UNIVERSITY OF VICTORIA

MIDTERM EXAMINATIONS JUNE 2005

DIGITAL VLSI SYSTEMS

CENG 465/ELEC 543

NAME: STUDENT NO.:

INSTRUCTOR: Dr. Fayez Gebali, P.Eng. SECTION: K01
DURATION: 50 MINUTES

TO BE ANSWERED ON THE PAPER
CLOSED BOOK EXAMINATION

STUDENTS MUST COUNT THE NUMBER OF PAGES IN THIS EXAMINATION PAPER BEFORE BEGINNING TO WRITE AND REPORT ANY DISCREPANCY TO THE INVIGILATOR.

THIS QUESTION PAPER HAS 9 PAGES.

Question	1	2	3	4	5	**Total**
Maximum Mark	20	24	16	15	15	90
Earned Marks						

Question 1 [20 Marks] Tue or false? **True False**

1. The execution of a VHDL process starts from the first statement below `BEGIN` and ends when the last statement before `END` is finished. ☐ FALSE
2. Signals in a VHDL process are assigned values only when a process suspends. TRUE ☐
3. Only signals can be used as information carriers between VHDL processes TRUE ☐
4. Inputs to a system are called *generics* and outputs are called *ports*. ☐ FALSE
5. Ports are signals. TRUE ☐
6. Each system specified in VHDL is a composition of an entity and an architecture. TRUE ☐
7. Structural specification states explicitly the functionality of an entity. ☐ FALSE
8. A package is a compact way of writing an entity-architecture pair. ☐ FALSE
9. All VHDL processes in architecture are active all the time when the architecture is active. ☐ FALSE
10. Moore's law implies less off-chip communications delay. ☐ FALSE
11. Declaration of an internal signal contains the name of the signal, its mode and type. ☐ FALSE
12. All signals declared in an entity are visible in all architectures assigned to this entity. TRUE ☐
13. A port must be either input or output ☐ FALSE
14. Variables can be declared in a procedure declarative part. TRUE ☐
15. Signals can be declared in a procedure declarative part. ☐ FALSE
16. It is possible to have more than one procedure or function with the same name. TRUE ☐
17. A procedure could have an empty parameter list. TRUE ☐
18. A procedure could be invoked outside a `PROCESS`. TRUE ☐
19. The `NEXT` statement is used in a `LOOP` to exit the loop iterations. ☐ FALSE
20. Assume the type *word* is an array of reals. The following aggregate is legal
    `VARIABLE x: word := (1.6, 2=: 2.0, OTHERS => 0.0);` ☐ FALSE

Question 2 [24 Marks] Assume the following VHDL type definitions

```
TYPE resistance IS RANGE 0 TO 1E6
    UNITS
        ohm;
    END UNITS resitance;
TYPE index IS RANGE 21 DOWNTO 11;
TYPE logic IS (unknown,low,undriven,high);
TYPE a IS ARRAY (1 TO 4, 31 DOWNTO 0) OF bit;
```

Find the attributes values indicated below

Attribute	**Value**	**Attribute**	**Value**
resistance'left	0 ohm	resistance'right	1E6 ohm
resistance'high	1E6 ohm	resistance'low	0 ohm
resistance'ascending	true		
index'left	21	index'right	11
index'low	11	index'high	21
logic'left	unknown	logic'low	unknown
logic'high	high	logic'right	high
logic'ascending	true		
a'left(1)	1	a'right(2)	0
a'low(1)	1	a'high(2)	31
a'range(1)	1 TO 4	a'reverse_range(2)	0 TO 31
a'length(1)	4	a'length(2)	32
a'ascending(1)	true	a'ascending(2)	false

Question 3 [16 Marks] Draw the output waveforms for the the signals $x1$, $y1$, $x2$ and $y2$ assuming the inputs vary as shown below.

```
LIBRARY ieee; USE IEEE.std_logic_1164.ALL;
ENTITY alu_1 IS
    PORT(
        a,b,c:  IN std_logic;
        x1,y1:  OUT std_logic);
END ENTITY alu_1;
ARCHITECTURE alu_1_arch OF alu_1 IS
    SIGNAL d:  std_logic;
BEGIN
    PROCESS (a,b,c) IS
    BEGIN
        d <= a;
        x1 <= c XOR d;
        d <= b;
        y1 <= c XOR d;
    END PROCESS;
END;
```

```
LIBRARY ieee; USE IEEE.std_logic_1164.ALL;
ENTITY alu_2 IS
    PORT(
        a,b,c:  IN std_logic;
        x2,y2:  OUT std_logic);
END ENTITY alu_2;
ARCHITECTURE alu_2_arch OF alu2 IS
BEGIN
    PROCESS (a,b,c) IS
        VARIABLE d:  std_logic;
    BEGIN
        d := a;
        x2 <= c XOR d;
        d := b;
        y2 <= c XOR d;
    END PROCESS;
END;
```

time (ns)	0	1	2	3
a	0	0	1	1
b	0	1	0	1
c	1	1	0	0
x1	U	1	1	0
y1	U	1	1	0
x2	1	1	1	1
y2	1	0	0	1

Question 4

4(a) [10 Marks] Write VHDL entity and architecture for an edge-triggered flip-flop and a two-input multiplexer having the following specifications:

Component	Inputs	Input Type	Output	Output Type
D-FF	d, clk	std_logic	q	std_logic
Mux	a, b, s	std_logic	c	std_logic

The two components must follow these specifications:

1. Use the RTL design style and the **selective** signal assignment statement for the mux.
2. Use the RTL design style and the **conditional** signal assignment statement for the D-FF.
3. Use the `rising_edge` function to detect the rising edge of the clock input for the D-FF.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY d_ff IS
    PORT(
        d,clk:  IN std_logic;
        q:  OUT std_logic);
END ENTITY d_ff;
ARCHITECTURE d_ff OF d_ff IS
BEGIN

    q <= d WHEN rising_edge(clk);

END ARCHITECTURE d_ff;
```

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY mux IS
    PORT(
        a,b,s:  IN std_logic;
        c:  OUT std_logic);
END ENTITY mux;
ARCHITECTURE mux OF mux IS
BEGIN

    WITH s SELECT
        c <= a WHEN '0',
            b WHEN '1',
            'U' WHEN OTHERS;

END ARCHITECTURE mux;
```

Question 4 Cont.
4(b) [5 Marks] Write VHDL entity and architecture for a circuit that uses the two components defined in question 4(a) and has the following I/O specifications

Signal	Mode	Type	Comment
d, clk	IN	std_logic	D-FF inputs
a, b, s	IN	std_logic	Mux inputs
q	OUT	std_logic	D-FF output
c	OUT	std_logic	Mux output

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY test IS
    PORT (
        d,clk,a,b,s:  IN std_logic;
        q,c:  OUT std_logic);
END ENTITY test;

ARCHITECTURE test OF test IS

    COMPONENT d_ff IS
        PORT(
            d,clk:  IN std_logic;
            q:  OUT std_logic);
    END COMPONENT;

    COMPONENT mux IS
        PORT(
            a,b,s:  IN std_logic;
            c:  OUT std_logic);
    END COMPONENT;

    --Configure which architecture
    FOR d_ff_instance:  d_ff
        USE ENTITY work.d_ff (d_ff);
    FOR mux_instance:  mux
        USE ENTITY work.mux (mux);

BEGIN
    d_ff_instance:  d_ff
        PORT MAP (d,clk,q);
    mux_instance:  mux
        PORT MAP (a,b,s,c);

END ARCHITECTURE test;
```

Question 5 [15 Marks] Your task is to write VHDL code for a 4-bit positive edge-triggered register entity. The register description is based on a procedure describing the operation of a register as was discussed in the lectures. This procedure should be declared in a package. The procedure has the following signals:

signal	**Mode**	**Type**
d	IN	std_logic_vector(3 DOWNTO 0)
clk	IN	std_logic
q	OUT	std_logic_vector(3 DOWNTO 0)

5(a) [5 Marks] Declare a procedure `dff` in the package `package_procedures`.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
PACKAGE package_procedure IS -- package declaration
    PROCEDURE dff (
        SIGNAL d:  IN std_logic_vector(3 DOWNTO 0);
        SIGNAL clk:  IN std_logic;
        SIGNAL q:  OUT std_logic_vector(3 DOWNTO 0));
END PACKAGE package_procedure;
```

Question 5 Cont.
5(b) [5 Marks] Define the functionality of the procedure in the package body.

```
PACKAGE BODY packaff_procedure IS -- package body
    PROCEDURE dff (
        SIGNAL d:  IN std_logic_vector(3 DOWNTO 0);
        SIGNAL clk:  IN std_logic;
        SIGNAL q:  OUT std_logic_vector(3 DOWNTO 0)) IS
    BEGIN

        IF rising_edge(clk) THEN
            q <= d;
        END IF;

    END PROCEDURE dff;
END PACKAGE BODY package_procedure;
```

Question 5 Cont.
5(c) [5 Marks] Define an entity `reg` that describing the 4-bit register and uses the procedure in questions 5(a) and 5(b) in its architecture as a concurrent statement.

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE work.package_procedure.ALL;
ENTITY dff_procedure IS
    PORT (
        d:  IN std_logic_vector(3 DOWNTO 0);
        clk:  IN std_logic;
        q:  OUT std_logic_vector(3 DOWNTO 0));
END ENTITY procedure;
ARCHITECTURE rtl OF procedure IS
BEGIN

    dff(
        d=>d, clk=>clk, q=>q);

END ARCHITECTURE rtl;
```

END