More Verilog

Exercise 1: Is a signal named **$\overline{\text{overload}}$** active-high or active-low? Is there an overload if this signal is high? What if the signal was named **overload**? $\overline{\text{overload}}$

active-low

H → false

"overload" is false → no overload.

overload

active-high

H → true

"overload" is true → yes, overload.

Exercise 2: Come up with active-high and an active-low names for a signal that is at 3 V when a door is open and 0 V when the door is closed.

open → active high

$\overline{\text{closed}}$ → active low

Exercise 3: If $\overline{\texttt{D}}$ is a word and $\overline{\texttt{D0}}$ is low, is the word an even or odd number?

for bits in $\overline{D}$

L = 1

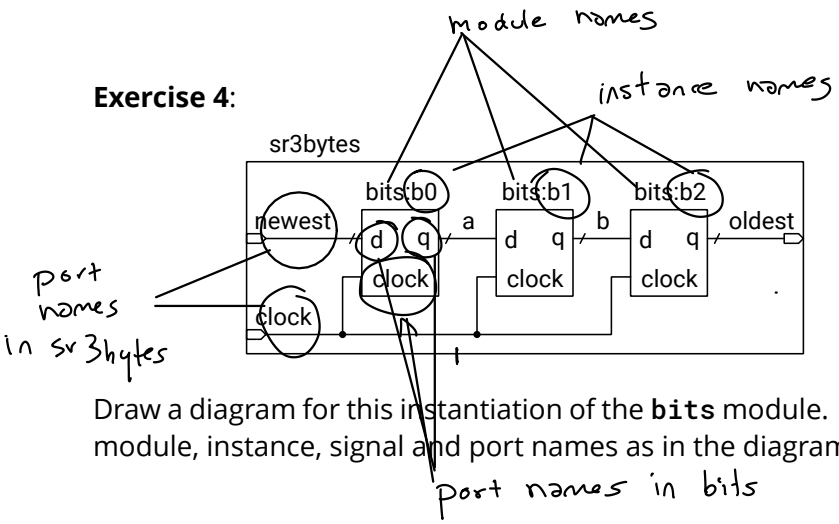
H = 0

(L.S. bit is low ≡ 1) . odd

$D_0 \equiv D[0]$ = L.S. bit of D.

(over bar means active-low not complement).

Exercise 4:



signals in sr3bytes: newest, oldest, clock, a, b

ports in sr3bytes newest, oldest, clock

Draw a diagram for this instantiation of the **bits** module. Label the module, instance, signal and port names as in the diagram above.

ports in bits: d, q, clock

Exercise 5:

```
module sr3bytes
   (
    input  logic [7:0] newest,
    output logic [7:0] oldest,
    input  logic clock
    ) ;

   localparam nbits = 8 ;

   logic [nbits-1:0] a, b ;

   // matching by order
   bits #(nbits) b0 (newest,a,clock);

   // matching by name (order does not matter)
   bits #(.nb(nbits)) b1 (.q(b),.clock,.d(a));

   // wildcards for names that match
   bits #(.nb(nbits)) b2 (.d(b),.q(oldest),.*);

endmodule
```

signals

instance names.

module names

module instantiations

port/signal

Identify the module instantiation statements in the code above. For each one, what is the instantiated module's name? The instance name?

Exercise 6: What are the values of the following expressions: `|4'b0001`, `^4'b1001`, `&4'b1111`, `&4'b1110`?

1

1^0^0^1 → 0

1

0

Exercise 7:

```
// concatenation:
  logic [3:0] x = { 2'b00, 2'b11 } ;
// array literal:
  logic [3:0] x[2] = '{ 4'b0011, 4'b1010 } ;
// replication within literal:
  logic [3:0] x[2] = '{2{ 2'b00, 2'b11 }} ;
```

x is 4'b0011

x[0] is 4'b0011, x[1] is 4'b1010

What are the initial values of **x** in the examples above?

should be '{2{ {2'b00,2'b11} }}
initial value of x[0] and x[1] is 4'b0011