Unit-5 Introduction to HDL(Verilog)

01 September 2022 09:58

Question Bank

1. What are the four abstraction relevel in Verilog?
2. What is synthesizable and non-synthesizable construct?
3. What testbench is explained?
4. How to represent number in Verilog language?
5. What are the two main data type in Verilog?

Brief History of Verilog

Verilog Hardware Description Language (HDL) 1984-Gateway Design Automation (Cadence Design Systems acquired it in 1989) and released the Verilog HDL to the public domain.

IEEE formed a standards working group in 1993 to create the standard and in December 1995 Verilog HDL officially became IEEE 1364-1995 standard (Verilog 1.0).

Presently IEEE-1364- 2005 Standard.

Basic of Verilog

- Logic circuit having four possible input values — logic high (1),logic low (0), don't care (x) and tristate (z) ; Verilog support it.
- Verilog is a case sensitive language.
- we can insert single line comment anywhere in the code using // which ends at the end of the line
- A block of comments starts with /* and it is terminated with */.

Reserve word in Verilog

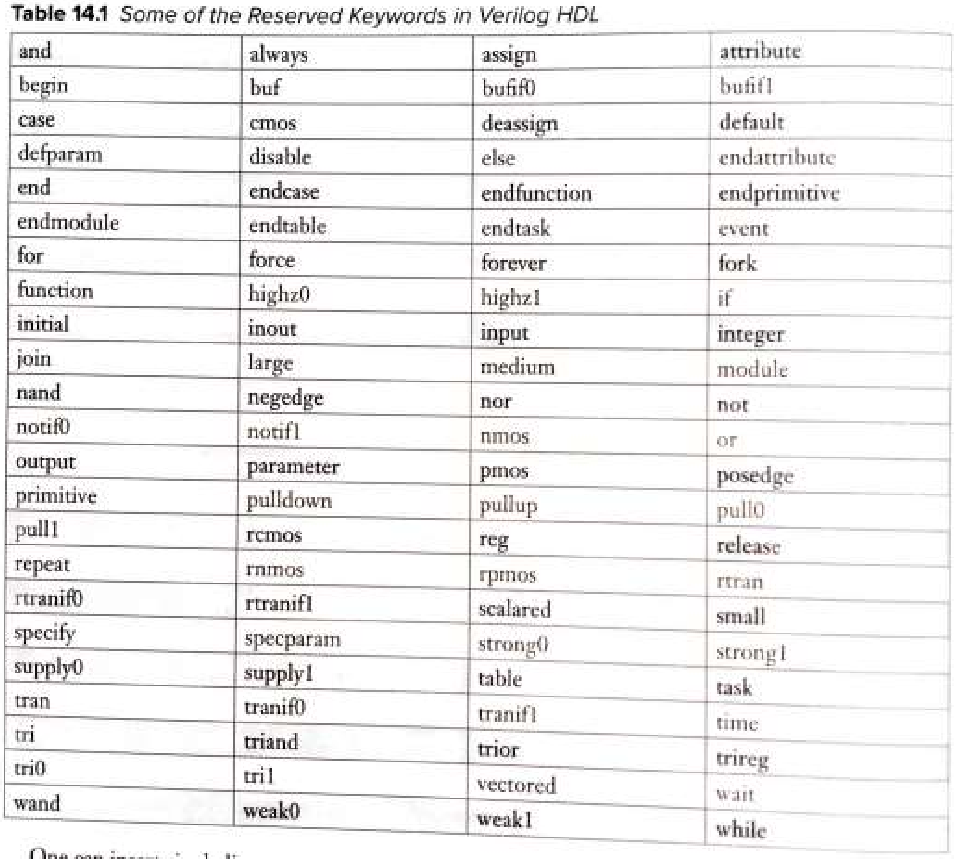


Table 14.1 *Some of the Reserved Keywords in Verilog HDL*

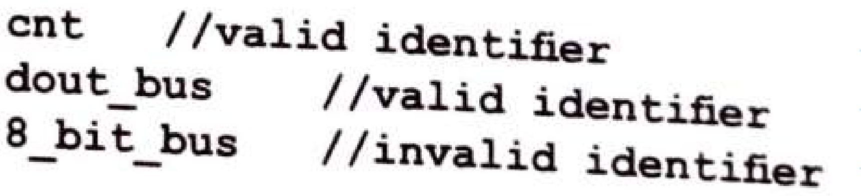
and	always	assign	attribute
begin	buf	bufif0	bufif1
case	cmos	deassign	default
defparam	disable	else	endattribute
end	endcase	endfunction	endprimitive
endmodule	endtable	endtask	event
for	force	forever	fork
function	highz0	highz1	if
initial	inout	input	integer
join	large	medium	module
nand	negedge	nor	not
notif0	notif1	nmos	or
output	parameter	pmos	posedge
primitive	pulldown	pullup	pull0
pull1	rcmos	reg	release
repeat	rnmos	rpmos	rtran
rtranif0	rtranif1	scalared	small
specify	specparam	strong0	strong1
supply0	supply1	table	task
tran	tranif0	tranif1	time
tri	triand	trior	trireg
tri0	tri1	vectored	wait
wand	weak0	weak1	while

One can insert

Identifier

User can provide a unique name to an object using an identifier so that it can be referenced.
Such an identifier can be any sequence of letters, digits, dollar signs ($) and underscore characters (_). But the identifier should not begin with a digit or $ character.

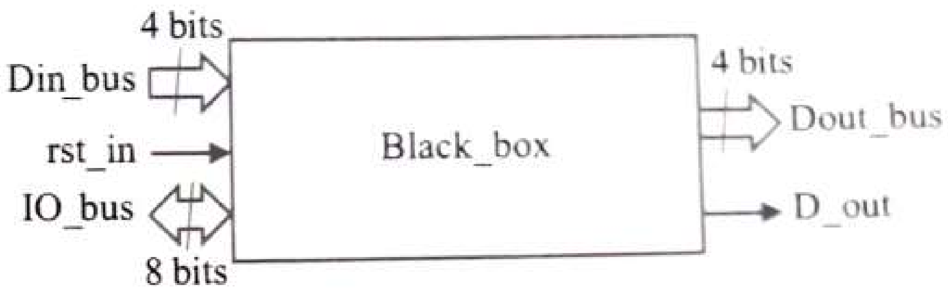
Example of the identifier



```
cnt         //valid identifier
dout_bus        //valid identifier
8_bit_bus       //invalid identifier
```

What is input/output pin?

The purpose of any hardware chip to process some input Signal and produce some output.
The problem statement define relationship between the input and output.
Initially designer do not know what is inside the chip therefore called black box.



A signal flow in a chip can be either unidirectional (only input, only output) or bidirectional (input—output)

Circuit laid inside the black box communicates with the external environment through this interface (IO pin).

For IO Verilog use reserve word input, output, inout keyword.

By Default input and output terminals are single bit and called as a scalar port. Sometime input or output more than one bit called a vector port.

input rst_in;			//input port (scalar) declaration
input [4:0] Din_bus;			//4 bits input port (vector) declaration
inout [7:0] IO_bus;			//8 bits IO port (vector) bidirectional bus declaration
output [3:0] Dout_bus;			//4 bits output port (vector) declaration
output D_out;			//output port (scalar) declaration

What are the four abstraction relevel?

Once the interface with this black box is clear, we can focus on the internal details and its design paradigm. Four levels of abstraction are provided in Verilog for the logic design purpose and designer are free to use it as per requirement.

- Behavioral or algorithmic level: It is the highest level of circuit abstraction which is provided in Verilog. The relationship between input and output, i.e., behavior of a black box is coded as per algorithm using procedural constructs. So, the hardware design code looks like C program and the inner details of the design are unclear.

- Dataflow level: Here, the designer has an idea how data is flowing in between the registers and how the data is being processed. Usually, designer has a rough or block level idea about the hardware to be implemented to attain the functionality. But complete gate level circuit is unclear and the hardware Implementation job is left on the synthesis tools. Most of the time the combination of behavior and dataflow- modelling known as Register Transfer Level (RTL) modelling is used to realize the black box design by coding the data flow through the logic circuit.

- Gate level: It is also known as structural design. The designer has clear idea about the logic gates used and their pin-to-pin connectivity. Entire functionality of a black box is described using logic gates and there is no ambiguity about the connections to be made. But, if the gate count of a circuit is more than hundreds of gates, keeping track of the connectivity is very difficult and time—consuming work which is contradictory to the objectives of using HDL. Further, if there is any change is design specification need to rework on the entire design.

- Switch level: It is the lowest level of abstraction which is supported by Verilog. It aims at the library cell development. It is possible to implement entire design using switches, i.e., transistors as every gate is made up of switches. Similar to gate level modelling, it is difficult to modify such a circuit to accommodate small changes in the design specifications. It is very difficult to migrate the design with change in technology node again and again. Further, today's designs comprise of millions of gates and laying so many transistors without any issues is over ambitious task.

What is synthesizable and non-synthesizable construct?

Test bench can have both synthesizable as well as non-synthesizable constructs.
But design file should have only synthesizable constructs as we want to map the designed circuit on a silicon chip.

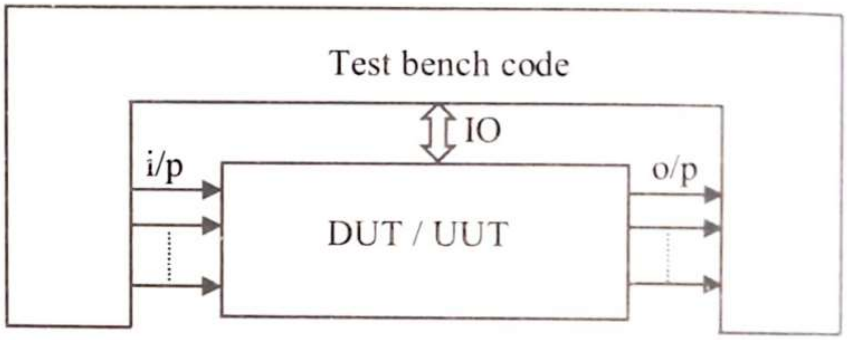
non-synthesizable constructs (example-time delay) used in test bench to test the circuit, need not to use other language only for test purpose.

Synthesizable constructs are divided in two group

- concurrent statements-Preferably concurrent statements are used while designing a combinational logic circuit
- sequential statements- one can use sequential statements to design a combinational or a sequential logic circuit.

What testbench is explained?

Program to test the design which made by us.
we can create a circuit testing environment using synthesizable as well as non-synthesizable constructs Is called test bench.



With the help of testbench regenerate set of 1 and 0 and apply to input terminal of the black box . And TB capture the change in the output terminal and display the result in various forms like webform or text data.

In a testbench there is no input and output therefore, the module declaration of a test bench will have empty port list.

Between module and end module keyword we cannot use another module declaration.

All the input of DUT are declared as a 'reg' in the testbench code and all output are declared as a wire in TB it is reverse as programs.

What are the two main data type in Verilog?

In combination Design - input 'wire', output 'reg' ; sequential opposite input 'reg' and output 'wire'

Two main data types 'wire' and 'register'. We can use 'wire' or 'reg' keyword to declare these two data type.

When concurrent statements are used to implement design functionality the LHS side signals must be declared as 'wire.'
This declaration is incapable of holding the signal value therefore , a continuous net driver is required.
This satisfies the requirement of the ,combinational logic gates wherein instantaneous changes at the input ports must be sensed to produce output signal.

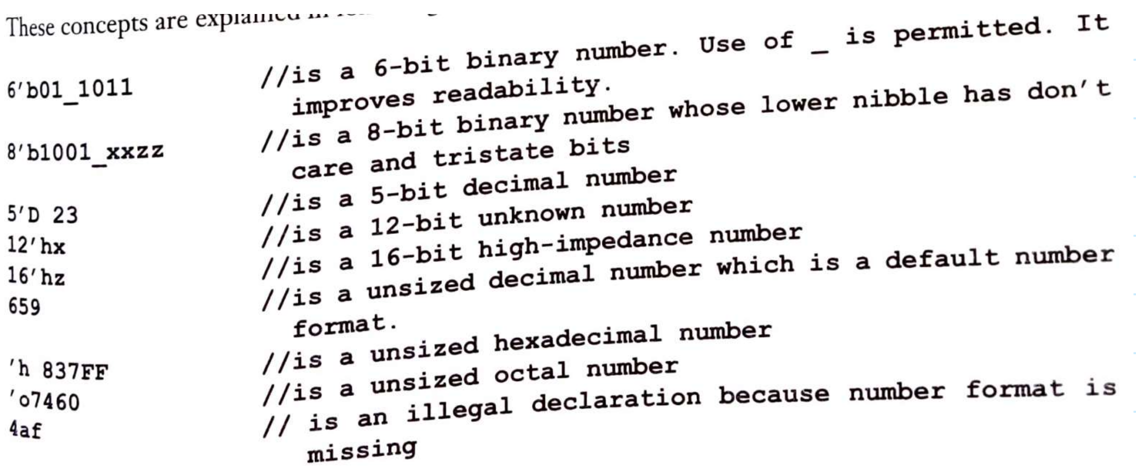
'reg'- it is capable of holding the signal value until the previous value is not overwritten.
This keyword creates a memory location in the tool wherein the current signal value is stored. Since the signal value is in the temporary memory location, one can change its value by replacing the previous data. This is the actual requirement of the sequential circuits which change the circuit output based on either the clock edge or asynchronous signals used in the circuit design. Therefore, sequential statements are to implement a sequential circuit design fictionality with the LHS side signals declared as a 'reg.'

How to represent number in Verilog language?

Four types of number systems —binary, octal, decimal, and hexadecimal
Number base format (i.e., radix) is case insensitive so one can use b or B to represent a binary number. Similarly, o, d, and h can be used to octal, decimal, and hexadecimal numbers, respectively.

<size>'<radix><number>

Size field how many bits(data width)
Example



These concepts are explained in

```
6'b01_1011        //is a 6-bit binary number. Use of _ is permitted. It
                    improves readability.
8'b1001_xxzz      //is a 8-bit binary number whose lower nibble has don't
                    care and tristate bits
5'D 23            //is a 5-bit decimal number
12'hx             //is a 12-bit unknown number
16'hz             //is a 16-bit high-impedance number
659               //is a unsized decimal number which is a default number
                    format.
'h 837FF          //is a unsized hexadecimal number
'o7460            //is a unsized octal number
4af               // is an illegal declaration because number format is
                     missing
```