

Identifiers, Data Types, and Operators

- Identifiers
 - basic identifiers: start with a letter, do not end with “_”
 - case insensitive
- Data Objects
 - signals
 - constants
 - variables
 - files

1

VHDL Standard Data Types

Type	Range of values	Example declaration
integer	implementation defined	signal index: integer := 0;
real	implementation defined	variable val: real := 1.0;
boolean	(TRUE, FALSE)	variable test: boolean :=TRUE;
character	defined in package STANDARD	variable term: character := '@';
bit	0, 1	signal In1: bit := '0';
bit_vector	array with each element of type bit	variable PC: bit_vector (31 downto 0)
time	implementation defined	variable delay: time := 25 ns ;
string	array with each element of type character	variable name : string (1 to 10) := "model name";
natural	0 to the maximum integer value in the implementation	variable index: natural := 0;
positive	1 to the maximum integer value in the implementation	variable index: positive := 1;

2

Data Types (cont.)

- Enumerated data types are particularly useful for constructing models of computing systems

- examples

```
type instr_opcode is ('add', 'sub', 'xor', 'nor', 'beq', 'lw', 'sw');
```

```
type state is ('empty', 'half_full', 'half_empty', 'empty');
```

- Array types

```
type byte is array (7 downto 0) of std_logic;
```

```
type word is array (31 downto 0) of std_logic;
```

```
type memory is array (0 to 4095) of word;
```

3

Physical Types

```
type time is range <implementation dependent>
```

```
units
```

```
fs; -- femtoseconds
```

```
ps = 1000 fs; -- picoseconds
```

```
ns = 1000 ps; -- nanoseconds
```

```
us = 1000 ns; -- microseconds
```

```
ms = 1000 us; -- milliseconds
```

```
s = 1000 ms; -- seconds
```

```
min = 60 s; -- minutes
```

```
hr = 60 min; -- hours
```

```
end units;
```

```
type power is range 1 to 1000000
```

```
units
```

```
uw;-- base unit is microwatts
```

```
mw = 1000 uw;-- milliwatts
```

```
w = 1000 mw; -- watts
```

```
kw = 1000000 mw-- kilowatts
```

```
mw = 1000 kw; -- megawatts
```

```
end units;
```

4

Modeling with Physical Types

- Use packages to encapsulate type definitions, type conversions functions and arithmetic functions for new types
- Examples
 - modeling power
 - modeling silicon area
 - modeling physical resources that are “cumulative”

5

Operators

- VHDL '93 vs. VHDL '87 operators

logical operators	and	or	nand	nor	xor	xnor
relational operators	=	/=	<	<=	>	>=
shift operators	sll	srl	sla	sra	rol	ror
addition operators	+	-	&			
unary operators	+	-				
multiplying operators	*	/	mod	rem		
miscellaneous operators	**	abs	not	&		

- VHDL text or language reference manual for less commonly used operators and types

6