OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Summer Examination 2014

CS2502: Logic Design

Professor Ian Gent
Professor Barry O'Sullivan
Dr Frank Böhme

Answer all questions.

Total marks: **80**

90 minutes

PLEASE DO NOT TURN THIS PAGE UNTIL INSTRUCTED TO DO SO.
ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER.

1. Recall the two canonical forms for switching functions, i.e. the *canonical sum of products* (CSOP) and the *canonical product of sums* (CPOS).

   These questions deal with general properties of these two forms and their components. **16 marks**

   a) Consider a switching function of three variables given by its CSOP as follows:

   $$Y = f(A, B, C) = m_2 + m_6 + m_7$$

   Give the CPOS form of f. Do *not use short notation* as above! Your answer must not contain notations like M_3. (*6 marks*)

   b) Consider some switching function g about which only the following is known: Its notation in CPOS form is shorter than its notation in CSOP form (i.e. the CPOS contains fewer literals).
   Which general conclusion about g can be made from the assertion above?
   Hint: Consider the truth table of g. (*5 marks*)

   c) Recall that *minterms* as well as *simplified products* are both *AND-combinations* of direct or inverted input variables.
   Explain the general difference between these two expressions. (*5 marks*)

2. Recall that *majority decoders* are a group of switching functions where the output Y always assumes the *predominating input value*.

   These questions deal with general properties of these functions. **16 marks**

   a) How many arguments do these functions have? Describe the constraints (if any) to the number of inputs of a majority decoder. (*5 marks*)

   b) Consider the two canonical representations (CSOP and CPOS) of these switching functions. Should either of these two forms be preferred because it has a shorter notation than the other?
   Justify your answer. (*5 marks*)

   c) Consider the two *simplified* representations of these functions, i.e. the SSOP form and the SPOS form. Will these simplified forms always have a shorter notation than the corresponding canonical forms (CSOP and CPOS)?
   Answer with *Yes* or *No* and justify your answer. (*6 marks*)

3. Consider a combinational circuit with 4 inputs A, B, C, D and one output Y. The output Y shall assume 1 if the 4bit number formed by $ABCD$ is *prime*. Y shall be 0 otherwise.

   As usual, we understand $ABCD$ as an unsigned integer in the range $0 \ldots 15$ with D being the *least significant bit*.

   For example, the bit pattern $ABCD = 1011$ corresponds to the decimal number 11 which means $Y = 1$ since 11 is a prime number. The bit pattern $ABCD = 1110$ corresponds to decimal 14 which means $Y = 0$ since 14 is not a prime number.

   Please recall that the *smallest prime number* is 2.

   The following questions deal with various aspects of the switching function $Y = f(A, B, C, D)$. **27 marks**

   a) Use a Karnaugh-map to to find the *simplified sum of products* (SSOP) of f. (*6 marks*)

   b) From now on, we assume that the number formed by $ABCD$ is always in the range $0 \ldots 9$. (The same constraint occurs in BCD-to-7-segment decoders.)
   Exploit this constraint. Use a Karnaugh-map with *don't care entries* to find another function f with a shorter SSOP form. (*5 marks*)

   c) Rewrite your result from b) into an expression that contains only OR and NOT operations and draw the corresponding circuit diagram.
   This circuit diagram should only contain NOR gates. (*5 marks*)

   d) Rewrite your result from b) into an expression that contains only AND and NOT operations and draw the corresponding circuit diagram.
   This circuit diagram should only contain NAND gates. (*5 marks*)

   e) Give a *simplified product of sums* (SPOS) form of f. The assumption from b) still applies. (*6 marks*)

4. Recall the XOR operation with three arguments, i.e. $Y = A \oplus B \oplus C$. The output Y assumes 1 if exactly one or all three inputs are 1. Y assumes 0 otherwise. In general, the XOR operator evaluates to 1 if and only if the *arithmetic sum* of all input values is *odd*.

   The questions below deal with a *sequential* realization of this XOR operation. This circuit is implemented as a *Moore-machine* with one binary input X and one binary output Y. Its input-output behaviour is described as follows: The output Y always depends on the three values of X which occurred at the *most recent three clock steps*. If 1 or all 3 of these most recent input values are 1 then the output Y shall assume 1. Y shall assume 0 otherwise.

   For reasons of simplicity, we don't pay attention to the output at the very first two clock steps. The machine has exactly *8 states* that should correspond to the binary number composed by the input values of the most recent three clock steps.
   For example, if the most recent sequence of input values was $1, 1, 0$ (0 being most recent), then the machine is in state 6 (because the binary representation of 6 is 110) and the output is $Y = 0$ because $1 \oplus 1 \oplus 0 = 0$. Suppose a new clock step occurs and X has meanwhile changed to 1. Then the machine enters state 7 (binary representation 111) and he output is now 1 because $1 \oplus 1 \oplus 1 = 1$. **21 marks**

   a) Determine and draw the *output table* of this Moore-machine, i.e. a table that shows how the output depends on the current state. (*5 marks*)

   b) Determine and draw the *state transition table* of this Moore-machine. (*6 marks*)

   c) Give an example of an input sequence that causes *alternating output*,
   e.g. $Y = 0, 1, 0, 1, 0, 1, \ldots$. (*6 marks*)

   d) How many flip-flop circuits would a circuit realization of this machine contain?
   Justify your answer. (*4 marks*)