Introduction to Digital Logic
EECS/CSE 31L

Homework 1
Combinational logic modeling

course instructor: Pooria M.Yaghini
prepared by: Kasra Moazemmi
Henry Samueli School of Engineering
University of California, Irvine

October 25, 2014

Due on Saturday 11/01/2014 , 11:00 pm

1 Data types (points 40)

s1 to s8 are VHDL signals. based on the assignments shown below, list which synthesizable predefined data types each of these signals can belong to.

A) s1 <= '0';
B) s2 <= 'Z';
C) s3 <= TRUE;
D) s4 <= "01000";
E) s5 <= "0100Z";
F) s6 <= ('0', '1', '0', '0', '0');
G) s7 <= (OTHERS =>'Z');
H) s8 <= 255;

2 Operations (points 30)

If a(7:0) = "00110011" and b(3:0) ="1111". Determine the result of following statements.

A) a(7 DOWNTO 4) NAND "0111"
B) a(7 DOWNTO 4) XOR NOT b
C) "1111" NOR b
D) b(2 DOWNTO 0) XNOR "101"
E) b SLL 2
F) a ROL 3

3 Generate (points 30)

In this question, we try to use GENERATE statement to construct a larger multiplexer using multiple instances of a basic 2x1MUX.

The first box shows the code for 2x1 MUX. The second box shows the code for larger MUX.

```
------The component (mux2x1)-------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
------------------------
ENTITY mux2x1 IS
 PORT (a, b, sel: IN STD_LOGIC;
            x: OUT STD_LOGIC);
END ENTITY;
------------------------
ARCHITECTURE mux2x1 OF mux2x1 IS
BEGIN
 x <= a WHEN sel='0' ELSE b;
END ARCHITECTURE;
------------------------
```

Try to fill the four missing parts(A,B,C and D) of code bellow to complete the design for 3-bit MUX(two 3-bit inputs and one 3-bit output).

```
------Main code-------
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
------------------------
ENTITY mux2x3 IS
 PORT (a, b: IN STD_LOGIC_VECTOR( 2 DOWNTO 0);
   sel : IN STD_LOGIC:
   x: OUT ???A???;
END ENTITY;
------------------------
ARCHITECTURE mux2x3 OF mux2x3 IS
 ------Component declaration------
 COMPONENT mux2x1 IS
  PORT (a, b, sel: IN STD_LOGIC;
       x:OUT STD_LOGIC):
 END ???B???;

BEGIN
 -------Component instantiation--------
 generate_mux2x: FOR i IN 0 TO 2 GENERATE
  comp: mux2x1 PORT MAP (a(i), b(i), ???C???, x(i));
 END ???D??? generate_mux2x3;
END ARCHITECTURE;
------------------------
```