

**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.

<b>Question</b>	1	2	3	4	5	<b>Total</b>
<b>Maximum Mark</b>	20	24	16	15	15	90
<b>Earned Marks</b>						

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

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

```

TYPE resistance IS RANGE 0 TO 1E6
  UNITS
    ohm;
  END UNITS resistance;
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		resistance'right	
resistance'high		resistance'low	
resistance'ascending			
index'left		index'right	
index'low		index'high	
logic'left		logic'low	
logic'high		logic'right	
logic'ascending			
a'left(1)		a'right(2)	
a'low(1)		a'high(2)	
a'range(1)		a'reverse_range(2)	
a'length(1)		a'length(2)	
a'ascending(1)		a'ascending(2)	

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

<pre> 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 &lt;= a;     x1 &lt;= c XOR d;     d &lt;= b;     y1 &lt;= c XOR d;   END PROCESS; END; </pre>	<pre> 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 alu_2 IS BEGIN   PROCESS (a,b,c) IS     VARIABLE d: std_logic;   BEGIN     d := a;     x2 &lt;= c XOR d;     d := b;     y2 &lt;= c XOR d;   END PROCESS; END; </pre>
---	---

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

**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.

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

**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:

<b>signal</b>	<b>Mode</b>	<b>Type</b>
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`.

Question 5 Cont.

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



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.

END