

Requirement specification (kravspecifikation)

NandSim – A tool for specifying and simulating integrated circuits with logic gates.

The customer is an integrated circuit chip (IC) manufacturer. The tool is to be designed used for software developers and as an interaction point between hardware developers and software developers.

The solution is a visual tool, that both software and hardware developers can use to build and simulate logic circuit models. The models consist of assembled logic gates. To organize complicated logic, the gates of the circuits can be organized into reusable component. The tool can simulate the modeled circuits in an interactive live simulation. Software developers should be able to develop software targetting the modeled circuits, and further be able to execute and test code in the simulation.

The goal is a tool that enable IC manufacturers, specifically software and hardware developers to design and develop circuit models and embedded software, as to reduce the development time for an IC solution, by decoupling the software and hardware development processes.

Requirements

As an IC developer using this solution, I should be able to:

- **★ assemble logic circuits visually by placing and wiring together primitive logic gates.** This includes a visual editor area, where logic gates and components can be inserted and dragged around, and where each gate and component can be wired together.
- **★ create circuit components built with logic gates that can be used like primitive logic gates.** To help organize and make complex circuits manageable, sub-sections of a circuit can be encapsulated in a component.
- **★ simulate the logic of a circuit.** A simulation consists of initializing a state according to the circuit and the inputs given. The simulation should be continuous, and support simulating state dependent on a previous state.
- **★ interact visually with a simulation of a circuit.** The simulation should support updating the state according to any changes made to the circuit.
- **Upload files with data input for complex circuits, including program and data files in suitable file formats such as binary or hexadecimal.** The purpose is to enable flashing of a simulated circuit with data and programs to be used in the simulation.
- **Monitor complex circuit output in suitable formats such as base 10 displays, hexadecimal or ASCII text.** To be able to monitor the produced outputs of a simulated circuits, the tools should have ways to display outputs appropriately.
- **Create and export component libraries.** Components created in one project should also be usable in other projects.

- **Save projects in the cloud with a user account.** Projects can be stored in the cloud using user accounts. With the same user account, projects can be accessed, edited and saved on multiple computers.
- **Collaborate with others using cloud projects and multiple user accounts.** Multiple people can collaborate on the same project, each using their own user account. This includes a strategy for work synchronization and work conflict resolution.

★ marked requirements demarcate the minimum viable product.

Non-functional requirements

The system should conform to the non-functional requirements of:

- **Simulating interactively without unreasonable delay.** The system should be able to simulate any circuit with less than 100 components in less than 500 milliseconds on modern hardware.
- **Preventing loss of work.** The system should employ a strategy to save the user's progress frequently enough, so that work is never lost. Immediately following any significant change, the system should be able to restart without loss of work.

Delimitation

For simplicity, both of use and of implementation, the solution will be a digital logic simulator. This is an abstraction removed from simulating electrical current in circuits with resistors, capacitors, etc. This limits the tool in terms of its applicability while still being suited for the case description.

The simulator will not handle unstable circuits. If a circuit contains a cycling connection in such a way that the circuit vacillates indefinitely, the circuit is said to be unstable. A general solution to detect unstable circuits ahead of time is mathematically impossible due to the halting problem, and heuristics-based detection methods come with their own complexity. It's decided that the simulator simply doesn't support unstable circuits, and it's thus the user's responsibility ensure that circuits are stable.