LANGUAGE CONSTRUCTS AND CONVENTIONS IN VERILOG

Dr.K.Sivasankaran
Associate Professor,
VLSI Division
School of Electronics Engineering,
VIT University

Outline

- White Space
- Operators
- Comments
- Identifiers
- Keywords
- Number Specification
- Strings

White Space

- In any design description the white space characters are included to improve readability.

Blank spaces ---> **\b**

Tabs ---> **\t**

New lines ---> **\n**

White space is not ignored in strings.

Example: **$display(** “The value of a=%b, b=%b, y=%b \n”, a,b,y);

Operators

- unary operators.
  - operates on a single operand.

  Example: **out = ~ a**;

- binary operators.
  - operates on two operands.

  Example: **out = a & b**;

- ternary operators.
  - operates on three operands.

  Example: **out = s ? a : b;**

Comments

- Improve readability and helps good documentation.

- Two comment structures are available in verilog:
  - single / one line comment

Example:

```
module d_ff (Q, dp, clk); //This is the design description of a D flip-flop.
//Here Q is the output.
// dp is the input and clk is the clock.
```

  - multiple line / block comment

Example:

```
/* this logic performs
     the reversal
     of bits */
```

- Multiple line comments cannot be nested.
  - /* This is /* a wrong comment */ structure */

Identifiers

- Identifiers are used to define language constructs.
- Identifiers refer objects to be referenced in the design.
- Identifiers are made of alphabets (both cases), numbers, the underscore '_' and the dollar sign '$'.
- They start with an alphabetic character or underscore.
- They cannot start with a number or with '$' which is reserved for system tasks.
- Identifiers are case sensitive i.e., identifiers differing in their case are distinct.
- An identifier say count is different from COUNT, count and cOuNT.

Identifiers

- **name, _name. Name, name1, name_$, . . .** all these are allowed as identifiers
- **name aa** not allowed as an identifier because of the blank (“name” and “aa” are interpreted as two different identifiers)
- **$name** not allowed as an identifier because of the presence of “$” as the first character.
- **1_name** not allowed as an identifier, since the numeral “1” is the first character
- **@name** not allowed as an identifier because of the presence of the character “@”.
- **A+b** not allowed as an identifier because of the presence of the character “+”.

Examples for Identifiers

Count

COUNT

_R2_D2

R56_68

FIVE$

$count	Illegal
12six_b	Illegal

```
module  and_gate  (a,b,y);
input a,b;
output y;
assign y= a & b;
endmodule
```

Identifier used during simulation

```
module  and_gate   (a,b,y);
Input  a,b ;
output y;
assign y= a & b;
endmodule
```

Identifier used in ports

Keywords

- The keywords define the language constructs.
- A keyword signifies an activity to be carried out, initiated, or terminated.
- As such, a programmer cannot use a keyword for any purpose other than that it is intended for.
- All keywords in Verilog are in small letters and require to be used as such.

Examples

module → signifies the beginning of a module definition.

endmodule → signifies the end of a module definition.

begin → signifies the beginning of a block of statements.

end → signifies the end of a block of statements.

if → signifies a conditional activity to be checked

while →signifies a conditional activity to be carried out.

Format

- Verilog HDL is case-sensitive.
- All the keywords in Verilog must be in lower case.
- Verilog constructs may be written across multiple lines, or on one line.

Number Specification

Sized numbers.

<size> ’<base format> <number>

Unsized numbers.

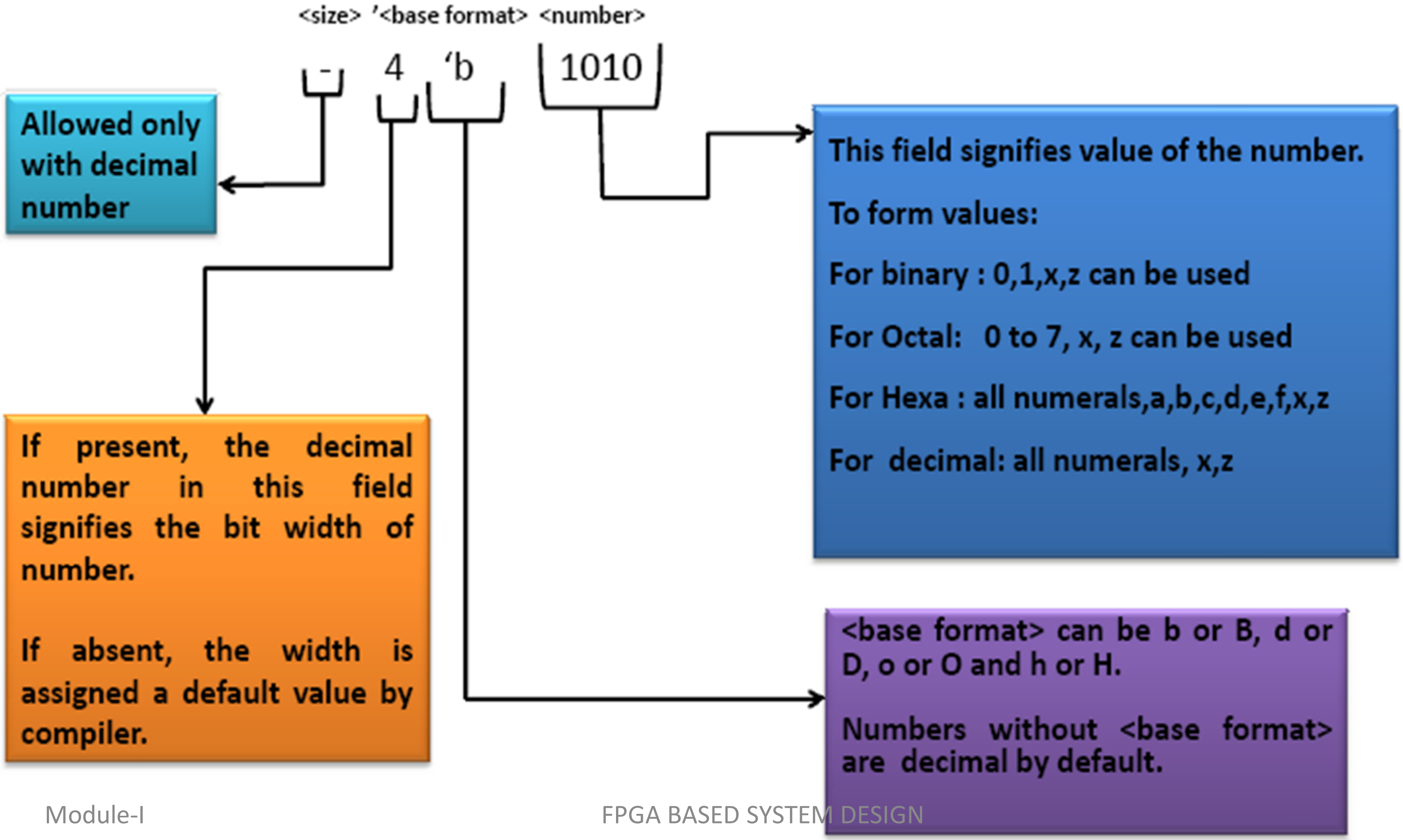
’<base format> <number>

<size> in decimal

<base format> can be b or B, d or D, o or O and h or H.

Numbers without <base format> are decimal by default.

Number specification



Allowed only with decimal number

If present, the decimal number in this field signifies the bit width of number.

If absent, the width is assigned a default value by compiler.

This field signifies value of the number.

To form values:

For binary : 0,1,x,z can be used

For Octal: 0 to 7, x, z can be used

For Hexa : all numerals,a,b,c,d,e,f,x,z

For decimal: all numerals, x,z

<base format> can be b or B, d or D, o or O and h or H.

Numbers without <base format> are decimal by default.

Number Specification -Example

Sized numbers :

```
4'b1111 // This is a 4-bit binary number
12'habc // This is a 12-bit hexadecimal number
16'd255 // This is a 16-bit decimal number.
```

Unsized numbers :

```
23456 // This is a 32-bit decimal number by default
'hc3 // This is a 32-bit hexadecimal number
 'o21 // This is a 32-bit octal number
```

Unknown and High Impedance Values

<u>Unknown & high impedance values</u>

- X or x for unknown values.
- Z or z for high impedance values.
- X or Z at the MSB has the self padding property.

<u>Examples:</u>

32 ’B z // this is a 32-bit high impedance number

6 ’h X // this is a 6-bit hex number

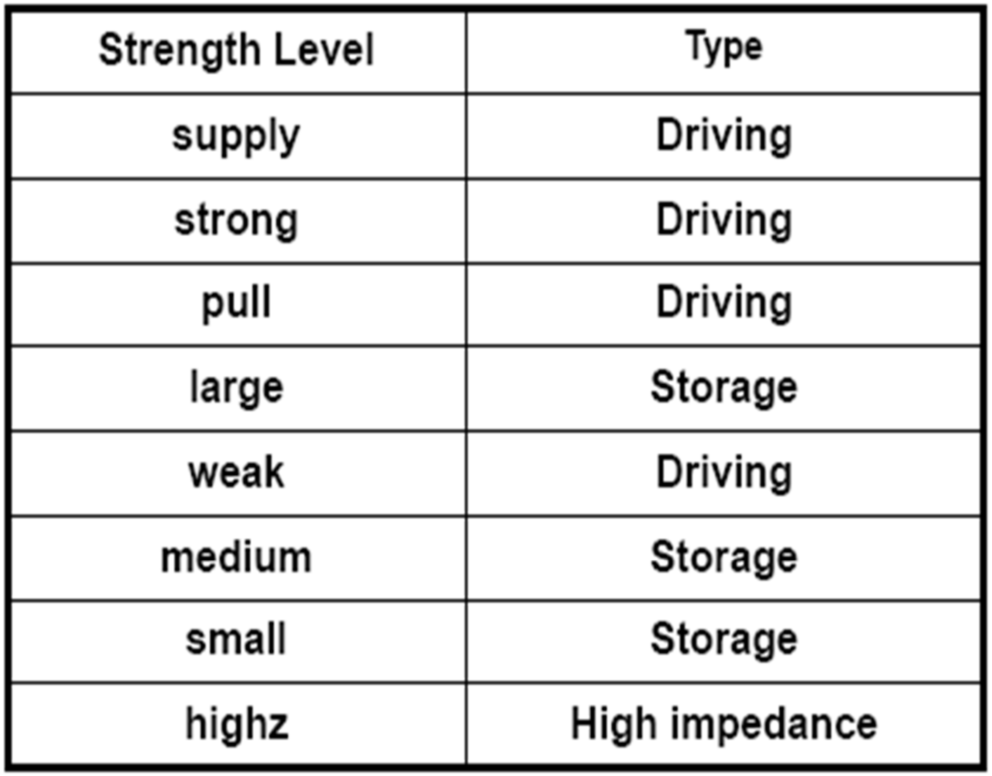
12 ’H 13x // this is a 12-bit hex number

Number Specification-Example

5’037	5-bit octal
4’D2	4-bit decimal
9’b11011x01	x signifies the concerned bit to be of unknown value.
9’o12z	equivalent to 001 010 **zzz**
7’Hx	7-bit x (x extended), i.e.… xxxxxxx
4’hz	4-bit z (z extended), i.e… zzzz
4’d-4	Not legal
-4’d7	Its value in 2’s complement form is 7.
8 ‘h 2A	Spaces allowed between size & ‘ character & between *base* and *value*
3’ b001	Not legal: no space allowed between ’ and base b
10’b10	Padded with 0 to the left, 0000000010
11’hb0	equivalent value is 000 1011 0000.
5’h**z**a	A 5-bit hex number. Its value is taken as **z** 1010.
3’b1001_0011	is same as 3’b011

Value Set

- In Verilog there are 4 values and 8 strength levels to model the real design



Strength Level	Type
supply	Driving
strong	Driving
pull	Driving
large	Storage
weak	Driving
medium	Storage
small	Storage
highz	High impedance

Integers

- It is a decimal number –signed or unsigned; an unsigned number is automatically taken as a positive number.

Example:

2

25

253

–253

The following are invalid since nondecimal representations are not permissible.

2a

B8

–2a

–B8

Real Numbers

Real numbers can be specified in decimal or scientific notation.

The decimal notation has the form

3.2

A number can be specified in scientific notation as

4.3e2

where 4.3 is the mantissa and 2 the exponent. The decimal equivalent of this number is 430.

Examples:

–4.3e2, –4.3e–2, and 4.3e–2.

Strings

- A string is a sequence of characters that are enclosed by double quotes.

- “Verilog classes are very interesting???”

- Spaces are not ignored in strings.

- Strings cannot be on multiple lines.

Strings in expressions

When a string of ASCII characters as above is an operand in an expression, it is treated as a binary number.

Example:

P = “numb”

assigns the binary value 0110 1110 0111 0101 0110 1101 0110 0010 to P

(0110 1110, 0111 0101, 0110 1101 and 0110 0010 are the 8-bit equivalents of the letters n, u, m, and b, respectively).

Review

- Multiple line comments can be nested – True/False
- List out rules for choosing an identifier name
- Verilog code is case insensitive – True/False
- Identify the valid number specification

  a. 1’b0 b. -4’b10 c. 3 ‘ b101 d. 3’bx e. 4’b10

- Strings can be given in multiple lines