DeltaLambdaSigma ΔΛΣ
System Requirements Specification

DeltaLambdaSigma, Ivan Sekyonda, CEO		
Rob Hiles	Matthew McHugh	Ben Yamada
Kes Johnson	Raj Karsalia	

Introduction

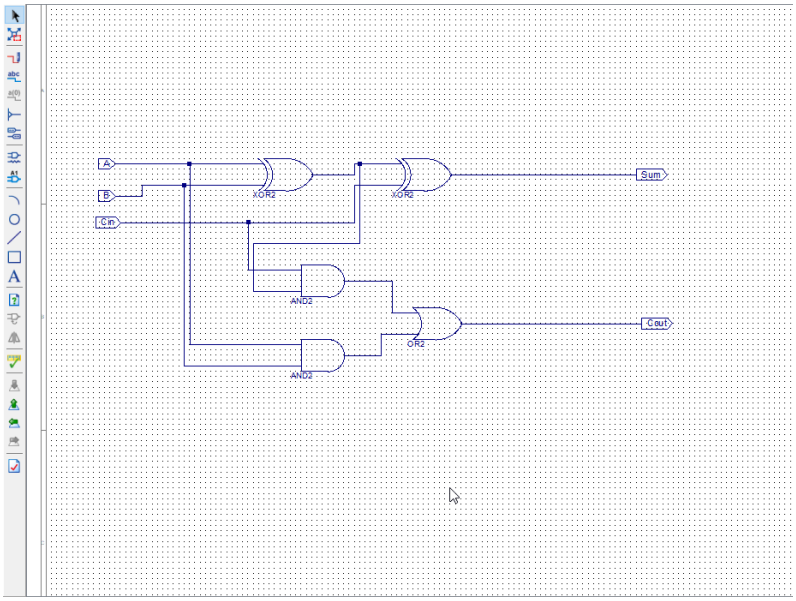
DeltaLambdaSigma, an Integrated Synthesis Environment (ISE), is a web application that allows users to design and simulate digital logic schematics. The Web application will serve as a viable replacement for the aging Xilinx software. The application provides (1) an interface for users to design circuits using components (such as inputs, outputs, wires, gates, busses, etc.) and (2) an interface for users to simulate and analyze the behavior of the circuit. The application also provides the functionality for the user to build and deploy their own symbols.

User stories

Schematic Design Interface

As a user, I want to have a workplace for my designs

- This is the main page where gates/other components can be placed.
- Fixed page size
- Example:



As a user, I want to have basic components

- This design and implement the following components

List of “Built In” Components
Wires (1-16 bit)
not
or (2-8 input)
and (2-8 input)
xor (2-8 input)
xand (2-8 input)
nor (2-8 input)

nand (2-8 input)
xnor (2-8 input)
Bus taps
Multiplexer with max size is 16 bits
Demultiplexer with max size is 16 bits
Buses with max size is 16 bits
LEDs that display on the schematic

As a user, I want to create my own components

- Components can be made from a user's previous designs.
- Comparable to “create symbol” functionality in Xilinx

As a user, I want to synthesize my schematics

- Allows for syntax checking of graphical design
- Comparable to “compiling” code

As a user, I want to be able to place components

- Components are able to be manipulated (size changes/rotations)
- Components are able to be different things (gates, wires, etc.)
- Copy and paste components

As a user, I want to save my design to a file (export).

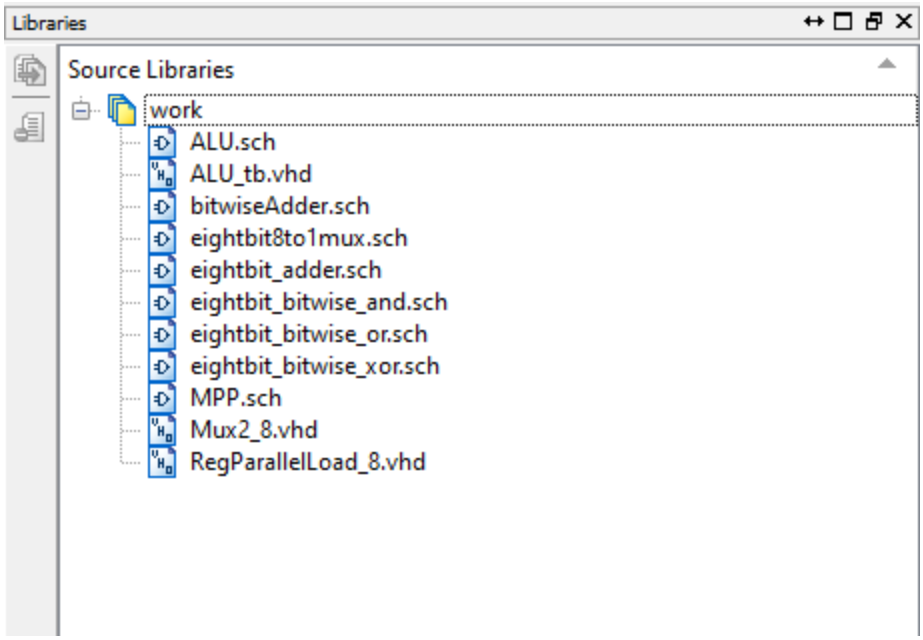
- Export the schematic to the users local computer
- “.sch” format

As a user, I want to import a file with my design (import).

- Import the schematic from the users local computer
- “.sch” format

As a user, I want to import a library with a series of designs.

- Import the library from the users local computer
- Example:

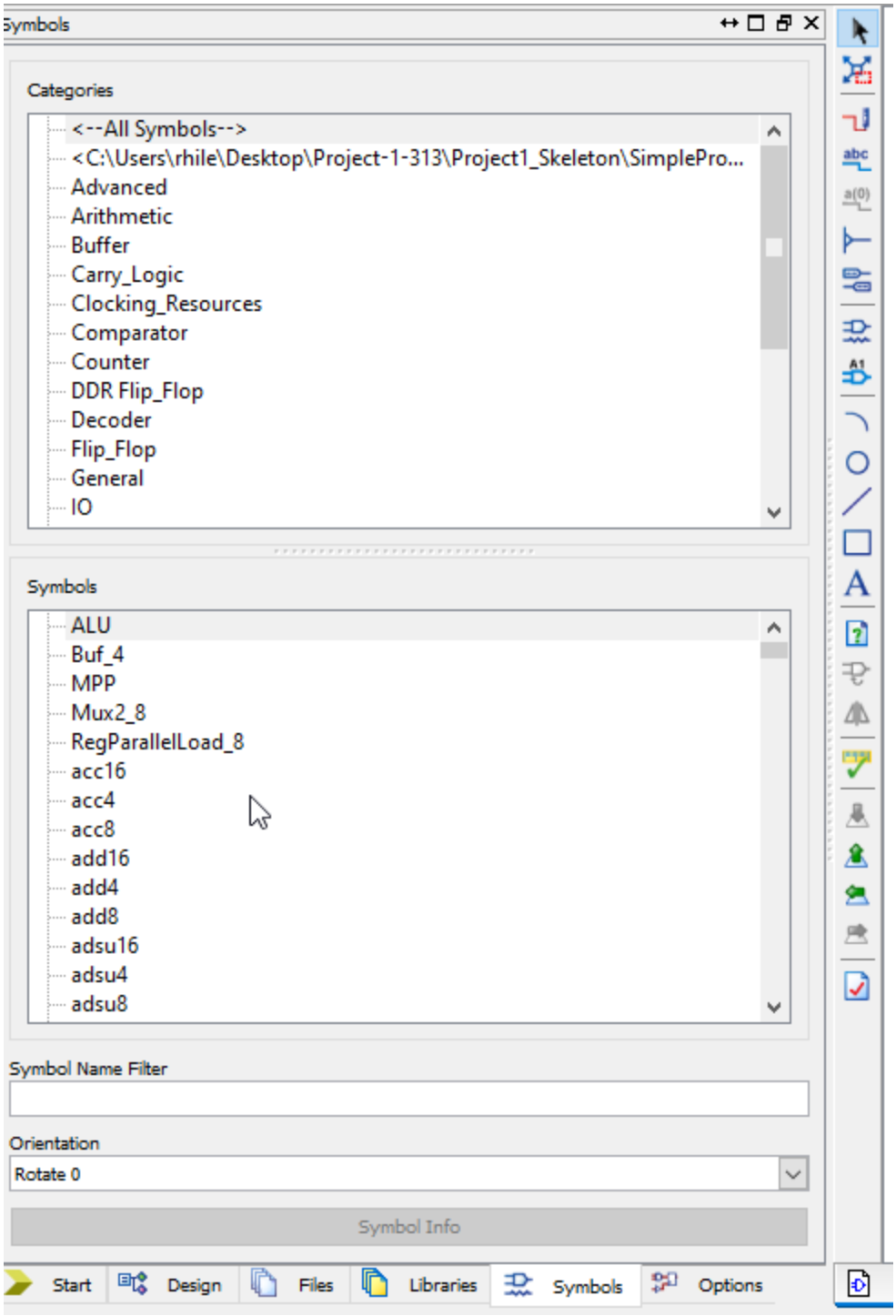


As a user, I want to have a toolbar

- Includes import/export buttons
- Simulation controls

As a user, I want to have a toolbox

- Includes component selection
- Example:



Simulation Interface

As a user, I want to simulate my design

- Be able to take a given schematic and a given testbench and view the output

As a user, I want to have a default testbench

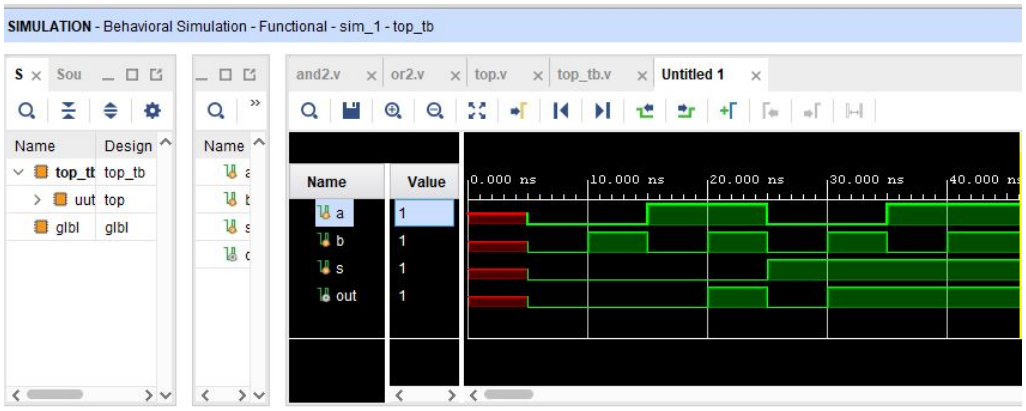
- System generates a default testbench if the user doesn't supply one

As a user, I want to be able to import a custom testbench (.vhdl)

- System is able to accept a custom testbench to simulate design

As a user, I want to view input/output waveforms from simulation

- For example…



HDL Console

As a backend service, I want to take the graphically represented schematic and translate it into hardware description language (HDL) code

- Takes the schematic and translates it into HDL code to be used by a HDL console

As a backend service, I want to take HDL code and a testbench and simulate the behavior

- Takes the schematic in HDL format and simulates it using a test bench in VHDL code

As a backend service, I want to generate a default testbench

- If the users does not supply a test bench a default one will be generated
- Generates a basic testbench in vhdl to simulate a circuit
- Tries all possible input combinations in a deterministic manner

Story Backlog

1. As a developer, I want to learn how to deploy a Django web application to the cloud.
2. As a developer, I want to learn how to build a pipeline for my web application.
3. As a developer, I want to figure out how to translate a graphical design/schematic to code.
4. As a developer, I want to figure out how to load/save a schematic design of a specific format (such as “.sch”).
5. As a developer, I want to figure out what file formats to use for schematics (.sch) and for workbenches (.vhdl)
6. As a developer, I want to host a web application that will handle at least 80 users simultaneously.
7. As a user, I want to have a workplace for my designs

8. As a user, I want to have basic components
9. As a user, I want to have a toolbox
10. As a user, I want to have a toolbar
11. As a user, I want to be able to place components
12. As a developer, I want to be able to run a test schematic
13. As a user, I want to save my design to a file.
14. As a user, I want to import a file with my design.
15. As a user, I want to simulate my design
16. As a backend service, I want to generate a default testbench
17. As a user, I want to have a default testbench
18. As a backend service, I want to take HDL code and a testbench and simulate the behavior
19. As a user, I want to create my own components
20. As a user, I want to import a library with a series of designs.
21. As a user, I want to be able to import a custom testbench (vhdl)
22. As a user, I want to view input/output waveforms from simulation
23. As a backend service, I want to take the graphically represented schematic and translate it into hardware description language (HDL) code
24. As a backend service, I want to take HDL code and a testbench and simulate the behavior
25. As a backend service, I want to generate a default testbench

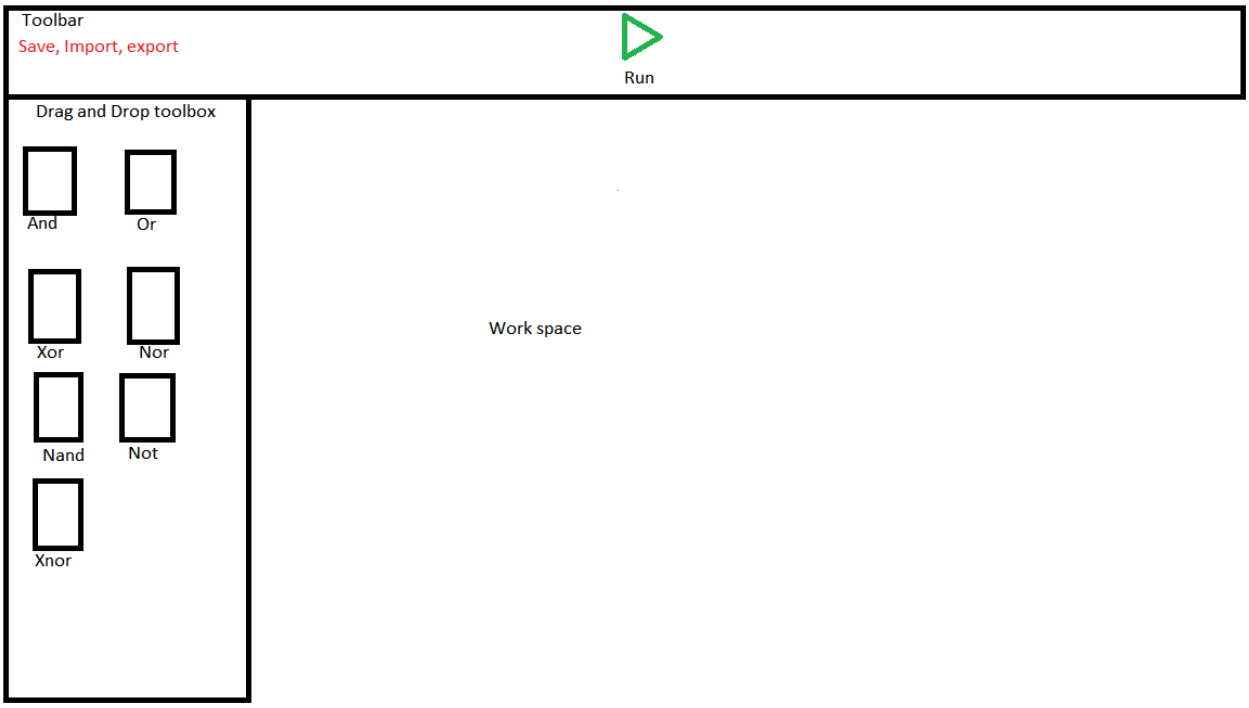
Development dependencies

- Backend components need to be built before frontend/graphic components

Stakeholder dependencies

- Simulation functionality (Waveforms)
- Blackboxes/symbol creation
- Import/export functionality

Diagrams



Risk analysis

The main risk facing DeltaLambdaSigma is the developer's inexperience with Integrated Synthesis Environments (ISEs). This inexperience may make identifying all the stories more difficult as that would require an in depth understanding of schematic design software. Another risk is the developer's inexperience with the Django framework, graphic design, and digital logic design.

Resources

1. Xilinx docs/SDK info
2. Logisim GitHub (open source)
3. Xilinx ISE Wiki Page