Lecture - 10

☆ Procedural Assignment (contd.)

* Recap :
  - always block
  - initial block
  - begin - end statement
  - if, if-else, if-else if-else statements
  - case statement

* "while" loop
  - Syntax :

```
while (expression)
        sequential statement;
```

  - sequential statement can contain "begin --- end" statement.

Scanned with CamScanner

* Example:

```
integer my_count = 0;
initial
begin
    while (my_count <= 255)
    begin
        $display ("My count: %d", my_count);
        my_count = my_count + 1;
    end
end
```

* "for" loop

- Syntax:

```
for (expr1; expr2; expr3)
    sequential statement;
```

- The "for" loop consists of three parts:
  i) An initial condition (expr1).
  ii) A check to see if the terminating condition is true (expr2).
  iii) A procedural assignment to ~~check~~ change the value of the control variable (expr3).

* Example:

```
reg [100:1] data;
integer i;
initial
    for (i=1; i<=100; i=i+1)
        data[i] = 1'b0;
```

- Note: "i++" is not available in Verilog.

Scanned with CamScanner

* "repeat"

- Syntax: repeat (expression)
  sequential-statement;

- Difference between "repeat" & "while"/"for"
  → "while"/"for" executes the loop based on some condition
  → "repeat" executes the loop a fixed no. of times.

- repeat (expression)
  sequential statement;

  expression → constant, variable, signal

  sequential statement → can also be "begin.... end".

- If "expression" is a variable, the its value is taken which was present at start of "repeat" block. i.e. during execution of repeat block, even if value of expression variable change, only the old value which was at the start of loop is considered.

* Example:

```
reg clock;
initial
begin
    clock = 1'b0;
    repeat (100)
        #5 clock = ~clock; //Generate only 50
                           //clock periods/cycles
end
```

Scanned with CamScanner

* "forever"

- Syntax : forever
  sequential-statement;

- "forever" is equivalent to while (1)

- Example

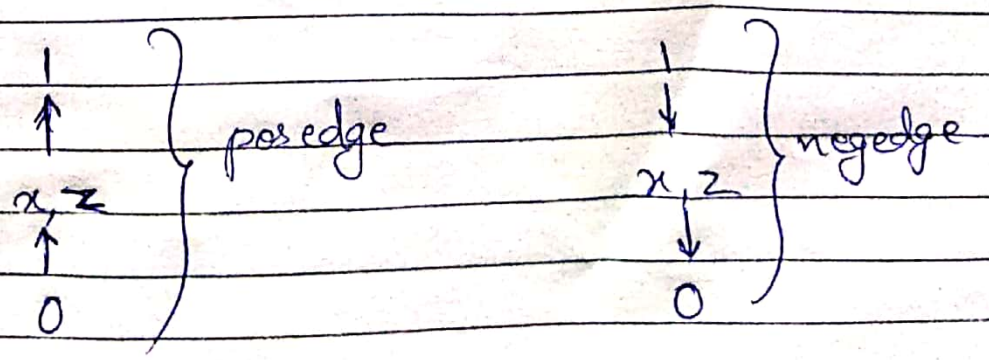
```
reg clk;
initial
begin
        clk = 1'b0;
        forever
                #5  clk = ~clk; //Period = 10 units
end
```

* Other Constructs in Verilog

- # (time-value) → makes a block suspend for "time-value" units of time

- @ (event-expression) → makes a block suspend until "event-expression" triggers.

- posedge → any transition from 0, x, z to '1'
  → also transition from '0' to x, z



Scanned with CamScanner

* Examples

@(in)	"in" changes
@(a or b or c)	any of a, b, c changes
@(a, b, c)	
@(posedge clk)	+ve edge of clk
@(*)	any variable changes

* Laboratory Tasks

1) D-flipflop with synchronous set & reset
2) D-flip flop with asynchronous set & reset

Scanned with CamScanner