Solutions to Quiz 1

There were four versions of Question 1 and two versions of Question 2. The values and the answers for all versions are given below.

Question 1

Fill the table below with the value of each expression as a Verilog numeric literal including the correct width and the correct value in *hexadecimal* base. Assume the following declarations:

```
logic [7:0] x ;
logic [3:0] y ;
```

and that `x` has the value `8'h38` (or `8'hA6`) and that `y` has the value `4'b0101`. The first row has been filled in as an example. You need not show your work or draw another box around the answer

expression	value
`x[3:0]`	`4'h8 (or 4'h6)`
`x[5:4] && y ^ y`	
`1 > 1 < 1`	
`x[5:4] & y`	
`{ y >> 4, x }`	
`y * ( y > x[7:4] )`	
`x[4] ? x[7] : !x[7] ? x[7:4] : x[7:0]`	

— or —

expression	value
`x[3:0]`	`4'h8 (or 4'h6)`
`x[5:4] || y ^ y`	
`1 >= 1 < 1`	
`y[3:2] | x`	
`{ y >> 4, x }`	
`y * ( y < x[7:4] )`	
`x[4] ? x[7] : !x[7] ? x[7:4] : x[7:0]`	

Answers

For `x = 8'ha6` (`8'b1010_0110`) and `y = 4'b0101`:

expression	value
`x[3:0]`	`4'h6`
`x[5:4] && y ^ y`	`1'h0`
`1 > 1 < 1`	`1'h1`
`x[5:4] & y`	`4'h0`
`{ y >> 4, x }`	`12'ha6`
`y * ( y > x[7:4] )`	`4'h0`
`x[4] ? x[7] : !x[7] ? x[7:4] : x[7:0]`	`8'ha6`

expression	value
`x[3:0]`	`4'h6`
`x[5:4] || y ^ y`	`1'h1`
`1 >= 1 < 1`	`1'h0`
`y[3:2] | x`	`8'ha7`
`{ y >> 4, x }`	`12'ha6`
`y * ( y < x[7:4] )`	`4'h5`
`x[4] ? x[7] : !x[7] ? x[7:4] : x[7:0]`	`8'ha6`

For `x = 8'h38` (`8'b0011_1000`) and `y = 4'b0101`:

expression	value
x[3:0]	4'h8
x[5:4] && y ^ y	1'h0
1 > 1 < 1	1'h1
x[5:4] & y	4'h1
{ y >> 4, x }	12'h38
y * (y > x[7:4])	4'h5
x[4] ? x[7] : !x[7] ? x[7:4] : x[7:0]	8'h0

expression	value
x[3:0]	4'h8
x[5:4] || y ^ y	1'h1
1 >= 1 < 1	1'h0
y[3:2] | x	8'h39
{ y >> 4, x }	12'h38
y * (y < x[7:4])	4'h0
x[4] ? x[7] : !x[7] ? x[7:4] : x[7:0]	8'h0

Question 2

Write a Verilog module named **`next`** that has a 16-bit **`logic`** input named **`cnt`** and a 16-bit logic output named **`cnt_next`**. If the value of **`cnt`** is greater than or equal to 4095 (or less than or equal to zero) then **`cnt_next`** should be set to zero (or 8190), otherwise it should be set to **`cnt`** plus one (or minus three). Follow the course coding guidelines but omit comments.

Answers

```
module next ( input logic [15:0] cnt,
              output logic [15:0] cnt_next ) ;

   assign cnt_next = cnt >= 4095 ? '0 : cnt + 1 ;

endmodule

module next_ ( input logic [15:0] cnt,
               output logic [15:0] cnt_next ) ;

   assign cnt_next = cnt <= 0 ? 8190 : cnt - 3 ;

endmodule
```