Faculty of Science
Department of Computer Science
Chair for Embedded Systems
Prof. Dr. O. Bringmann

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Digital Design and Synthesis of Embedded Systems

Summer Term 2025

Exercise 9 : Deadline 20.07.2025 - 23:59 o'clock

The solutions are submitted via ILIAS as a packed archive (ZIP/TAR.GZ). This ZIP contains the written answers to questions, the source code (not as a listing in the PDF), the simulation results as VCD files, and the Makefile. Each of your programming tasks **must** be executable with your Makefile or with a standard python interpreter. If your python solution uses non-standard packages you have to provide a requiretmens.txt file. If you do not follow this format, your handing may be graded with 0 points!

Exercise 1 Questions about the lecture [4 Points]

(a) What is a standard cell library?

(b) What are design constraints and how are they used in RTL synthesis?

(c) What is the result of the technology mapping algorithms for FPGAs and ASICs?

(d) Which step in an FPGA flow comes after the technology mapping?

Exercise 2 Technology Mapping [8 Points]

In the lecture, you have learned how a technology mapping to lookup tables (LUTs) or standard cells is possible using cone coverage or tree coverage. You will implement parts of these algorithms yourself in this assignment. The ESES graph library from the last exercise sheets is available to you for this purpose. Work on the following tasks:

(a) FPGA mapping to lookup tables. One of the first algorithms for mapping an arbitrary netlist to LUTs is called *FlowMap*. A simple implementation is given in the function `flow_map` in the Python file.

  (a) Implement the calculation of the set N_v. Its definition is as follows: Given a graph $G = (V, E)$, we define N_v as the set of all nodes u for which a path in G to v exists:

$$N_v = \{u \mid \exists\, \mathrm{path}(u, v) \in G\} \cup \{v\}$$

  (b) Implement the calculation of the set PI and PO. A primary input (PI) is a node that has no successor node. A primary output (PO) is a node that has no successor node.

  (c) Implement the calculation of the set $GATES$. A gate is a node in the graph that is neither PI nor PO.

  (d) Test the algorithm with the provided main function. Add the result to your submission. *Note:* The example is the example from the lecture.

  (e) Write two more test cases. Use a *Constant* node for all PIs.

  (f) Briefly describe which criteria are used to calculate the best possible LUT.

(b) Technology mapping on standard cells. You will find an incomplete implementation of the algorithm for standard cell mapping in the function `def tech_map`. The main function provides you with an implementation of the standard cell library from the lecture. You can extend this as you wish:

   (a) Implement the calculation of the quantity N_v analog to the previous task.

   (b) Test the algorithm with the provided main function. Add the result to your submission. *note:* The example is the example from the lecture.

   (c) Write two more test cases. Make sure that your graph only has one output node. You may need to extend the standard cell library.

   (d) The lecture describes that a node cover must be calculated in order to map standard cells to operations. Briefly explain how this calculation is done here.

(c) Do you notice any similarities in the algorithms ? If so, which ones.

Exercise 3 Memory Macros [8 Points]

Memory elements are one of the most expensive resources in terms of area and power consumption. A pure implementation of memory from registers (flop-flops), as we have done so far, is only used for very small memory sizes. You will find out the reason for this in this task. All the necessary design files in the `memory_macro` folder:

(a) In the directory containing the tasks, you will find a memory macro for the NANGATE technology in the `SRAM_...` subdir. Look at the design files and enter the following characteristics in a table: Number of words and word width, Area, max. clock rate, leakage and the dynamic power consumption during an operation.

(b) In the memory macro folder, you will find many different files that are required for an ASIC flow. Research what is normally contained in the files with the following extensions and briefly summarize the results of your research:

- `.gds` GDSII-File
- `.lef` LEF-File
- `.lib` Liberty-File
- `.sp` Spice-File

(c) You will find a Verilog file (ending `.v`) in the memory macro folder. Can you imagine why this file exists? Think of the different IP types that exist.

(d) In the folder of the task you will find an SRAM implementation in SV: `sram.sv`. Set the memory size the same as the memory macro and synthesize the design with the provided Makefile. Look at the generated post-synthesis netlist and find out from which standard cells the SRAM was implemented.

(e) Compare the synthesized design with the memory macro and display the result in a table (power, area).

(f) Comment on whether it is better to use a memory macro or an RTL implementation.