Technical University of Munich
School of Engineering and Design
Prof. Dr. Martin Werner

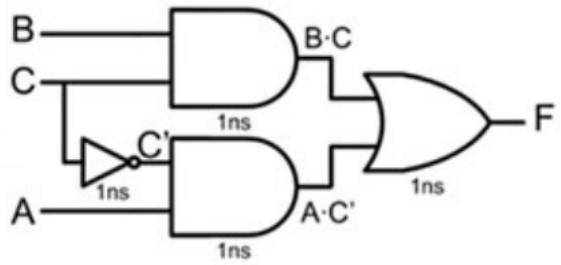
Computational Foundations II (Summer Term 2022)
Tutorial 2

Tasks marked with a star like **Optional Task*** are optional. Tasks marked like **Hard Task+** are given, but it is not expected that you solve them now. It is great if you learn to solve them during the lecture. Go back to them after a few weeks and see your own progress.

Learning Outcome: Timing Glitches, Medium Scale Circuits

Task 4: Timing Glitches

The Boolean algebra perspective on electronic circuits is a very powerful tool showing us that more or less every interesting functionality can be implemented by using only a single electronic building block (NAND). However, this perspective is incomplete. This task shows that the algebraic minimization of Boolean circuits can lead to so-called timing hazards or timing glitches and that real-world digital electronics might need to take into account these aspects.
When turning from a input-output perspective as represented by a truth table to a timing perspective, one first defines a stimulus (often called testbench) which is a temporal trace of all input signal lines. A good stimulus consists of all states and all important transitions, at the moment, we will create only a stimulus for three bits which shows a problem.
We follow an example from the given book `https://link-springer-com.eaccess.ub.tum.de/book/10.1007/978-3-030-13605-5`, which you can freely access. Try to follow this task before reading Chapter 4.5 in this book.
Consider the following circuit:

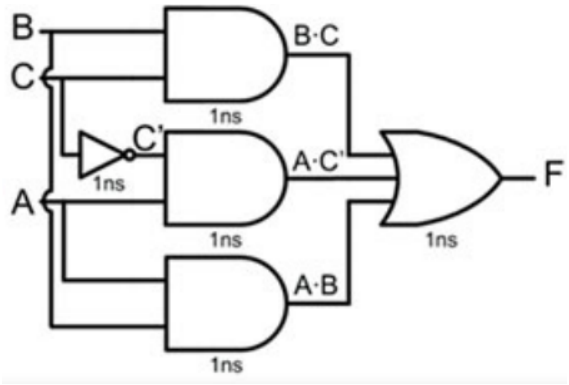


in which the timings are understood to be delays in the following sense: if a change on an input line implies a change to the output lines, then this change is performed 1ns later.

- Reconstruct the expression of this circuit
- Draw a Karnaugh map with A on the vertical axis and a Gray code for BC on the horizontal axis.
- Draw a time-signal diagram for 7ns with given signals $A = 1$ and $B = 1$ for all times. Signal C starts as $C = 1$ and changes to $C = 0$ at t=2ns. Considering delays, derive the change of all signal lines in the circuit, namely, BC, $\overline{C} == C'$, $A \cdot \overline{C}$, and finally F.

  Explain all changes to signals at times.

– With the same stimulus as in the previous task, analyze the behaviour of the output of



Task 5: Multiplexer

In the lecture, we have seen the definition of multiplexer and demultiplexer. Implement a 1-to-4 multiplexer and a 4-to-1 demultiplexer for a 2-bit address bus. Feel free to use multi-input AND and OR gates.

Task 6: Adder

– Implement a half-adder and a full adder using OR, AND, and NOT and implement a full adder using only NAND operations

– Stitch together a ripple-carry adder for 8 bit. This contains a half adder for the two lowest bits and then a full adder for each bit. The carry output is fed into the next-higher adder.

– Check that the adder is correct

– If we add timing delays, how many delays do we need to wait? What is the depth of the circuit? If you have a gate technology which can implement gate logic at 1MHz, what is the speed of the addition?

This task shows a problem of gate logic, namely, that it easily accumulates a depth and that the input signals of all bits have to be held for a long duration until the last addition has happened. In ASIC integrated circuits, this is not a severe problem, as the internal frequencies can be quite high, but for reconfigurable devices like FPGAs, where the hardware description is implemented at runtime, these pose scalability problems. Therefore, modern architectures of FPGAs can implement complex Boolean circuits (using a lookup table) and provide so-called “fast carry chains”. Interested students can use a search engine to find out more.