# 3 Design and simulation of digital circuits using computer aided design software (I)

## 3.1 Objectives

Logisim offers a dedicated environment for design and simulation of digital circuits. Its structural elements are presented, and the handling of active controls as part of the simulation environment. The design steps for several circuits and their simulation sessions are detailed as guide for using this tool and its facilities. This work is mainly focused on schematic editor and symbol editor as well as the management of the circuits in the library attached to the project.

## 3.2 The Logisim application

Logisim (v. 2.7.1) offers support for logic design of digital circuits and functional simulation with values applied on input terminals, while highlighting the results on output terminals. The design of the circuit has a graphical representation called *logic diagram*, which contains the symbols of the inner circuits and their interconnections. Logisim contains a set of basic logic gates and several other circuits. Nevertheless, the user has the possibility to extend them with more complex circuits by means of hierarchy design and encapsulation in reusable libraries. The design phase is twofold:

1. The implementation of logic diagram in the Canvas of the schematic editor by defining the structure of the circuit. In this phase, Logisim allows the enabling/disabling of the simulation engine.
2. The design of the circuit symbol, which enables its integration inside other circuits. This is representative for the hierarchy design using abstraction levels of progressive complexity.

### 3.2.1 The schematic editor

The schematic editor is available immediately after launching Logisim. A logic diagram can be implemented in the Canvas (Figure 3. 1). The zoom in/zoom out control 

in the lower-left corner can be used to change the size of the elements in the diagram.

#### 3.2.1.1 Elements of the logic diagram and their packaging into libraries

The Explorer Pane (also called Toolbox) in the lateral side of the Canvas shows the elements (tools/circuitry) that can be placed on the logic diagram. The elements are packed into libraries. The topmost library is called *Untitled*, same as project name. Initially, it contains the circuit called *main*, which is currently under development. Pressing ⊞ on the left side of a library name, will highlight the list of tools and circuits inside. For instance, the *Gates* library contains the basic logic gates. The other libraries contain more complex circuitry or useful tools required for the development of logic diagrams.

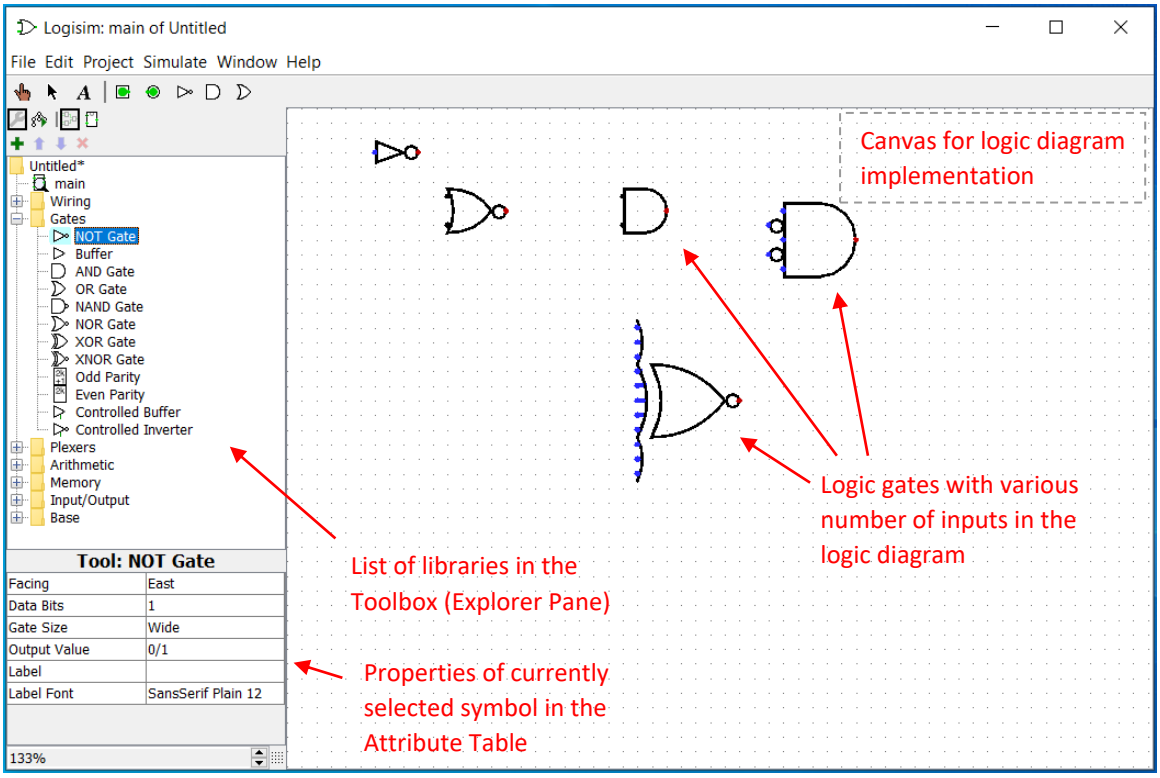



Figure 3. 1 The schematic editor in Logisim

### 3.2.1.2 Symbols and their properties

The symbols are graphical representations of the available circuits and tools. They can be inserted in the logic diagram by selection, followed by a click in the Canvas. Also, they can be drag-and-dropped to other positions. A symbol in the diagram can be multiplied with Copy, Paste actions available in the Edit section of the menu bar. The keyboard shortcuts for Copy-Paste are *Ctrl+C* followed by *Ctrl+V*, or simply *Ctrl+Insert*. When a symbol is selected, either in the Toolbar or the Canvas, the Explorer Pane will display its properties in the Attributes Table below the library list. For instance, by selecting the inverter gate (NOT) the list of properties will contain the following set of attributes (Figure 3. 1):

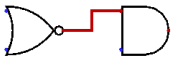
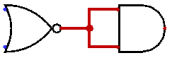
- `Facing` – orientation of the symbol (used to rotate the symbol);
- `Data Bits` – number of bits for the logic gate;
- `Gate Size` – symbol dimension in the diagram;
- `Output Value` – the output behavior: "0/1" is most common;
- `Label` – an optional text used as ID in the diagram;
- `Label Font` – font size for label.

The list of properties may change by the symbol type, but there is a common list of properties for most of them. The attribute values can be modified, which may affect the functionality of the circuit, sometimes reflected by minor changes of the symbol's appearance in the Canvas. For instance, a NAND gate with attribute `Number Of Inputs` = 2 has the symbol  ; with 5 inputs, the symbol changes its shape to  ; if inputs

2 and 4 are inverted, with attribute values `Negate 2 = Yes` and `Negate 4 = Yes`, a small circle appears nearby:  .

### 3.2.1.3 Connections between elements on the diagram

When placed in the diagram, inputs and outputs of the elements are represented by dots visible on the boundaries of the symbol with a different color. When the mouse is hover around an input or output pin a green circle highlights its presence  .

To connect inputs and outputs, the user should move the mouse from one pin to another, while keeping the left button pressed; a connection appears like this:  . To create a branch in a certain position of the wire the user can start a new wire from that point to an unconnected input/output:  . The starting point of a branch takes the form of a visible dot on the wire. There are two types of connections allowed on the diagram:

- direct connection from output to input;
- connection from output to wire or from wire to input. As a side note, two connected wires always fuse to one wire.

A cable with more wires can be created with the Splitter element available in the *Wiring* library. Attributes `Fan Out` and `Bit Width In` must be set to the number of wires. **Note**: Any element on the diagram (symbol or wire segment) can be removed by right-clicking it and choosing Delete in the context menu or by pressing *Del* key after selection.

### 3.2.1.4 Inputs with fixed value

The *Wiring* library contains the Power and Ground elements. Power generates a "1" (VCC) and Ground represents the mass "0" (GND). Figure 3. 2 highlights the usage of such inputs. Because these values propagate through the circuitry, they produce visible effects on outputs and wires: the wires carrying "1" are colored in light-green, while those carrying "0" are in dark-green.

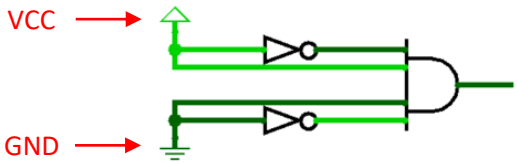



Figure 3. 2 Circuit with inputs connected to VCC and GND

### 3.2.1.5 The text tool

The text tool Text Tool can be accessed from the *Base* library or from the Toolbar in the upper side  . It can be used to insert a block of text in the

diagram, without any functional effect. The text has a descriptive role or can be used to highlight relevant labels.

#### 3.2.1.6 Input and output terminals

Any logic diagram requires input and output terminals. Input terminals are required to send signals to the elements of the circuit. Output terminals capture the results. The number of input and output terminals may vary with the structure of the circuit. During simulation, the terminals highlight their values. In Logisim, the input terminal Pin is available in the *Wiring* library. In case the `Output` attribute is changed to False the terminal type switches to output, and its shape turns circular , as is Figure 3. 3. A terminal can have one or more bits defined by `Data Bits` attribute. For 1-bit terminals `Data Bits` should be set to 1. The terminal can be named using attribute `Label`.

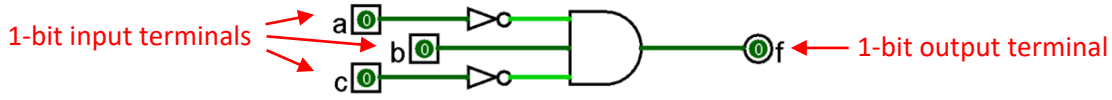



Figure 3. 3 A diagram with input and output terminals highlighting their values

#### 3.2.1.7 Simulation in Logisim

When placing input terminals, their values propagate throughout the circuitry and generate corresponding effects. Since edit mode is enabled at startup, it is not possible to change their values. The simulation mode can be enabled by using the Poke Tool from the *Base* library or from the Toolbar . Afterwards, any click on the input terminals will toggle their values between "0" and "1". The return to edit mode is possible using either Select Tool or Edit Tool from the *Base* library or from the Toolbar: .

When the simulation engine detects errors or incomplete connections, the output terminal will signal unknown or erroneous values. All options linked to simulation can be found in the Simulate menu. Simulation Enabled or *Ctrl+E* will toggle between starting/stopping the simulation. Reset Simulation or *Ctrl+R* applies a reset on the simulation and clears all values to "0".

#### 3.2.1.8 The Toolbar of the schematic editor

The Toolbar situated beneath the menu bar offers quick access to a subset of elements that can be placed in the diagram. The Toolbar has the following structure , but it is possible to bring appropriate changes by accessing Options in the Project menu. A new window will appear, where the Toolbar tab allows access to configuring Toolbar components.

### 3.2.2 The symbol editor

The circuit from the logic diagram can be encapsulated in a symbolic representation having a rectangular shape with input and output pins (representing

terminals). The symbol is saved in the project library; hence it is possible to be used inside other diagrams, with the functionality of the circuit it represents. Any communication with the circuit is possible solely through the pins of the symbol. The switch from schematic editor to symbol editor, and back, is possible using the commands Edit Circuit Appearance and Edit Circuit Layout from the Project menu. A quick switch is possible using the shortcut buttons from the Toolbar:  . For instance, the symbol generated automatically for the diagram in Figure 3. 3 is highlighted in Figure 3. 4.

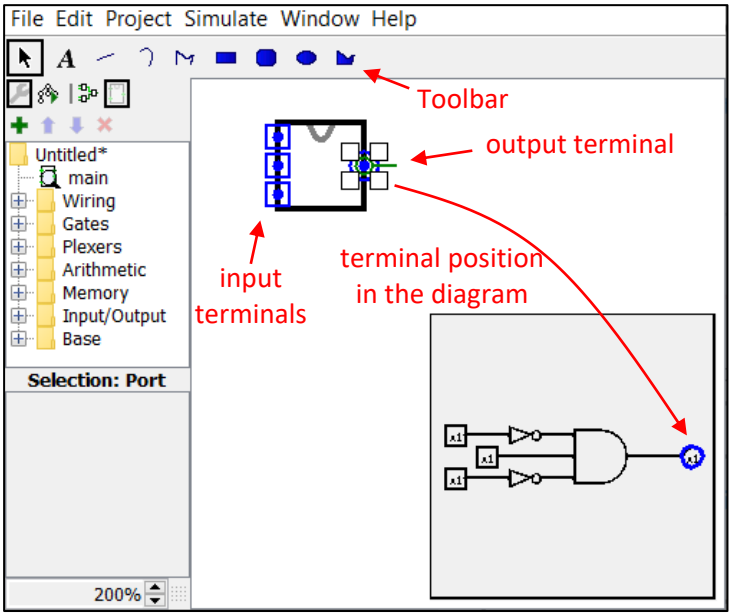



Figure 3. 4 The symbol editor associated with current diagram

The symbol and its graphical elements can be modified. The input terminals on the symbol have a rectangular blue shape, while the outputs are circular. Selecting a terminal pin from the symbol will highlight its position in the logic diagram. Optionally, the label of the terminals or the name of the diagram can be placed on the symbol using the text tool from the Toolbar of the symbol editor:  .

### 3.2.3 Managing the circuits in the project library

The project library can support one or several circuits. The Add circuit  command on top of the list of libraries can be used to add a new circuit. Logisim will ask its name and will generate an empty diagram. The up-down buttons  can be used to move the current circuit into another position within the library. A circuit can be deleted using the remove button  .

It is possible to switch between the circuits within a library by double-clicking their symbol in the Toolbox. The schematic editor will load the corresponding diagram in edit mode.

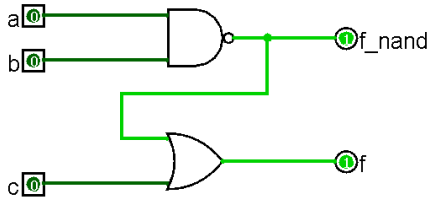
The File menu contains options to Save the project, Open a project or create a New project. A saved project will save its library in the project file.

**Note**: It is important to save the project as often as possible to keep the work safe.

Selecting a symbol in the current library will display its properties. Changing the attribute `Circuit Name` will rename the circuit and its symbol within the library.

### 3.3 Assignments

1. Create a new project in Logisim and design a circuit that contains the NOT, AND, XOR and OR gates from the *Gates* library. Excepting NOT, the gates will be configured to accept two 1-bit inputs. Add two input terminals linked to all gates and four output terminals, one for each gate. Test the logic gates using the simulator and their truth tables.
2. Add a new circuit to the project and design the logic diagram in the figure below. Calculate the truth table. Apply all possible combinations of values on a, b, c and compare the results with the truth table. Switch to symbol editor and alter the symbol generated automatically by adding the names of the terminals using the text tool from the Toolbar.




3. Add a new circuit for each of the following logic functions. Test the circuits using the simulator, by confronting the results with the corresponding truth table for all possible combinations of input values:

a) $f_1 = a + \bar{b} + c$
b) $f_2 = (a + b) \cdot (a + c)$
c) $f_3 = a + \overline{b \cdot \bar{c}}$
d) $f_4 = b \cdot (\overline{a + c} + b \cdot \bar{c})$
e) $f_5 = \overline{a \cdot \overline{\overline{b \cdot c}}}$
f) $f_6 = \bar{a} + \bar{b} + \overline{a + c}$
g) $f_7 = \overline{a \cdot b} + b \cdot \bar{c}$