Sri SAI RAM ENGINEERING COLLEGE

INSTITUTE OF TECHNOLOGY

West Tambaram, Chennai - 44

YEAR	SEM
II	III

CS 8351

DIGITAL PRINCIPLES AND SYSTEM DESIGN
(Common to CSE & IT)

UNIT NO. III

3.8 HDL Models of Sequential circuits

Version: 1.0

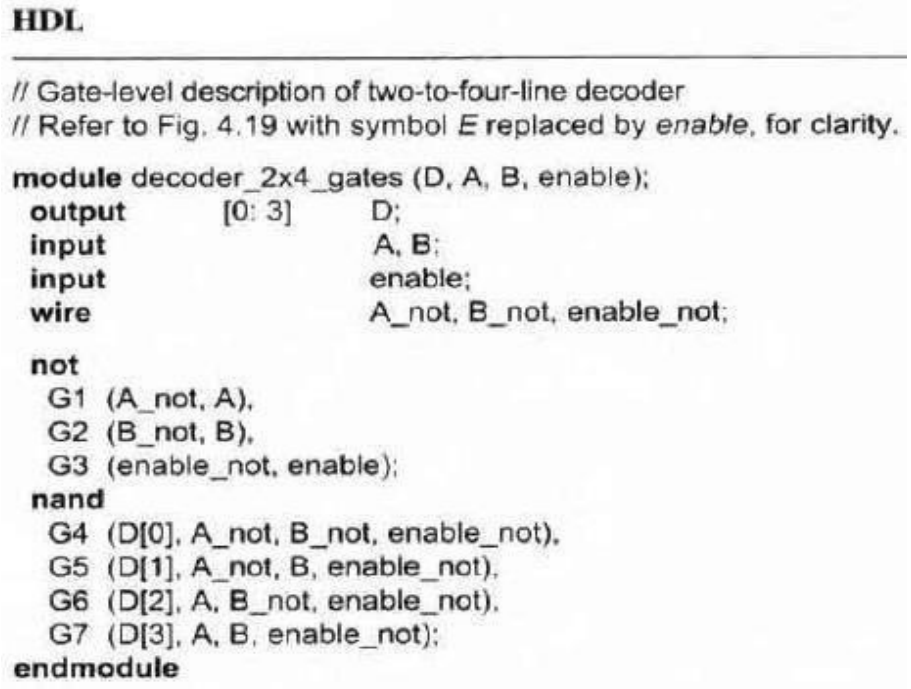
HDL MODELS

The Verilog HDL model can be described in any one of the following modeling styles,

- ✓ Gate level modeling-using instantiations of predefined and user defined primitive gates.
- ✓ Dataflow modeling using continuous assignment with the keyword **assign**.
- ✓ Behavioral modeling using procedural assignment statements with the keyword **always**.

➢ Gate level modeling

In this type, a circuit is specified by its logic gates and their interconnections. Gate level modeling provides a textual description of a schematic diagram. The verilog HDL includes 12 basic gates as predefined primitives. They are and, nand, or, nor, xor, xnor, not & buf.



HDL

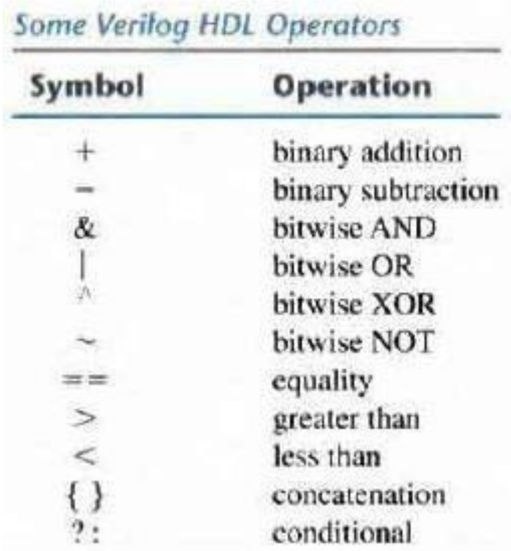
```
// Gate-level description of two-to-four-line decoder
// Refer to Fig. 4.19 with symbol E replaced by enable, for clarity.

module decoder_2x4_gates (D, A, B, enable);
 output        [0: 3]       D;
 input                      A, B;
 input                      enable;
 wire                       A_not, B_not, enable_not;

 not
   G1  (A_not, A),
   G2  (B_not, B),
   G3  (enable_not, enable);
 nand
   G4  (D[0], A_not, B_not, enable_not),
   G5  (D[1], A_not, B, enable_not),
   G6  (D[2], A, B_not, enable_not),
   G7  (D[3], A, B, enable_not);
endmodule
```

- **Dataflow modeling**

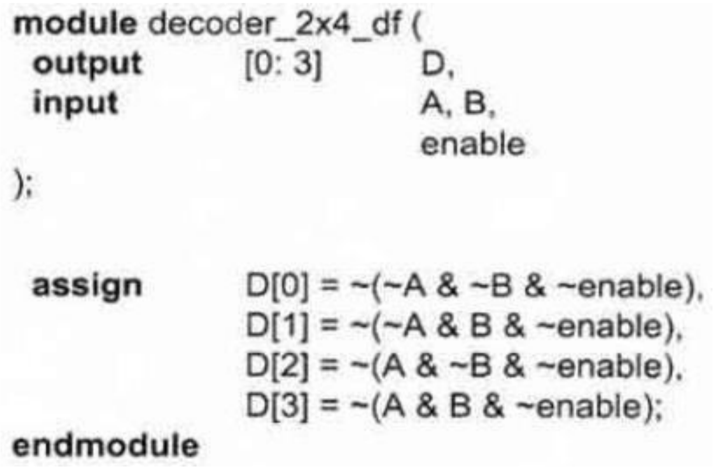
Dataflow modeling of combinational logic uses a number of operators that act on operands to produce desired results. Verilog HDL provides about 30 different operators. Dataflow modeling uses continuous assignments and the keyword **assign**. A continuous assignment is a statement that assigns a value to a net. The data type family **net** is used to represent a physical connection between circuit elements.



Some Verilog HDL Operators

Symbol	Operation
+	binary addition
−	binary subtraction
&	bitwise AND
|	bitwise OR
^	bitwise XOR
~	bitwise NOT
==	equality
>	greater than
<	less than
{ }	concatenation
? :	conditional

HDL for 2-to-4 line decoder

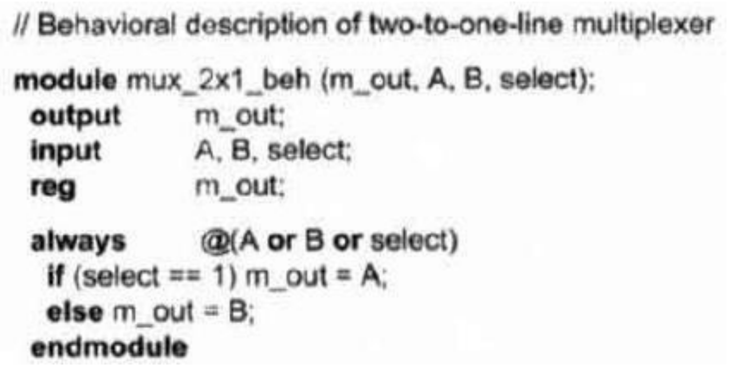


```
module decoder_2x4_df (
  output        [0: 3]       D,
  input                      A, B,
                             enable
);

  assign        D[0] = ~(~A & ~B & ~enable),
                D[1] = ~(~A & B & ~enable),
                D[2] = ~(A & ~B & ~enable),
                D[3] = ~(A & B & ~enable);
endmodule
```

- **Behavioral modeling**

Behavioral modeling represents digital circuits at a functional and algorithmic level. It is used mostly to describe sequential circuits, but can also be used to describe combinational circuits. Behavioral descriptions use the keyword **always,** followed by an optional event control expression and a list of procedural assignment statements.



```
// Behavioral description of two-to-one-line multiplexer

module mux_2x1_beh (m_out, A, B, select);
  output        m_out;
  input         A, B, select;
  reg           m_out;

  always         @(A or B or select)
    if (select == 1) m_out = A;
    else m_out = B;
  endmodule
```