Institut for Matematik og Datalogi
Syddansk Universitet

# FF500 Take Home Exam — Computer Science 2015

This is your FF500 take home exam. This project is based on your courses DM549, DM550, and DM534, though most directly on DM534 since it overlaps with both of the other two. It covers stating algorithms precisely and being able to follow the execution of an algorithm (as is required in programming), circuit design (Boolean algebra and logic), and LaTeX. The project will be graded on a Pass/Fail basis.

The project is due at **8:15 on Tuesday, September 22**.

You may write it either in Danish or English, but write it clearly. It must be made in LaTeX, though you may draw some circuits by hand, scan them in to make PDF files, and then include the PDF files in your LaTeXcode. Write your full name, your section number (D1, D2, or D3), and your “instruktor”s name (Kristine Vitting Klinkby Knudsen, Mathias W. Svendsen, or Jesper With Mikkelsen) clearly on the first page of your project (on the top, if it is not a cover page). You should turn in your project as a PDF file via Blackboard. The assignment/project hand-in is in the menu for the course DM534 and is called “SDU Assignment”. Turn in using your correct section, D1, D2 or D3. Keep the receipt it gives you proving that you turned your project in on time. **Blackboard will not allow you to turn in an assignment/project late.** (Ask for help early if you need help for submitting.)

You will be able to pick up your graded project from Joan Boyar in her office, starting at 15:00 on Wednesday, September 30. (Check your email to see if it is available earlier in the day.) If your assignment is not approved, you must redo it (even though you will have other assignments to work on in this course and your others at the same time) and turn in a corrected version by 8:15 on Monday, October 5, again turning in a PDF file through SDU Assignment via Blackboard. The corrected versions will be graded by 10:00 on October 19.

Note that it is important to always pick up your projects (or later in the course, assignments), since this is how you will find out if they have been approved and there will generally be useful comments on them to help you improve your performance.

In order to have this project approved, you must do all parts of the assignment, making only minor errors. Your answers must be written so that they are complete and easy to understand.

Cheating on this project is viewed as cheating on an exam. You are allowed to talk about course material with your fellow students, but working together on this project is cheating. If you have questions about the project, come to Joan Boyar or your "instruktor" for DM534.

Please note that this project is a compulsory part of your first-year examination. If you fail to hand in the project on time or do not pass it, you will be called in to talk with an administrator, and you will be automatically signed up for the re-exam. If you do not pass this project, the re-exam will be a different project, which will be available through the course homepage by 10:00 on October 19. It will be due at 8:15 on October 26. Again, it should be turned in as PDF file through Blackboard for DM534.

## The Project

Do the following problems. and write your solutions in LaTeX. Do not include the statements of the problems or other information not asked for. Write complete, clear, easy to read, concise answers.

1. Consider the following the algorithm:

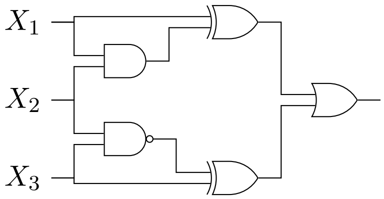
```
**CALC**$(M, N)$:
{ Input: two positive integers $M, N$ }

    $Q := 0$
    **while** $M \geq N$
    **begin**
        $M := M - N$
        $Q := Q + 1$
    **end**
    return($Q$)
```

   (a) Suppose the inputs to the algorithm are $M = 18$ and $N = 5$. Show the sequence of values for $M$, $N$ and $Q$ that are computed by the algorithm and also the result. More generally, what does the algorithm calculate?

   (b) If the line "**while** $M \geq N$ **do**" is changed to "**while** $M > 0$ **do**", what effect will this have on the algorithm? Which inputs will make the algorithm still give the same result as the original?

(c) Using pseudocode represent the original algorithm in another way by removing the **while** structure and using a **repeat-until** structure instead. Your new pseudocode should produce the same result as the original in all cases.

2. Consider the following circuit:



(a) Give a truth table for this circuit, including the outputs of all five gates as separate columns. Use LATEXcode for this (do not import a file).

(b) Express the output of this circuit as a Boolean formula. Use math mode in LATEX.

(c) Draw another circuit which would produce the same output without using any XOR gates, without increasing the number of gates. This may be drawn neatly by hand, scanned in to make a PDF file, and imported to LATEXas a PDF file. Give a truth table for this circuit, including the outputs of all of your gates as separate columns. Use LATEXcode for this part (do not import a file).

3. Either do the first two problems below or the third one. The third one is somewhat more challenging. In all cases, explain your solutions (a truth table, including the outputs of the most important gates, not just the output, may help in doing this; and if you use Disjunctive Normal Form or some of De Morgan's Laws, state that). The circuits themselves may be drawn neatly by hand, scanned in to make PDF files, and included in your LATEXcode as PDF files.

(a) Design and draw a circuit containing only AND, OR and NOT gates (each gate having at most two inputs) which takes three bits as input and outputs a 1 if the input is 001, 011, 100 or 111, and a 0 otherwise. (In the student resources for the DM534 textbook, under the Activities for Chapter 1, there is a simulator for logic circuits which you could use to check your circuit. It is time consuming to use, though.) The circuit you create should be clearly marked as to which input goes where. Call the inputs $a$, $b$, and $c$, going from left to right.

(b) Design and draw a circuit containing only AND, OR and NOT gates (each gate having at most two inputs) which takes four bits as input and outputs a 1 if the input is 0001, 1011 or 1101, and a 0 otherwise. The circuit you create should be clearly marked as to which input goes where. Call the inputs $a$, $b$, $c$, and $d$, going from left to right.

(c) Design and draw a circuit containing only AND and XOR gates (each gate having two inputs) which takes seven bits as input and outputs a 1 if the number of ones in the input is 2, 3, 6, or 7, and a 0 otherwise. Use only three AND gates. How many XOR gates do you need? Hint: Look at (and use) the problem from the discussion section where you were asked to minimize the number of AND gates. Then consider how to represent in binary how many ones there are in a set of three bits. Then add two numbers, each of which has two bits, plus one extra bit.

4. What is the hexadecimal number 1FC in binary? What is it in decimal? Show your calculations.

5. Include your LaTeXcode for this project. Make sure lines are broken properly so it can be read.