

عملگرها

Operators

عملگرها در VHDL

افزایش اولویت

xnor	xor	nor	nand	or	and	عملگرهای منطقی
>=	>	<=	<	/=	=	عملگرهای رابطه ای
ror	rol	sra	sla	srl	ssl	عملگرهای شیفت
			&	-	+	عملگرهای جمعی
				-	+	علامت
		rem	mod	/	*	عملگرهای ضربی
			not	abs	**	عملگرهای متفرقه

عملگرهای منطقی

• عملگرهای منطقی:

- روی اشیا از نوع bit، bit- vector، boolean
- برای std-logic، std-logic-vector، signed و unsigned گرانبار شده‌اند

```
entity OPR_EX_E is
    port( X,Y : in bit;
          Z   : out bit);
end entity OPR_EX_E;

architecture OPR_EX_A of OPR_EX_E is
    signal DATA_BUS1, DATA_BUS2, DATA_BUS3 : std_logic_vector(7
downto 0);
begin
    Z <= X and Y;
    DATA_BUS3 <= DATA_BUS1 xor DATA_BUS2;
end architecture OPR_EX_A;
```

عملگرهای رابطه‌ای

• عملگرهای رابطه‌ای:

□ برای کلیه انواع داده‌ی استاندارد

□ نتیجه: boolean

```
variable A : bit_vector(2 downto 0);  
variable B : bit_vector(0 to 2);  
if(A > B) then ... ;
```

```
variable A : bit_vector(2 downto 0);  
variable B : bit_vector(0 to 2);  
if(A(2) > B(0) or  
   ((A(2) = B(0)) and (A(1) > B(1))) or  
   ((A(2) = B(0)) and (A(1) = B(1)) and (A(0) > B(2)))) then ... ;
```

عملگرهای رابطه‌ای

• عملگرهای رابطه‌ای:

- ❑ مقایسه بردارهای با اندازه متفاوت ← مقایسه از سمت چپ
- ❑ با صفر پر کنید.

```
signal X : bit_vector(7 downto 0);  
signal Y : bit_vector(3 downto 0);  
:  
if(X >= "0000" & Y) generate ... ;
```

- ❑ مقایسه داده‌های شمارشی: سمت چپ کوچک‌تر

```
type boolean is (false,true);  
type std_ulogic is ('U', ..., '0', '1', ...);
```

عملگرهای شیفت

• عملگرهای شیفت:

□ منطقی

- عنصر سمت چپ: '0' (در حالت کلی: عنصر سمت چپ نوع داده شمارشی)

□ ریاضی

- عنصر سمت چپ: عنصر سمت چپ (تقسیم بر ۲، ۴، ۸، ...)

□ گردشی

- عنصر سمت چپ: عنصر سمت راست

```
signal S1,S2 : bit_vector(7 downto 0);  
:  
S2 <= S1 ror 1;  
S2 <= S1 sra 1;  
S2 <= S1 srl 1;
```

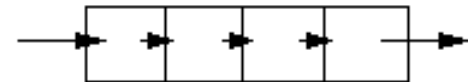
عملگرهای شیفت

- Logical Shift

sll

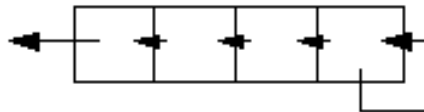


srl

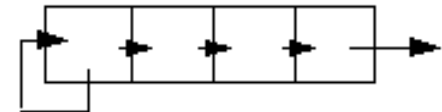


- Arithmetic Shift

sla

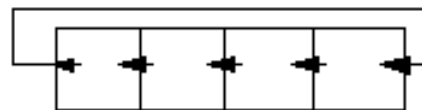


sra

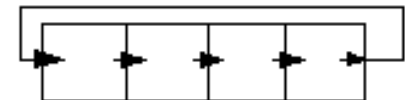


- Rotation

rol



ror



عملگرهای جمعی

• عملگرهای جمعی:

□ + و -

□ الحاق (concatenation)

- فقط سمت راست انتساب

- جمله دوم: با array aggregation هم امکان پذیر است

```
:  
signal S_A1, S_A2 : bit_vector (3 downto 0);  
signal S_A3 : bit_vector (7 downto 0);  
signal S1, S2, S3 : bit;  
:  
  S_A3 <= S_A1 & S_A2;  
  S_A1 <= S1 & S2 & '0' & S3;  
  S_A3 <= S_A1 & S1 & '1' & S2 & S3;  
  S_A3 <= S_A1 & "0000";
```


عملگرهای ضربی

• عملگرهای ضربی:

☐ * و /

☐ توان (**)

☐ rem و mod: باقیمانده

☐ فقط برای عملوندهای صحیح

– متفاوت وقتی که $A < 0$ یا $B < 0$

$\text{sign}(A \text{ rem } B) = \text{sign}(A)$ –

$\text{sign}(A \text{ mod } B) = \text{sign}(B)$ –