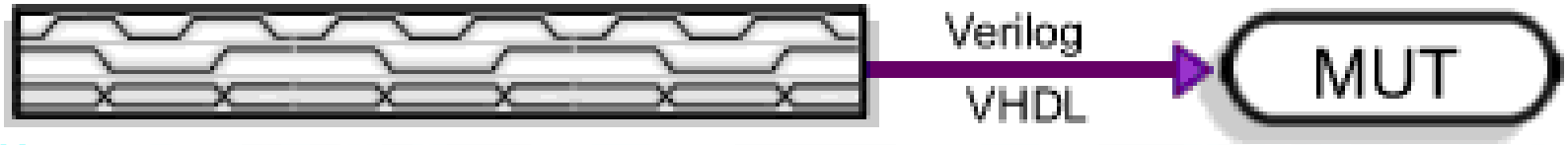
طراحی کامپیوتری سیستم‌های دیجیتال

جلسه سوم



بررسی درستی

- برای شبیه سازی یک مدار در زبان های توصیف سخت افزاری باید ورودی هایی به مدار اعمال نمود تا بر اساس آن خروجی تولید کند.
- نوشتن ماژول تحریک این امکان را برای ما فراهم می کند.
- ماژول تحریک حداقل شامل یک مولد سیگنال است و نمونه دهی ماژولی است که درستی آن را می خواهیم مورد تحقیق قرار دهیم.
- این توصیف سخت افزاری، که سیگنال ورودی (محرک) برای یک طراحی فراهم می کند، یک "میز تست" نامیده می شود.



میز تست Test Bench

ماژول تحریک درگاه ورودی و خروجی ندارد چون با محیط اش تعاملی ندارد.

ماژول تحریک می تواند به صورت یک برنامه جداگانه و مستقل نوشته شود یا به صورت یک ماژول در برنامه اصلی وجود داشته باشد.

گزاره های بلوکی میز تست از بالا به پایین و از چپ به راست به ترتیب شبیه سازی می شوند.

این گزاره ها در هر بلوک در میان کلمات کلیدی begin وend محاصره شده اند.

میز تست

میز تست به صورت یک برنامه مستقل

```
// Test bench for Simple_Circuit_prop_delay

module  t_Simple_Circuit_prop_delay;
  wire     D, E;
  reg      A, B, C;

Simple_Circuit_prop_delay M1 (A, B, C, D, E); // Instance name required

initial
  begin
    A = 1'b0; B = 1'b0; C = 1'b0;
    #100 A = 1'b1; B = 1'b1; C = 1'b1;
  end

  initial #200 $finish;
endmodule
```

میز تست

ورودی ها در میز تست با کلمه کلیدی reg و خروجی ها با کلمه کلیدی wire اعلام می شود.

initial معمولا برای شروع توصیف شکل موج در میز تست به کار می رود.

کلمه کلیدی initial همچنین برای آغاز شبیه سازی به کار می رود.

ممکن است یک میز تست حاوی بیش از یک گزاره initial باشد.

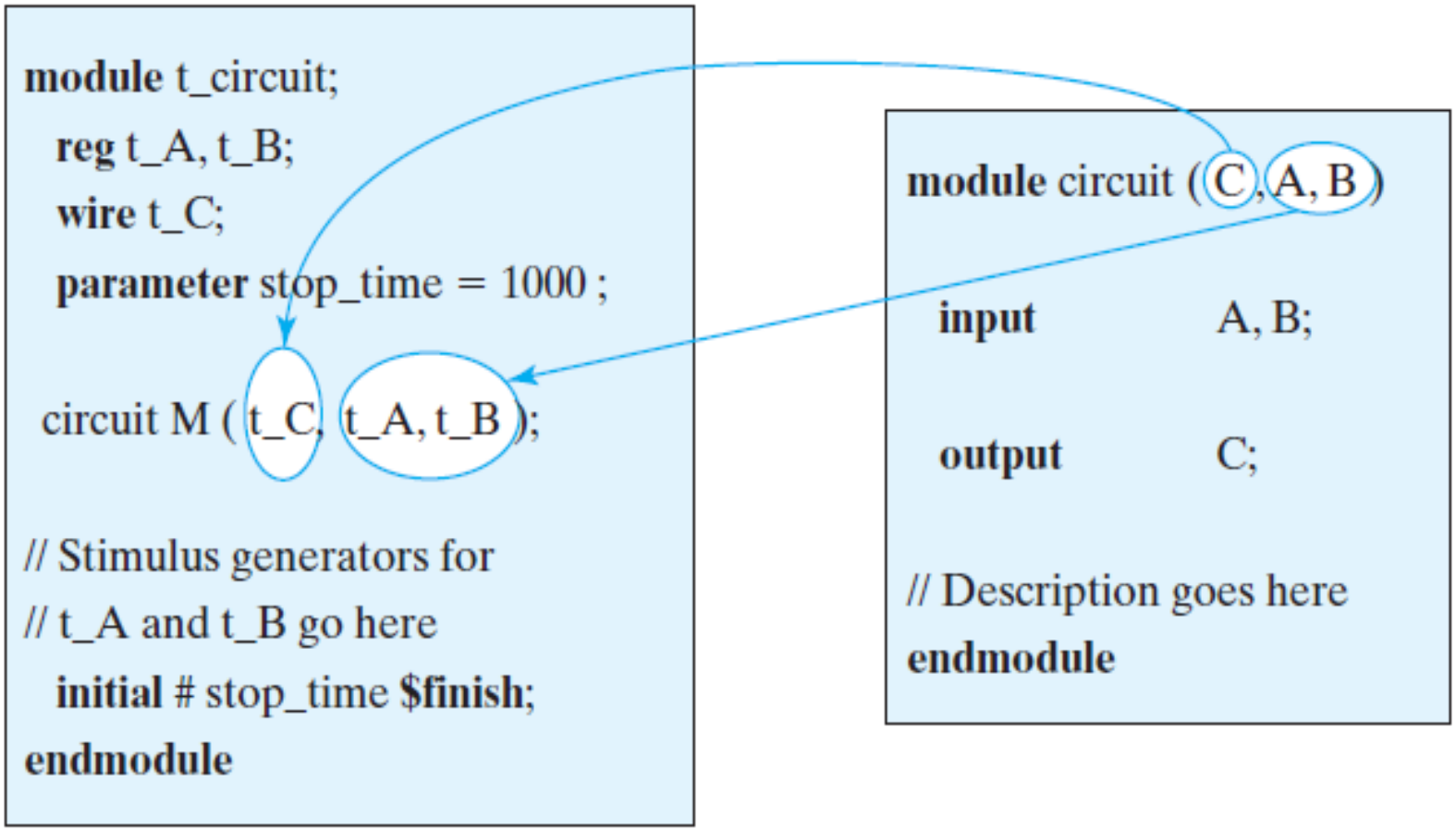
بلوک initial اگر تنها شامل یک دستور باشد، نیاز به begin/end ندارد.

$finish خاتمه شبیه سازی را مشخص می کند.

$stop توقف شبیه سازی را باعث می شود.

میز تست

ارتباط بین ماژول تحریک و طراحی



نمایش اعداد در وری لاگ

< ارقام معنا دار>< مبنای عدد نویسی >’<اندازه بر حسب بیت>

```
12’b100010101001
12’o4251
12’h8A9
12’d2217
```

بدون مشخص کردن اندازه

```
’b100010110
’o426
’h116
278
```

اعداد منفی

-5 ⟶ -4’b101 ⟶ 1011

به فرم مکمل دو ذخیره می شود

کمک به خوانایی اعداد با استفاده از (_)

12’b100010101001⟶12’b1000_1010_1001

تکرار با always

کلمه کلیدی always یک رفتار چرخه ای با عث می شود.

رفتار always به صورت نا محدود اجرا و باز اجرا می شود تا زمانی که شبیه سازی متوقف شود.

یک ماژول می تواند دارای بیش از یک گزینه always با شد.

بلوک always اگر تنها شامل یک دستور باشد، نیاز به begin/end ندارد.

```
initial
 begin
  clock = 1'b0;
 end

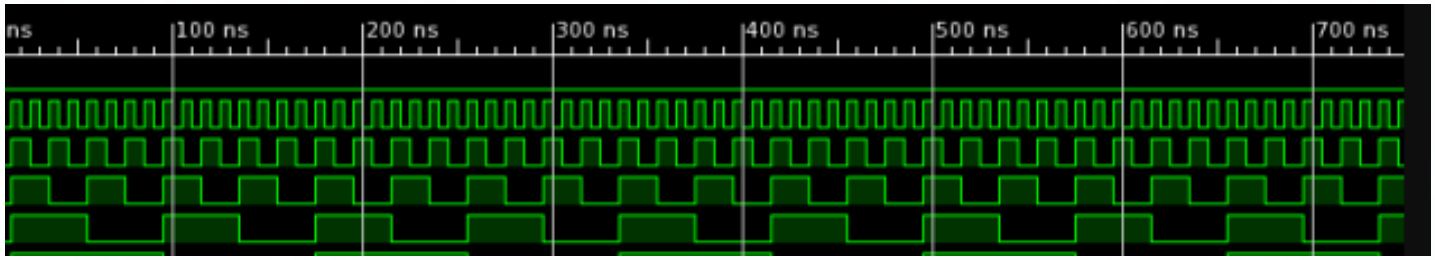
initial #200 $finish;
always #10 clock = ~clock;
```

دستورات تکرار

دستور Repeat

این دستور تعداد تکرار را مشخص می کند.

```
initial
  begin
    clock = 1'b0;
    repeat (30)
      #10 clock = ~clock;
  end
```



دستور Forever

این دستور تا پایان زمان شبیه سازی دستورات مورد نظر را اجرا می کند.

```
initial #200 $finish;
initial begin t_clock = 0; forever #5 t_clock = ~t_clock; end
```

وظایف سیستمی System Tasks

$time

- مدت زمان شبیه سازی را نمایش می دهد با واحد زمانی مشخص شده توسط timescale.

$monitor

- نمایش متغییرها در حین شبیه سازی هر زمان که یک متغییر تغییر می کند.

$display

- نمایش موارد خواسته شده همراه با رفتن به خط بعد.

$write

- مانند display ولی بدون رفتن به خط بعد.

نمایش خروجی ها

ساختار نحوی برای $display, $monitor, $write به صورت زیر است:

Task-name(format specification , argument list)

```
$display ("%d %b %b", C, A, B);
```

مثال

```
$display ("time = %0d A = %b", $time, A    );
```

خروجی

```
time = 3 A = 10
```

نمایش در خروجی

دستور $monitor

مثال

```
$monitor ("A = %b E = %b F = %b time = %0d", A, E, F, $time);
initial #500 $finish;
```

Simulation log:

```
A = xxxx  E = x F = x time =  0
A = 0000 E = x F = 0 time =  10
A = 0001 E = 0 F = 0 time =  20
A = 0010 E = 0 F = 0 time =  30
A = 0011  E = 0 F = 0 time =  40
A = 0100 E = 0 F = 0 time =  50
A = 0101 E = 1 F = 0 time =  60
A = 0110 E = 1 F = 0 time =  70
A = 0111  E = 1 F = 0 time =  80
A = 1000 E = 1 F = 0 time =  90
A = 1001 E = 0 F = 0 time =  100
A = 1010 E = 0 F = 0 time =  110
A = 1011  E = 0 F = 0 time =  120
A = 1100 E = 0 F = 0 time =  130
A = 1101 E = 1 F = 0 time =  140
A = 1101 E = 1 F = 1 time =  150
A = 0000 E = 1 F = 0 time =  160
```

نمایش در خروجی

دستور $monitor

در هر لحظه فقط یک دستور monitor می تواند فعال باشد.

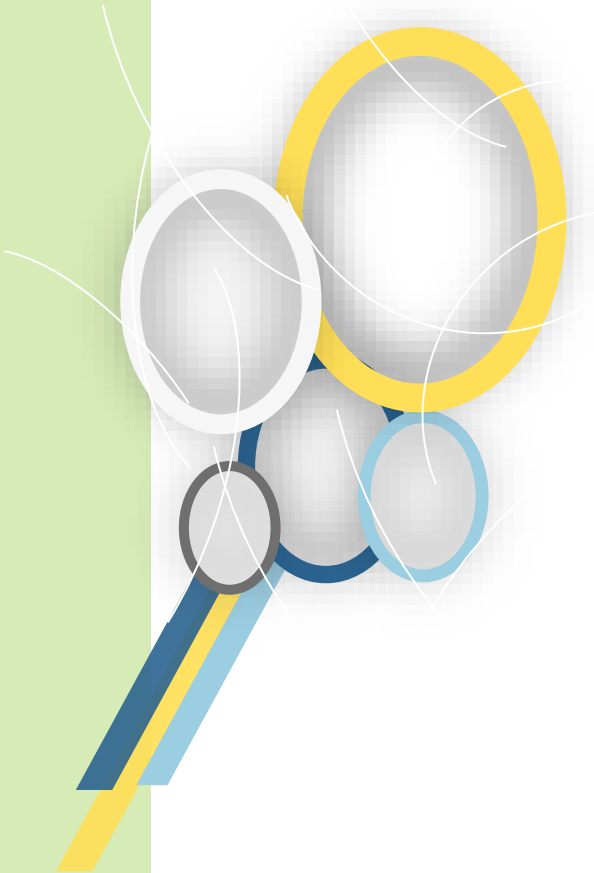
برای داشتن چند monitoring در یک برنامه می توان یکی از آنها را با توجه به نیاز فعال کرد و دستورات مانیتورینگ دیگر را غیرفعال نمود.

- غیر فعال کردن

$monitoroff

- فعال کردن

$monitoron



خسته نباشید