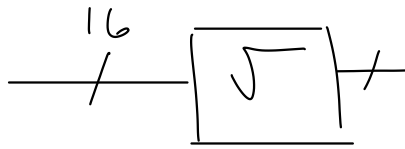


Simulation



Exercise 1:

1. typical inputs, 4, 8, 16
2. minimum and maximum valid inputs, 0 $2^{15} - 1 = 0x7fff$
3. invalid inputs, and -1
4. randomly-chosen values. (use `vand()`)

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.

wait for change in x

Exercise 2: What's the difference between `wait(x) y='1';` and `@(x) y='1';`?

wait for x to be true (non-zero)

X	wait(x)	@(x)
1 → 0	N	Y
0 → 1	Y	Y
X → 0	N	Y
1 → 1	Y	N

Exercise 3: How could you:

(a) terminate the simulation if a test vector failed?

Use `$stop` instead of / in addition to `$display`

(b) change the clock frequency to 10 MHz?

#0.05us (or #50ns)

(c) print each test vector as it's read? *use `$display` after `$scanf`*

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

*{ 1 0 0 } two test vectors lines
{ 1 0 0 } that assert reset*