Ollscoil Chathair
Bhaile Átha Cliath
Dublin City University

AUGUST/RESIT EXAMINATIONS 2021/2022

MODULE: EE496 - Computer Architecture and HDL

PROGRAMME(S):

ECE	BEng Electronic & Computer Engineering
ECSAO	Study Abroad (Engineering & Computing)

YEAR OF STUDY: 4,O

EXAMINER(S):

Xiaojun Wang	(Internal)	(Ext:5808)
Dr. Josep R. Casas	(External)	External

TIME ALLOWED: 2 Hours

INSTRUCTIONS: **Answer 4 questions. All questions carry equal marks.**

PLEASE DO NOT TURN OVER THIS PAGE UNTIL YOU ARE INSTRUCTED TO DO SO.
The use of programmable or text storing calculators is expressly forbidden.
Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

There are no additional requirements for this paper.

QUESTION 1 ***[TOTAL MARKS: 25]***

Q 1(a) **[8 Marks]**

Briefly describe the hardware description languages `VHDL`, `Verilog` and `SystemVerilog`, and give a brief comparison between them.

Q 1(b) **[5 Marks]**

Briefly discuss the main differences between `variables` and `signals` in VHDL.

Q 1(c) **[7 Marks]**

What is the difference between a `variable` and a shared variable in VHDL? Using `shared variables` can cause what problem to a VHDL model? Explain the problem with an example.

Q 1(d) **[5 Marks]**

Write a VHDL fragment that extends a 16-bit **signed** type signal `a(15 downto 0)` to 32 bits and assigns it to a 32-bit **signed** type signal `y(31 downto 0)`, maintaining the original decimal value of `a`.

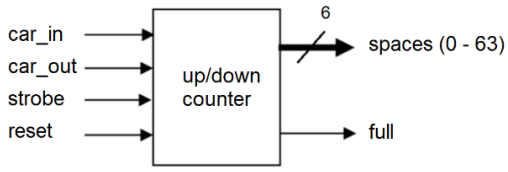
[End of Question 1]

QUESTION 2 ***[TOTAL MARKS: 25]***

An engineer uses VHDL to design a controller displaying the available parking spaces for a car park.

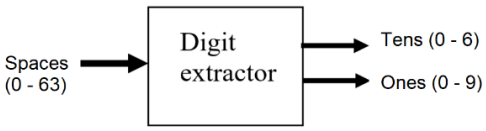
Q 2(a) **[10 Marks]**

The car park has **63** parking spaces (i.e. it can be represented with 6-bits). The output signal `spaces` indicate the currently available parking spaces. The vacant parking spaces are decreased by one when a car drives into the car park and are increased by one when a car leaves. Suppose a positive pulse is generated externally on the `strobe` input signal whenever a vehicle goes into or leaves the car park. The `reset` signal is active low, which resets spaces to the initially 63 available parking spaces. The controller sets the output signal `full` to logic '1' when the car park is full. Write a synthesisable RTL VHDL model for the counter block shown below.



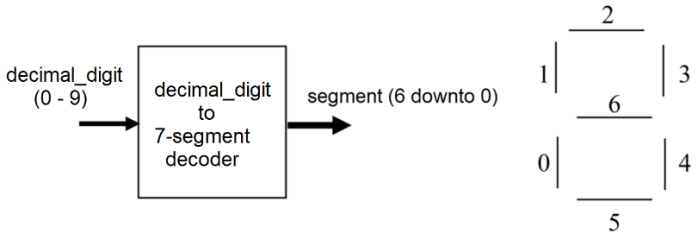
Q 2(b) **[10 Marks]**

It is necessary to extract the digits for tens and ones from the number of available parking spaces so that the four digits can each be displayed using a 7-segment display. Write an RTL VHDL model for this digit extract block shown below.



Q 2(c) **[5 Marks]**

A decimal digit to a 7-segment decoder converts a decimal digit to an appropriate code selecting the display segments shown below. A segment illuminates when its corresponding bit is at logic level `'0'`; otherwise, it is off. Wite an RTL model for the decoder shown below.



[End of Question 2]

QUESTION 3 ***[TOTAL MARKS: 25]***

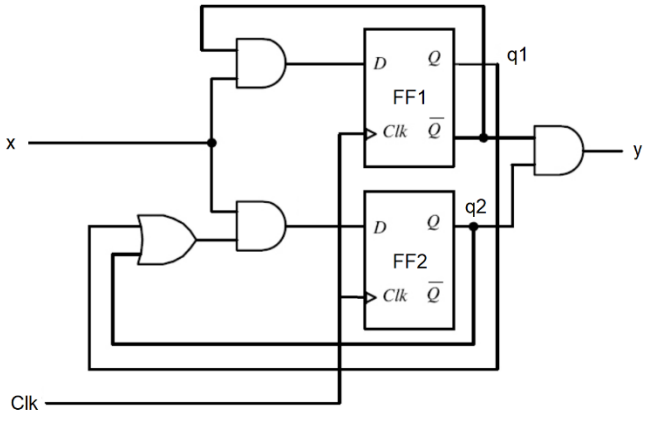
Q 3(a) **[15 Marks]**

Draw a logic circuit inferred by the following VHDL model and explain why to create each circuit component.

```
library ieee;
use ieee.std_logic_1164.all;
entity var_sig is
   port (a   : in  std_logic;
         b   : in  std_logic;
         c   : in  std_logic;
         en  : in  std_logic;
         oe  : in  std_logic;
         clk : in  std_logic;
         rst : in  std_logic;
         d   : out std_logic
        );
end var_sig;

architecture rtl of var_sig is
  signal s1 : std_logic;
  signal s2 : std_logic;
  signal s3 : std_logic;
begin
  p1: process (clk, rst)
        variable v1 : std_logic;
      begin
         if rst = '0' then
            v1 := '0';
            s1  <= '0';
         elsif rising_edge(clk) then
            v1 := a nand b;
            s1 <= v1 or c;
         end if;
      end process p1;

  p2: s2 <= s1 when en = '1';

  p3: s3 <= s2 when rising_edge(clk);

  p4: d <= s3 when oe = '1' else 'Z';
end rtl;
```

Q 3(b) **[10 Marks]**

Write a VHDL description for the following logic circuit. The inputs to the model are `x` and the clock signal `clk`. The model output is `y`.



[End of Question 3]

QUESTION 4 ***[TOTAL MARKS: 25]***

Q 4(a) **[10 Marks]**

Below is the state table of a Finite State Machine (FSM). The FSM has a data input signal `x` and a clock input signal `clk`. It also has an active-low asynchronous reset signal `rst`, when at logic '0' resets the FSM to its initial state A. The output signal is `y`. Write a synthesisable VHDL description of the FSM.

State Table

Current state	Input x = 0		Input x = 1	
	Next state	Output y	Next state	Output y
A	C	1	D	0
B	B	1	A	0
C	A	0	B	1
D	B	1	C	1

Q 4(b) **[5 Marks]**

Is the state machine represented by the above State Table a Mealy or a Moore machine? Why?

Q 4(c) **[5 Marks]**

Briefly describe the *one-hot encoding* scheme in an FSM design and explain its advantages when implementing the FSM with an FPGA. Then, give the one-hot encoding of the states in Q4(a).

Q 4(d) **[5 Marks]**

Briefly describe the functionality of a VHDL testbench and how to construct one.

[End of Question 4]

QUESTION 5 ***[TOTAL MARKS: 25]***

Q 5(a) **[5 Marks]**

Assume an 8-bit `signed` (i.e. two's complement) number `N` has a `hexadecimal` value `X"83"`. What is the equivalent `decimal` value of `N`? What is the hexadecimal representation of `N` when it is extended to 16-bit while maintaining the same equivalent decimal value?

Q 5(b) **[5 Marks]**

Assume `M` is a 16-bit `signed` (i.e. two's complement) constant with `hexadecimal` value `X"FE0C"`. Assume `K` is an 8-bit `signed` variable. Answer the following questions after executing the following statement.

```
K := resize(M, K'length);
```

i) What is the hexadecimal value of `K`?
ii) What is the equivalent decimal value of `K`?
iii) Has `K` maintained the same value as that of `M`?

Q 5(c) **[5 Marks]**

Assuming the datapath of a multi-cycle computer has five pipeline stages.

1. Hardware to support an instruction ***fetch***
2. Hardware to support an instruction ***decode***
3. Hardware to support instruction ***execution*** (i.e. the ALU)
4. Hardware to support a ***memory*** read or write
5. Hardware to support the ***write-back*** of the ALU result to the register file

Assume that each of the above pipeline stages takes the amount of time specified in the table below.

Fetch	Decode	Execute	Memory	Write Back
200 ps	120 ps	150 ps	200 ps	50 ps

What would the ***clock period*** be for the *entire datapath*?

Q 5(d) **[5 Marks]**

For the *pipelined datapath* in Q5(c), assuming no hazards or stalls, how long does it take to execute ***one*** instruction, or what is the instruction **latency**?

Q 5(e) **[5 Marks]**

What is the **throughput** of the pipelined datapath in Q5(c)?

[End of Question 5]

[END OF EXAM]