DAY 36 - 111 DAYS VERIFICATION CHALLENGE

Topic: Operators

Skill: Verilog

DAY 36 CHALLENGE:

1. Describe following operators with an example:

I. Arithmetic:

Character	Operation performed	Example
+	Add	b + c = 11
-	Subtrac	b - c = 9, -b=-10
/	Divide	b / a = 2
*	Multiply	a * b = 50
%	Modulus	b % a = 0

II. Logical:

Character	Operation performed	Example
!	Not true	!(a && b) = 1'b1
&&	Both expressions true	a && b = 1'b0
||	One ore both expressions true	a || b = 1'b1

III. Relational:

>	Greater than	a > b = 1'b0
<	Smaller than	a < b = 1'b1
>=	Greater than or equal	a >= d = 1'bX
<=	Smaller than or equal	a <= e = 1'bX

IV. Equality:

==	Equality	a == b = 1'b0
!=	Inequality	a != b = 1'b1
===	Case equality	e === e = 1'b1
!===	Case inequality	a !== d = 1'b1

V. Bitwise:

Character	Operation performed	Example
~	Invert each bit	~a = 3'b010
&	And each bit	b & c = 3'b010
|	Or each bit	a | b = 3'b111
^	Xor each bit	a ^ b = 3'b011
^~ or ~^	Xnor each bit	a ^~ b = 3'b100

VI. Reduction:

Character	Operation performed	Example
&	And all bits	&a = 1'b0, &d = 1'b0
~&	Nand all bits	~&a = 1'b1
|	Or all bits	|a = 1'b1, |c = 1'bX
~|	Nor all bits	~|a= 1'b0
^	Xor all bits	^a = 1'b1
^~ or ~^	Xnor all bits	~^a = 1'b0

VII. Shift:

Character	Operation performed	Example
>>	Shift right	b >> 1 results 4?b010X
<<	Shift left	a << 2 results 4?b1000

VIII. Concatenation:

{}	Concatenate	c = {a,b} = 8'101010x0

IX. Replication:

{{}}	Replicate	{3{2'b10}}= 6'b101010

X. Conditional:

?:	Conditions testing	test cond. ? if true do this or if not do this

2. Explain Operator Precedence in Verilog

Operator precedence determines the order in which operators are evaluated in expressions. Operators with higher precedence are evaluated before operators with lower precedence. When operators have the same precedence, their associativity (left-to-right or right-to-left) determines the order of evaluation.

Operators precedence
+, -, !, ~ (Unary)
+,- (Binary)
<<, >>
<,>,<=,>=
==, !=
&
^, ^~ or ~^
|
&&
||
?:

3. Design following in Verilog:

a. 3-bit ring counter

```
module three_bit_ring_counter(
    input clk,
    input rst,
    output reg [2:0] count
    );

always @(posedge clk)
    begin
       if(rst)
        begin
            count = 3'b001 ;
        end
        else
        begin
            count = {count[1:0], count[2]} ;
        end
    end
endmodule
```

b. Negative-edge triggered D-FF with clear

```
module D_FF_clear(
    input clk,
    input clr,
    input data_in,
    output reg data_out
    );
```

```
always @(negedge clk or posedge clr )
    begin
        if(clr)
            begin
                data_out <= 0 ;
            end
        else
            begin
                data_out <= data_in ;
            end
    end
endmodule
```

c. 4-bit asynchronous counter

```
module three_bit_ring_counter(
    input clk,
    input rst,
    output reg [3:0] count
    );

always @(posedge clk)
    begin
       if(rst)
        begin
            count = 0 ;
        end
        else
        begin
            count = count + 1 ;
        end
    end
endmodule
```

d. 4-bit comparator which compares two 4-bit no. & prints the largest no.

```
module four_bit_comparator(
    input [3:0] in_data1,
    input [3:0] in_data2,
    output reg [3:0] max_data
    );
always @(*)
    begin
        if(in_data1 >= in_data2)
            begin
                max_data <= in_data1 ;
            end
        else max_data <= in_data2 ;
    end
endmodule
```