**Conditional Statements:**

Conditional statements are used for making decisions based upon certain conditions. These conditions are used to decide whether or not a statement should be executed. Keywords if and else are used for conditional statements. There are three types of conditional statements.

**Type 1 conditional statement. No else statement.**

Statement executes or does not execute.

if (<expression>) true\_statement ;

**Type 2 conditional statement. One else statement**

Either true\_statement or false\_statement is evaluated

if (<expression>) true\_statement ; else false\_statement ;

**Type 3 conditional statement. Nested if-else-if.**

Choice of multiple statements. Only one is executed.

if (<expression1>) true\_statement1 ;

else if (<expression2>) true\_statement2 ;

else if (<expression3>) true\_statement3 ;

else default\_statement ;

In the following example one Verilog code is given for a 2:4 decoder using if-else structure.

module decoder(Do, Din, En);

input [1:0] Din;

input En;

output [3:0] Do;

reg [3:0] Do;

always @(En or Din)

begin

if (En)

begin

if (Din == 2'b00)

Do= 4'b0001;

else if (Din == 2'b01)

Do = 4'b0010;

else if (Din == 2'b10)

Do = 4'b0100;

else if (Din == 2'b11)

Do = 4'b1000;

else

$display("Error!");

end

end

endmodule

**Testbench Code:**

module decoder\_tb\_v;

reg [1:0] Din\_t;

reg En\_t;

wire [3:0] Do\_t;

decoder24 uut(

.Do(Do\_t),

.Din(Din\_t),

.En(En\_t)

);

initial begin

// Initialize Inputs

En = 1;

Din =2'b00; #20;

Din = 2'b01; #20;

Din = 2'b10; #20;

Din = 2'b11; #20;

end

initial

begin

$dumpfile("dump.vcd");

$dumpvars(1);

end

endmodule

**Conditional statement: case Statement**

In type 3 conditional statement of if-else, there were many alternatives, from which one was chosen. The nested if-else-if can become unwieldy if there are too many alternatives. A shortcut to achieve the same result is to use the case statement.

case Statement:

The keywords **case**, **endcase**, and **default** are used in the case statement.

case (expression)

alternative1: statement1;

alternative2: statement2;

alternative3: statement3;

...

...

default: default\_statement;

endcase

An example has been given here to design the 2:4 decoder using case statement.

module decoder\_case(Do, Din, En);

input En;

input [1:0] Din;

output [3:0]Do;

reg [3:0]Do;

always @ (En or Din)

begin

if (En)

begin

case (Din)

2'b00: Do = 4'b0001;

2'b01: Do = 4'b0010;

2'b10: Do = 4'b0100;

2'b11: Do = 4'b1000;

default: $display("Error!");

endcase

end

end

endmodule