Intro to Verilog

TABLE 2-2
Verilog Primitives for Combinational Logic Gates

Gate primitive	Example instance
`and`	`and (F, X, Y);`
`or`	`or (F, X, Y);`
`not`	`not (F, Y);`
`nand`	`nand (F, X, Y);`
`nor`	`nor (F, X, Y);`
`xor`	`xor (F, X, Y);`
`xnor`	`xnor (F, X, Y);`

TABLE 2-4
Verilog Bitwise Logic Operators

Verilog operator symbol	Operator function	Example
~	Bitwise not	`F = ~X;`
&	Bitwise and	`F = X & Y;`
|	Bitwise or	`F = X | Y;`
^	Bitwise xor	`F = X ^ Y;`
~^, ^~	Bitwise xnor	`F = X ~^ Y;`

TABLE 2-5
Truth Table for the Function $L = D\overline{X} + A$

D	X	A	L
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

FIGURE 2-5 Logic Circuit Diagram for L = DX + A

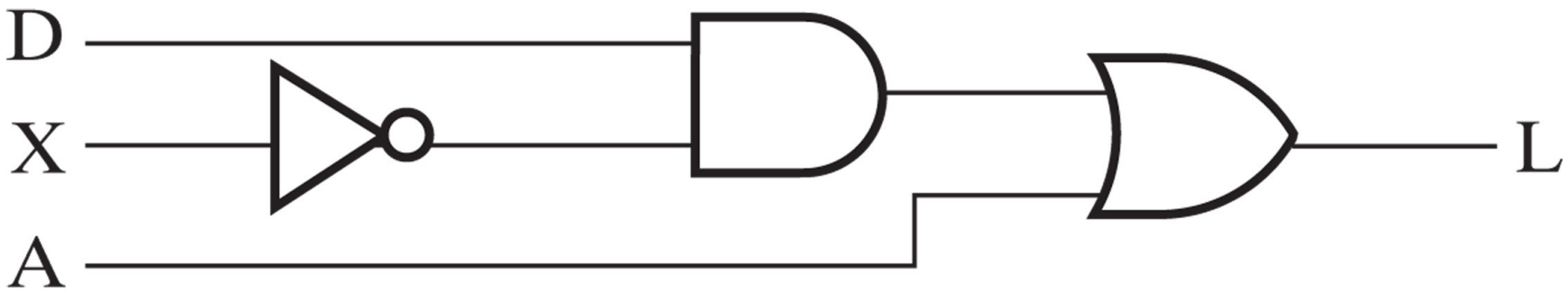

FIGURE 2-6 Verilog Model for the Logic Circuit of Figure 2-5

```
module fig2_5 (L, D, X, A);
    input D, X, A;
    output L;
    wire X_n, t2;

    not (X_n, X);
    and (t2, D, X_n);
    or (L, t2, A);
endmodule
```

FIGURE 2-27 Gate level schematic for a two-bit greater-than comparator circuit

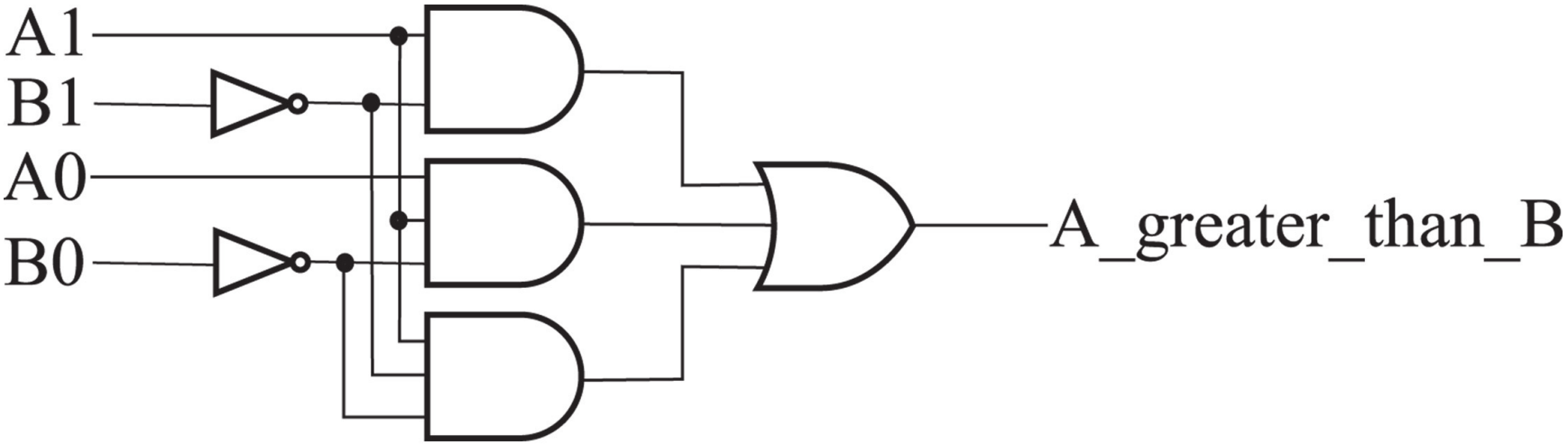

FIGURE 2-33 Structural Verilog Description of Two-Bit Greater-Than Circuit

```
// Two-bit greater-than circuit: Verilog structural model                    //  1
// See Figure 2-27 for logic diagram                                         //  2
module comparator_greater_than_structural(A, B, A_greater_than_B);           //  3
 input [1:0] A, B;                                                          //  4
 output A_greater_than_B;                                                   //  5
 wire B0_n, B1_n, and0_out, and1_out, and2_out;                             //  6
  not                                                                       //  7
   inv0(B0_n,  B[0]),  inv1(B1_n,  B[1]);                                   //  8
  and                                                                       //  9
   and0(and0_out,  A[1],  B1_n),                                            // 10
   and1(and1_out,  A[1],  A[0],  B0_n),                                     // 11
   and2(and2_out,  A[0],  B1_n,  B0_n);                                     // 12
  or                                                                        // 13
   or0(A_greater_than_B,  and0_out,  and1_out,  and2_out);                  // 14
endmodule                                                                   // 15
```

FIGURE 2-34 Dataflow Verilog Description of Two-Bit Greater-Than Comparator

```
// Two-bit greater-than circuit: Dataflow model                    //  1
// See Figure 2-27 for logic diagram                               //  2
module comparator_greater_than_dataflow(A, B, A_greater_than_B);   //  3
 input [1:0] A, B;                                                 //  4
 output A_greater_than_B;                                          //  5
 wire B1_n, B0_n, and0_out, and1_out, and2_out;                    //  6
 assign B1_n = ~B[1];                                              //  7
 assign B0_n = ~B[0];                                              //  8
 assign and0_out = A[1] & B1_n;                                    //  9
 assign and1_out = A[1] & A[0] & B0_n;                             // 10
 assign and2_out = A[0] & B1_n & B0_n;                             // 11
 assign A_greater_than_B = and0_out | and1_out | and2_out;         // 12
endmodule                                                          // 13
```

FIGURE 2-35 Conditional Dataflow Verilog Description of Two-Bit Greater-Than Circuit

```
// Two-bit greater-than circuit: Conditional model                    // 1
// See Figure 2-27 for logic diagram                                  // 2
module comparator_greater_than_conditional2(A, B, A_greater_than_B);  // 3
 input [1:0] A, B;                                                    // 4
 output A_greater_than_B;                                             // 5
 assign A_greater_than_B = (A > B)? 1'b1 :                            // 6
      1'b0;                                                           // 7
endmodule                                                             // 8
```

FIGURE 2-37 Behavioral Verilog Description of Two-Bit Greater-Than Circuit

```
// Two-bit greater-than circuit: Behavioral model                    // 1
// See Figure 2-27 for logic diagram                                 // 2
module comparator_greater_than_behavioral(A, B, A_greater_than_B);   // 3
 input [1:0] A, B;                                                   // 4
 output A_greater_than_B;                                            // 5
 assign A_greater_than_B = A > B;                                    // 6
endmodule                                                            // 7
```

THE END