**Exercise 1: ceil\_log2 function**

1. **VHDL code**

**LIBRARY** ieee;

**USE** ieee.std\_logic\_1164.**all**;

**USE** ieee.numeric\_std.**all**;

**use** ieee.math\_real.**all**;

**PACKAGE** lab4 **IS**

**FUNCTION** count\_leading\_zeros(*x* : UNSIGNED) **return** UNSIGNED;

**FUNCTION** ceil\_log2(*x* : UNSIGNED) **return** UNSIGNED;

**END** **PACKAGE** lab4;

**package** **body** lab4 **is**

**FUNCTION** count\_leading\_zeros(*x* : UNSIGNED) **return** UNSIGNED **IS**

**VARIABLE** count : UNSIGNED(*x*'*RANGE*) := to\_unsigned(0, *x*'*LENGTH*);

**BEGIN**

**FOR** i **IN** *x*'*RANGE* **LOOP**

**CASE** *x*(i) **IS**

**WHEN** *'0'* => count := count + 1;

**WHEN** **OTHERS** => **EXIT**;

**END** **CASE**;

**END** **LOOP**;

**RETURN** count;

**END** **FUNCTION** count\_leading\_zeros;

**FUNCTION** ceil\_log2(*x* : UNSIGNED) **return** UNSIGNED **IS**

-- as it turns out this can be done with a count leading zeros procedure which is

-- very close to our count leading ones procedure

**VARIABLE** temp : UNSIGNED(*x*'*RANGE*) := to\_unsigned(0, *x*'*LENGTH*);

**begin**

**IF** *x* = to\_unsigned(0, *x*'*LENGTH*) **THEN**

**return** to\_unsigned(0, *x*'*LENGTH*);

**ELSE**

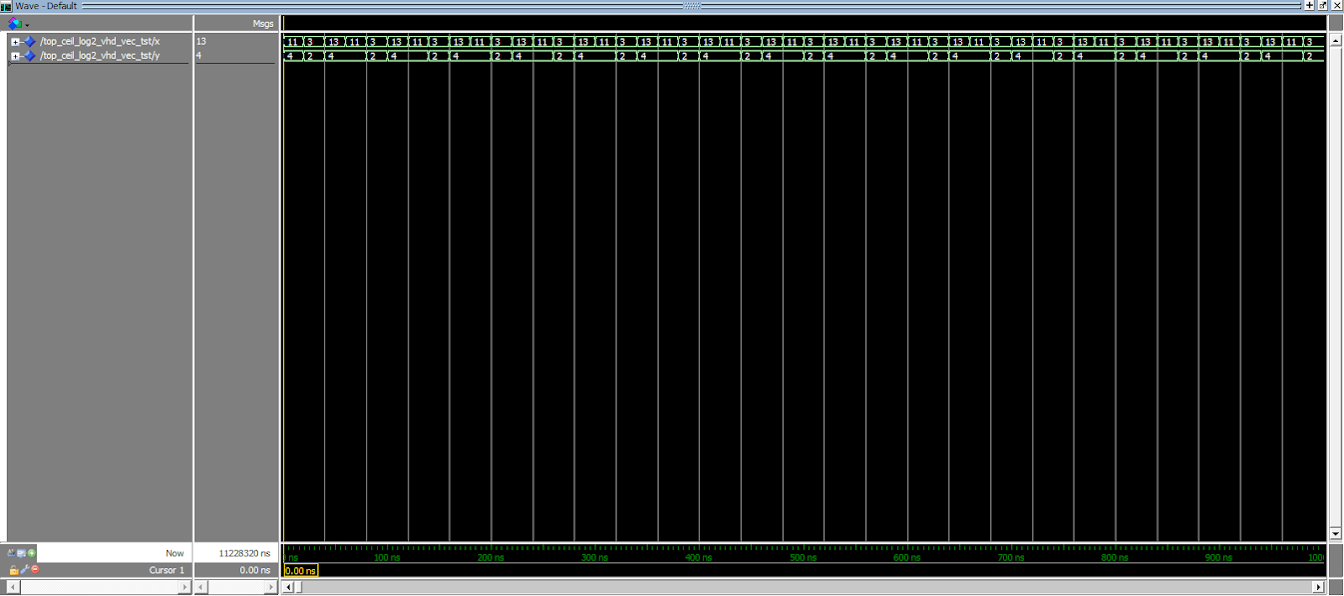
temp := count\_leading\_zeros(*x* - to\_unsigned(1, *x*'*LENGTH*));

**return** to\_unsigned(*x*'*HIGH*, *x*'*LENGTH*) - temp + to\_unsigned(1, *x*'*LENGTH*);

**END** **IF**;

**end** **FUNCTION** ceil\_log2;

**end** lab4;

1. 

**Exercise 2: natural\_to\_thermometer converter**

**LIBRARY** ieee;

**USE** ieee.std\_logic\_1164.**ALL**;

**USE** ieee.numeric\_std.**ALL**;

**USE** ieee.math\_real.**ALL**;

**PACKAGE** lab4 **IS**

**FUNCTION** natural\_to\_thermometer (*x*: UNSIGNED) **RETURN** BIT\_VECTOR;

**END** **PACKAGE** lab4;

**PACKAGE** **BODY** lab4 **IS**

**FUNCTION** natural\_to\_thermometer(*x*: UNSIGNED) **RETURN** BIT\_VECTOR **IS**

**VARIABLE** temp: BIT\_VECTOR((2\*\**x*'*LENGTH* - 1) **DOWNTO** 0) := (**OTHERS** => *'0'*);

**BEGIN**

**FOR** i **IN** 0 **TO** temp'*left* **LOOP**

**IF** i < *x* **THEN**

temp(i) := *'1'*;

**ELSE**

**EXIT**;

**END** **IF**;

**END** **LOOP**;

**RETURN** temp;

**END** **FUNCTION** natural\_to\_thermometer;

**END** **PACKAGE** **BODY** lab4;

**LIBRARY** ieee;

**USE** ieee.std\_logic\_1164.**ALL**;

**USE** ieee.numeric\_std.**ALL**;

**USE** ieee.math\_real.**ALL**;

**USE** work.lab4.**all**;

**ENTITY** top\_natural\_to\_thermometer **IS**

**GENERIC**(

-- number of input bits

*BITS* : INTEGER := 7

);

**PORT**(

-- input in stl format

x : **IN** STD\_LOGIC\_VECTOR(*BITS* - 1 **DOWNTO** 0) := (**OTHERS** => *'0'*);

--num of bits, take the log base 2, then convert to an int. we redefine this because vhdl sucks

y : **OUT** STD\_LOGIC\_VECTOR((2 \*\* *BITS*) - 1 **DOWNTO** 0) := (**OTHERS** => *'0'*)

);

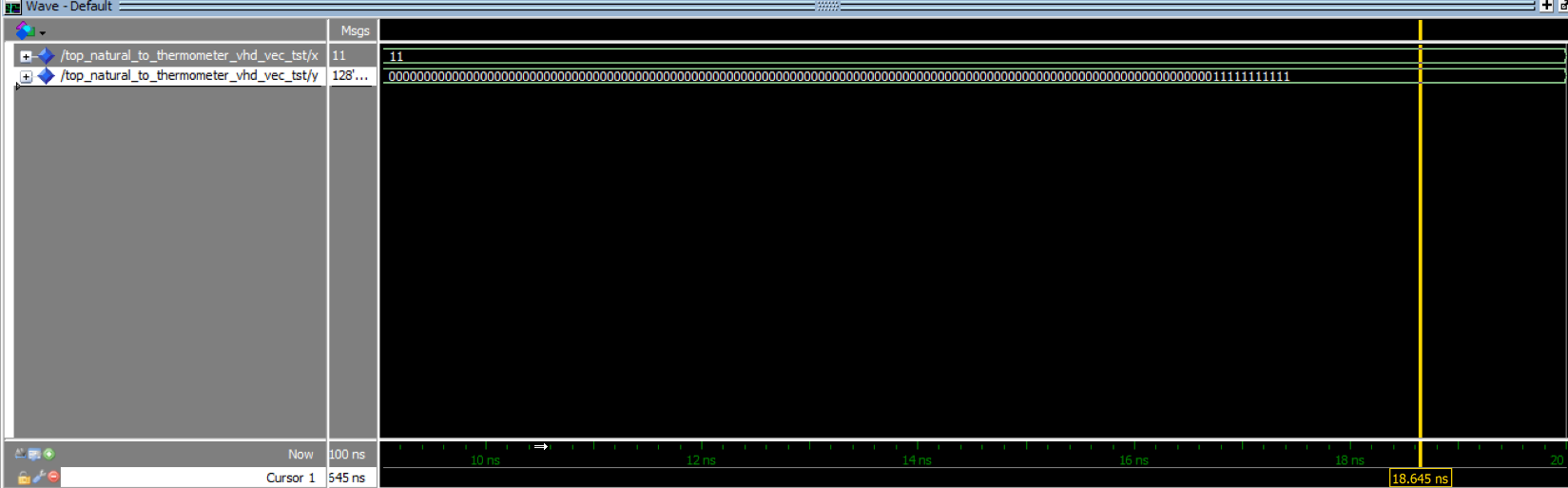
**END** **ENTITY** top\_natural\_to\_thermometer;

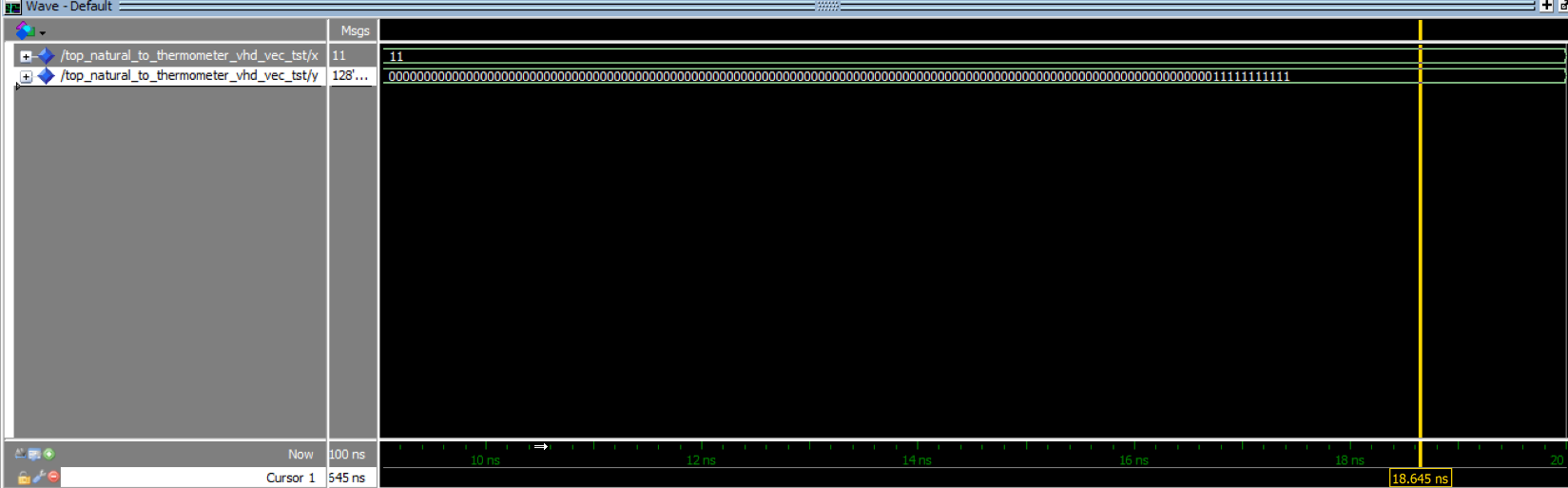
**ARCHITECTURE** arch **OF** top\_natural\_to\_thermometer **IS**

**BEGIN**

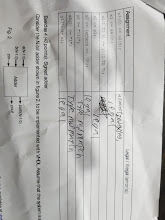
y <= to\_stdlogicvector(natural\_to\_thermometer(UNSIGNED(x)));

**END** **ARCHITECTURE** arch;





**Exercise 3: illegal assignments`**

****

**Exercise 4: Signed Adder**

b) VHDL code  
**library** ieee;

**use** ieee.std\_logic\_1164.**all**;

**use** ieee.numeric\_std.**all**;

**entity** adder **is**

**port**(

a, b, cin : **in** std\_logic;

sum, cout : **out** std\_logic

);

**end** **entity** adder;

**architecture** RTL **of** adder **is**

**begin**

sum <= a **XOR** b **XOR** cin;

cout <= (a **and** b) **or** (cin **and** a) **or** (cin **and** b);

**end** **architecture** RTL;

--------------------------------------------

**library** ieee;

**use** ieee.std\_logic\_1164.**all**;

**use** ieee.numeric\_std.**all**;

**use** work.**all**;

**entity** generic\_ripple **is**

**generic**(

*N* : integer := 8);

**port**(

a, b : **in** std\_logic\_vector(*N* - 1 **downto** 0);

cin : **in** std\_logic;

sum : **out** std\_logic\_vector(*N* - 1 **downto** 0);

cout : **out** std\_logic

);

**end** **entity** generic\_ripple;

**architecture** RTL **of** generic\_ripple **is**

**signal** temp : std\_logic\_vector(*N* **downto** 0);

**begin**

temp(0) <= cin;

cout <= temp(*N*);

signed\_adder : **for** i **in** 0 **to** *N* - 1 **generate**

full\_adder\_i : **entity** work.adder

**PORT** **MAP**(

temp(i), a(i), b(i), sum(i), temp(i + 1)

);

**end** **generate**;

**end** **architecture** RTL;

d) Simulation results (N = 8)

