Instructions

1. All questions should be answered on this test.

2. Raise your hand if the problem statements are not clear.

3. Students are expected to adhere to all provisions of the Virginia Tech Honor System. Sign the pledge below affirming that you have followed the honor code. By signing the pledge you are stating that you have not discussed any portion of this exam with any other student during the exam period, that you have not used notes or any other unauthorized material while taking the exam, that you have not copied any answers or portions of answers from another student and that you have not allowed anyone to copy yours.

Pledged: _____________________________
1. 
   a) (4 pts.) What does VHDL stand for?

   b) (4 pts) Name two types of delay used in signal assignment statements and describe each one.
   1.
   2.

   c) (3 pts) During the design cycle of going from a specification to a physical implementation, the design is represented at various levels. What are the three steps that take place in going from one level to the next?
   1.
   2.
   3.

   d) (6 pts.) What are two major differences between functions and procedures?

   e) (4 pts.) VHDL provides two mechanisms for sharing and re-using functions, procedures, data types, etc. What are they? (Hint: IEEE is an example of one of the mechanisms; std_logic_1164 is an example of the other.)
   1.
   2.

   f) (4 pts) A __________________________ is a VHDL model created to test another model; it instantiates the model to be tested and then applies inputs to it.
2. For each part, two architectures are given. Tell if the two architectures are equivalent, i.e. do they give exactly the same behavior for the entity? Briefly explain why or why not.

a) (5 pts.) (Assume that a, i0, and i1 are defined in the entity declaration as inputs of type std_logic and q is an output of type std_logic.)

```vhdl
architecture first of mux is
  signal s: std_logic;
begin
  process (a, i0, i1)
  begin
    s <= '0';
    if  (a = '1') then
      s <= '1';
    end if;
    case s is
    when '0' =>
      q <= i0;
    when '1' =>
      q <= i1;
    when others =>
      q <= 'X';
    end case;
  end process;
end first;
```

```vhdl
architecture second of mux is
begin
  process (a, i0, i1)
  variable s: std_logic;
  begin
    s := '0';
    if  (a = '1') then
      s := '1';
    end if;
    case s is
    when '0' =>
      q <= i0;
    when '1' =>
      q <= i1;
    when others =>
      q <= 'X';
    end case;
  end process;
end second;
```

b) (5 pts.) Assume that a, b, c, d are defined in the entity to be inputs of type bit and that z is defined to be an output of type bit.

```vhdl
architecture first of q2b is
begin
  z <= a after 5 ns when b = '1'
  else c after 5 ns when d = '0'
  else '0';
end first;
```

```vhdl
architecture second of q2b is
begin
  process(b, d)
  begin
    if (b = '1') then
      z <= a after 5 ns;
    elsif (d = '0') then
      z <= c after 5 ns;
    else
      z <= '0';
    end if;
  end process;
end second;
```
3. (20 pts.) Create a function named all_same for the err_cond type shown below. The input to the function should be of type err_cond_vector. The function should return a '1' when all of elements of the vector are '1', a '0' when all of them are a '0', and an 'E' indicating an error condition when they are not all '1' or all '0'. You may create other type definitions if you need them.

```vhd
type err_cond is ('0', '1', 'E');
type err_cond_vector is array (natural range <>) of err_cond;
```
4. (15 pts.) Given the entity and architecture below, briefly describe what happens if the simulation is run for 100 ns. What report messages, if any, are printed by the simulator? For any message that is printed, at what simulation time is it printed?

entity f02_q4 is
end f02_q4;

architecture oh_behave of f02_q4 is
signal sample: bit;
begin
process
begin
sample <= '1';
wait for 20 ns;
assert (sample = '0')
report "top process"
severity note;
wait;
end process;

process
begin
wait for 10 ns;
wait until sample = '1';
assert (sample = '0')
report "bottom process"
severity note;
wait;
end process;
end oh_behave;
5. (10 pts.) Consider the following type and signal declarations:

```vhdl
type car is (chevy, ford, buick, honda, toyota, rambler);
type car_vector is array (natural range <>) of car;
type car_2d is array (car, car'high downto car'low) of positive;
subtype subcar is car range car'val(2) downto car'left;

signal t1: car_2d;
signal t2: car_vector(2 downto 0) := (rambler, honda, chevy);
signal t3: subcar;
```

For each of the expressions below, indicate whether or not it is a valid expression, i.e. whether or not it could appear on the right-hand side of an assignment to a signal or variable of some type. If it is a valid expression, give its value at time \( t = 0 \).

a) car_2d'left

b) t2'left

c) t2'low

d) car'pos(1)

e) t1(honda, chevy)

f) car_2d'right(2)

g) t1(chevy, chevy) >= t2(1)

h) car_vector'high

i) subcar'left

j) subcar'high
6. (20 pts.) Write an architecture for the combinational circuit below. The inverter has a delay of 2 ns, while the other gates have a delay of 5 ns. The entity declaration has been given for you. You may use any VHDL constructs that you want and any style of architecture. Your code should be efficient and easily understandable.

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

entity   gates is
  port (a, b, c, d: in std_logic;
        z: out std_logic);
end gates;
```

```vhdl
library IEEE;
use IEEE.std_logic_1164.all;

description of inverter delay

entity   gates is
  port (a, b, c, d: in std_logic;
        z: out std_logic);
end gates;
```