

ENTITY generic-adder IS

Generic (data-width : integer := 16);

PORT (a,b: IN std_logic_vector(data-width-1 DOWNTO 0);

cin: IN std_logic;

sum: OUT std_logic_vector(data-width-1 DOWNTO 0);

cout: OUT std_logic);

END generic-adder;

ARCHITECTURE concurrent OF adder IS

SIGNAL c : std_logic_vector (0 TO data_width); // بیت آخر می نویسد

BEGIN

L1: FOR i IN 0 TO data_width-1 GENERATE

sum(i) <= a(i) XOR b(i) XOR c(i);

c(i+1) <= (a(i) AND b(i)) OR (a(i) AND c(i)) OR (b(i) AND c(i));

END GENERATE L1;

c(0) <= cin;

cout <= c(data_width);

END concurrent;

نوشتن testbench :

در صورتی که در همان فایل testbench می نویسیم باید کد آن به این entity باشد.

LIBRARY ieee;

USE ieee.std_logic_1164.ALL;

ENTITY generic_adder_tb IS END generic_adder_tb;

ARCHITECTURE concurrent_tb OF adder_tb IS

CONSTANT data_w : integer := 32;

COMPONENT generic_adder IS ...

END COMPONENT;

SIGNAL a_t : std_logic_vector (data_w-1 DOWNTO 0);

...

BEGIN

-- U1: generic_adder GENERIC MAP (32) PORT MAP (a_t, b_t, cin_t, sum_t, cout_t);

-- به جای 32 می توانیم 32 را به 16 یا 8 یا 4 یا 2 یا 1 تغییر دهیم

راه بهتر برای عدم رعایت ترتیب --

U1: generic_adder

GENERIC_MAP (data-width => data_w)

PORT MAP (

b => b_t,

a => a_t,

cin => cin_t,

sum => sum_t,

cout => cout_t);

a_t <= X"00000000", X"00000001" AFTER 15ns;

b_t <= X"00000000", X"0000000B" AFTER 10ns;

cin_t <= '0', '1' AFTER 39ns;

END concurrent;

مثال پردازش یک تصویر

فرض کنیم یک عکس 19×19 حاوی ۳۵۹ خانه داریم.

برای نوشتن یک فیلتر به صورت موازی می توانیم

به صورت زیر عمل کنیم.

L1: FOR i IN 0 TO 15 GENERATE

L2: FOR j IN 0 TO 15 GENERATE

$x(i,j) = (x(i-1,j-1) + \dots + x(i+1,j+1)) / 9$; 1 -- میانگین یک مربع 9 خانگی

END GENERATE L2;

END GENERATE L1;

البته برای سطرهای اول و آخر، سطر اول و آخر می توان FOR های جدا نوشت یا یک سری دستور و دستور به اطراف عکس اضافه کرد و آن را 18×18 کرد.

تمرین یک 32-bit Array Multiplier with GENERATE statement

به همراه test-bench، script (که generic باشد)

سعی شود یاد و تا حلقه باشد و سطر اول و آخر جدا باشد.

ادامی درس

چا، سنه (۱۸، ۱، ۹۹) (کونیزا، فصل ۵ تا ۱)

There is no sequential hardware

However, a sequential subset is designed in HDLs to simplify the hardware design process.

مثال:

ENTITY MUX4x1 IS

PORT (a, b, c, d : IN std_logic

sel : IN std_logic_vector (1 DOWNTO 0);

z : OUT std_logic);

END MUX4x1;

ARCHITECTURE sequential OF MUX4x1 IS

BEGIN

نکات:

PROCESS (a, b, c, d, sel) → sensitivity list

- در مدار Combinational، خروجی به همی ورودی ها

BEGIN

حساس است.

IF sel = "00" THEN

- process های مختلف با هم موازی هستند ولی که

z <= a;

input:

درون آن ها، تریتی است.

a, b, c, d, sel

ELSEIF sel = "01" THEN

z <= b;

- بین input و output حساسیت فرق است.

...

END IF;

END PROCESS;

END sequential;

مثال:

ENTITY adder16 IS

PORT (a, b : IN std_logic_vector (15 DOWNTO 0);

cin : IN std_logic;

cout : OUT std_logic;

sum : OUT std_logic_vector (15 DOWNTO 0));

END adder16;

ARCHITECTURE sequential OF adder16 IS

SIGNAL c : std_logic_vector (0 TO 16);

BEGIN

PROCESS (a, b, cin)

BEGIN

FOR i in 0 TO 15 LOOP

sum(i) <= $a(i) \oplus b(i) \oplus c(i)$

c(i+1) <= c(i) AND sum(i)

END LOOP;

END PROCESS;

c(0) <= cin;

cout <= c(16);

END sequential;

در زبان VHDL شرطی در نظر گرفته شده است که رویه اجرای داخل process ها موازی می شود اگر چه ظاهراً
سریال است.

فرض می کنیم اجرای کل process، منطبق با زمان می برد.
یک process زمانی اجرای شود که sensitivity list تغییر کند.

نکته:

وقتی مقدارها assign می شود، یادداشت می شود و تنها در انتهای کد، مقدارها در یک لحظه اعمال می شود. به عبارتی در طول اجرای process، مقدارها به سیگنال اعمال نمی شود و در انتها، یکبار به اعمال می شود.

```
PROCESS (a, b, c, d, sel)
```

```
BEGIN
```

```
a <= '1';
```

```
b <= '1';
```

```
c <= '0';
```

```
b <= '0';
```

```
END PROCESS; → a=1, b=0, c=0
```

سوال: این محدودیت کجا تأثیری نگذاشت؟

```
PROCESS (a, b, c, d)
```

```
BEGIN
```

```
a <= '1';
```

```
b <= '1';
```

```
c <= a;
```

```
b <= '0';
```

```
END PROCESS; → a=1, b=0, c=مقدار قبلی
```

سوال: آیا AFTER در process مشکل دارد؟ خیر. صرفاً در انتهای آن که مقادیر نهایی مشخص می شوند، این مقدارها در نظر گرفته می شوند. (اعمال آخرین AFTER، assignment)

نکته: در سخت افزار، combinational و sequential داریم اما این با مفاهیم کنونی sequential و concurrent در VHDL فرق دارد.

ENTITY dcd3x8 IS

: مثال

PORT (a : IN std_logic_vector(2 DOWNTO 0);

z0 : OUT std_logic;

z7 : OUT std_logic);

END ENTITY dcd3x8;

ARCHITECTURE seq of dcd3x8 IS

BEGIN

PROCESS (a)

BEGIN

IF a = "000" THEN

z0 <= '1'; z1 <= '0'; ..., z7 <= '0';

ELSEIF a = "001" THEN

...

ENDIF;

END PROCESS;

END seq;

: مثال

ENTITY dff IS

PORT (d, clk, rst : IN std_logic;

q, nq : OUT std_logic);

END dff;

ARCHITECTUR sequential OF dff IS BEGIN

PROCESS (clk, rst) BEGIN

IF (rst = '1') THEN

q <= '0'; nq <= '1';

ELSEIF clk'EVENT AND clk = '1' THEN

q <= d; nq <= ~~NOT~~ q;

END IF;

NOT d;

END PROCESS;

END sequential;

ورودی های یک process، سیگنال هایی هستند که در داخل process، در شرط ها چک می شوند یا سمت راست
اشتباه ها استفاده می شوند.

اگر sensitivity با input برابر نباشد، سخت افزار، combin. نیست.

اگر اشتباه nq را به خارج به صورت زیر ببریم:

$$nq \leq \text{NOT } q$$

خطا دارد چون q خروجی است و نمی توان مقدار آن را خواند می توانیم آن را با فرقی تعریف کنیم