UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Course:	**INF3430/4431**
Date:	**3. December 2018**
Time:	**09.00-13.00**
The exam consists of 6 pages.	
Appendix:	**1**
Allowed aids:	**All printed and written including calculator.**

Please verify that the exam document is complete before you start.

The exam consists for multiple choice assignments 1-3, 6 and 8-9. These shall be answered on the appendix sheet. Assignments 4-5, 7 and 10 shall be answered on standard sheets.

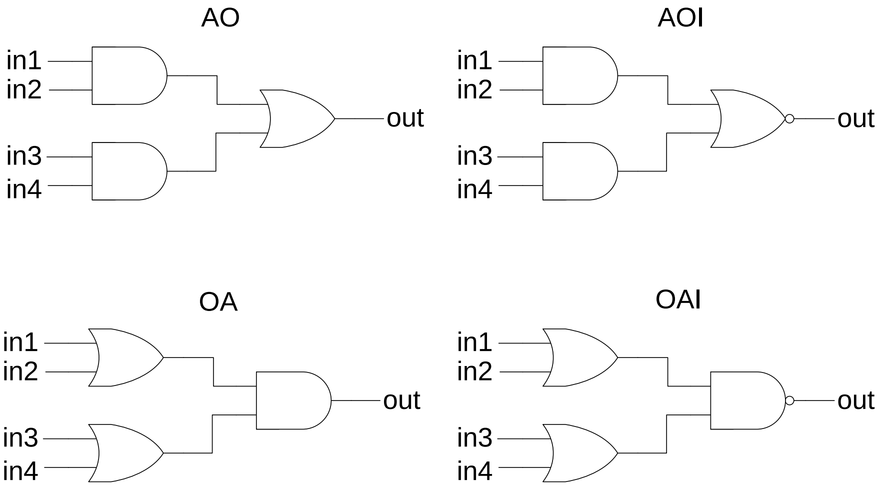
Each assignment has a score noted by the assignment number.

It is not necessary to repeat VHDL source code given in the assignments.

Multiple choice assignments:
Each assignment consists of a topic and some assertions given with a capital letter.
The assignments are answered by clearly marking X in the correct column for correct answer (i.e. that an assertion is true) on the appendix sheet. There is always at least one true assertion. To achieve maximum score, you shall mark each true assertion. 1 point is given for each correct answer. -1 point is given for each incorrect answer. You may use the left column in the text as a draft.
Appendix 1 with your candidate number is your answer.

Assignment 1 (3 %)

The figure below depicts the combinatorial circuits and-or (AO), and-or-invert (AOI), or-and (OA) and or-and-invert (OAI).



A 4-input Xilinx LUT with the content **"F888"** (hex) implements a:	A	and-or (AO)	
	B	and-or-invert (AOI)	
	C	or-and (OA)	
	D	or-and-invert (OAI)	

Assignment 2 (3 %)

FPGA technology	A	MicroBlaze is a soft processor core.	
	B	AXI4 is a shared processor bus.	
	C	AXI4Lite has independent read and write channels.	
	D	After some time with meta stability flip flops will always return to a steady state with the value '1'.	

Assignment 3 (3 %)

FPGA technology	A	In a combinational process, all signals that are read and assigned a value in the process shall be present in the sensitivity list.	
	B	In a sequential process with asynchronous reset, only reset and the clock signal shall be present in the sensitivity list.	
	C	SRAM-based FPGAs can be reprogrammed after the customer has received the product.	
	D	ASIC's can be reprogrammed	

Assignment 4 (6 %)

In this assignment a the module *compute_comb* which computes *a* multiplied with *b* added with *c* multiplied with *d*, in other words,. *result= (a*b)* + *(c*d)* shall be implemented. The entity compute_comb is presented below. The computation shall be using unsigned arithmetic operation on the operands a, b, c and d.

The output *result* shall have the required number of bits such that the result is always correct, and overflows do not occur. Change the signal *result* in the *entity compute_comb* such that the declaration of std_logic_vector has the correct number of bits.

Implement the architecture to the module *compute_comb* as listed under as a combinational process in synthesizable VHDL.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity compute_comb is
  port
     (a      : in  std_logic_vector(15 downto 0);
      b      : in  std_logic_vector(15 downto 0);
      c      : in  std_logic_vector(15 downto 0);
      d      : in  std_logic_vector(15 downto 0);
      result : out std_logic_vector(?? downto 0)); -- Change “??”
end entity compute_comb;

architecture rtl of compute_comb is
begin

  < Implement the compute_comb process >

end architecture rtl;
```

Assignment 5 (15 %)

The module *compute_comb* in assignment 4 shall be modified such that it is a synchronous module *compute_seq* with the reset signal *rst* and clock signal *clk*. The module shall compute result= (a*b) + (c*d). The inputs a, b, c and d are synchronous to the clock signal clk. The output *result* shall also now have the required number of bits such that overflows do not occur. The entity to compute_seq is listed below.

The signal *vdata* determines when the inputs *a, b, c* and *d* has valid data, i.e.. *vdata='*1*' when a,b,c* and *d* are valid. The output signal *vresult* shal be ‘0’ when the output *result* is not valid and ‘1’ when the output signal *result* is valid. When the output signal *result* is not valid the value shall be ‘0’.

The implementation does not meet timing closure with the selected technology and the required clock frequency. Therefore, the design has to be pipelined. Multiplication and addition has to be performed on different clock periods in order to meet timing closure. Multiple multiplications and additions can be executed in parallel in each clock period.

Implement the module compute_seq as listed under as a sequential process in synthesizable VHDL..

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity compute_seq is
  port (
    rst     : in  std_logic;
    clk     : in  std_logic;
    a       : in  std_logic_vector(15 downto 0);
    b       : in  std_logic_vector(15 downto 0);
    c       : in  std_logic_vector(15 downto 0);
    d       : in  std_logic_vector(15 downto 0);
    vdata   : in  std_logic;
    result  : out std_logic_vector(?? downto 0);
    vresult : out std_logic);
end entity compute_seq;

architecture rtl of compute_seq is

  < Implement the missing VHDL code>

end architecture rtl;
```

Assignment 6 (4 %)

How many registers (flip flops) does the implemented pipelined architecture rtl in Assignment 5 use.	A	96 registers	
	B	97 registers	
	C	98 registers	
	D	99 registers	

Assignment 7 (15 %)

The VHDL code listed below contains a *package mypack*, a *package somesubprograms*, the unfinished *function numbermonthdays* and the unfinished *prodecure days*.

a) (6 %)

Implement the unfinished *function numbermonthdays* in *package body* such that it returns the correct number of days in a month. For leap years February has 29 days otherwise it has 28 days. January, March, May, July, August, October, December has 31 days. The other months has 30 days. Note that month_type is declared in package mypack thus the datatype is defined..

b) (9 %)

Implement the unfinished *procedure days* which shall use *function numbermonthdays* to compute the number of days in a month. It shall also compute the number of days in a quarter. Furthermore, it shall compute the number of days in the quarter that the month belongs to. For example, February consists of 29 days for leap years. In that quarter, January, February, March contains 90 days (31+29+31=90).

```
package mypack is
 type month_type is (JAN, FEB, MAR, APR, MAY, JUN,
                     JUL, AUG, SEP, OCT, NOV, DEC);
end mypack;


library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.mypack.all;


package somesubprograms is

  function numbermonthdays (leap_year : boolean;
                            month     : month_type)
    return unsigned;


  procedure days (leap_year   : in  boolean;
                  month       : in  month_type;
                  monthdays   : out unsigned(4 downto 0);
                  quarterdays : out unsigned(6 downto 0));

end package;


package body somesubprograms is

  < Implement the function numbermonthdays >

  < Implement the procedure days >

end package body;
```

Assignment 8 (3 %)

Which data type are parameters declared as **in mode** in the procedure days in Assignment 7?	A	constant	
	B	variable	
	C	signal	

Assignment 9 (3 %)

Which data type are parameters declared as **out mode** in procedure days in Assignment 7?	A	constant	
	B	variable	
	C	signal	

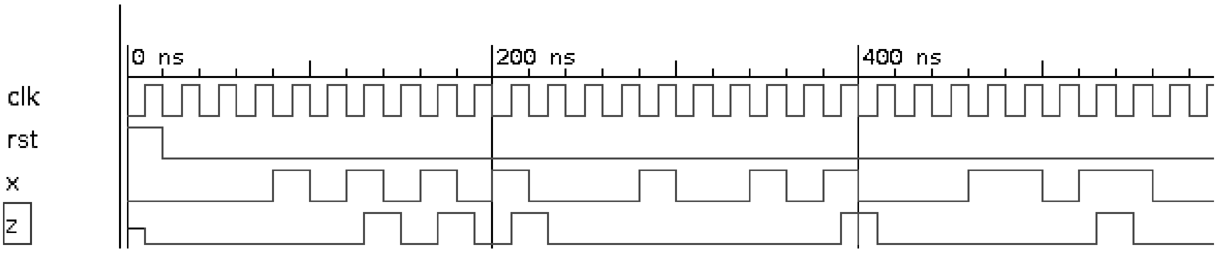
Assignment 10 (45 %)

In this assignment a Mealy type finite state machine shall be implemented. The state machine is called a 101-detector.

The state machine works as follows:
The output signal z is high when the 101-pattern is detected on the input signal x.

The input signal x is synchronous to the clock signal *clk*.

The timing diagram depicted below shows an example of input and output values of the state machine:



a) Score 15 %

Draw an ASM-diagram which depicts the finite state machine described above.

b) Score 15 %

Implement the finite state machine from assignment **11a)** as a two-process state machine in synthesizable VHDL.

The signals shall be of type std_logic, and the implementation shall use synchronous reset.

c) Score 15 %

Implement a VHDL test bench. The test bench shall not be self-checking.

Use 0x55 («01010101»), 0x4a («01001010») and 0x6c («01101100») as input values to x, where MSB bit 7 is clocked into the 101-detector first, and the LSB bit 0 last. This pattern is depicted in the timing diagram above..

Appendix 1.
INF3430/INF4431. Result for candidate number: _________

Assignment	A	B	C	D
1				
2				
3				
6				■
8				
9				