Simulation

Exercise 1:

1. typical inputs, 64 → 8
   4 → 2
2. minimum and maximum valid inputs, min - 0
   max = $2^{16}-1$ = 32767
3. invalid inputs, and -1 = 16'h ffff
   32768 = 16'h8000
4. randomly-chosen values. 3, 1297 . . .

Give examples of appropriate test inputs for each of the above categories if you were testing a circuit that computed the square root of a 16-bit signed number.

Exercise 2: What's the difference between:

`always @(x) y = '1;`, sets y to 1 whenever x changes

`wait(x) y='1;`, sets y to 1 when x is non-zero

and `@(x) y='1;`? waits until x changes and sets y to 1 (once)

Exercise 3: How could you:

(a) terminate the simulation if a test vector failed? $finish or $stop

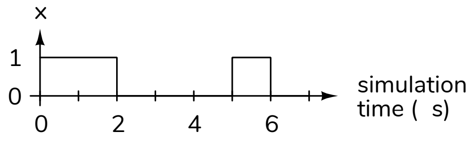
(b) change the clock frequency to 10 MHz? always #0.05us clk=~clk ;

(c) print each test vector as it's read? add $write() immediately after $fread()

(d) assert the reset input for two clock cycles?

reset = '1 ;
@(negedge clk) ;
@(negedge clk) ;
reset = '0 ;

Exercise 4:



x = '1 ;
#2us x = '0 ;
#3us x = '1 ;
#1us x = '0 ;

What statements could you use in an initial block to create this waveform on the signal **x**?