

input clk, ResetL, Pulse_A, Pulse B;

output Duty;

output [1:0] compare;