BLOCKING & NON-BLOCKING ASSIGNMENTS

(Procedural Assignment Statements)

Blocking Assignments:

Sequential Execution:

- Blocking assignments are executed sequentially, one after the other, in the order they appear in the code.
- The next statement will not begin execution until the current statement is completed.

Symbol:

- The assignment operator = is used for blocking assignments.

Immediate Update:

- The right-hand side (RHS) of a blocking assignment is evaluated immediately, and the left-hand side (LHS) is updated immediately.
- Both assign and evaluate at active region.

Example:

```
module swap; reg
 [3:0] a,b;
 reg [3:0] temp;
 initial begin
  a=10;b=11;
   $display("____USING BLOCKING SWAP THE NUMBERS __ ");
   #10;
   temp=b;
   b=a;
   a=temp;
#20 $flnish();
 end
 initial
```

$monitor($time," values of a=%d,b=%d",a,b); endmodule

Output:

____USING BLOCKING SWAP THE NUMBERS_____
0 values of a=10,b=11
10 values of a=11,b=10

Non-Blocking Assignments:

Parallel Execution:

- Non-blocking assignments allow for parallel execution of statements within the same always block.
- All non-blocking assignments within the block are scheduled to occur simultaneously without waiting for the result of the previous one.

Symbol:

- The non-blocking assignment operator <= is used for non-blocking assignments.

Delayed Update:

- In a non-blocking assignment, the RHS is evaluated immediately(Active Region), but the actual update of the LHS is delayed until the end of the current time step(NBA Region).

Example:

```
module swap; reg
 [3:0]a,b; initial
 begin
  $display("_____USING NON BLOCKING SWAP A NUMBERS___");
  a=11;b=10; #10;
  a<=b;
  b<=a;
```

```
  #20 $flnish();
 end
 initial
  $monitor($time,"values of a=%d,b=%d",a,b);
endmodule
```

Output:

```
_______USING NON BLOCKING SWAP A
          NUMBERS_____ 0values of a=11,b=10
          10values of a=10,b=11
```