
icestudio Documentation

Release 0.3.3-rc

Jesús Arroyo Torrens

Aug 17, 2018

1	Installation	3
1.1	GNU/Linux	3
1.2	Mac OS	3
1.3	Windows	4
2	Quick Start	7
2.1	Setup the toolchain	7
2.2	Setup the drivers	8
2.3	Upload a design	9
3	User Guide	15
3.1	Menu	16
3.2	Blocks menu	21
3.3	Design	22
3.4	Take a snapshot	31
4	How to...	33
4.1	Install the toolchain	33
4.2	Update the toolchain	34
4.3	Install the drivers	34
4.4	Create a project	35
4.5	Show the FPGA resources	48
4.6	Upload a bitstream	49
4.7	Show the commands output	52
4.8	Create a block	53
4.9	Add a project as block	56
4.10	Add a collection	59
4.11	Select a collection	59
4.12	View the selected collection info	59
4.13	Load external collections	59
4.14	Create a collection package	60
4.15	Include a list file	60
4.16	Include a verilog (header) file	61
4.17	View the board rules	61
4.18	Disable the board rules	62
4.19	Configure a remote host	64
4.20	Close the application	65

5	Setup	67
5.1	Download	67
5.2	Install	67
5.3	Execute	67
5.4	Languages	67
5.5	Package	68
5.6	Apio configuration	68
6	Project	69
6.1	Definition	69
6.2	Package	71
6.3	Samples	72
7	Blocks	97
7.1	Definition	97
7.2	Basic blocks	97
7.3	Generic blocks	106
8	Board Rules	109
8.1	Input rules	110
8.2	Output rules	110
9	Compilers	113
9.1	Implementation	113
9.2	Sample	114

Icestudio is a visual editor for open FPGA boards. It is built on top of the [Icestorm project](#). This IDE is available for GNU/Linux, Windows and Mac OS X.

Supported boards:

HX1K

- [IceZUM Alhambra](#)
- [Nandland Go board](#)
- [iCEstick Evaluation Kit](#)

HX8K

- [Alhambra II](#)
- [BlackIce](#)
- [BlackIce II](#)
- [icoBOARD 1.0](#)
- [Kéfir I iCE40-HX4K](#)
- [iCE40-HX8K Breakout Board](#)

LP8K

- [TinyFPGA B2](#)
- [TinyFPGA BX](#)

Source code: <https://github.com/FPGAwards/icestudio>

1.1 GNU/Linux

1. Install [Python 2.7](#) and **xclip** (to enable Copy/Paste).
2. Download the [AppImage](#) file and make it executable:

```
$ chmod a+x icestudio-0.3.3*.AppImage
```

Hint: You can download the ZIP release and install it in the system using the script [linux_installer.sh](#). This script registers the *.ice* files as *Icestudio project*. There is also a [linux_uninstaller.sh](#) to revert the previous configuration.

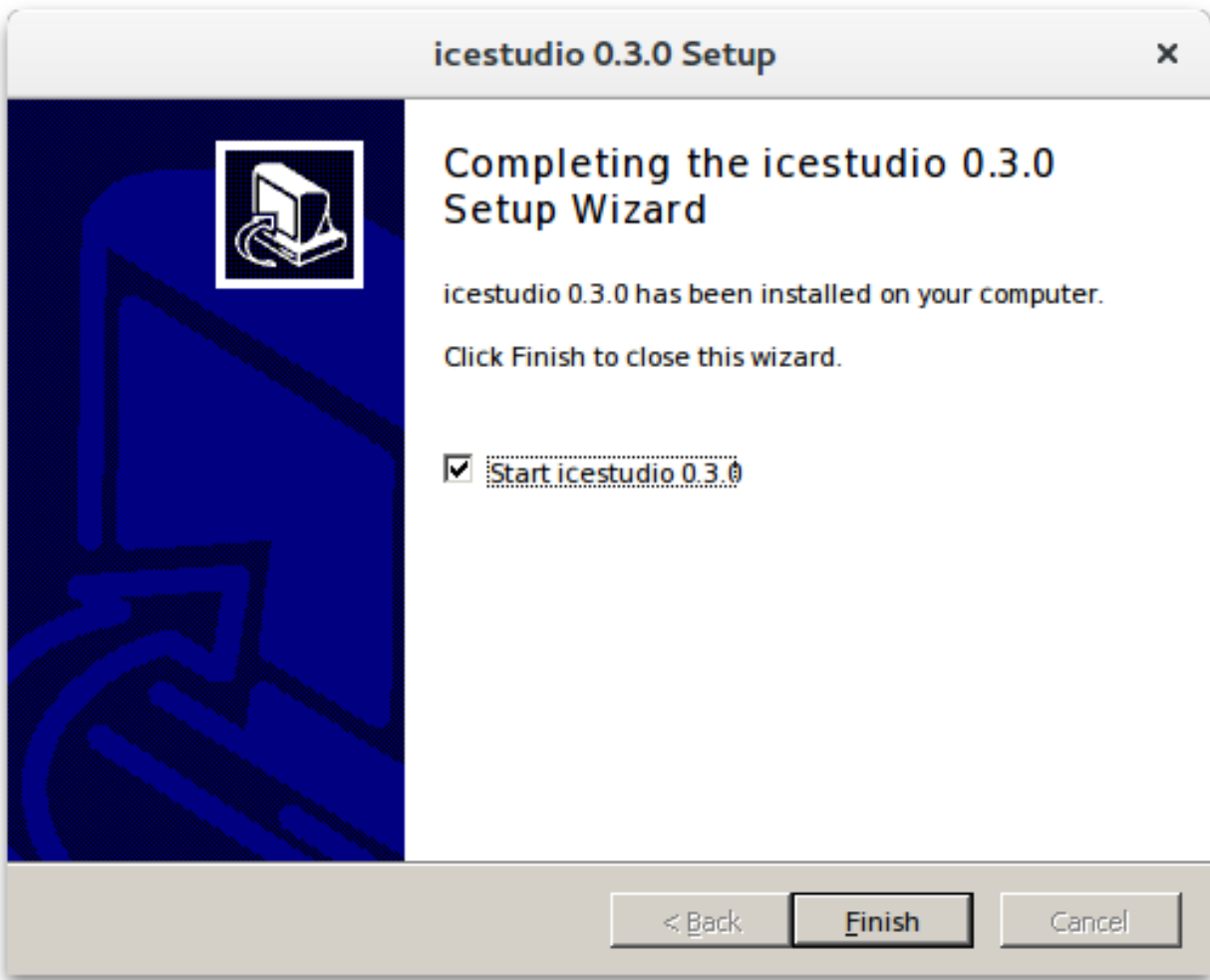
1.2 Mac OS

1. Install [Python 2.7](#).
2. Install [Homebrew](#).
3. Download and execute the [DMG package](#).



1.3 Windows

1. Download and execute the [Windows installer](#).

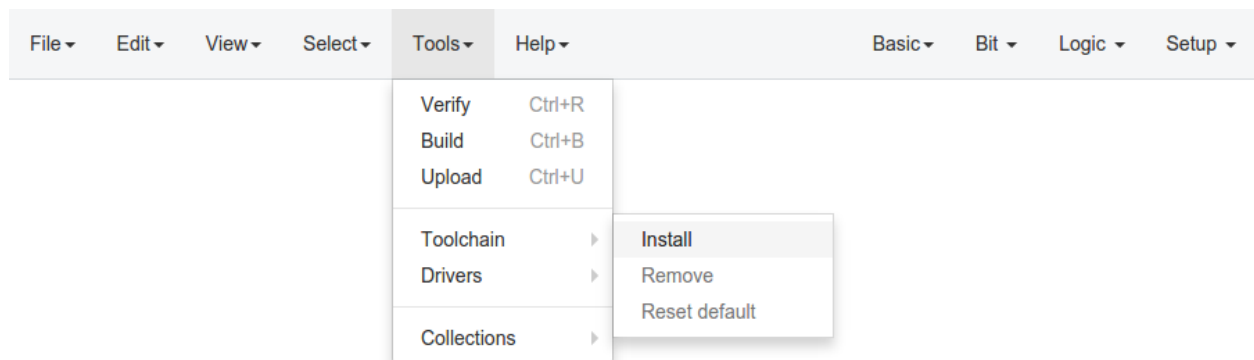


Note: Python 2.7 will be installed automatically if it is not installed. This installer registers the *.ice* files as *Icestudio project*.

Warning: If the error *Failed building wheel for apio [...]* appears *pip* installer may be blocked. Disable the antivirus and try again.

2.1 Setup the toolchain

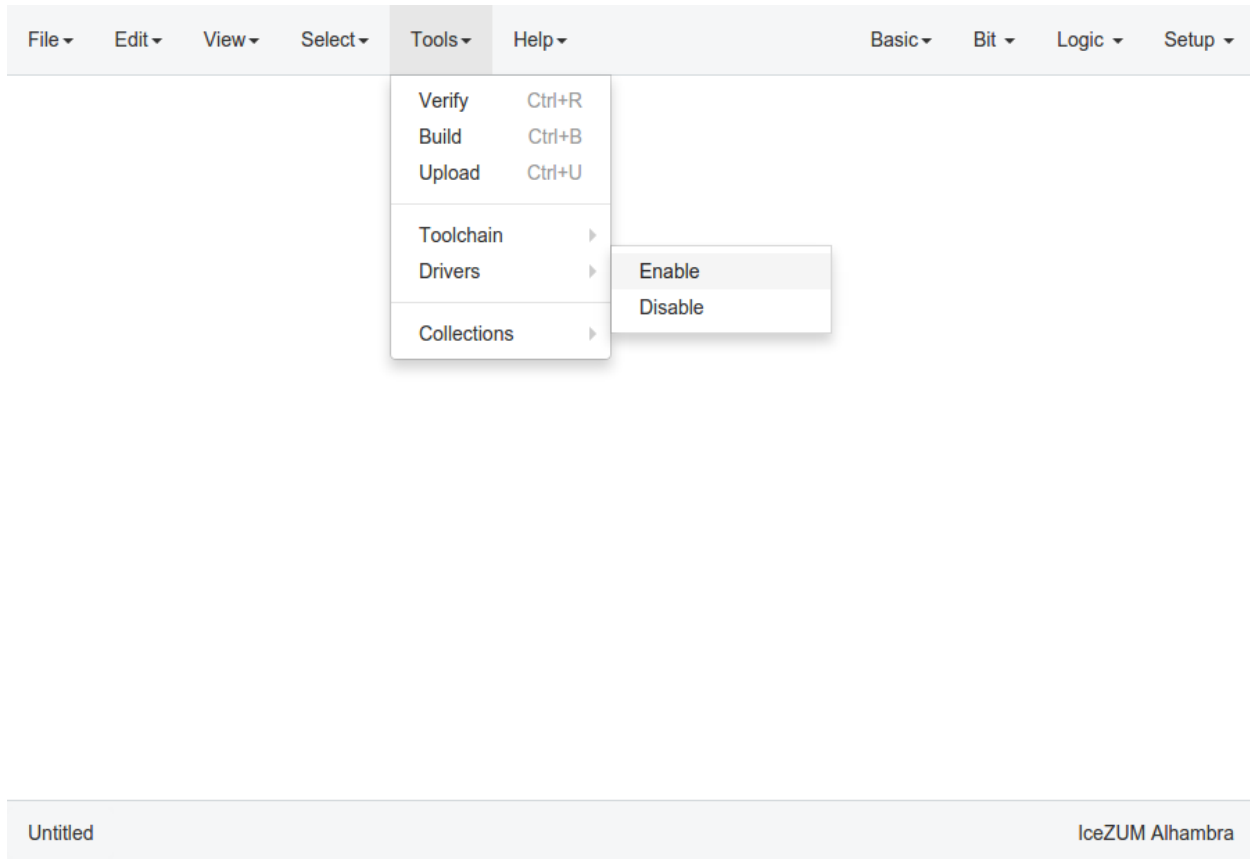
Go to **Tools > Toolchain > Install**



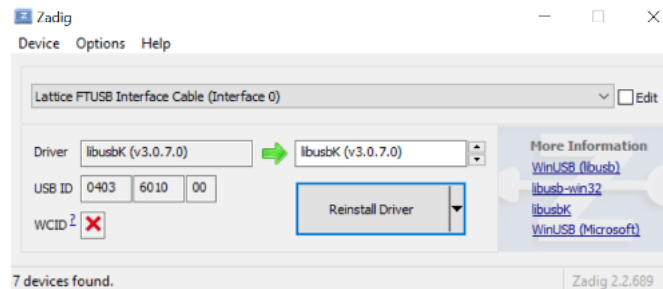
Apio backend and all its needed packages will be installed. This operation does not require Internet connection.

2.2 Setup the drivers

Connect your FPGA board and select **Tools > Drivers > Enable**. This operation requires **administrator privileges**.



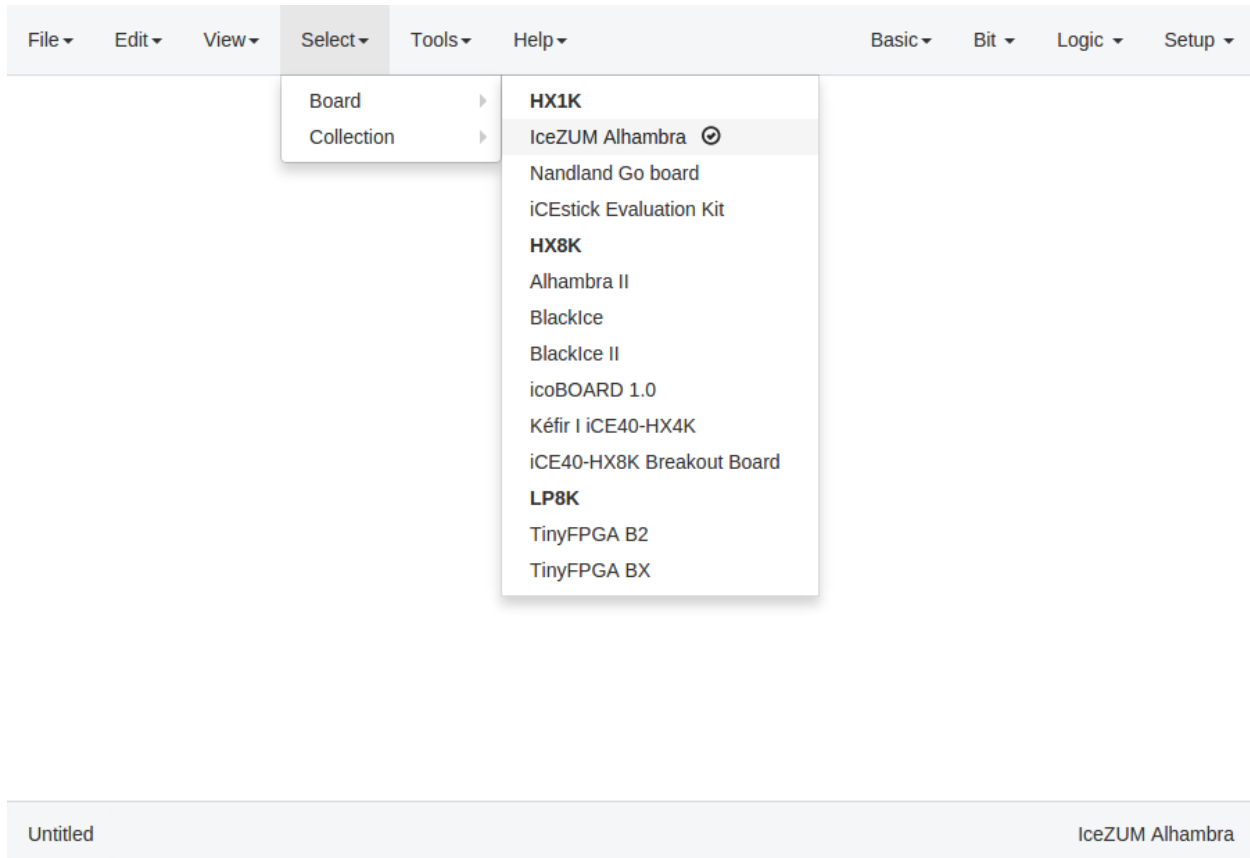
Note: In Windows, an external application (Zadig) is launched to replace the existing FTDI driver of the **Interface 0** by **libusbK**.



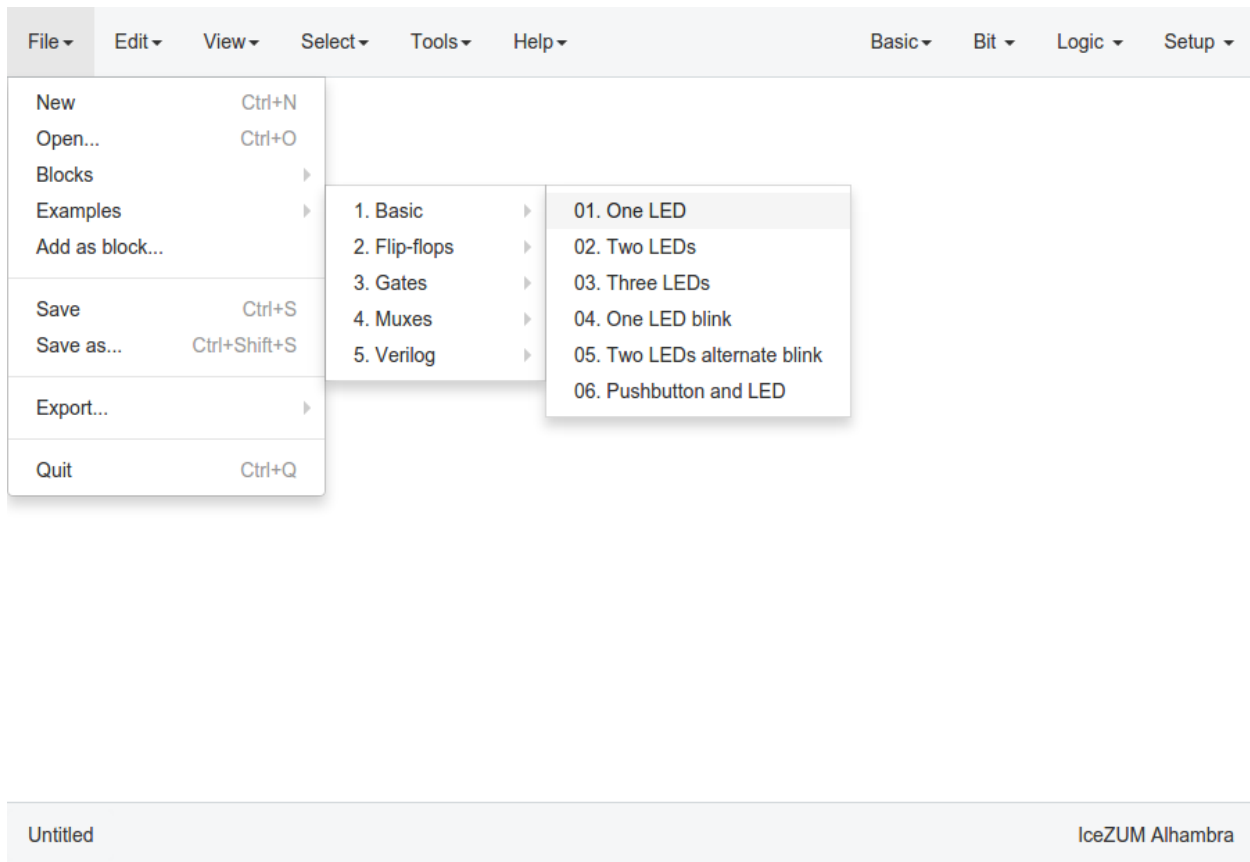
In MacOS this operation requires Internet connection to allow *Homebrew* to install *libffi* and *libftdi* packages.

2.3 Upload a design

Go to **Select > Board > IceZUM Alhambra**



Go to **File > Examples > 1. Basic > 01. One LED**



File ▾ Edit ▾ View ▾ Select ▾ Tools ▾ Help ▾ Basic ▾ Bit ▾ Logic ▾ Setup ▾

Turn on a LED



01. One LED

IceZUM Alhambra

Then, you can verify, build or upload the project in **Tools > Verify | Build | Upload**.

File ▾ Edit ▾ View ▾ Select ▾ Tools ▾ Help ▾

Basic ▾ Bit ▾ Logic ▾ Setup ▾

Turn on a LED

1

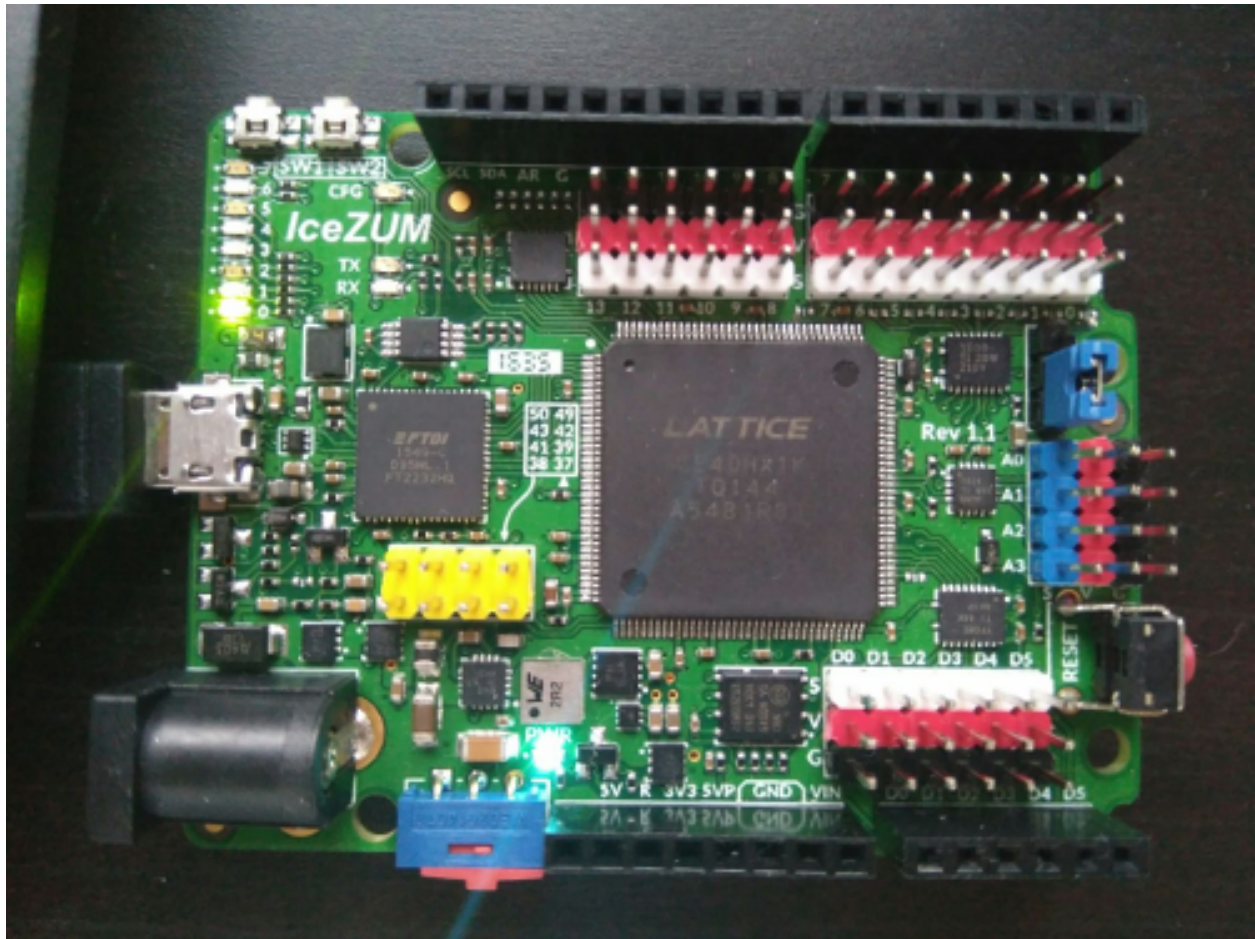
LED
LED0 × ▾

Upload done

01. One LED

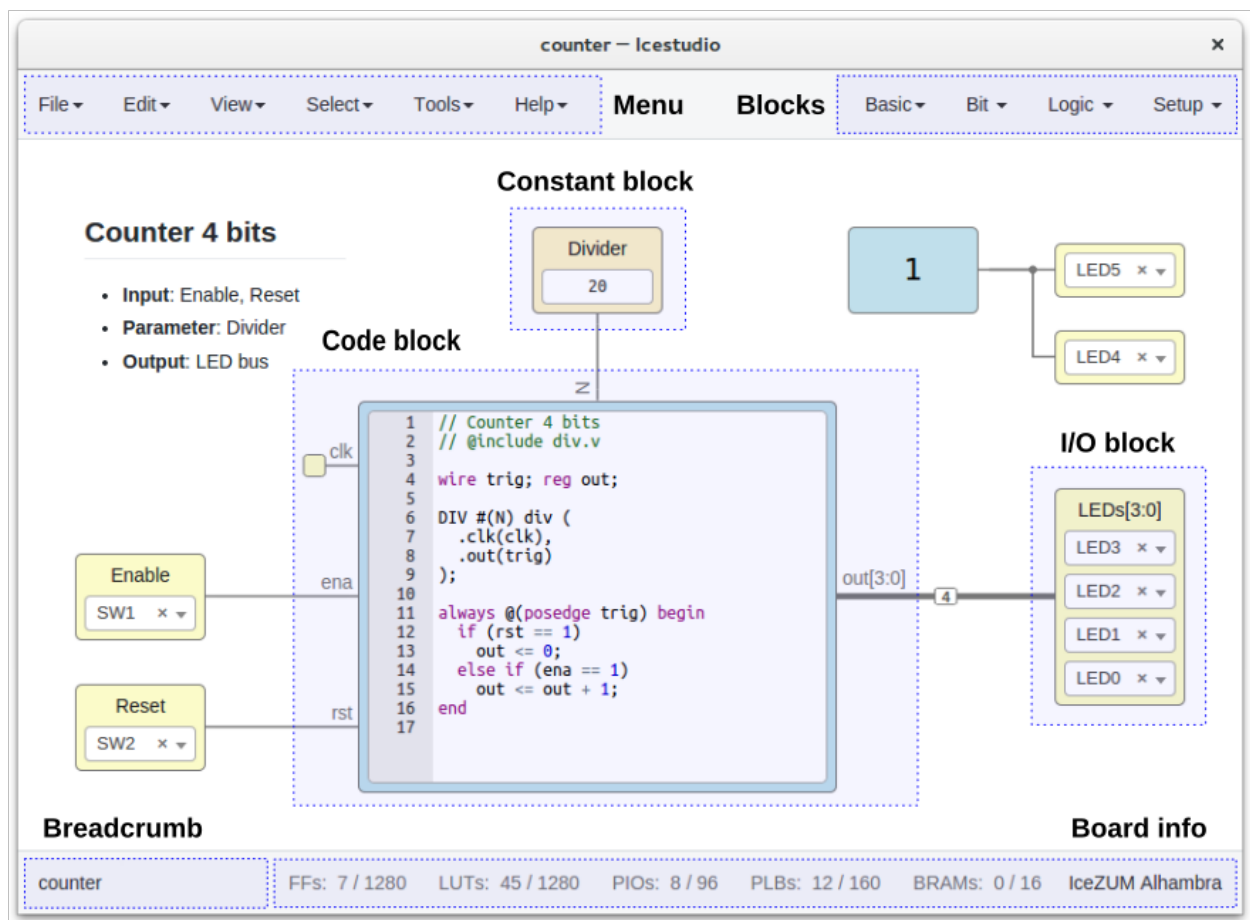
IceZUM Alhambra

Here is the FPGA board with the **LED0** turned on.



CHAPTER 3

User Guide



3.1 Menu

3.1.1 File

Action	Description	Shorcuts	Mac OS
New	Create a new window	Ctrl+N	+N
Open...	Open a project	Ctrl+O	+O
Blocks	Open the blocks from the selected collection		
Examples	Open the examples from the selected collection		
Add as block...	Add a project as a block		
Save	Save the current project	Ctrl+S	+S
Save as...	Save the current with a new name	Ctrl+Shift+S	Shift++S
Export...	Export multiple output files		
Quit	Close the application	Ctrl+Q	+Q

Note: The exportable files are: Verilog, PCF, Testbench, GTKWave, BLIF, ASC and Bitstream

3.1.2 Edit

Action	Description	Shortcut	Mac OS
Undo	Undo the last change	Ctrl+Z	+Z
Redo	Revert the last undo	Ctrl+Y Ctrl+Shift+Z	+Y Shift++Z
Cut	Cut selected blocks	Ctrl+X	+X
Copy	Copy selected blocks	Ctrl+C	+C
Paste	Paste copied blocks	Ctrl+V	+V
Select all	Select all the blocks	Ctrl+A	+A
Fit content	Fit the content into the screen	Ctrl+l	+l

Preferences

Language

The supported languages are:

- English
- Spanish
- French
- Chinese
- Galician
- Basque
- Catalan

Board rules

Enable or disable globally the board rules. These rules allow to automate tasks such as default port connections or default pin values. For example, the IceZUM Alhambra rules are:

- All not connected “clk” ports are connected to the internal CLK signal.
- All not used LEDs are turned off.

More information in the [Board rules](#) section.

External collections

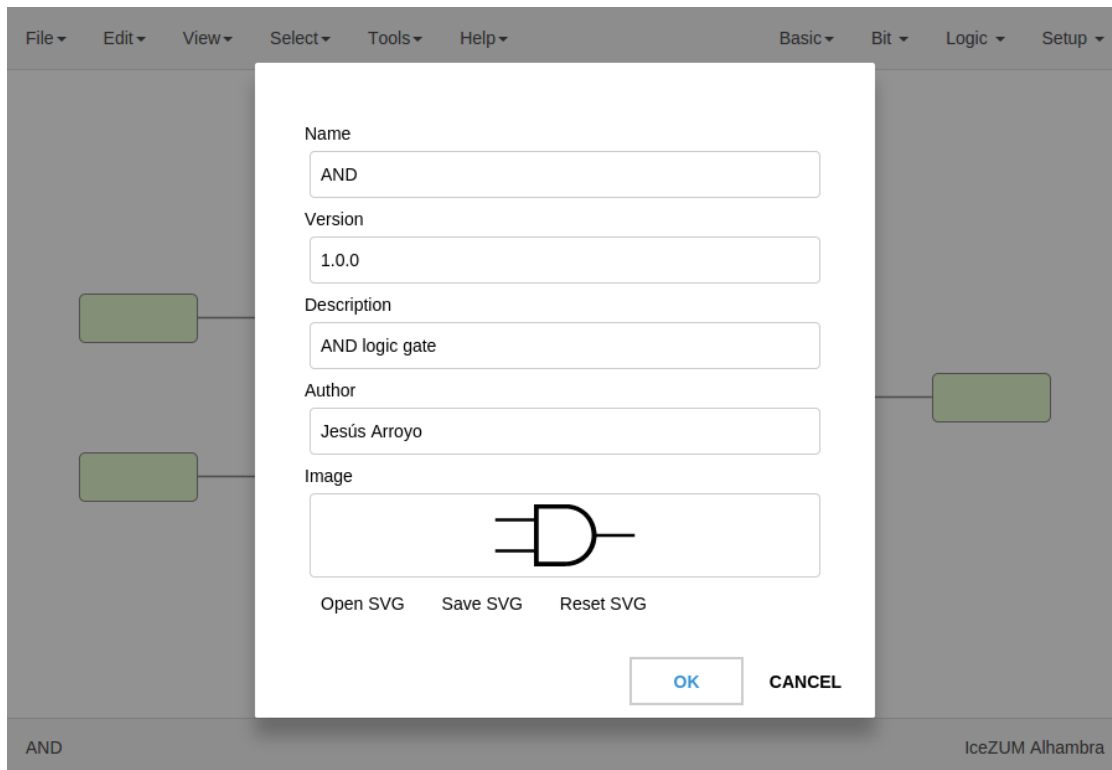
Set the external collections path in your system. This absolute route should contain Icestudio collections (directories or symbolic links to directories) that will be loaded as external collections.

Remote hostname

Set the hostname of a remote device with an FPGA board connected. The format is `user@host`. For example, `pi@192.168.0.22`. Verify, Build and Upload functions will be executed in this host, that must have apio pre-configured.

Project information

Contains all the information about the project:



- Name: project name
- Version: version number
- Description: information shown in the block tooltip
- Author: creator of the project
- Image: SVG shown in the block body

3.1.3 View

Action	Description
PCF	Show the board PCF file
Pinout	Show the board SVG pinout
Datasheet	Open a web browser with the board datasheet
Board rules	Show the current board rules
Collection info	Show the current collection README file
Toolchain output	Show the output of the latest executed command
FPGA resources	Show / Hide the used FPGA resources

3.1.4 Select

Board

Select the FPGA board. The supported boards are:

HX1K

- IceZUM Alhambra
- Nandland Go board
- iCEstick Evaluation Kit

HX8K

- Alhambra II
- BlackIce
- BlackIce II
- icoBOARD 1.0
- K fir I iCE40-HX4K
- iCE40-HX8K Breakout Board

LP8K

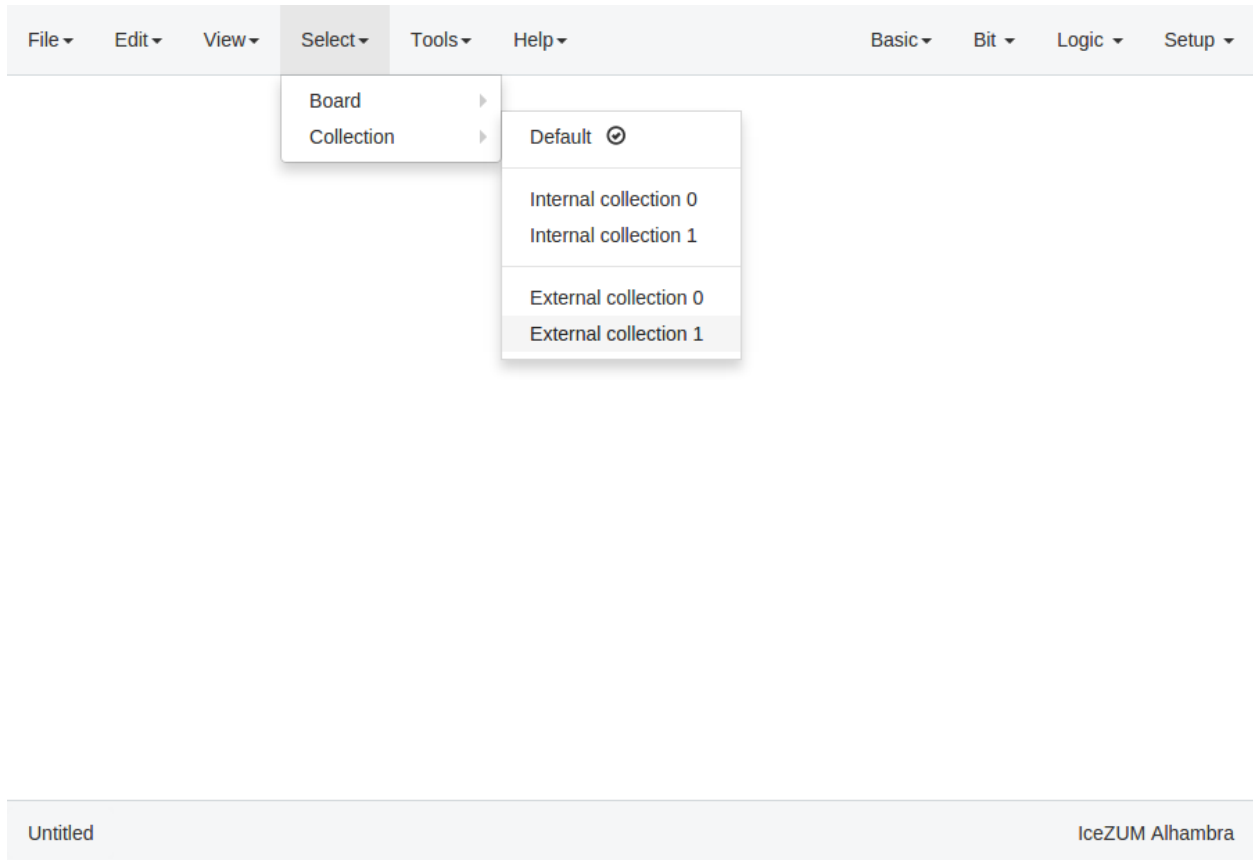
- TinyFPGA B2
- TinyFPGA BX

When a board is selected all I/O block combos are updated and its current values reset.

Hint: This information is stored in the *app/resources/boards* directory. In order to support a new board just create a new directory with the *info.json*, *pinout.pcf* and *pinout.svg* (optional) files with its information. The *pinout.json* file must be generated from the *pinout.pcf* using the *generator.py* script.

Collection

Select the Collection from the internal collections installed (from **Tools > Collections**) or the external collections (from **Edit > Preferences > External collections**).



A collection is composed by **blocks** and **examples**. When a collection is selected, the following sections are updated with the collection content:

- File > Blocks
- File > Examples
- Menu blocks

Note: The *Default collection* is always available, and contains the blocks and examples distributed within the application.

3.1.5 Tools

Action	Description	Shortcut	Mac OS
Verify	Check the generated verilog code	Ctrl+R	+R
Build	Synthesize the bitstream from the design	Ctrl+B	+B
Upload	Synthesize (if required) and upload the bitstream to the FPGA	Ctrl+U	+U

Toolchain

Action	Description
Install/Update	Install a virtualenv, apio and the required apio packages. It requires Python 2.7
Remove	Remove the toolchain directories
Reset default	Restore the default toolchain distributed within Icestudio
Apio version	Show the current apio version

Drivers

Action	Description
Enable	Launch the FTDI drivers configuration. Each OS has a different process
Disable	Revert the FTDI drivers configuration. Each OS has a different process

Collections

Action	Description
Add	Add a ZIP file with one or more collections
Reload	Reload all the collections from the files
Remove	Remove the selected collection
Remove all	Remove all the collections

Note: A collection is composed by **blocks** and **examples** sorted by categories (directories). The **package.json** file is required and contains information about the collection. The **locale** directory is optional and contains the translations for the blocks and examples. A collection has the following structure:

```
Collection/  
├── blocks/  
├── examples/  
├── locale/  
├── LICENSE  
├── package.json  
└── README.md
```

A ZIP file of collections contains one or more *Collection directories* at the main level. A collection can be selected in **Select > Collections**.

More information in the [Default collection](#).

Hint: When a ZIP file of collections is added to Icestudio, it is installed in `~/.icestudio/collections`.

3.1.6 Help

Action	Description
View license	Open the Icestudio license in a web browser
Version	Show the Icestudio current version
Documentation	Open the Icestudio documentation in a web browser
Source code	Open the Icestudio source code in a web browser
Community forum	Open the FPGAwars forum in a web browser
About Icestudio	Information about the application

3.2 Blocks menu

3.2.1 Basic

It contains the basic blocks:

- **Input**: show a dialog to insert the name and type of the input block.
- **Output**: show a dialog to insert the name and type of the output block.
- **Constant**: show a dialog to insert the name and type of the constant block.
- **Memory**: show a dialog to insert the name and type of the memory block.
- **Code**: show a dialog to insert the ports and parameters of the code block.
- **Information**: create an empty text box block.

Note: **Input** and **output** ports can be set to **virtual**. Virtual ports are used to independent-FPGA projects. Also, they can be configured as a **bus** by adding the notation `[x:y]` to the port name.

Note: **Constant** and **memory** blocks can be set to **local**. Local parameters are not exposed when the project is added as a block.

Hint: Multiple **input**, **output**, **constant** and **memory** blocks can be created using the *comma* separator. For example: `x, y, z` will create 3 blocks with those names. FPGA I/O ports values are set in the block combo box. These values can be set by searching and also unset by doing click on the cross. Double click over **input**, **output**, **constant** or **memory** block allows to modify the block name and type. In **code** block ports definition, multiple *input* and *output* ports, and *parameters*, can be created also using the *comma* separator.

3.2.2 Blocks

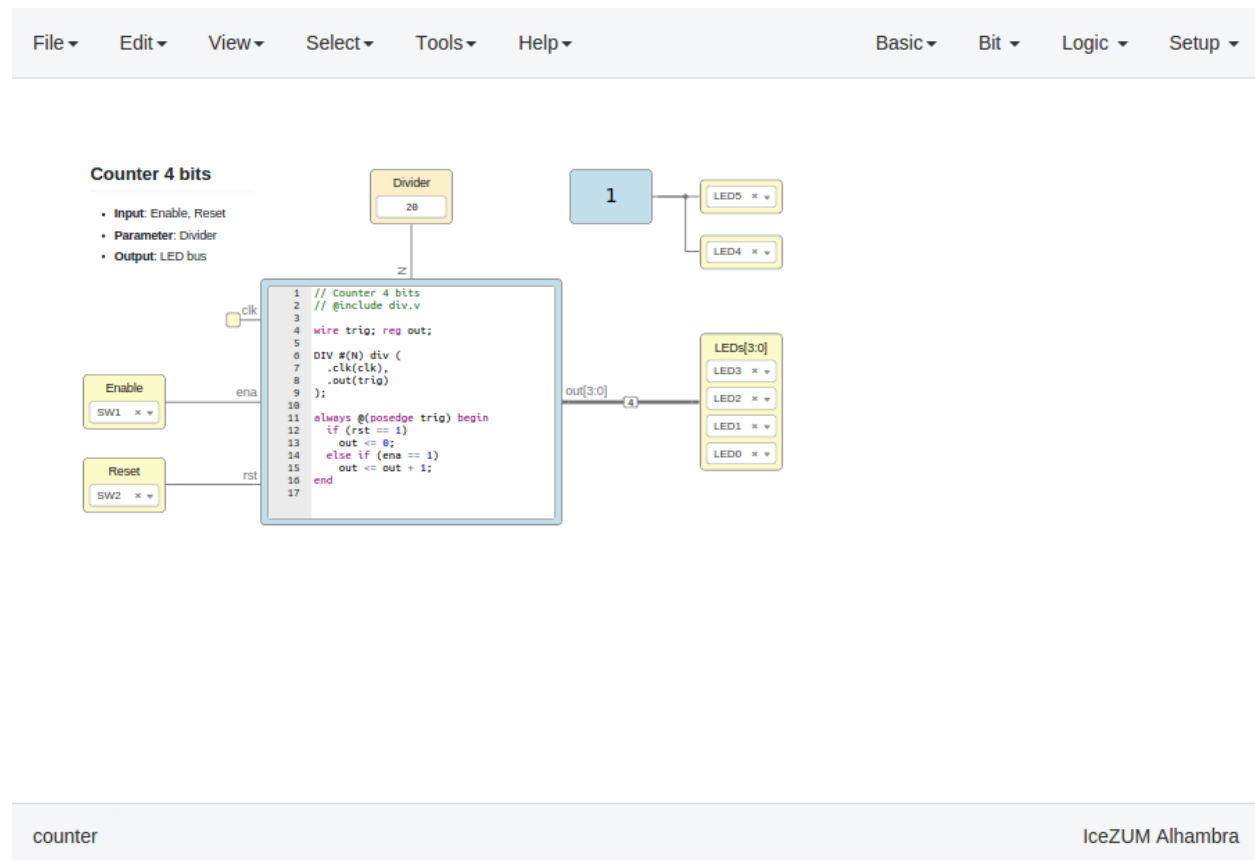
It contains all stored blocks sorted by categories. This menu is generated when the application starts. It can show the Default collection or any installed collection.

3.3 Design

This is the main panel. It contains the blocks and the wires.

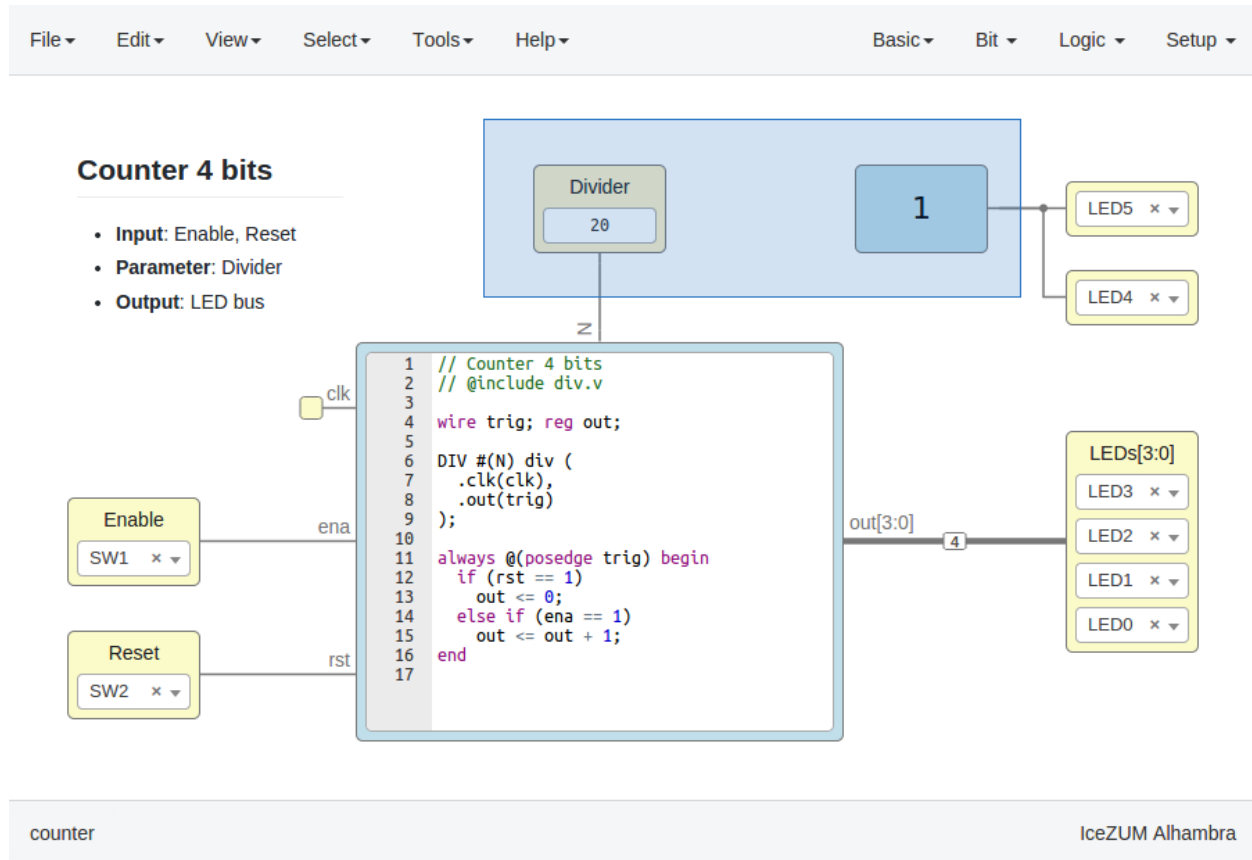
3.3.1 Pan & Zoom

Pan is performed using the **Ctrl + mouse left button** or **mouse right button** over the background. Zoom is performed using **mouse wheel**.



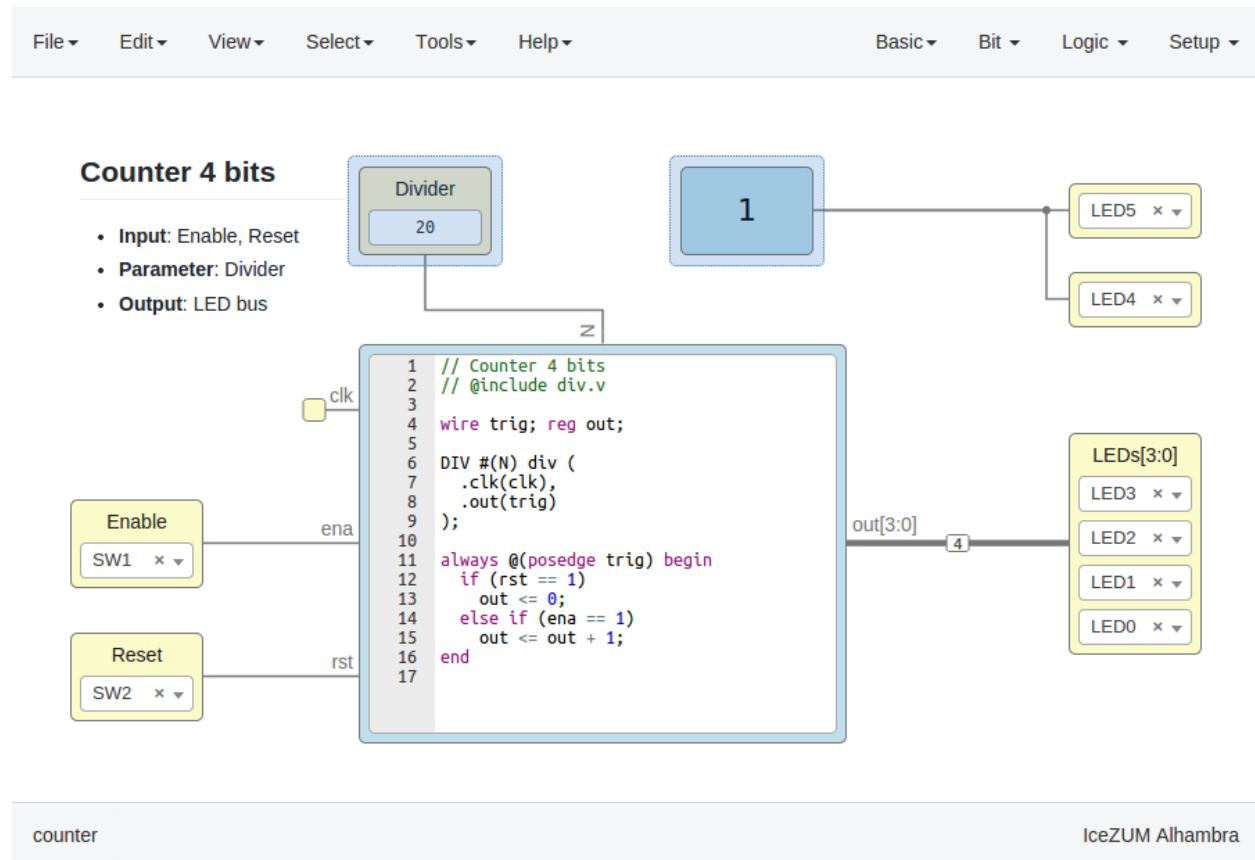
3.3.2 Select

Block selection is performed with the **mouse left button**. Blocks can be **selected/unselected** individually using **Shift+left-click**. In addition, several blocks can be selected by a **selection box**. A selection is canceled when the background is **left-clicked**.



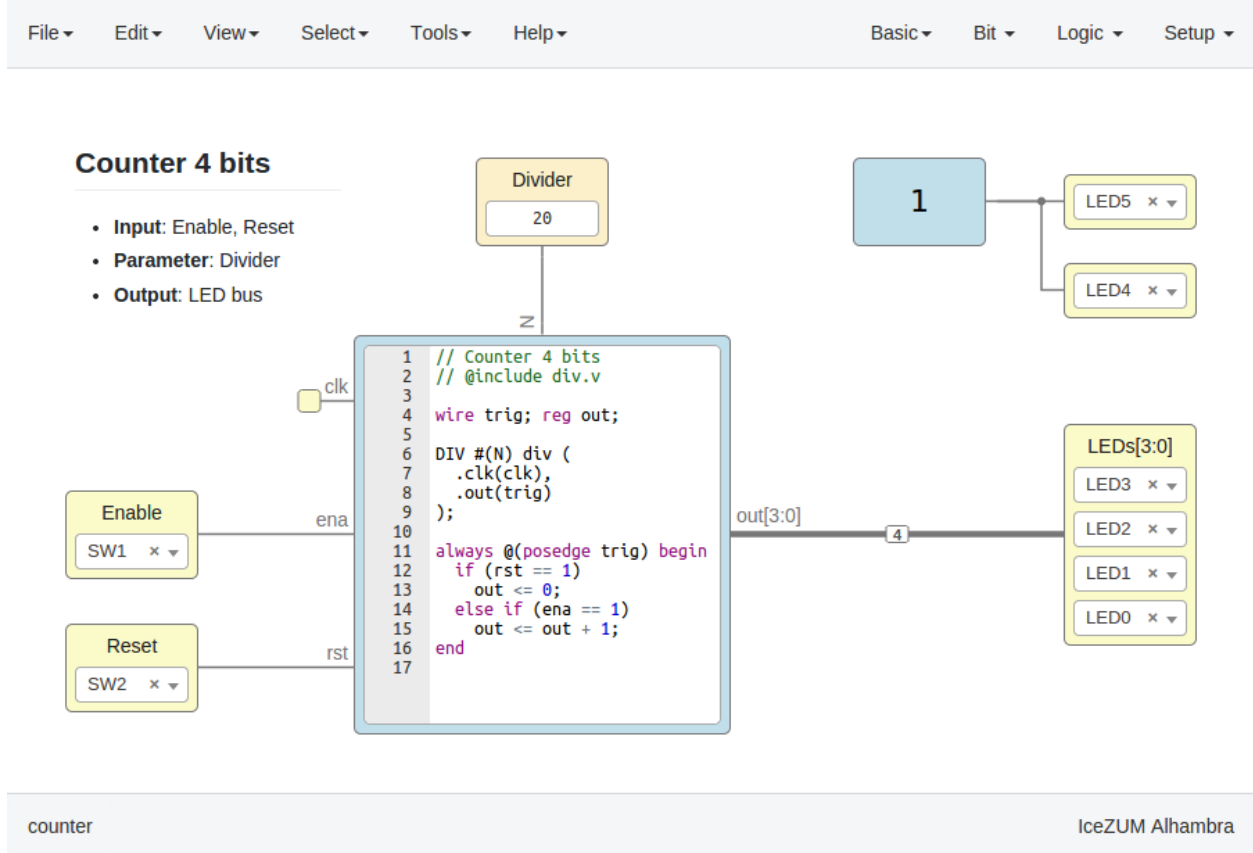
3.3.3 Move blocks

Any block or blocks selection can be moved in the design using the **mouse left button** over the block or the selection. Also a blocks selection can be moved with the **arrow keys**.



3.3.4 Resize text blocks

Memory, **Code**, and **Information** blocks can be resized with the resize tool in the **bottom-right corner** of the block.



3.3.5 Block examination

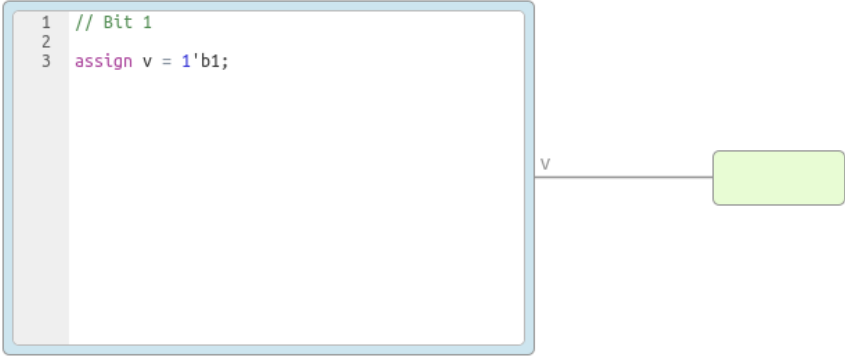
Non-basic blocks can be read only examined by **double clicking** the block using the **mouse left button**. This is a recursive action. In order to go back, click on the **< back** link or press the **back key**.

During the examination, pan, zoom and code navigation are enabled. Also the 'Fit content' action.

File ▾ Edit ▾ View ▾ Select ▾ Tools ▾ Help ▾ Basic ▾ Bit ▾ Logic ▾ Setup ▾

Bit 1 - v1.0.0 Read only
[< back](#)

```
1 // Bit 1
2
3 assign v = 1'b1;
```

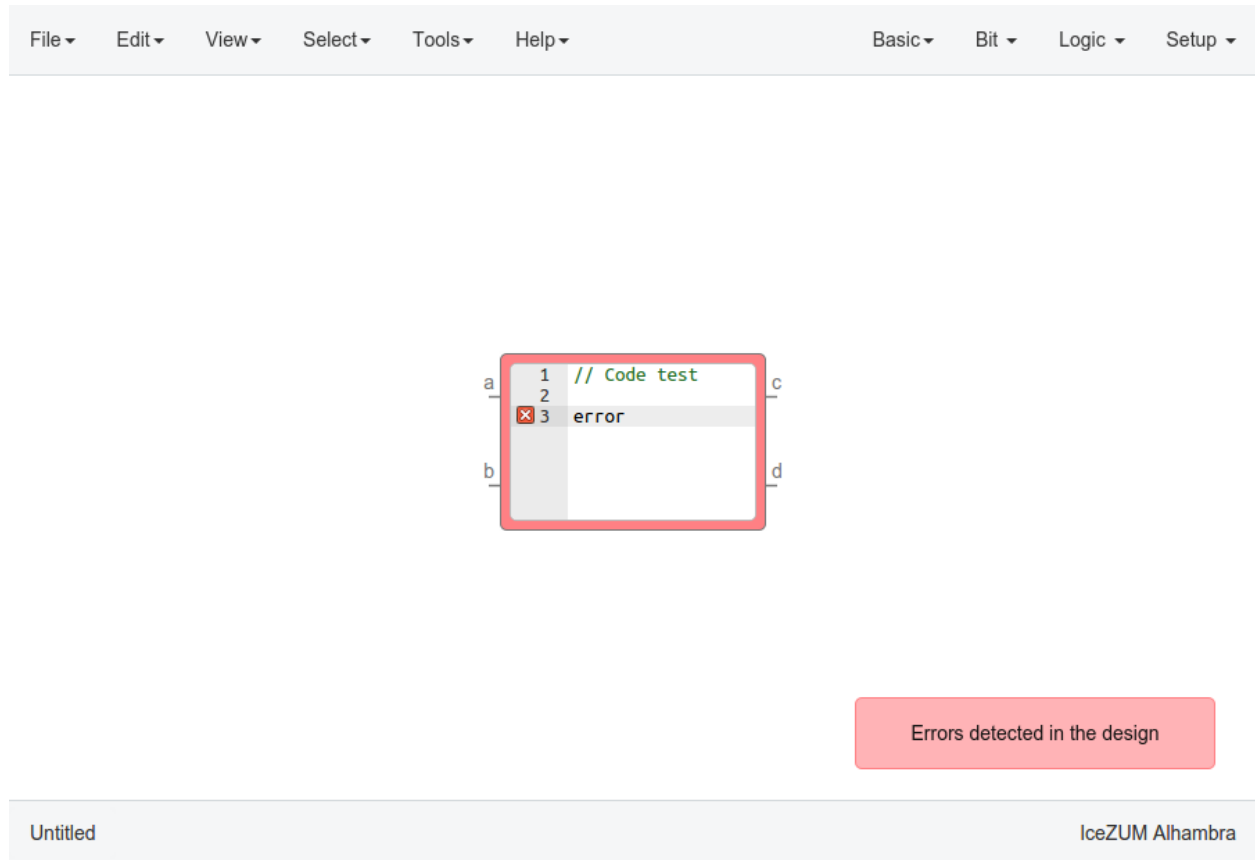


[counter](#) / Bit 1 IceZUM Alhambra

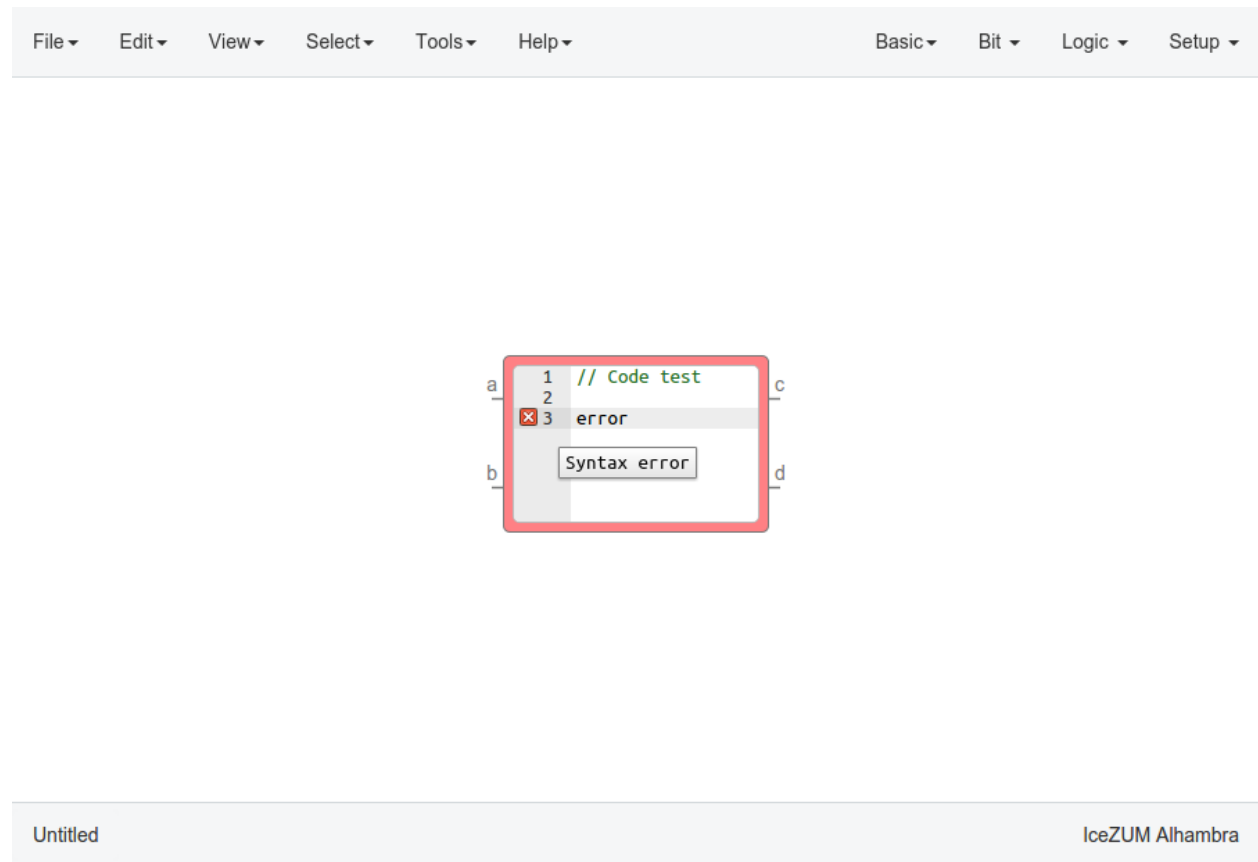
Note: The examination path is stored in the **breadcrumbs**. This allows to go back to any previous block.

3.3.6 Verilog error detection

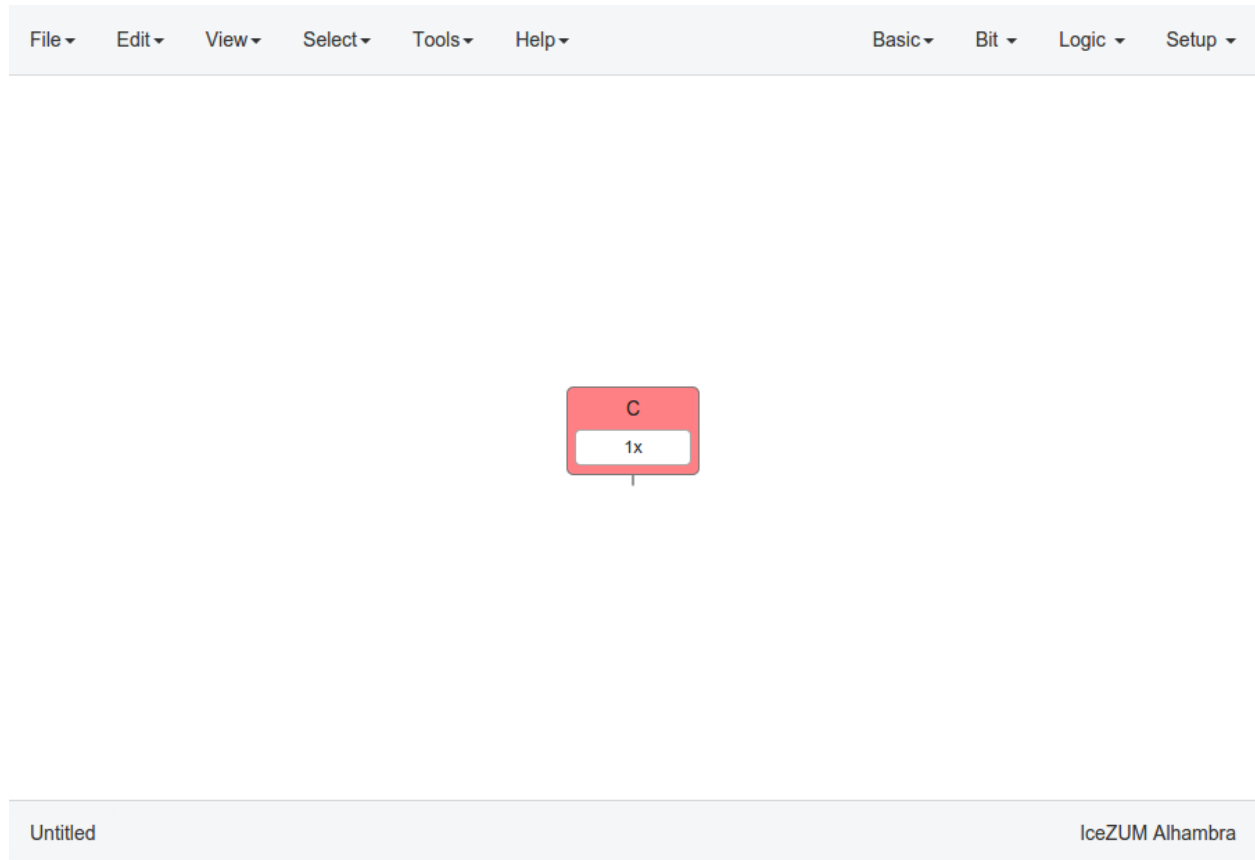
The Verify, Build and Upload errors are captured and showed in the design with a notification.



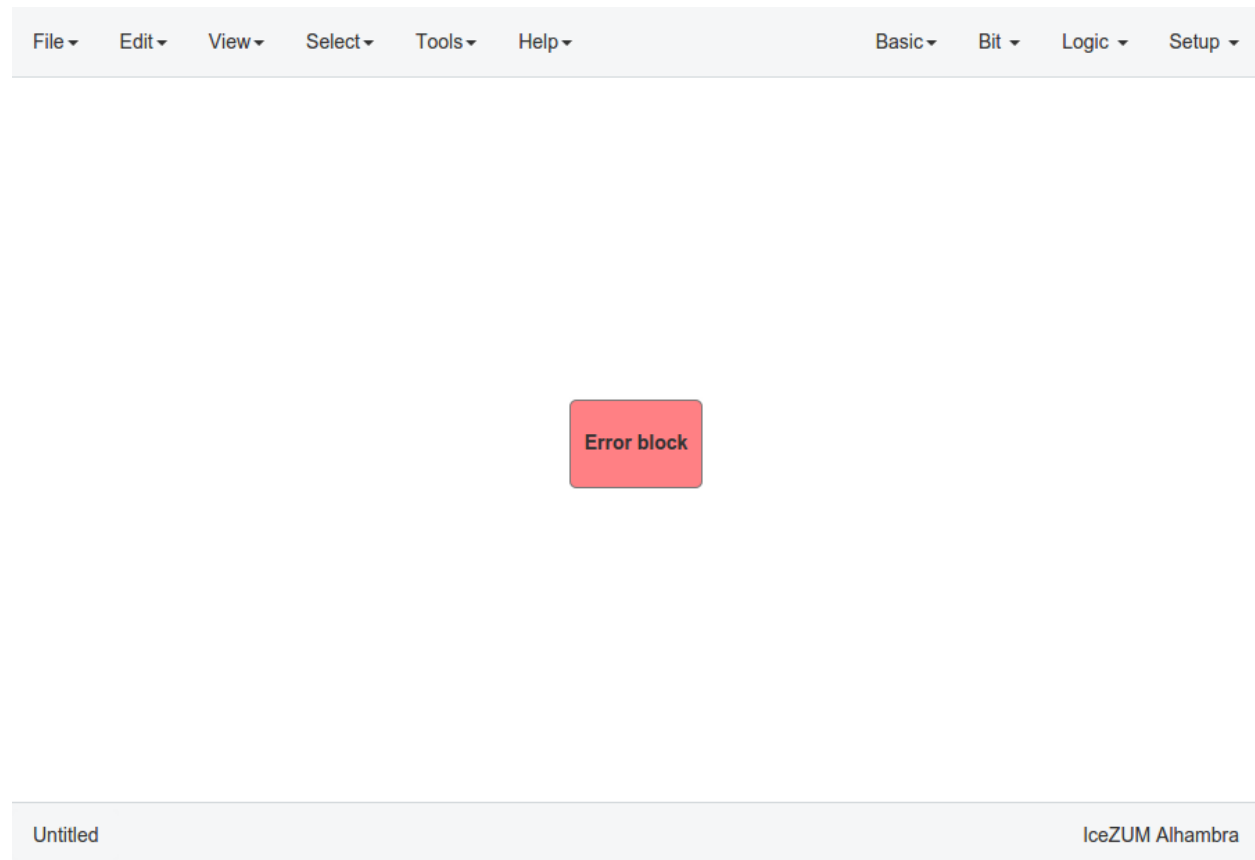
If the error comes from a Code block it is marked in red an inline annotation is set:



If the error comes from a Constant block it is marked in red.



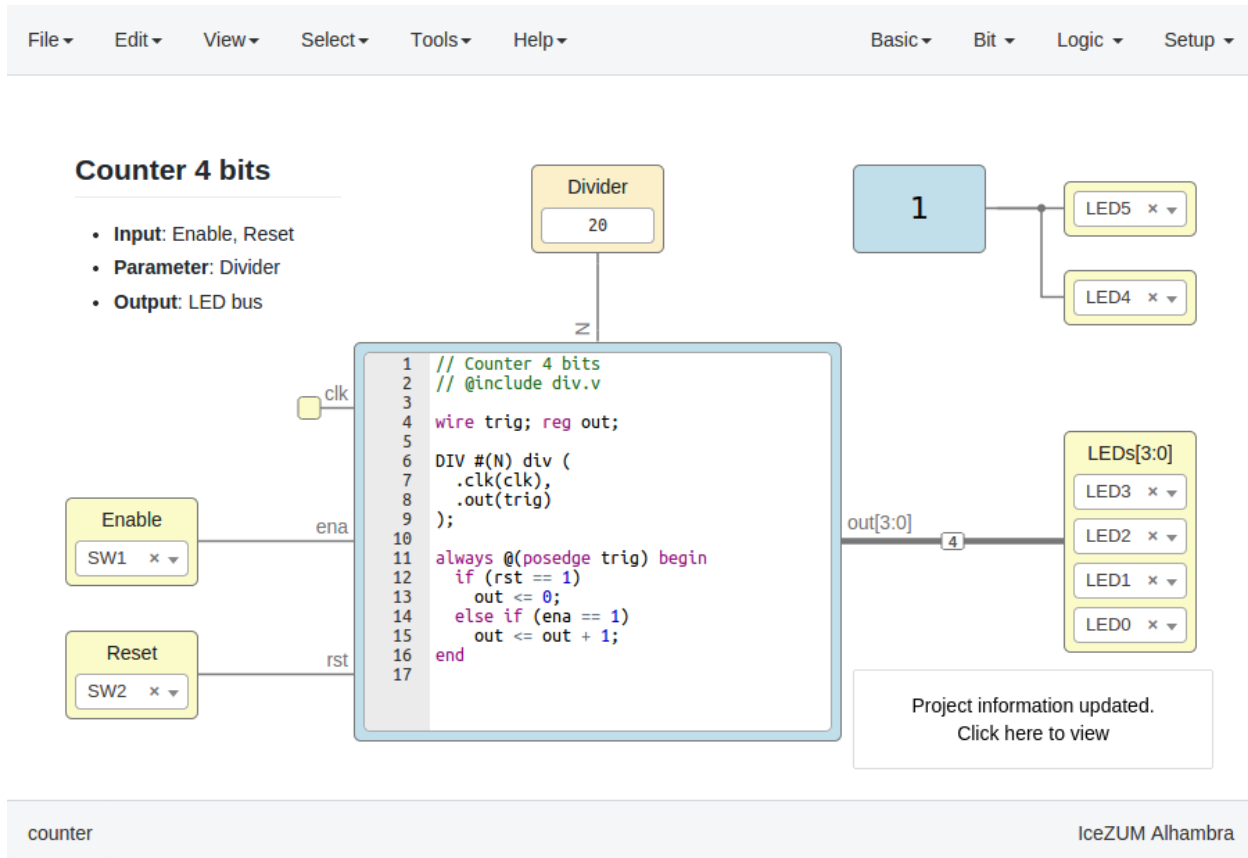
If the error comes from a Generic block it is marked in red.



3.3.7 Undo/Redo

Icestudio allows to undo/redo the following actions:

- Add or remove a block.
- Add or remove a wire.
- Move a block or a blocks selection.
- Edit an I/O block: name, type and value.
- Edit a Constant block: name, type and value.
- Edit a Memory block: name, type, format and value.
- Edit a Code block: ports, parameters and content.
- Edit an Information block: type and content.
- Change the board.
- Change the language.
- Change the Project information: when undo/redo a Project information change, a clickable notification will appear to access the Project information section.



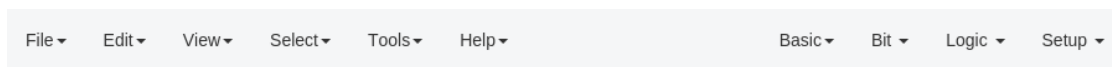
3.4 Take a snapshot

Taking a **png** snapshot of the application is as easy as press **Ctrl+P**. A save dialog appears to set the name and the path of the captured image.

4.1 Install the toolchain

1. **Install Python 2.7**
2. **Launch the toolchain installation process**

Go to **Tools > Toolchain > Install**. Be patient for the toolchain installation.



Toolchain installed

Click here to **setup the drivers**

Untitled

IceZUM Alhambra

Note: When the toolchain is installed, the menu option changes to **Tools > Toolchain > Update**. Also, the toolchain can be restored to default in **Tools > Toolchain > Reset default**.

4.2 Update the toolchain

1. **Connect to the Internet**
2. **Launch the toolchain updating process**

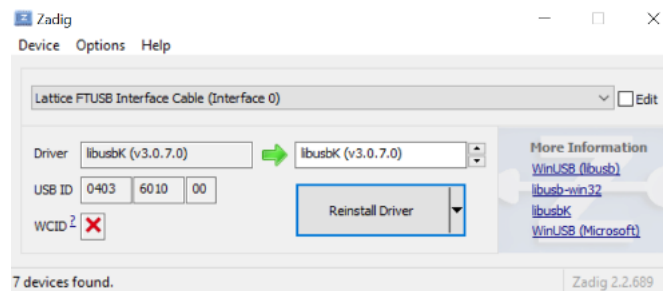
Go to **Tools > Toolchain > Update**. Be patient for the toolchain update.

4.3 Install the drivers

1. **Install the toolchain** (required for Windows)
2. **Enable the FTDI drivers**

Go to **Tools > Drivers > Enable**. Each OS has a different process. This configuration requires administration privileges.

Note: In Windows, an external application (Zadig) is launched to replace the existing FTDI driver of the **Interface 0** by **libusbK**.



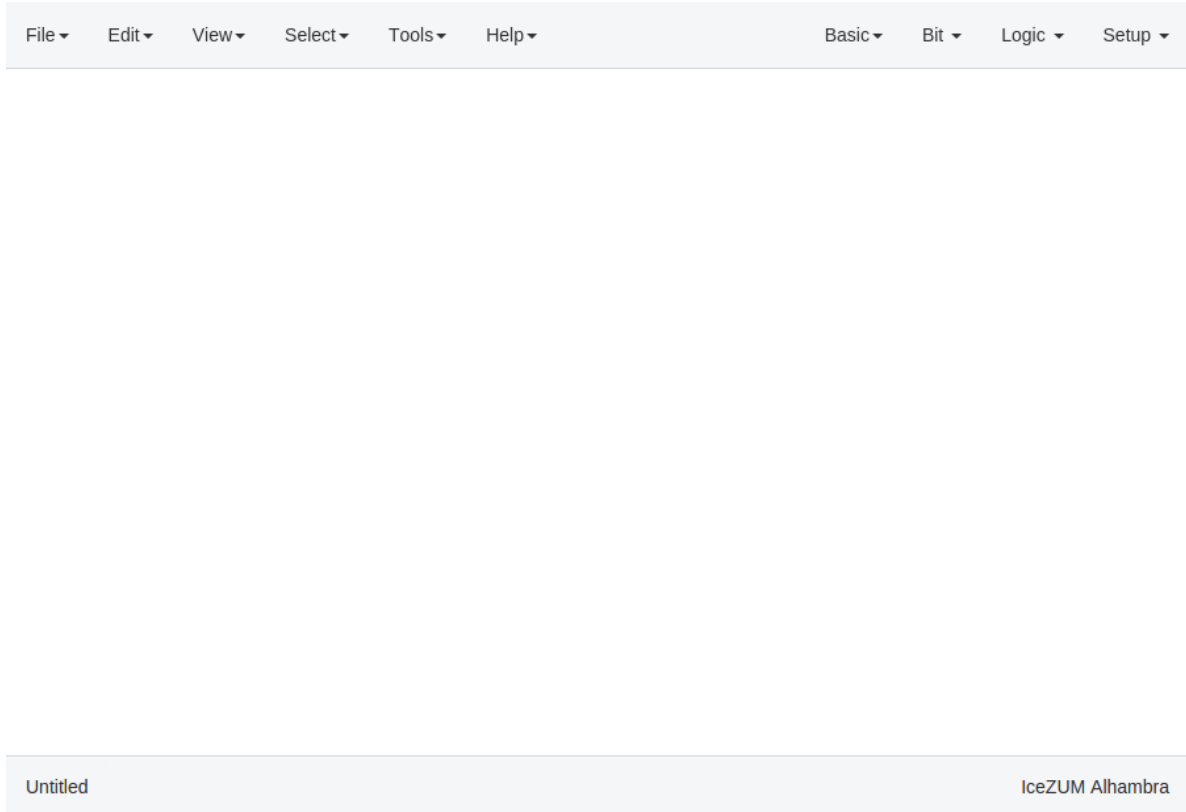
In MacOS this operation requires Internet connection to allow *Homebrew* to install *libffi* and *libftdi* packages.

Hint: To revert the drivers configuration go to **Tools > Drivers > Disable**

4.4 Create a project

1. Create a new project

Go to **Edit > New**. A new window will be opened.



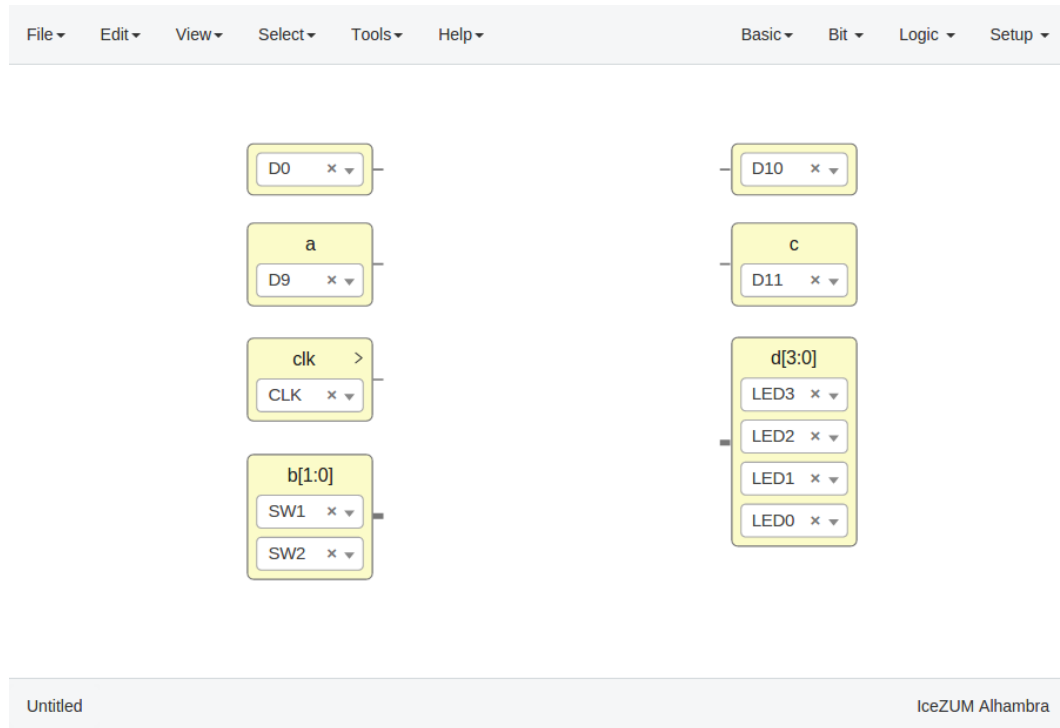
2. Add blocks

There are different types of blocks:

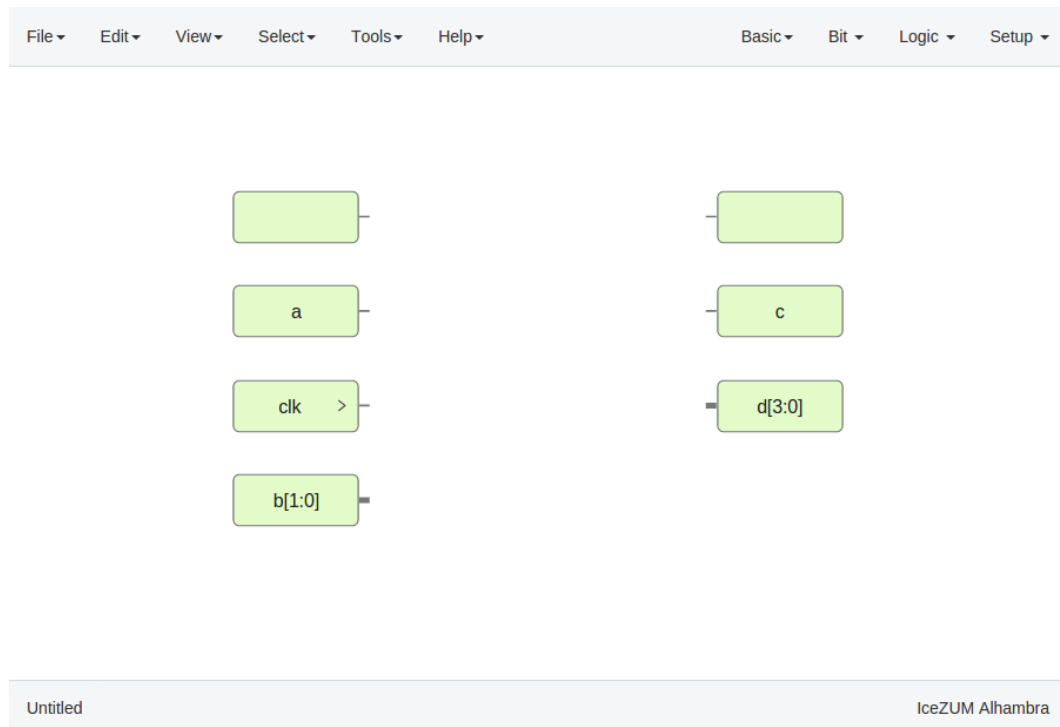
1. *Input/Output blocks*

Click on **Basic > Input** or **Basic > Output**, write the block's name and press OK or Enter.

Also, it can be configured as **buses** using the $[x:y]$ notation (x is the most significant bit).



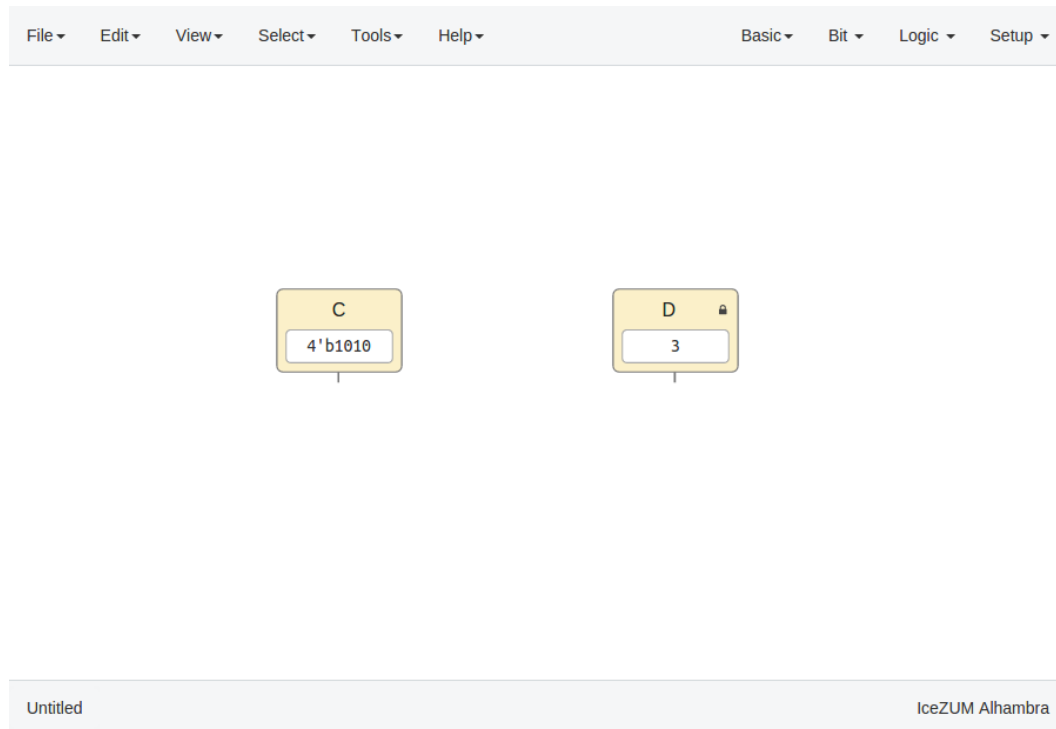
If these blocks are used to build generic blocks, they should be configured as **virtual** (green). Then, the FPGA pin selector won't be shown.



2. Constant blocks

Click on **Basic > Constant**, write the block's name and press OK or Enter.

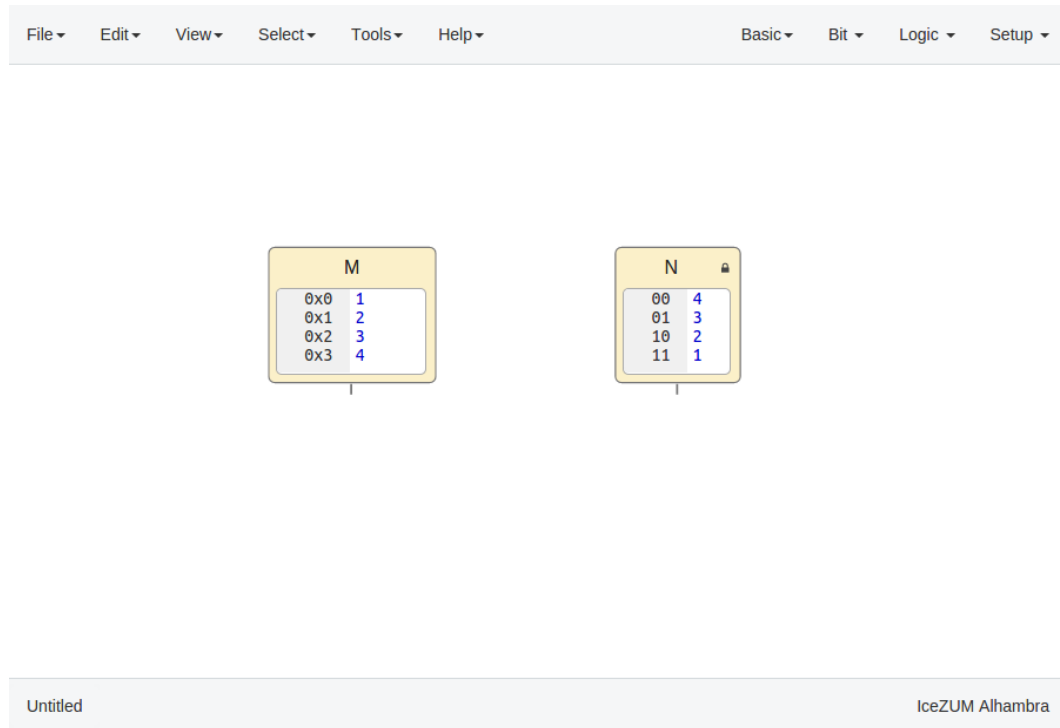
These blocks can be configured as **local**. Then, this parameter won't be exported.



3. Memory blocks

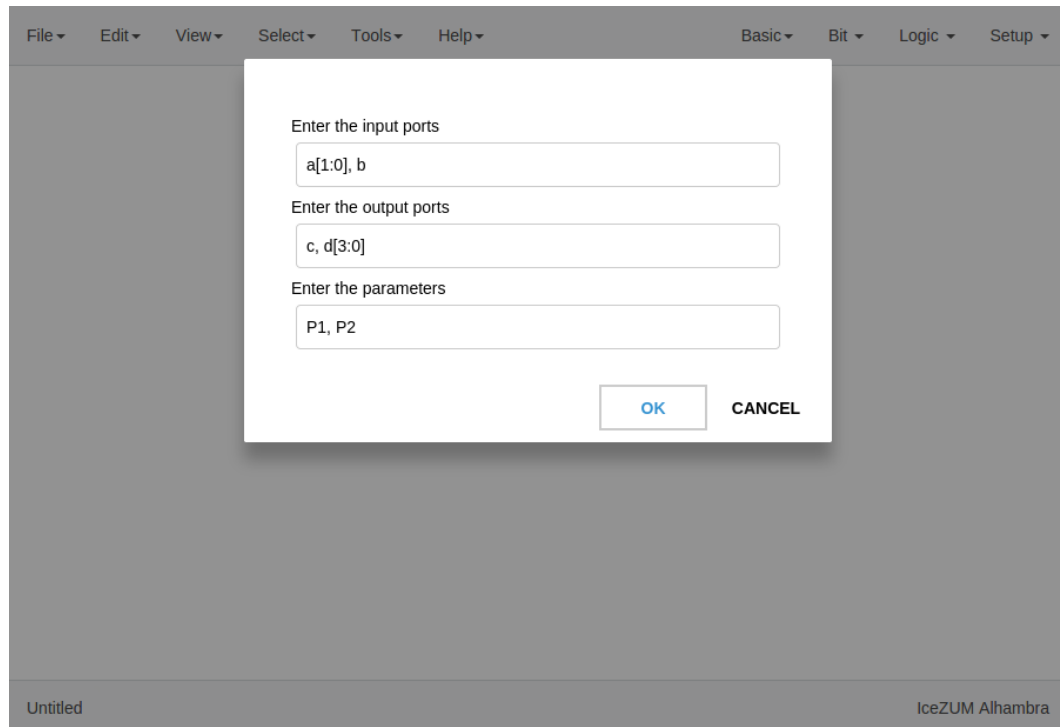
Click on **Basic > Memory**, write the block's name and press OK or Enter.

These blocks can be configured as **local**. Then, this parameter won't be exported. Also you can update the **address format** of the memory to be *binary*, *decimal* or *hexadecimal*.



4. Code blocks

Click on **Basic > Code**, add the code ports. Port names are separated by a comma. E.g.: a, b.



This block contains a text editor to write your module in verilog code. Module header and footer are not required.

File
Edit
View
Select
Tools
Help
Basic
Bit
Logic
Setup

Untitled
IceZUM Alhambra

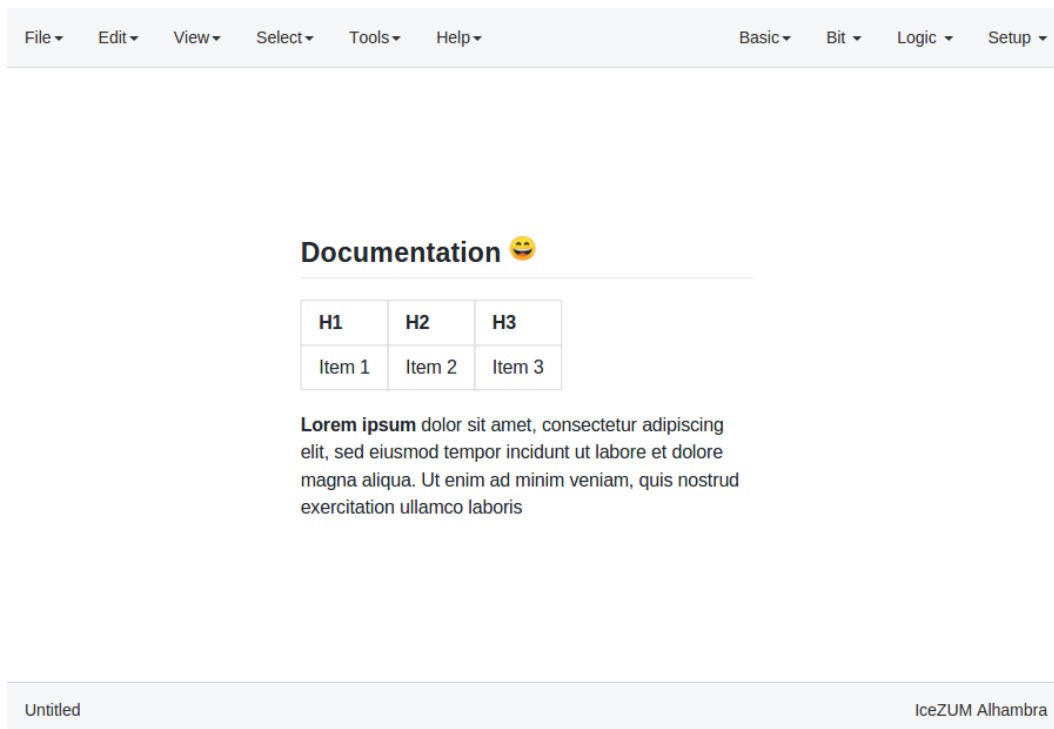
5. Info blocks

Click on **Basic > Info**.

This block contains a text editor to add comments about the project in **Markdown** or **HTML**.



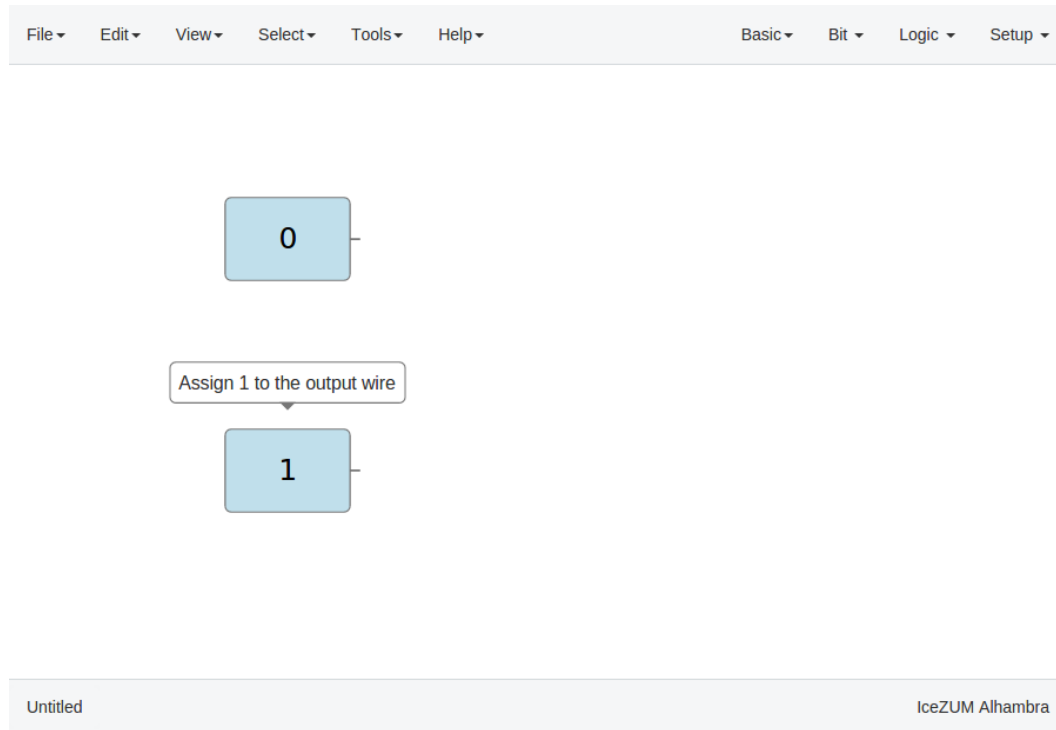
It can be rendered simply by double-clicking the block.



6. Bit blocks

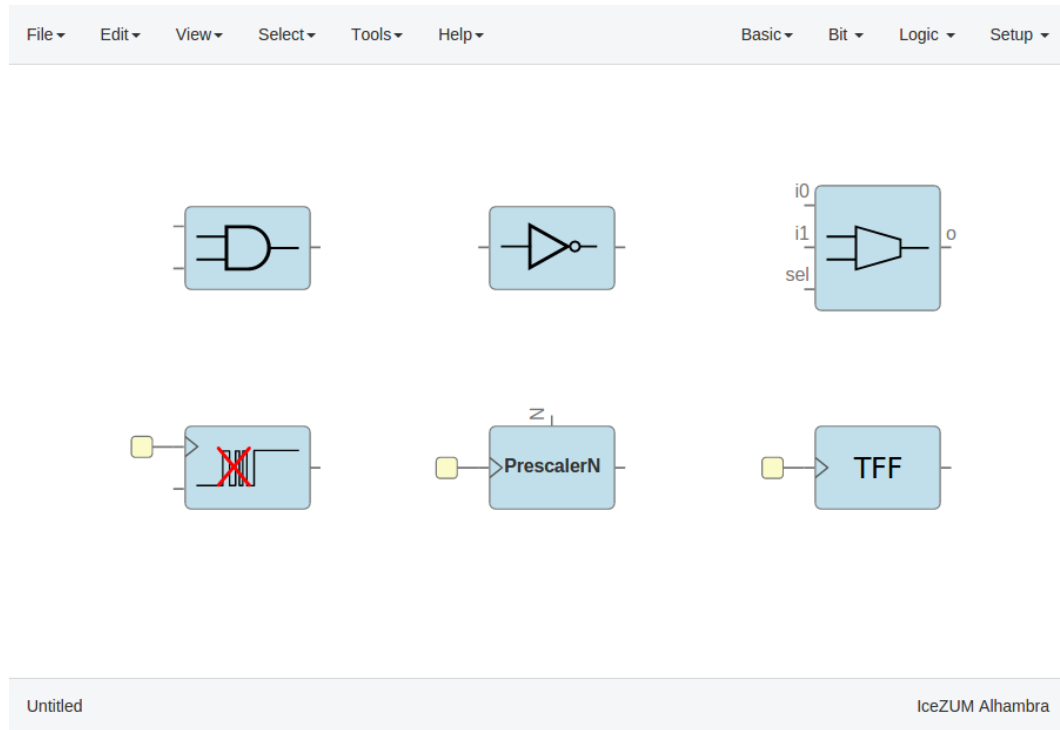
Click on **Bit > 0** or **Bit > 1**.

These blocks are low and high logic drivers.



7. Logic blocks

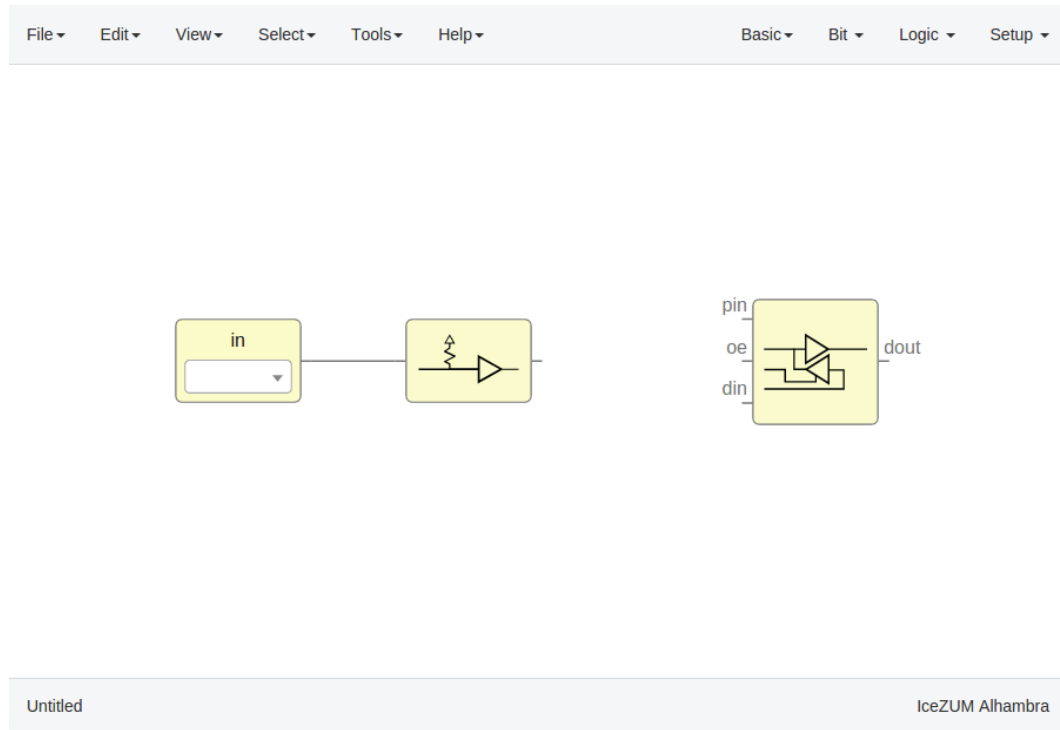
Go to the **Logic** menu and select a block. This menu contains **Gates**, **Combinational blocks** and **Sequential blocks**.



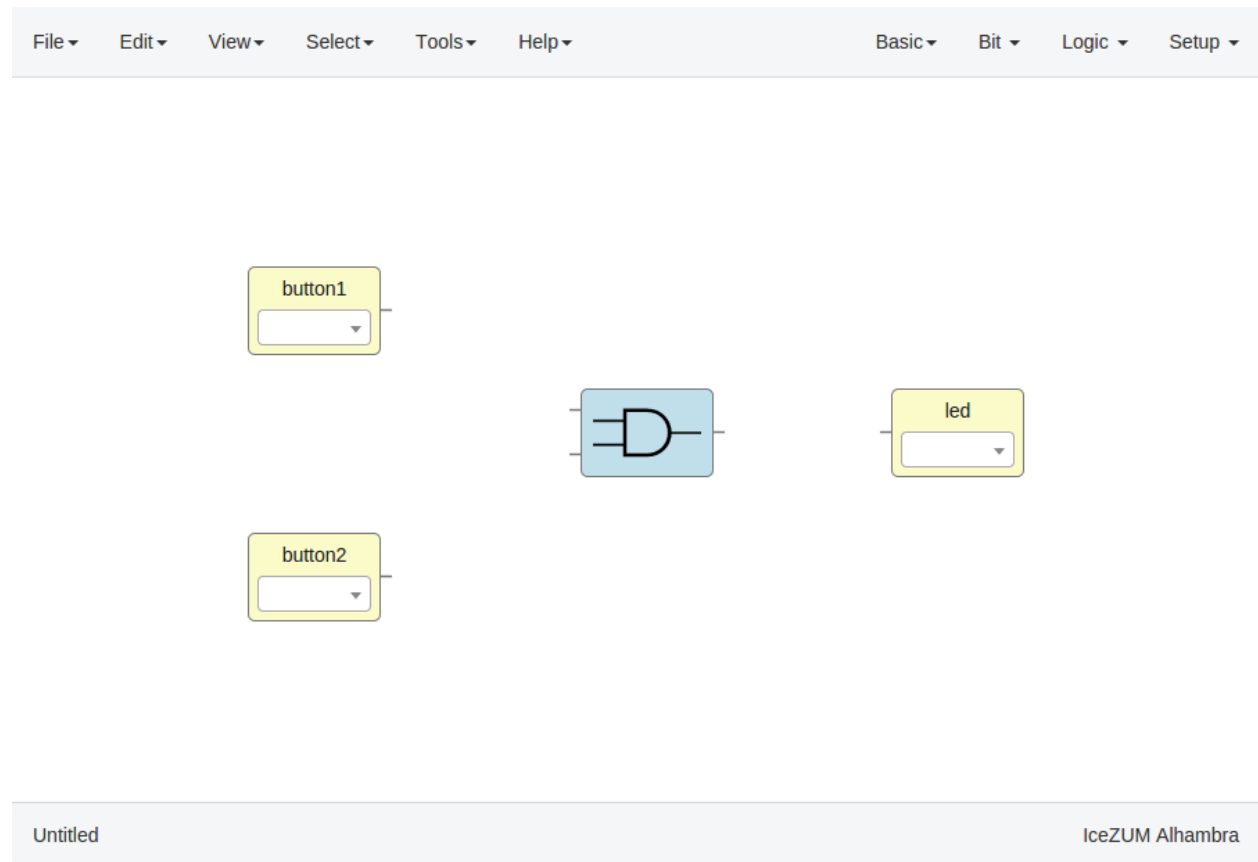
8. Setup blocks

Click on **Setup > Pull up** or **Setup > Tri-state**.

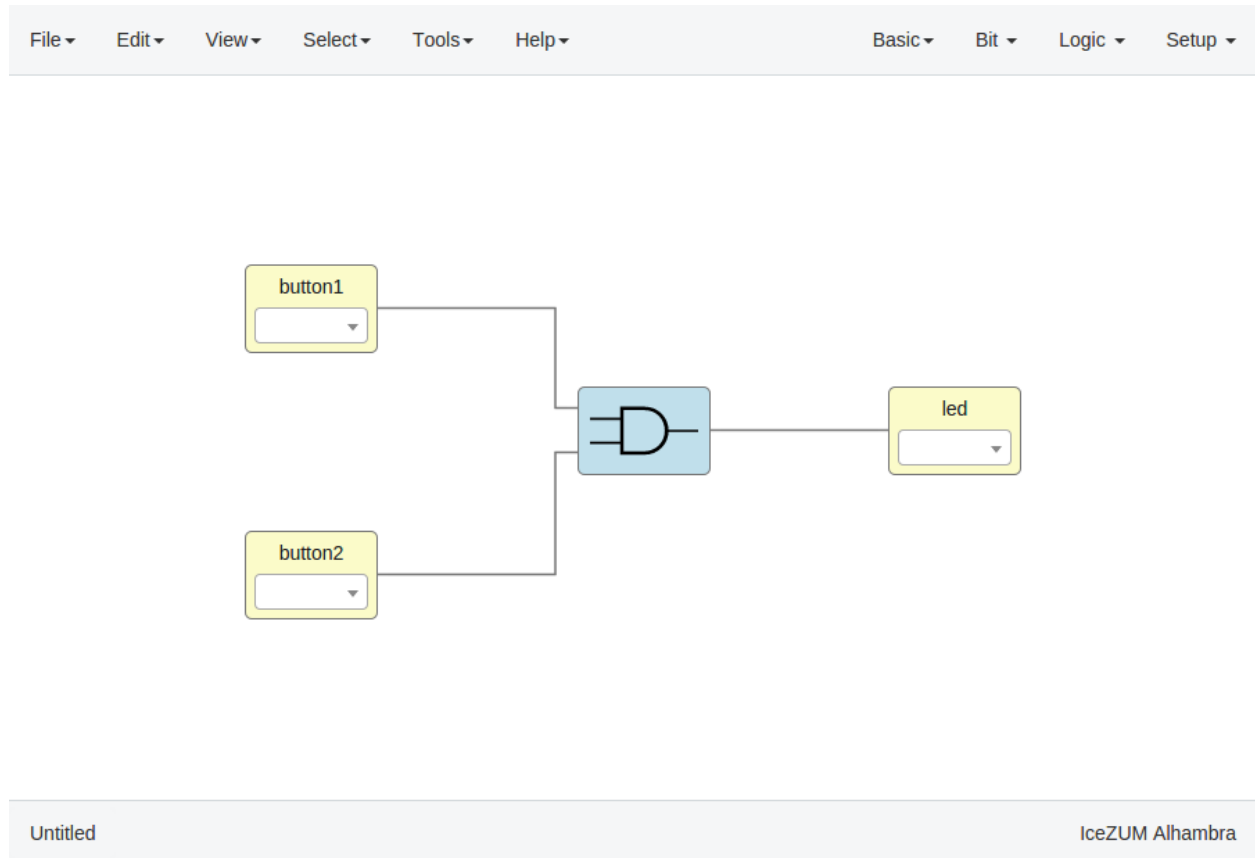
The *Pull up* block must be connected to input ports in order to configure a pull up in the FPGA.



In this example we are going to implement an AND logic gate with its input/output pins connected to the FPGA I/O.

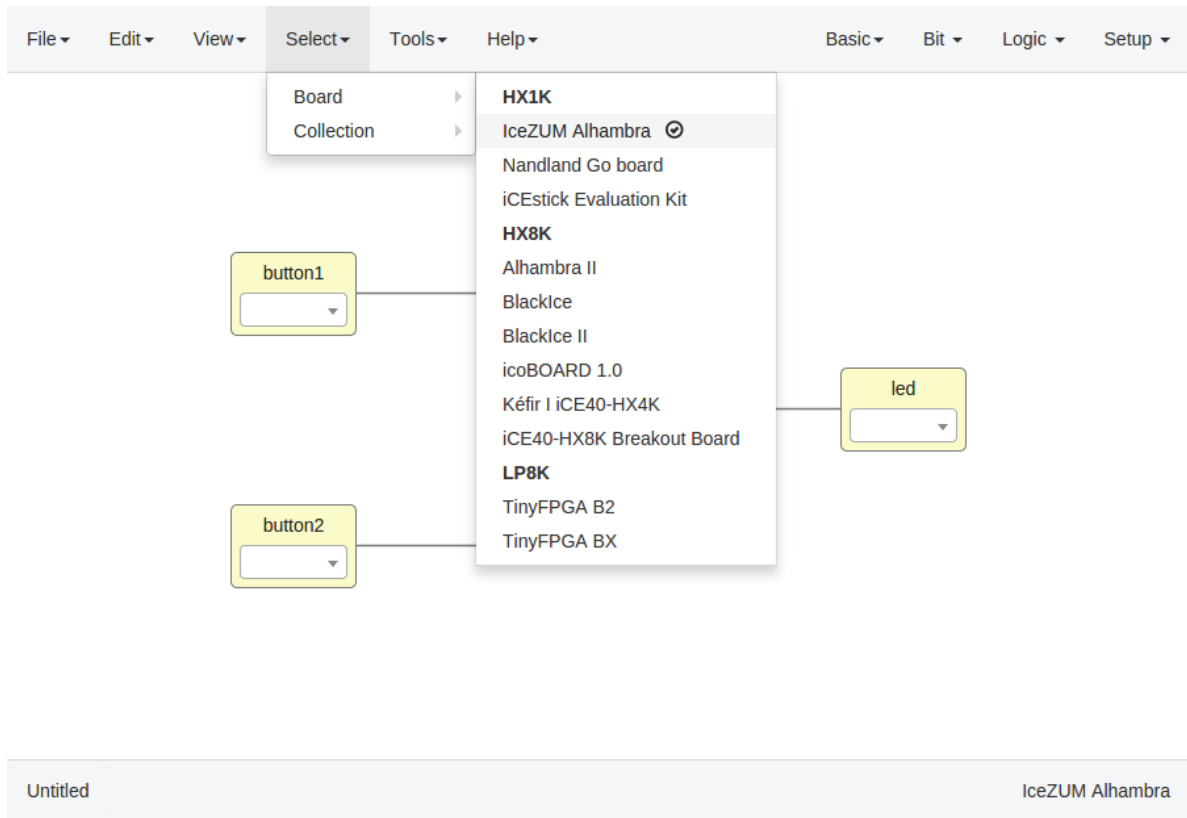


3. Connect the blocks



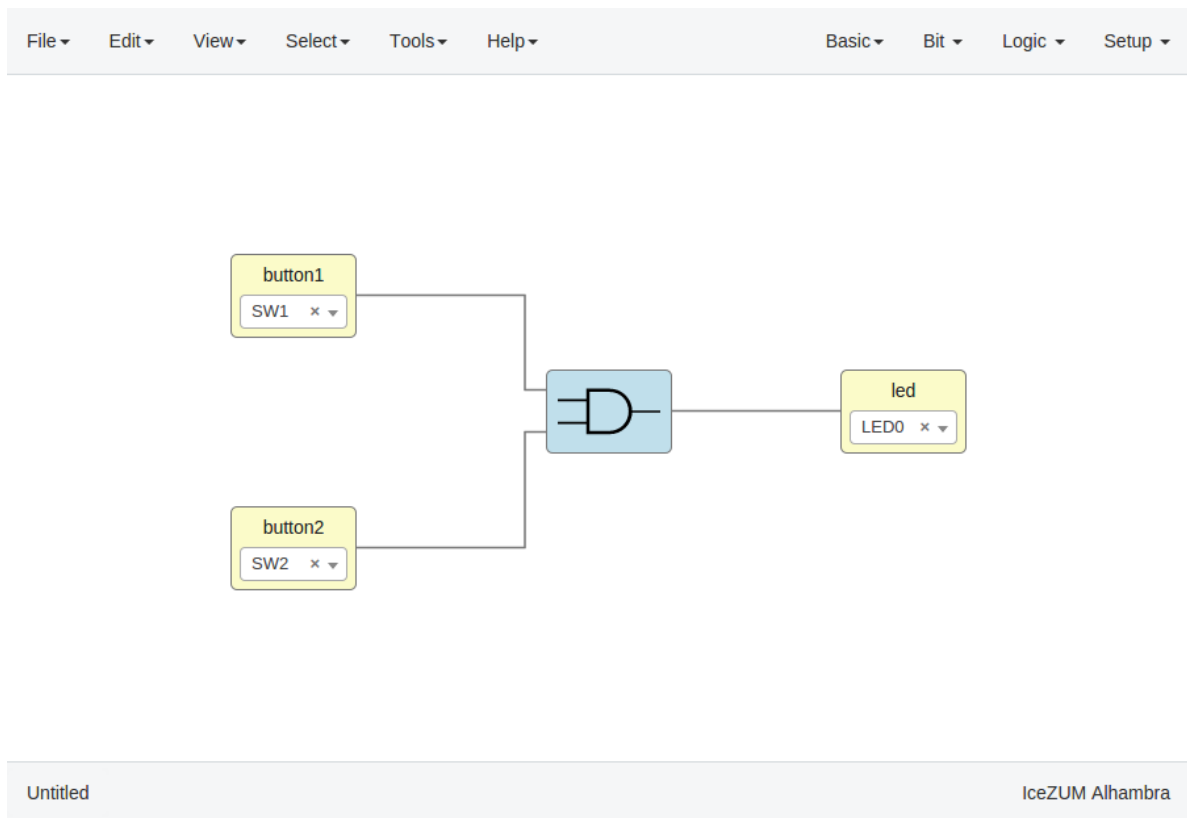
4. Select your board

Go to **Select > Board** and select the board from the list.



5. Set FPGA I/O pins

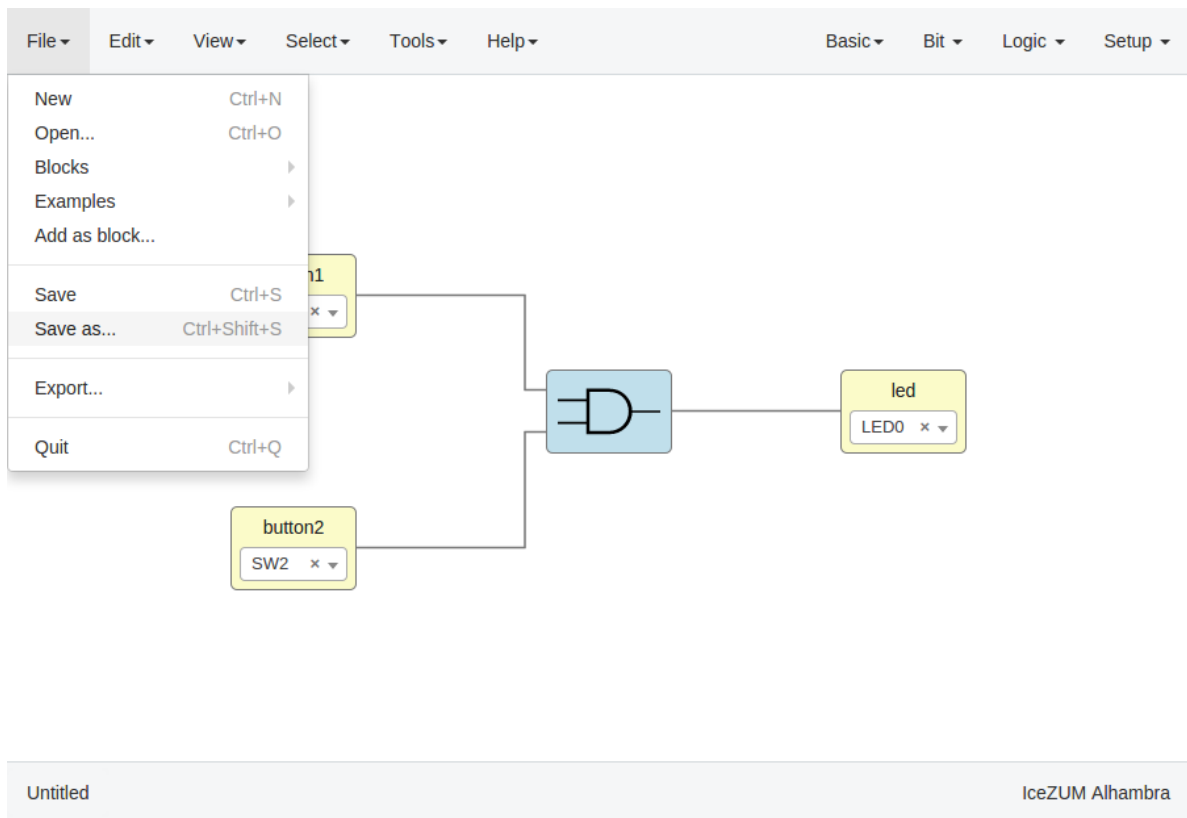
Select all Input/Output blocks' pins.



6. Save the project

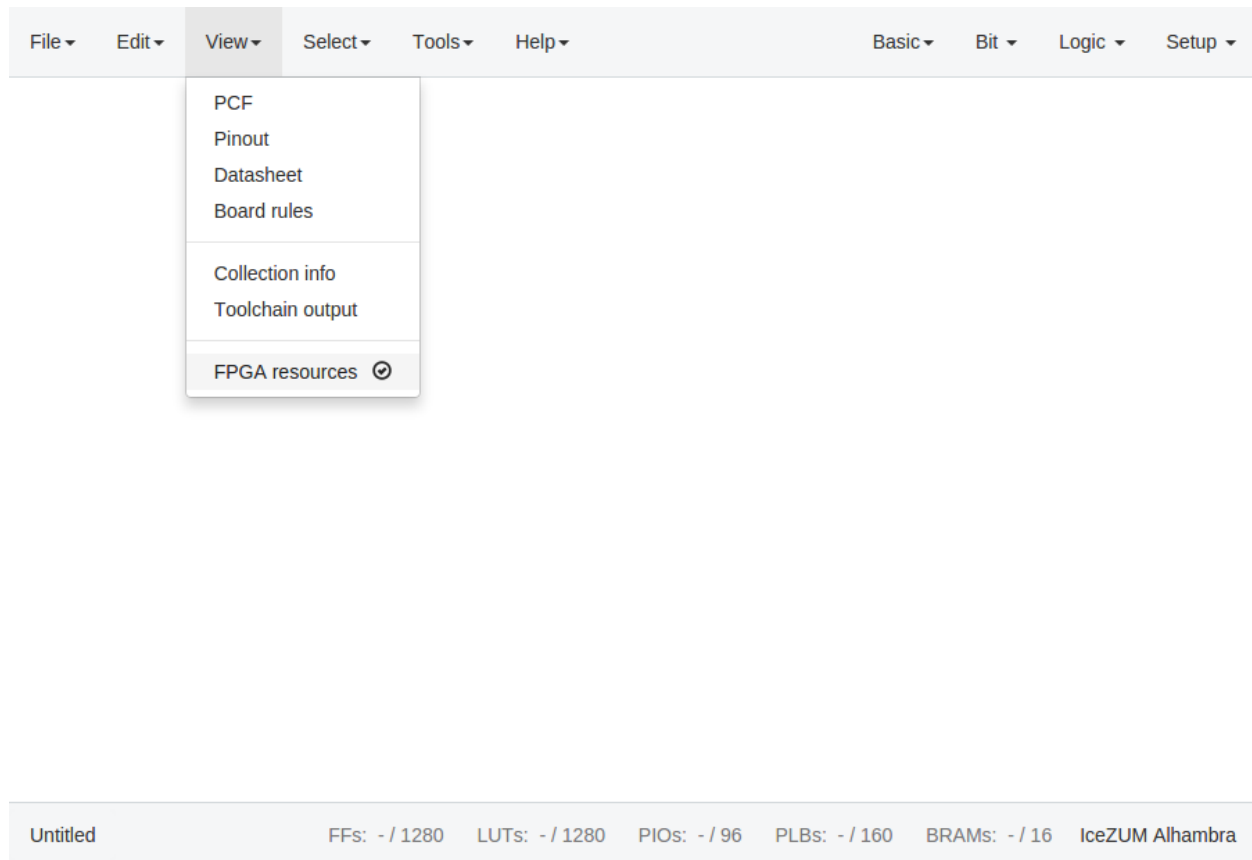
Go to **Edit > Save as** and select the project name, for example *myProject*.

It will be saved as an **.jce** file.



4.5 Show the FPGA resources

Go to **View > FPGA resources**



4.6 Upload a bitstream

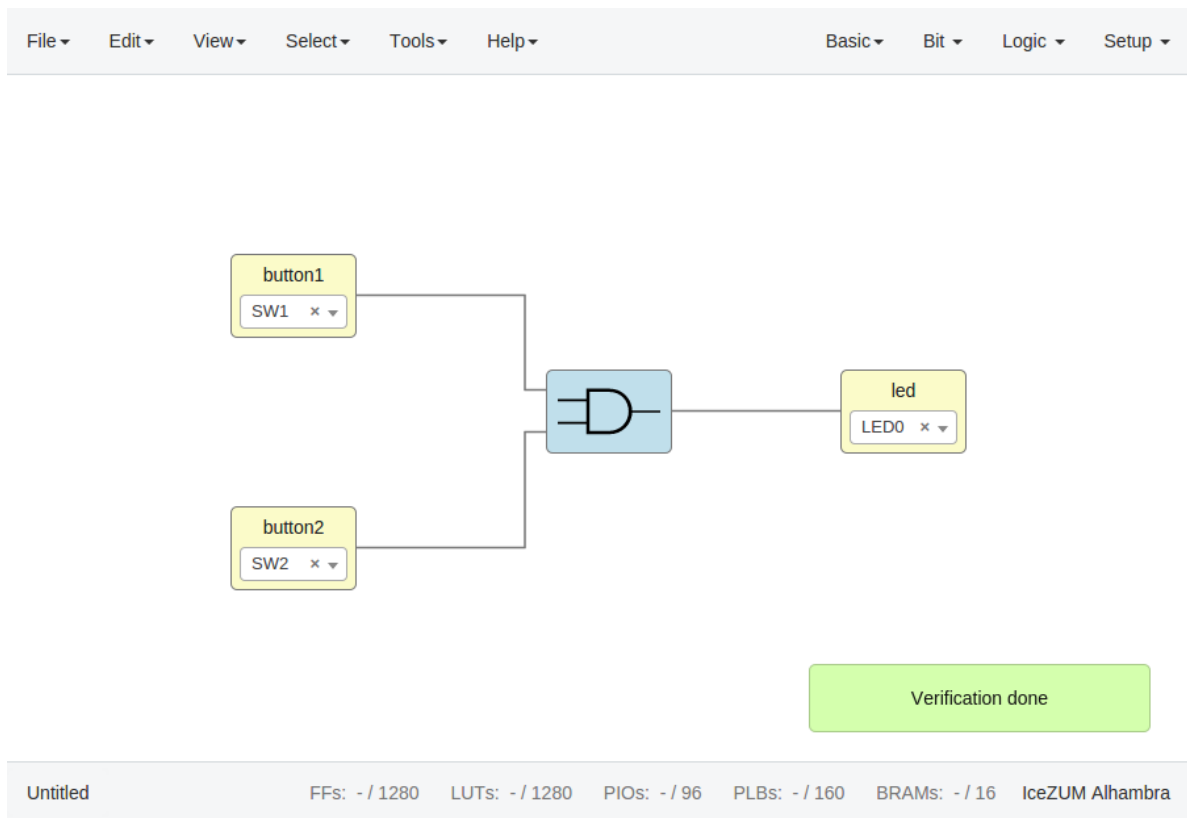
1. Open a project

Go to **Edit > Open...** and select an **.ice** file.

2. Verify the project

Go to **Tools > Verify**.

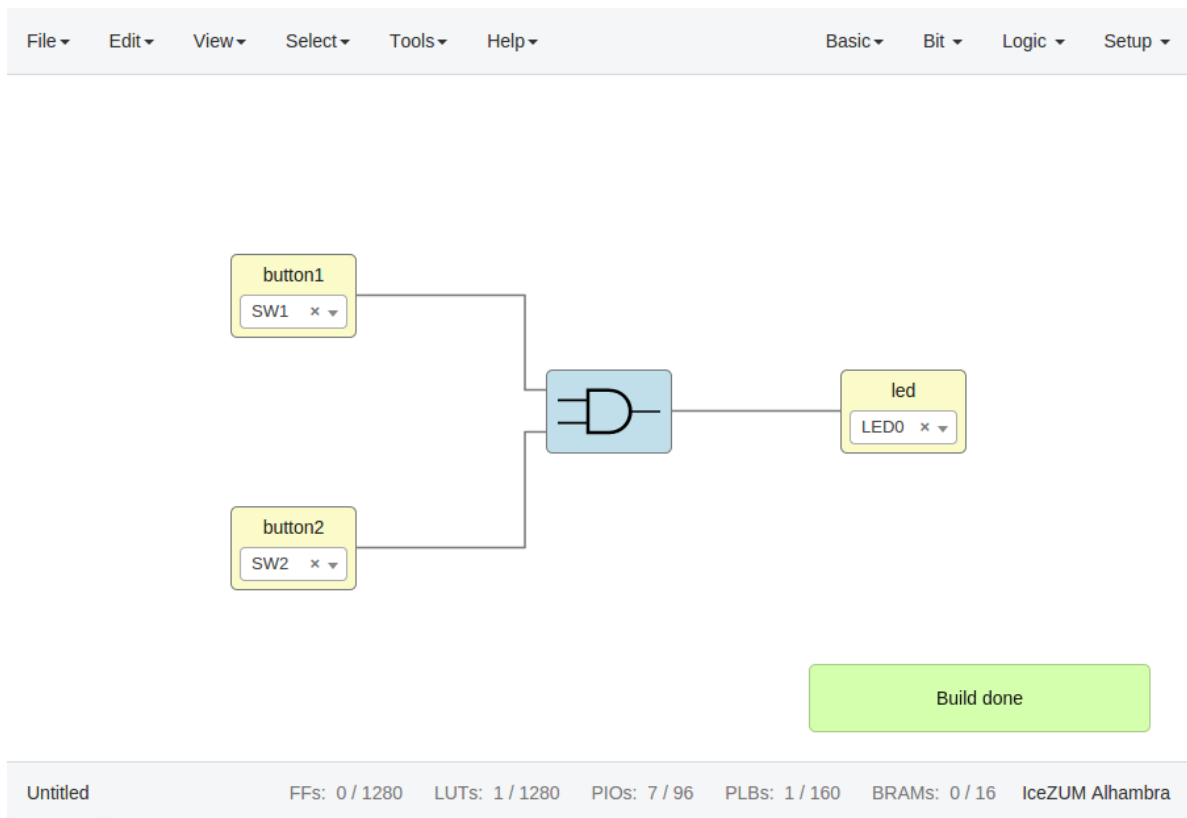
This option checks the generated verilog code using `apio verify`.



3. Build the project

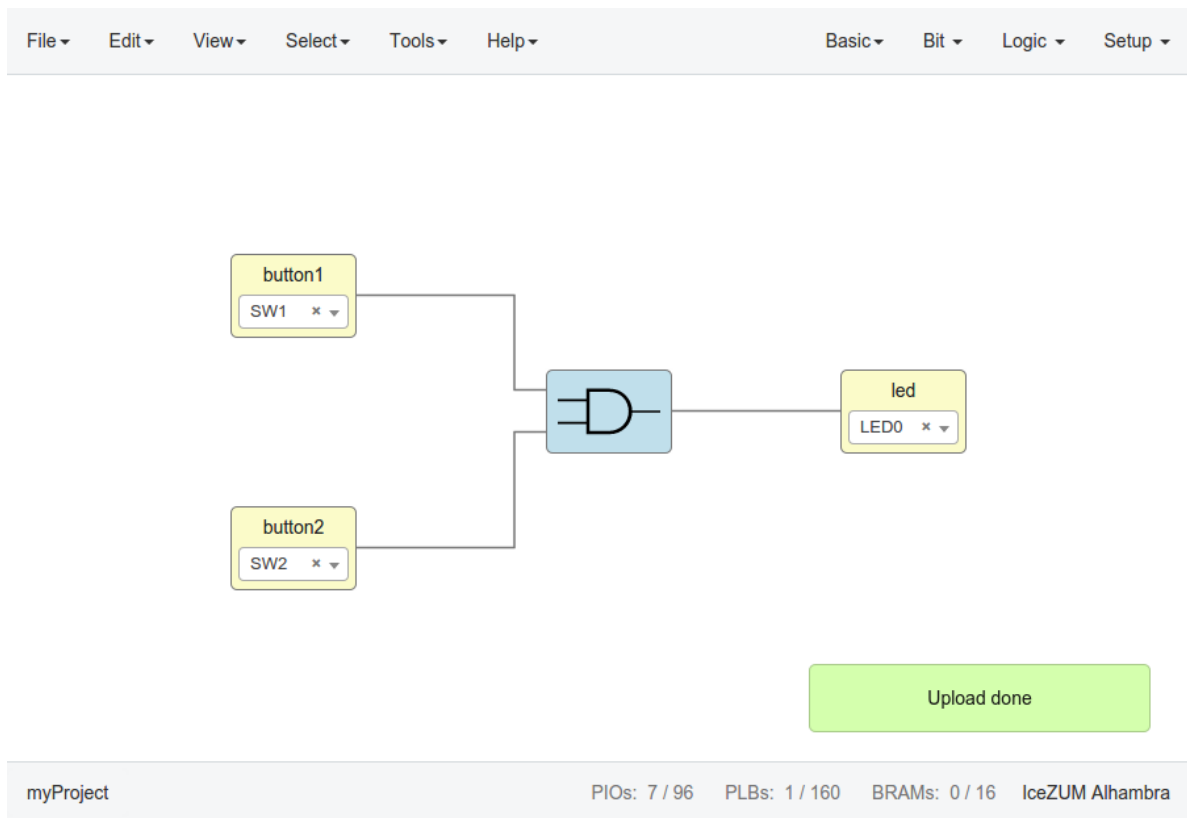
Go to **Tools > Build**.

This option generates a bitstream using `apio build`.



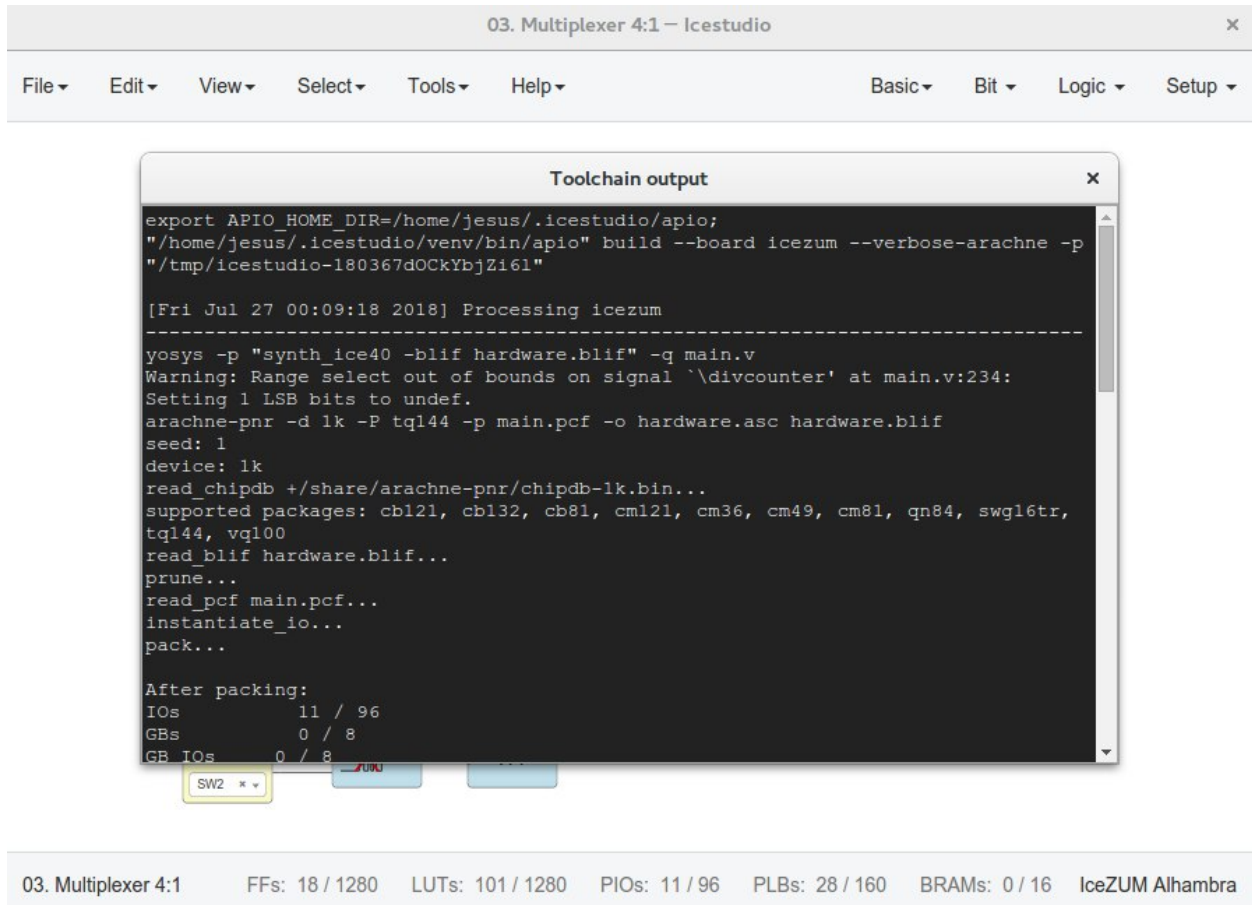
4. Upload the project

Connect your FPGA board and press **Tools > Upload**. This option uses `apio upload`.



4.7 Show the commands output

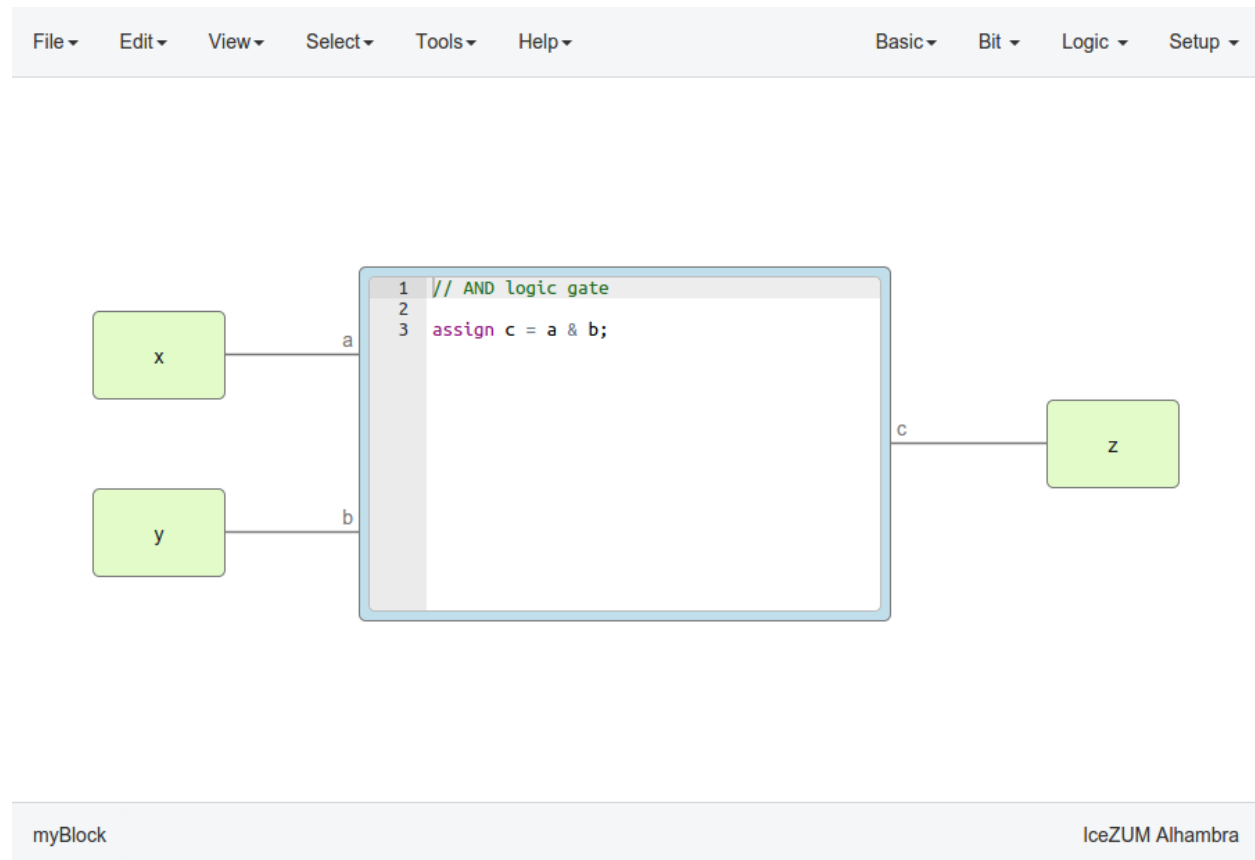
After executing *Tools > Verify*, *Tools > Build* or *Tools > Upload* you can see the executed command and the output in a new windows opened from **View > Toolchain output**.



4.8 Create a block

1. Open a project

Go to **Edit > Open project** and select an **.ice** file. It is recommended to set all the I/O ports non-FPGA ports (green) to create a block.

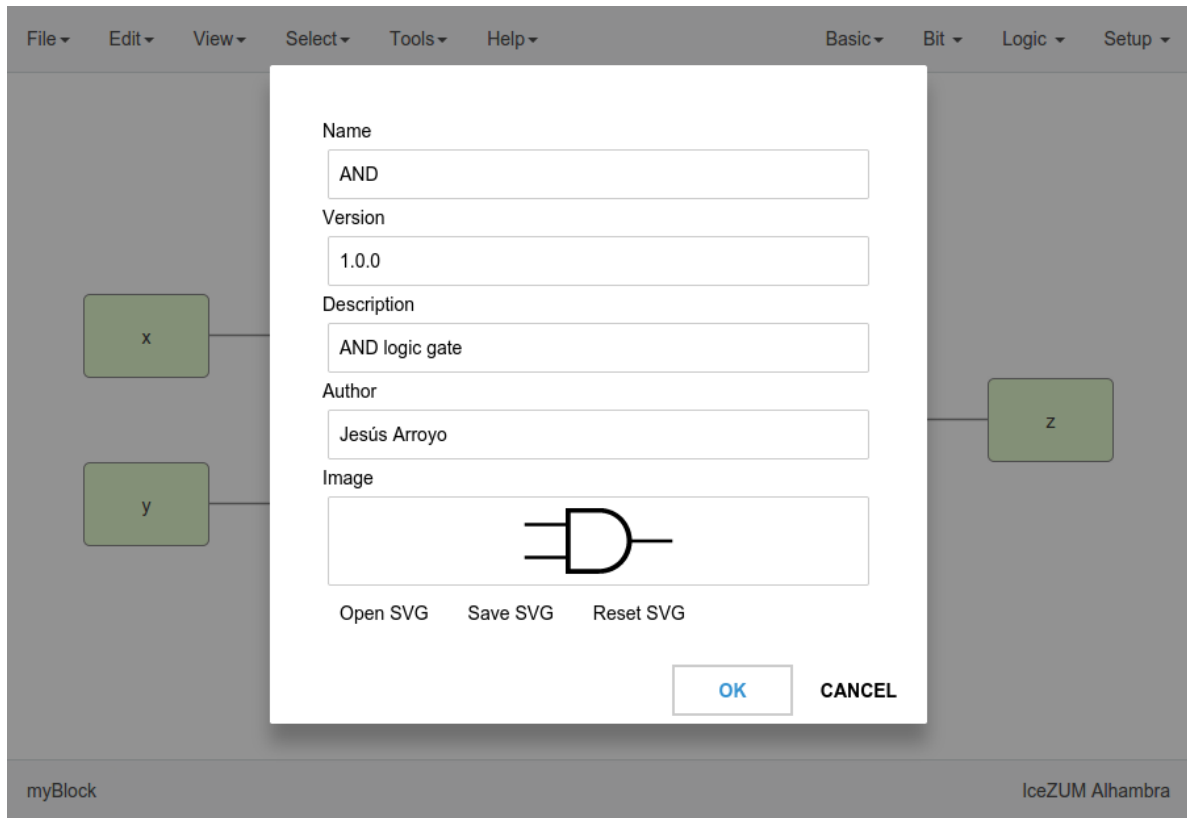


2. Verify the project

Go to **Tools > Verify**.

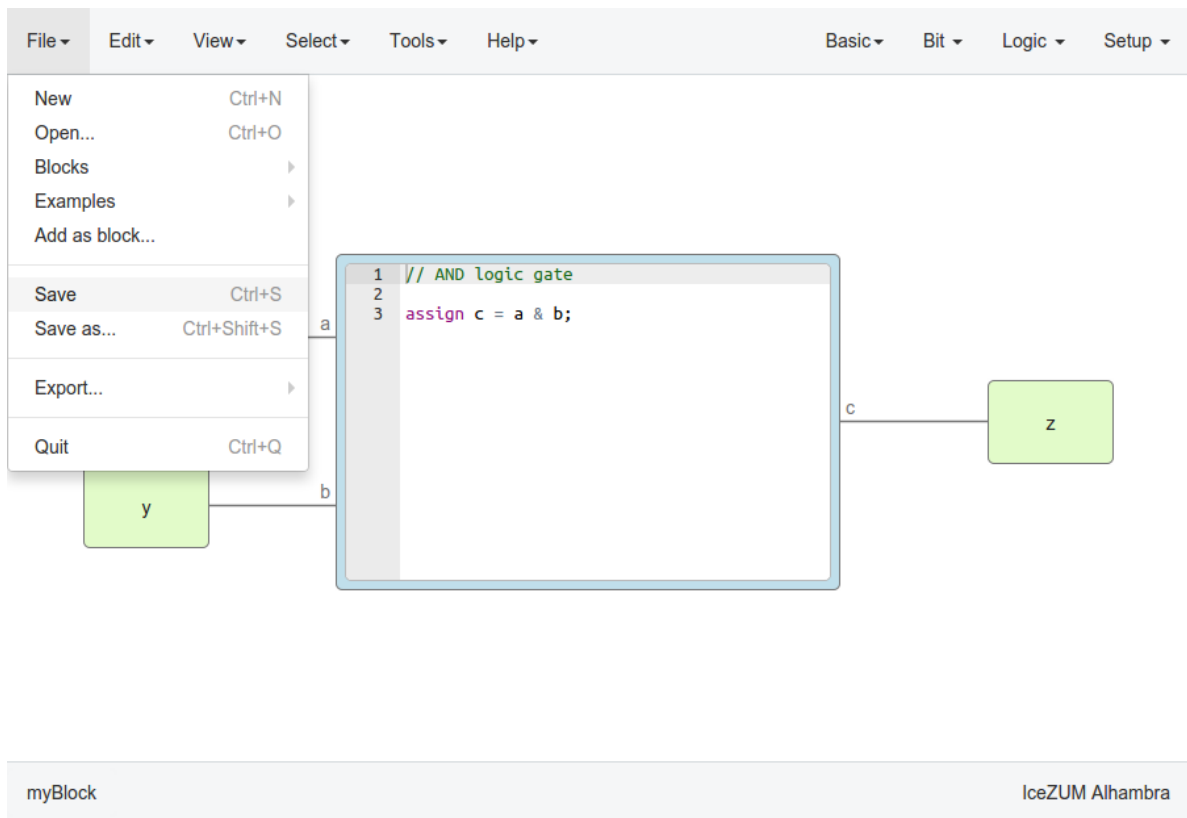
2. Add the project information

Go to **Edit > Preferences > Project information**.



3. Save the project

Go to **Edit > Save**.

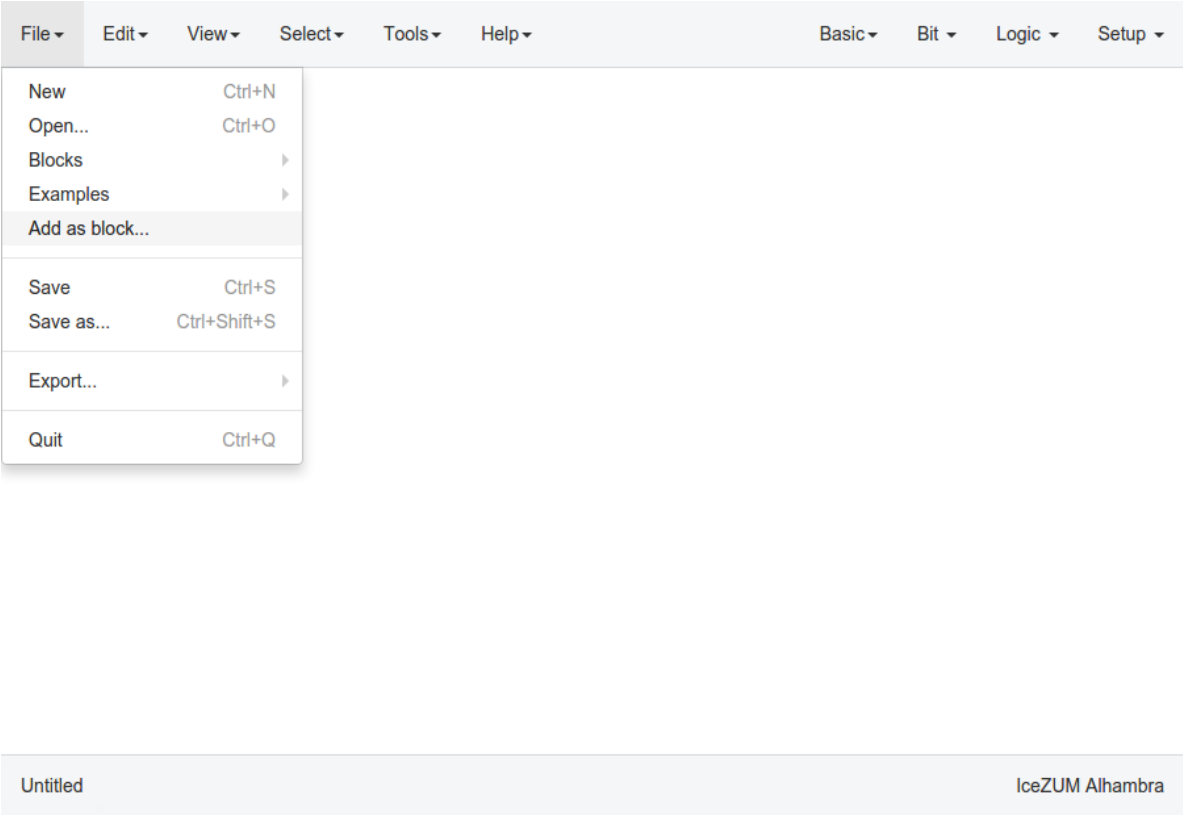


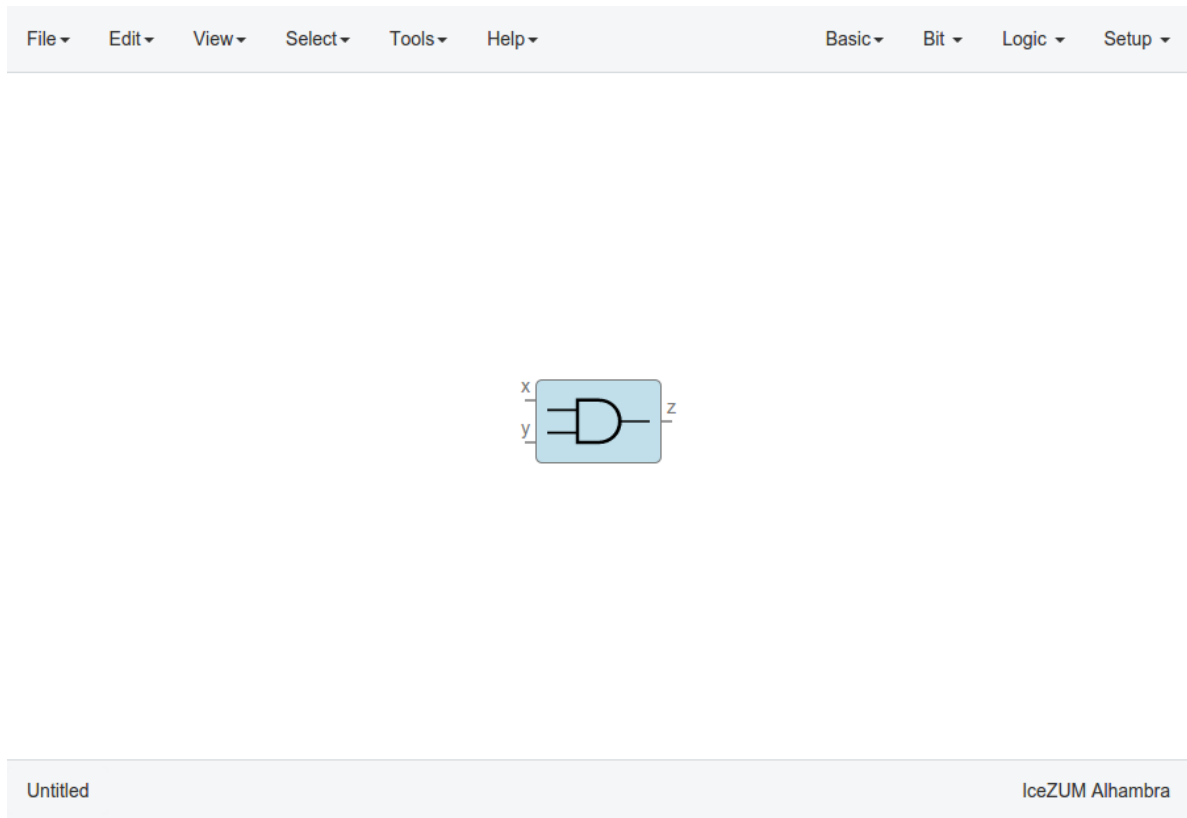
4.9 Add a project as block

1. **Open or create a new project**

2. **Import the custom block**

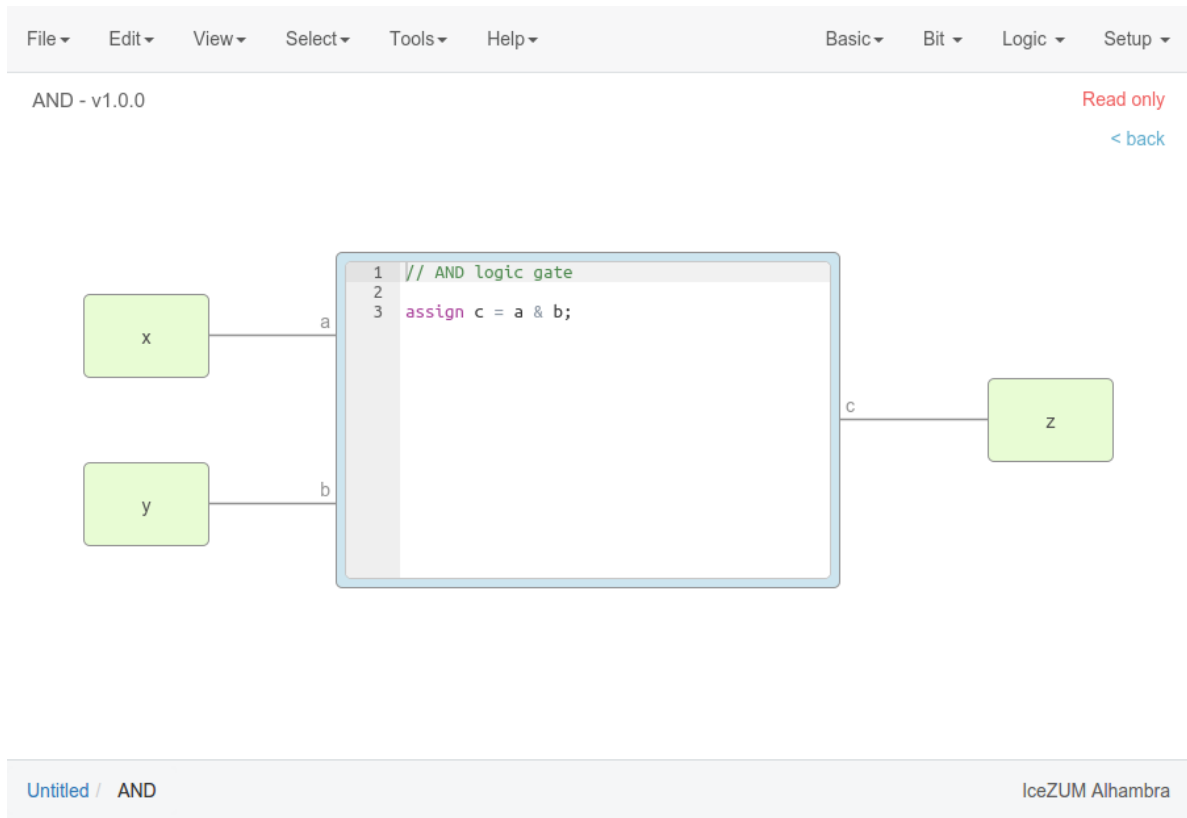
Go to **Edit > Add as block...** and select an **.ice** file.





3. Examine the custom block

Complex blocks can be examined by double clicking the block.



4.10 Add a collection

Go to **Tools > Collections > Add** and select a collection package (ZIP file).

4.11 Select a collection

Go to **Select > Collections**. Select a collection. The first item is the “Default” collection that is the one stored in the application, then the installed internal collections and finally the external collections.

4.12 View the selected collection info

Go to **View > Collection info**. A new window will appear with the README.md file content.

4.13 Load external collections

Go to **Edit > Preferences > External collection** and insert an absolute path of your system containing Icestudio collections.

4.14 Create a collection package

1. Create one or more collections

You can use the `icm cli tool` to create and update a collection.

```
Collection/
├── blocks
│   ├── category1
│   │   ├── block1.ice
│   │   └── subcategory1
│   │       ├── block11.ice
│   │       └── block12.ice
│   └── category2
│       └── block2.ice
├── examples
│   ├── example1.ice
│   ├── example2.ice
│   └── example3.ice
├── locale
│   ├── en
│   │   └── en.po
│   ├── es_ES
│   │   └── es_ES.po
│   └── translation.js
├── LICENSE
├── package.json
└── README.md
```

2. ZIP all your collections

Create a ZIP file with all your created collections at the main level.

```
Collections.zip
├── Collections 1
│   └── ...
└── Collections 2
    └── ...
```

Note: The file `package.json` must exist, and also the `blocks` directory and/or the `examples` directory. The `locale` directory is optional. More information in the [Default collection](#).

4.15 Include a list file

If your code block contains a list file(s), for example:

```
$readmemh("rom.list", rom);
```

1. **Save the ice project**
2. **Copy the list file(s) in the project directory**
3. **Build and upload the project**

Also you can include explicitly a list file in the header of a code block:

```
// @include rom.list
```

4.16 Include a verilog (header) file

If your code block includes a verilog (header) file(s), for example:

```
// @include lib.vh  
// @include math.v  
  
`include "lib.vh"
```

1. **Save the ice project**
2. **Copy the verilog (header) file(s) in the project's directory**
3. **Build and upload the project**

4.17 View the board rules

Go to **View > Board rules**

IceZUM Alhambra - Rules



Input rules

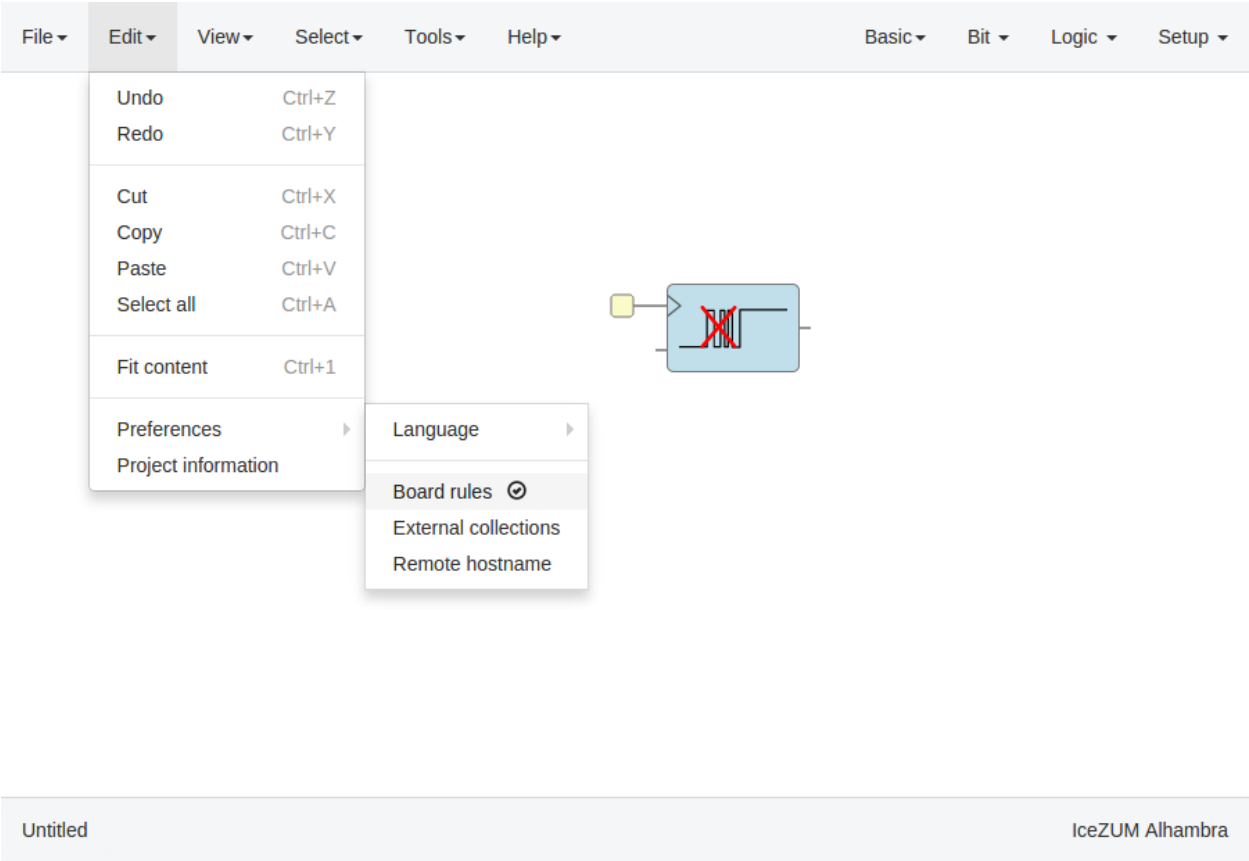
- Input `clk` ports → FPGA pin **21**

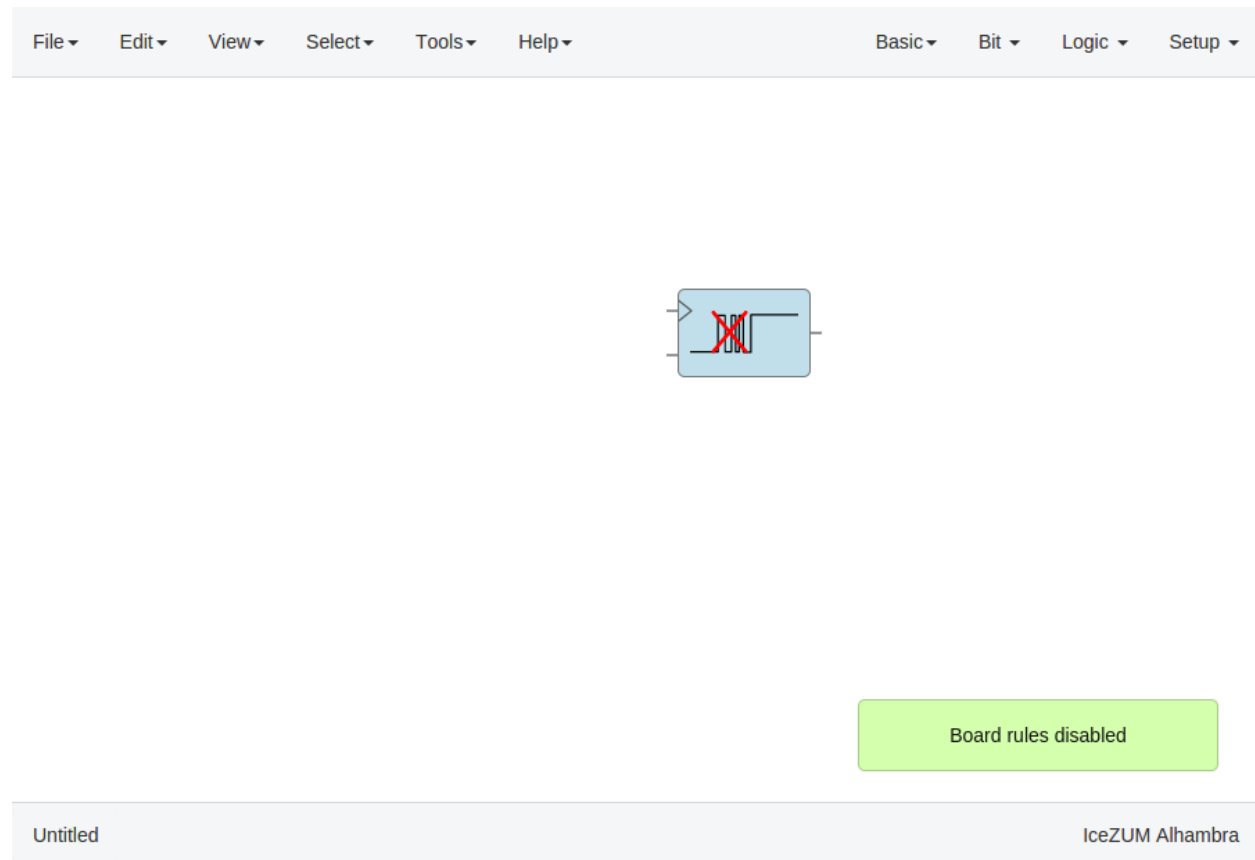
Output rules

- FPGA pin **95** → value `0`
- FPGA pin **96** → value `0`
- FPGA pin **97** → value `0`
- FPGA pin **98** → value `0`
- FPGA pin **99** → value `0`
- FPGA pin **101** → value `0`
- FPGA pin **102** → value `0`
- FPGA pin **104** → value `0`

4.18 Disable the board rules

Go to **Edit > Preferences > Board rules**





4.19 Configure a remote host

If you want to use a RPi, eg `pi@192.168.0.22`, or another computer from Icestudio as a client, first configure the host:

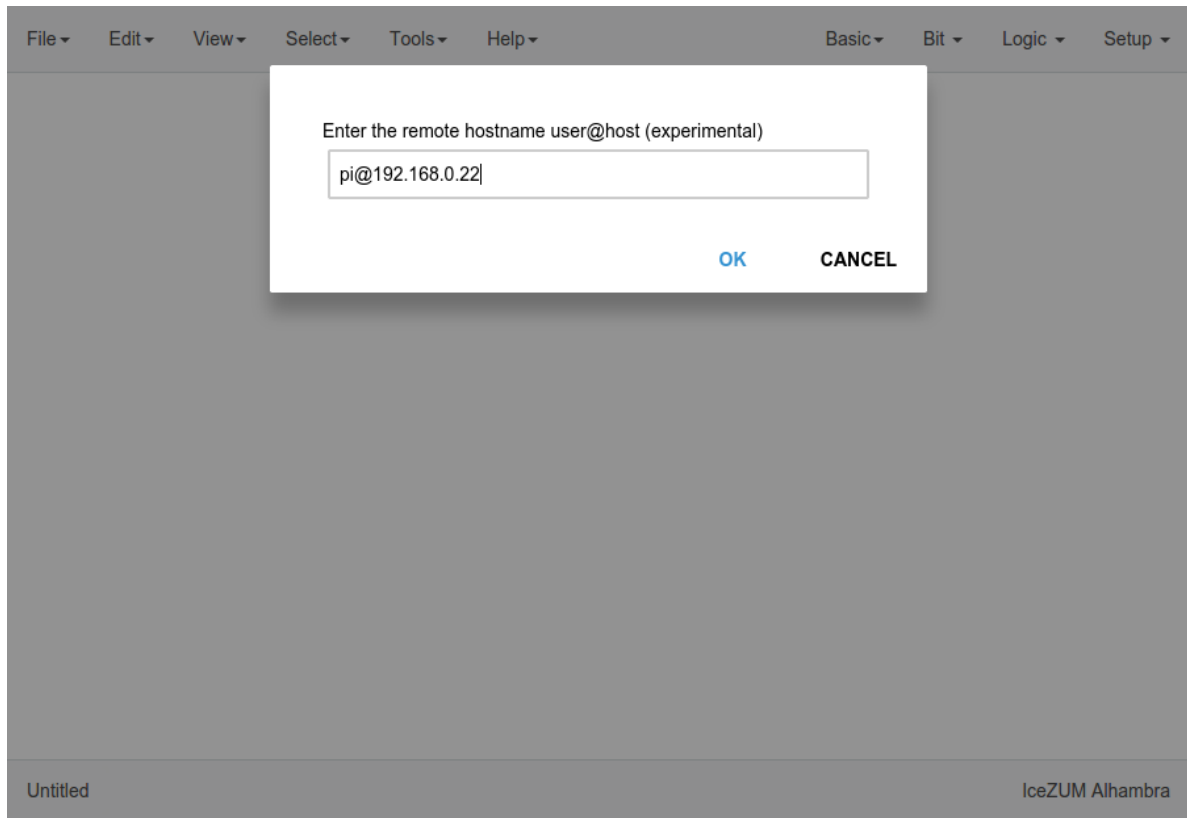
1. **Copy your SSH public key into the server**

```
$ ssh-keygen
$ ssh-copy-id -i .ssh/id_rsa.pub pi@192.168.0.22
```

2. **Install apio in the server**

```
$ ssh pi@192.168.0.22
$ sudo pip install -U apio
$ apio install --all
$ apio drivers --enable # For FTDI devices
```

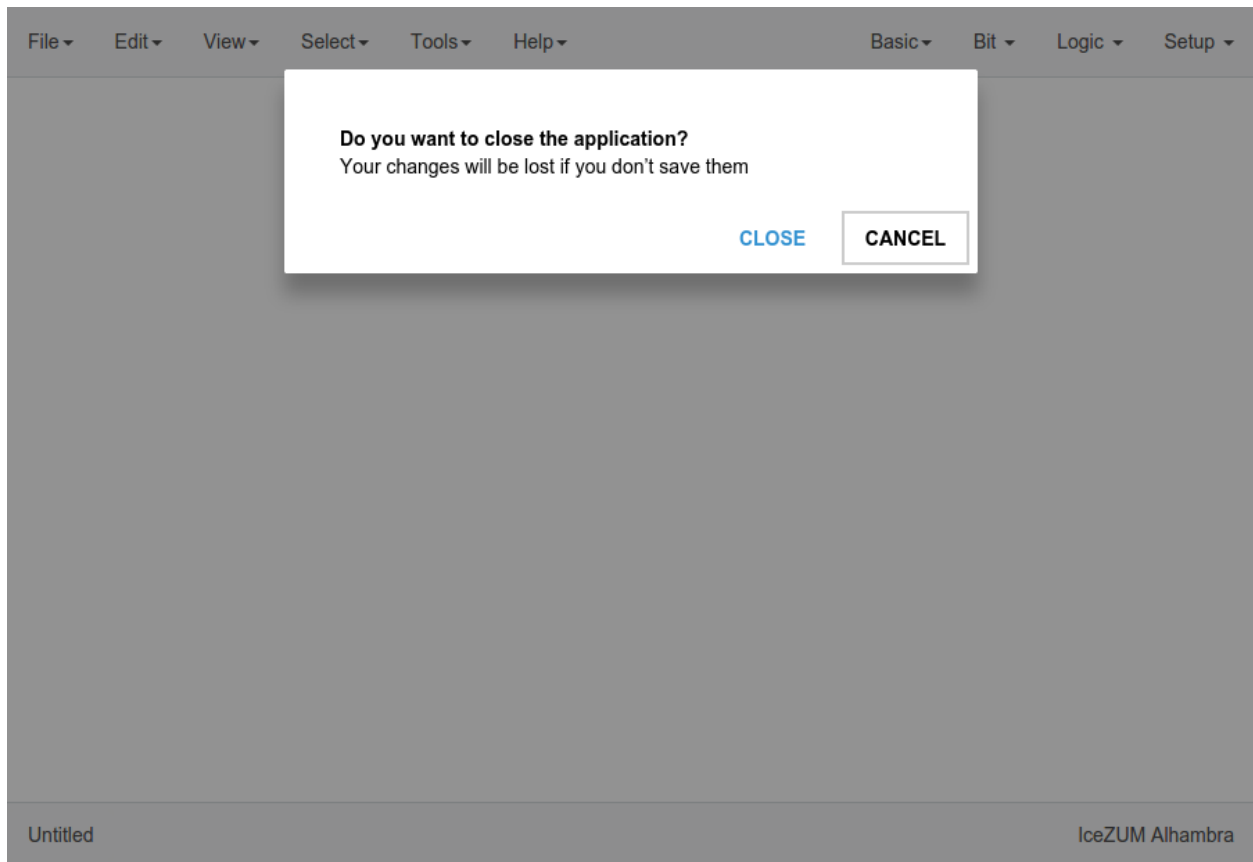
3. **Enter the host name in Icestudio, Edit > Remote hostname**



4. Now, **Verify**, **Build** and **Upload** tools will run in the selected host

4.20 Close the application

Go to **File > Quit** or click the application's close button. If there are unsaved changes an alert will appear to confirm or cancel the action:



This is a quick start guide to setup the development environment for Icestudio.

Install [Python 2.7](#) and [Node.js](#).

[Atom](#) editor with [linter-jshint](#) is recommended.

5.1 Download

```
git clone https://github.com/FPGAwards/icestudio.git
cd icestudio
```

5.2 Install

```
npm install
```

5.3 Execute

```
npm start
```

5.4 Languages

Add or update the [app translations](#) using [Poedit](#).

Developer note: use `npm run gettext` to extract the labels from the code.

5.5 Package

```
npm run dist
```

Target OS	Development OS	Output files
GNU/Linux	GNU/Linux	(linux32,linux64).zip, (linux32,linux64).AppImage
Windows	GNU/Linux	(win32,win64).zip, (win32,win64).exe
Mac OS	Mac OS	(osx32,osx64).zip, osx64.dmg

5.6 Apio configuration

Apio backend is configured in the `app/package.json` file:

- `apio.min`: minimum version ($\geq$)
- `apio.max`: maximum version ($<$)
- `apio.extras`: list of external Python programmers (*blackiceprog*, *tinyfpgab*)
- `apio.external`: load an external Apio package instead of the default one (e.g. */path/to/my/apio*)
- `apio.branch`: install Apio from the repository branch instead of PyPI.

An external Apio package can be also set on runtime using the `ICESTUDIO_APIO` environment variable.

6.1 Definition

- Version: 1.2.
- Package: project information.
- Design: board information and circuit design.
- Dependencies: all used dependencies in one level.

Extension: **.jce**

```
{
  "version": "1.2",
  "package": {
    "name": "",
    "version": "",
    "description": "",
    "author": "",
    "image": ""
  },
  "design": {
    "board": "",
    "graph": {
      "blocks": [],
      "wires": []
    }
  },
  "dependencies": {}
}
```

6.1.1 Block instances

```
{
  "id": "",
  "type": "",
  "data": {},
  "position": {
    "x": 0,
    "y": 0
  },
  "size": {
    "width": 0,
    "height": 0
  }
}
```

6.1.2 Wire instances

Wire

```
{
  "source": {
    "block": "",
    "port": ""
  },
  "target": {
    "block": "",
    "port": ""
  },
  "vertices": [
    {
      "x": 0,
      "y": 0
    }
  ]
}
```

Bus

```
{
  "source": {
    "block": "",
    "port": ""
  },
  "target": {
    "block": "",
    "port": ""
  },
  "vertices": [
    {
      "x": 0,
      "y": 0
    }
  ]
}
```

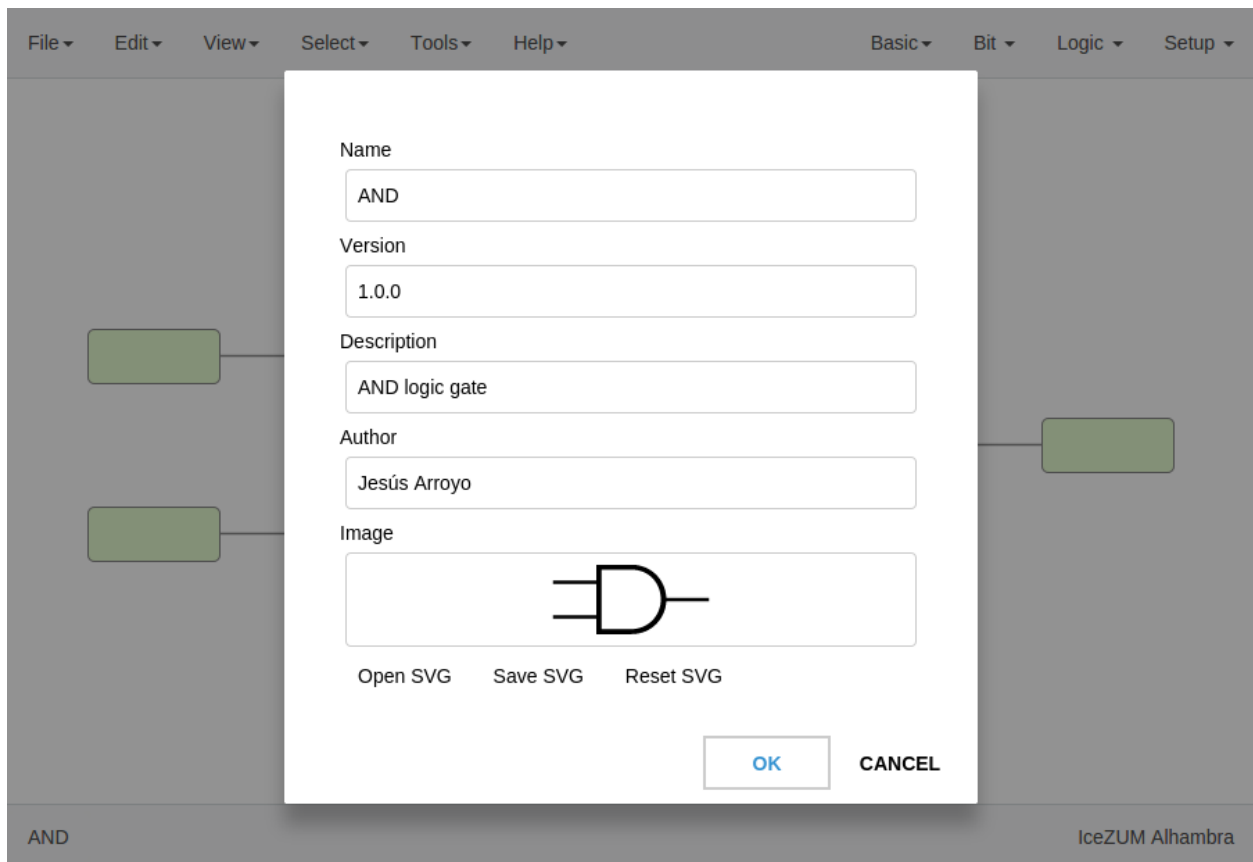
(continues on next page)

(continued from previous page)

```
},  
  "size": 2  
}
```

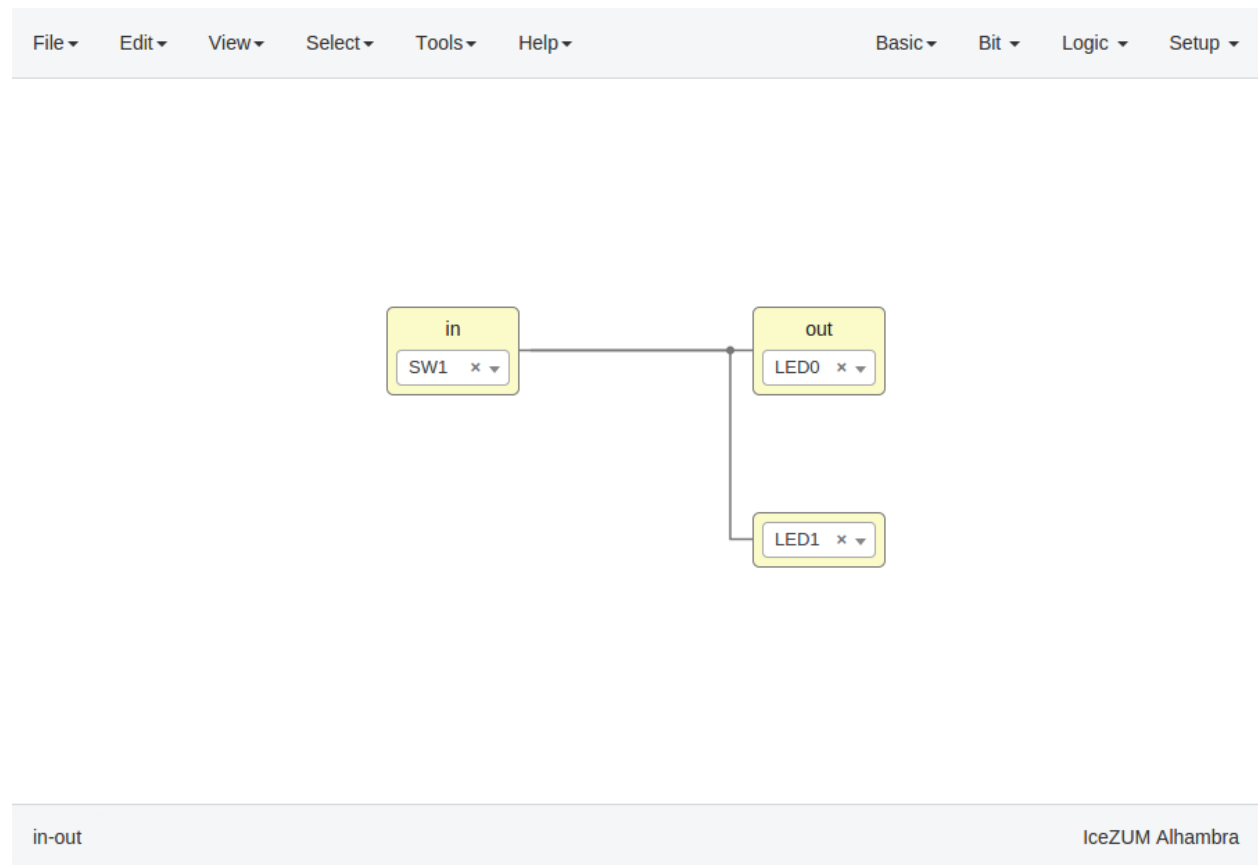
6.2 Package

- Name
- Version
- Description
- Author
- Image (SVG)



6.3 Samples

6.3.1 1. in-out



File: **in-out.ice**

Show/Hide code

```
{
  "version": "1.2",
  "package": {
    "name": "in-out",
    "version": "1.0",
    "description": "Assign the input to both outputs",
    "author": "Jesús Arroyo",
    "image": ""
  },
}
```

(continues on next page)

(continued from previous page)

```

"design": {
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "b25e5929-162c-4631-8d04-156e0b382590",
        "type": "basic.input",
        "data": {
          "name": "in",
          "pins": [
            {
              "index": "0",
              "name": "SW1",
              "value": "10"
            }
          ],
          "virtual": false
        },
        "position": {
          "x": 224,
          "y": 112
        }
      },
      {
        "id": "2a8315b1-437e-40b7-adfb-ff961a0aa8f6",
        "type": "basic.output",
        "data": {
          "name": "out",
          "pins": [
            {
              "index": "0",
              "name": "LED0",
              "value": "95"
            }
          ],
          "virtual": false
        },
        "position": {
          "x": 488,
          "y": 112
        }
      },
      {
        "id": "f8ffb071-9a46-4b46-86d2-cd5b83bae395",
        "type": "basic.output",
        "data": {
          "name": "",
          "pins": [
            {
              "index": "0",
              "name": "LED1",
              "value": "96"
            }
          ],
          "virtual": false
        },
        "position": {

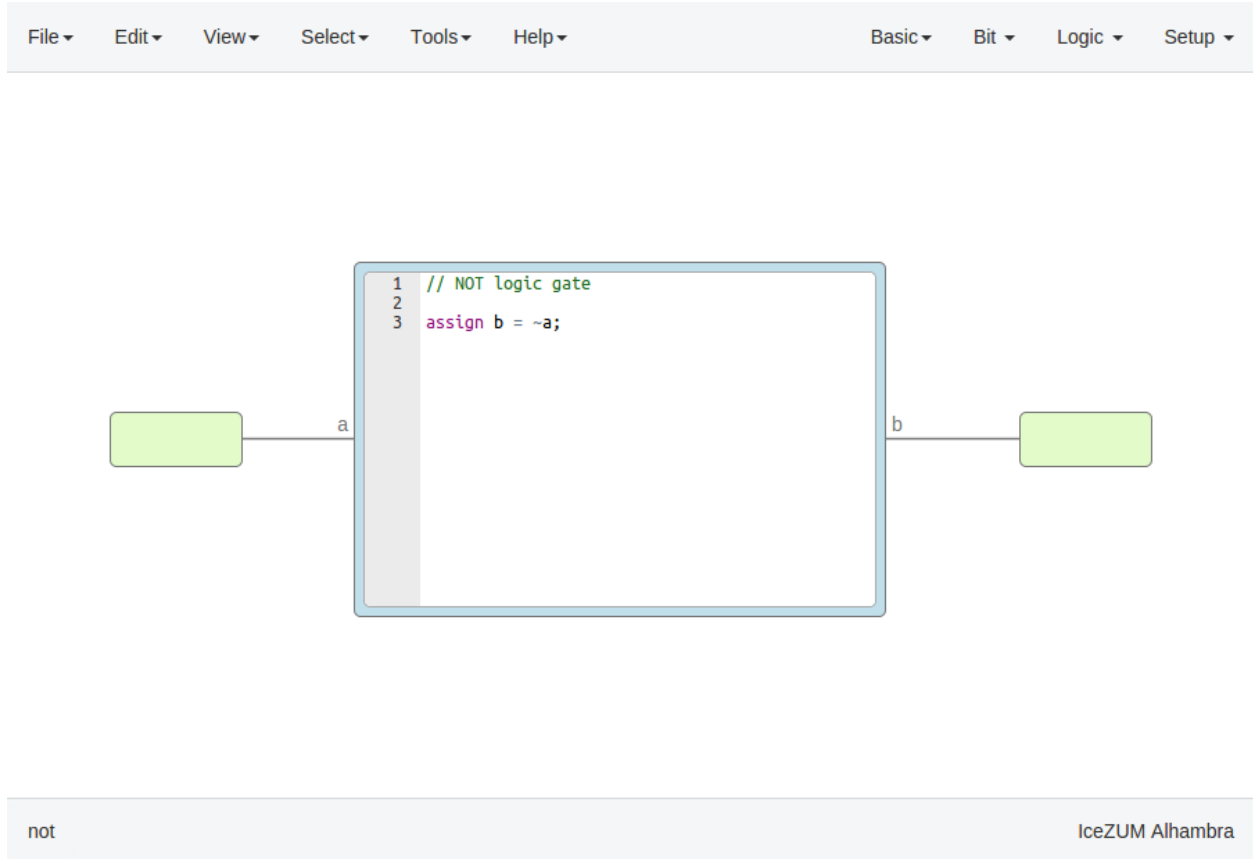
```

(continues on next page)

(continued from previous page)

```
        "x": 488,  
        "y": 248  
    }  
  }  
],  
"wires": [  
  {  
    "source": {  
      "block": "b25e5929-162c-4631-8d04-156e0b382590",  
      "port": "out"  
    },  
    "target": {  
      "block": "2a8315b1-437e-40b7-adfb-ff961a0aa8f6",  
      "port": "in"  
    }  
  },  
  {  
    "source": {  
      "block": "b25e5929-162c-4631-8d04-156e0b382590",  
      "port": "out"  
    },  
    "target": {  
      "block": "f8fffb071-9a46-4b46-86d2-cd5b83bae395",  
      "port": "in"  
    }  
  }  
]  
}  
},  
"dependencies": {}  
}
```

6.3.2 2. not



File: **not.ice**

Show/Hide code

```
{
  "version": "1.2",
  "package": {
    "name": "Not",
    "version": "1.0",
    "description": "NOT logic gate",
    "author": "Jesús Arroyo",
    "image": "%3Csvg%20xmlns=%22http://www.w3.org/2000/svg%22%20width=%2291.33%22
    %20height=%2245.752%22%20version=%221%22%3E%3Cpath%20d=%22M0%2020.446h27v2H0zm70.
    322.001h15.3v2h-15.3z%22/%3E%3Cpath%20d=%22M66.05%2026.746c-2.9%200-5.3-2.4-5.3-5.
    3s2.4-5.3%205.3-5.3%205.3%202.4%205.3%205.3-2.4%205.3-5.3%205.3zm0-8.6c-1.8%200-3.3
    %201.5-3.3%203.3%200%201.8%201.5%203.3%203.3%203.3%201.8%200%203.3-1.5%203.3-3.3
    %200-1.8-1.5-3.3-3.3-3.3z%22/%3E%3Cpath%20d=%22M25.962%202.563l33.624%2018.883l25.
    962%2040.33V2.563z%22%20fill=%22none%22%20stroke=%22#000%22%20stroke-width=%223%22/
    %3E%3C/svg%3E"
  }
}
```

(continues on next page)

(continued from previous page)

```

},
"design": {
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "a4058fa5-b66e-4e5e-b542-28d7c3e9d3cd",
        "type": "basic.input",
        "data": {
          "name": "",
          "pins": [
            {
              "index": "0",
              "name": "",
              "value": 0
            }
          ],
          "virtual": true
        },
        "position": {
          "x": 72,
          "y": 184
        }
      },
      {
        "id": "07895985-9d14-4a6f-8f2d-b2a6ddf61852",
        "type": "basic.output",
        "data": {
          "name": "",
          "pins": [
            {
              "index": "0",
              "name": "",
              "value": 0
            }
          ],
          "virtual": true
        },
        "position": {
          "x": 728,
          "y": 184
        }
      },
      {
        "id": "364b95cc-e8ff-4c65-b332-d6125c5968ee",
        "type": "basic.code",
        "data": {
          "code": "// NOT logic gate\n\nassign b = ~a;",
          "params": [],
          "ports": {
            "in": [
              {
                "name": "a"
              }
            ],
            "out": [

```

(continues on next page)

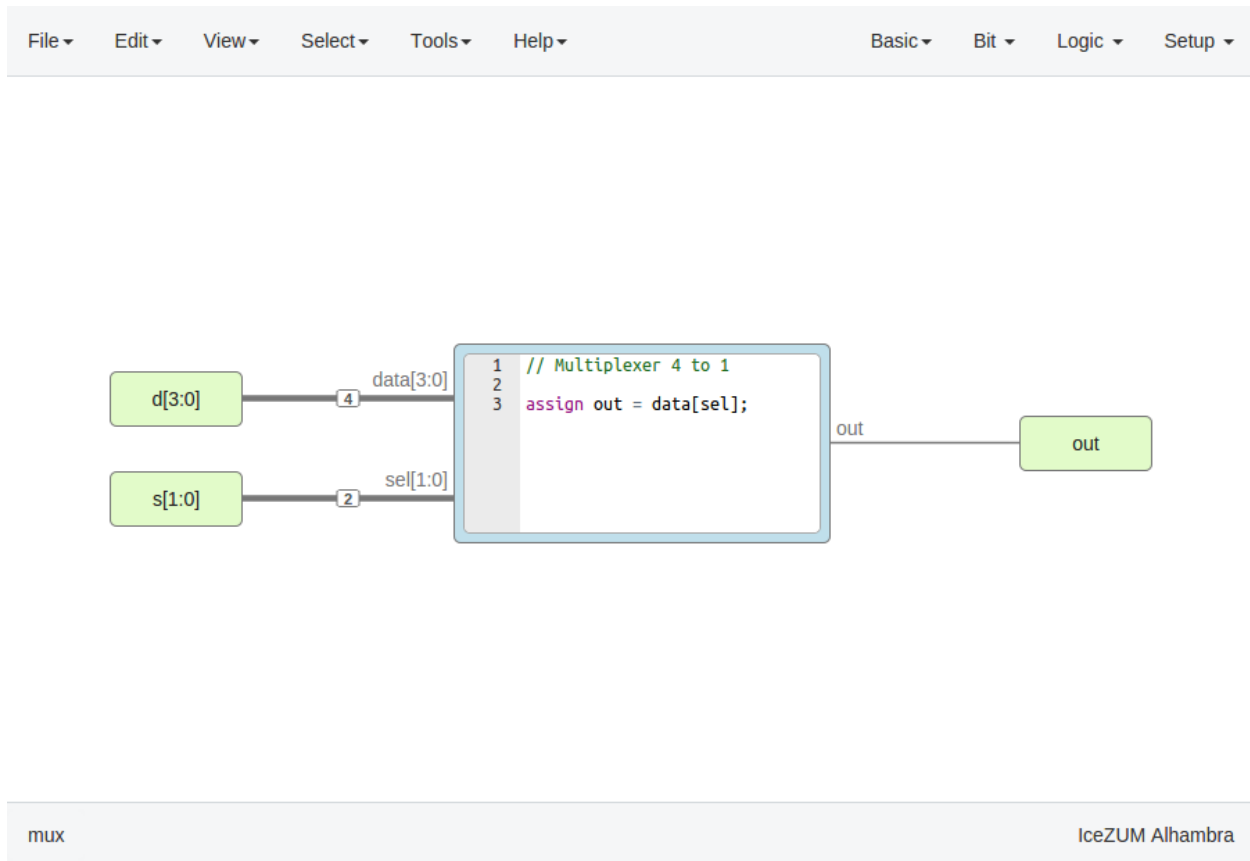
(continued from previous page)

```

        "name": "b"
      }
    ]
  },
  "position": {
    "x": 248,
    "y": 88
  },
  "size": {
    "width": 384,
    "height": 256
  }
},
],
"wires": [
  {
    "source": {
      "block": "a4058fa5-b66e-4e5e-b542-28d7c3e9d3cd",
      "port": "out"
    },
    "target": {
      "block": "364b95cc-e8ff-4c65-b332-d6125c5968ee",
      "port": "a"
    }
  },
  {
    "source": {
      "block": "364b95cc-e8ff-4c65-b332-d6125c5968ee",
      "port": "b"
    },
    "target": {
      "block": "07895985-9d14-4a6f-8f2d-b2a6ddf61852",
      "port": "in"
    }
  }
]
},
"dependencies": {}
}

```

6.3.3 3. mux



File: **mux.ice**

Show/Hide code

```

{
  "version": "1.2",
  "package": {
    "name": "Mux4:1",
    "version": "1.1",
    "description": "Multiplexer 4 to 1",
    "author": "Jesús Arroyo",
    "image": "%3Csvg%20xmlns=%22http://www.w3.org/2000/svg%22%20viewBox=%22-252%20400.
→ 9%2081%2040%22%20width=%2281%22%20height=%2240%22%3E%3Cpath%20d=%22M-191%20419.9v-7.
→ 21-41-11.8v40141-11.7v-7.4zm-39%2018.5v-35137%2010.8v13.5z%22/%3E%3C/svg%3E"
  },
  "design": {

```

(continues on next page)

(continued from previous page)

```

"board": "icezum",
"graph": {
  "blocks": [
    {
      "id": "95f8c313-6e18-4ee3-b9cf-7266dec53c93",
      "type": "basic.input",
      "data": {
        "name": "d",
        "range": "[3:0]",
        "pins": [
          {
            "index": "3",
            "name": "",
            "value": 0
          },
          {
            "index": "2",
            "name": "",
            "value": 0
          },
          {
            "index": "1",
            "name": "",
            "value": 0
          },
          {
            "index": "0",
            "name": "",
            "value": 0
          }
        ]
      },
      "virtual": true
    },
    {
      "position": {
        "x": 64,
        "y": 160
      }
    },
    {
      "id": "60d40fc8-3388-4066-8f0a-af17e179a9bd",
      "type": "basic.output",
      "data": {
        "name": "out",
        "pins": [
          {
            "index": "0",
            "name": "",
            "value": 0
          }
        ]
      },
      "virtual": true
    },
    {
      "position": {
        "x": 720,
        "y": 192
      }
    }
  ],

```

(continues on next page)

(continued from previous page)

```

{
  "id": "f6528039-852b-41f9-aa41-268994b3f631",
  "type": "basic.input",
  "data": {
    "name": "s",
    "range": "[1:0]",
    "pins": [
      {
        "index": "1",
        "name": "",
        "value": 0
      },
      {
        "index": "0",
        "name": "",
        "value": 0
      }
    ],
    "virtual": true
  },
  "position": {
    "x": 64,
    "y": 232
  }
},
{
  "id": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
  "type": "basic.code",
  "data": {
    "code": "// Multiplexer 4 to 1\n\nassign out = data[sel];",
    "params": [],
    "ports": {
      "in": [
        {
          "name": "data",
          "range": "[3:0]",
          "size": 4
        },
        {
          "name": "sel",
          "range": "[1:0]",
          "size": 2
        }
      ],
      "out": [
        {
          "name": "out"
        }
      ]
    }
  },
  "position": {
    "x": 312,
    "y": 152
  },
  "size": {
    "width": 272,

```

(continues on next page)

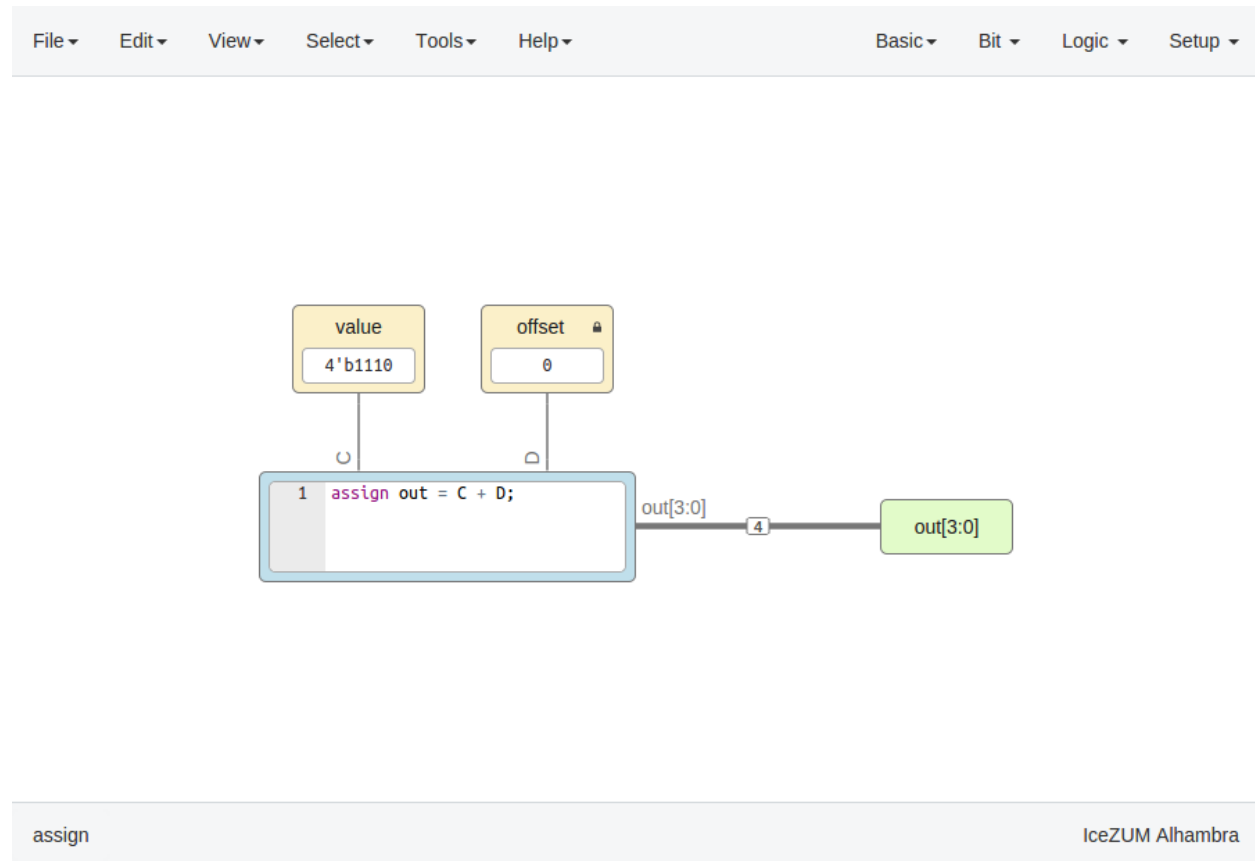
(continued from previous page)

```

        "height": 144
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "95f8c313-6e18-4ee3-b9cf-7266dec53c93",
        "port": "out"
      },
      "target": {
        "block": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
        "port": "data"
      },
      "vertices": [],
      "size": 4
    },
    {
      "source": {
        "block": "f6528039-852b-41f9-aa41-268994b3f631",
        "port": "out"
      },
      "target": {
        "block": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
        "port": "sel"
      },
      "size": 2
    },
    {
      "source": {
        "block": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
        "port": "out"
      },
      "target": {
        "block": "60d40fc8-3388-4066-8f0a-af17e179a9bd",
        "port": "in"
      }
    }
  ]
},
"dependencies": {}
}

```

6.3.4 4. assign



File: assign.ice

Show/Hide code

```
{
  "version": "1.2",
  "package": {
    "name": "Assign",
    "version": "1.1",
    "description": "Assign the value plus an offset to the 4bit output",
    "author": "Jesús Arroyo",
    "image": "%3Csvg%20xmlns=%22http://www.w3.org/2000/svg%22%20width=%22557.531%22
    %20height=%22417.407%22%20viewBox=%220%200%20522.68539%20391.31919%22%3E%3Ctext
    %20style=%22line-height:125%25;text-align:center%22%20x=%22388.929%22%20y=%22571.69
    %22%20font-weight=%22400%22%20font-size=%22382.156%22%20font-family=%22sans-serif%22
    %20letter-spacing=%220%22%20word-spacing=%220%22%20text-anchor=%22middle%22
    %20transform=%22translate(-127.586%20-256.42)%22%3E%3Ctspan%20x=%22388.929%22%20y=
    %22571.69%22%3E%3C/tspan%3E%3C/text%3E%3C/svg%3E"
  }
}
```

(continues on next page)

(continued from previous page)

```

},
"design": {
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "ef743d41-5941-4831-becd-0d930c4eed54",
        "type": "basic.output",
        "data": {
          "name": "out",
          "range": "[3:0]",
          "pins": [
            {
              "index": "3",
              "name": "LED3",
              "value": "98"
            },
            {
              "index": "2",
              "name": "LED2",
              "value": "97"
            },
            {
              "index": "1",
              "name": "LED1",
              "value": "96"
            },
            {
              "index": "0",
              "name": "LED0",
              "value": "95"
            }
          ]
        },
        "virtual": true
      },
      {
        "position": {
          "x": 616,
          "y": 240
        }
      },
      {
        "id": "909655b9-5ef0-4c45-9494-c0238d2e4732",
        "type": "basic.constant",
        "data": {
          "name": "value",
          "value": "4'b1110",
          "local": false
        },
        "position": {
          "x": 192,
          "y": 112
        }
      },
      {
        "id": "7e351e09-634d-407c-ab7e-452519468292",
        "type": "basic.constant",
        "data": {

```

(continues on next page)

(continued from previous page)

```

        "name": "offset",
        "value": "0",
        "local": true
    },
    "position": {
        "x": 328,
        "y": 112
    }
},
{
    "id": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
    "type": "basic.code",
    "data": {
        "code": "assign out = C + D;",
        "params": [
            {
                "name": "C"
            },
            {
                "name": "D"
            }
        ],
        "ports": {
            "in": [],
            "out": [
                {
                    "name": "out",
                    "range": "[3:0]",
                    "size": 4
                }
            ]
        }
    },
    "position": {
        "x": 168,
        "y": 232
    },
    "size": {
        "width": 272,
        "height": 80
    }
},
{
    "wires": [
        {
            "source": {
                "block": "909655b9-5ef0-4c45-9494-c0238d2e4732",
                "port": "constant-out"
            },
            "target": {
                "block": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
                "port": "C"
            }
        },
        {
            "source": {
                "block": "7e351e09-634d-407c-ab7e-452519468292",

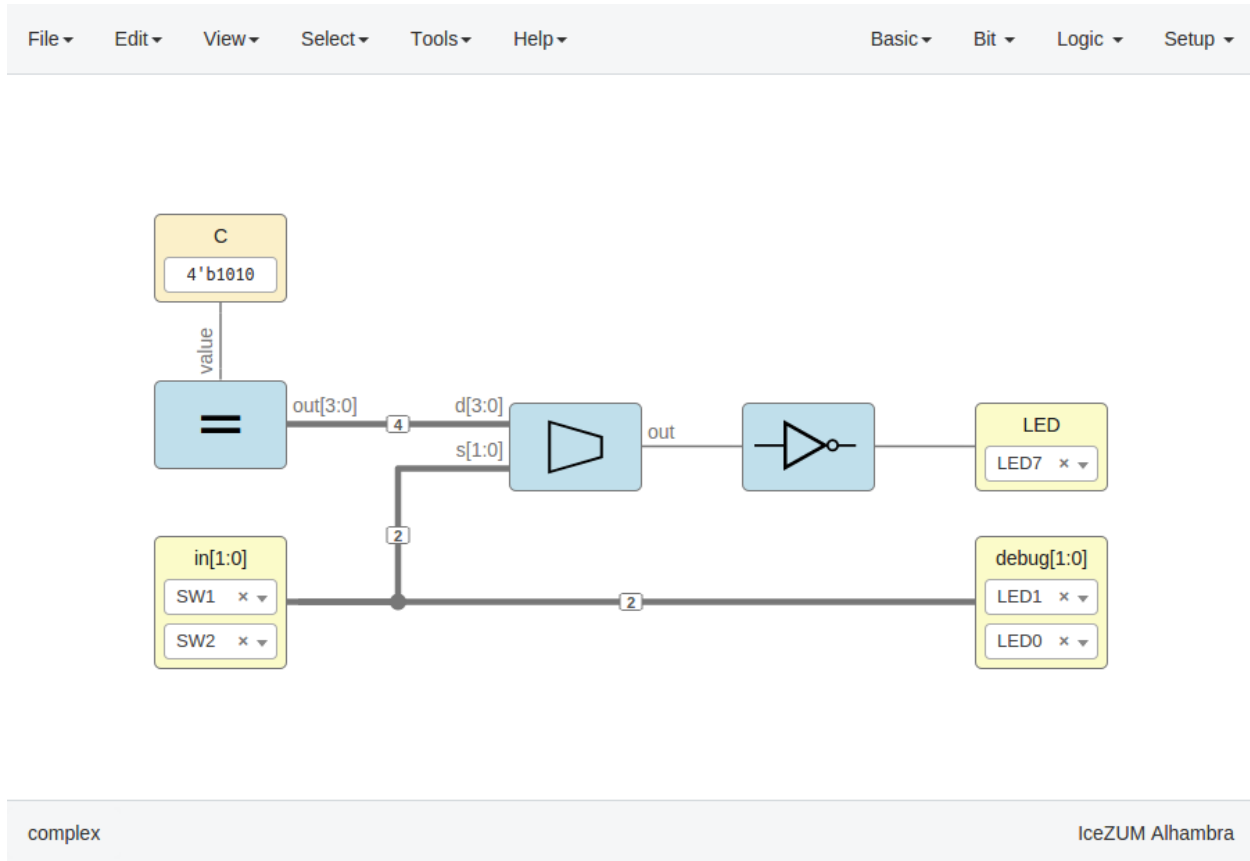
```

(continues on next page)

(continued from previous page)

```
    "port": "constant-out"
  },
  "target": {
    "block": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
    "port": "D"
  }
},
{
  "source": {
    "block": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
    "port": "out"
  },
  "target": {
    "block": "ef743d41-5941-4831-becd-0d930c4eed54",
    "port": "in"
  },
  "size": 4
}
]
}
},
"dependencies": {}
}
```

6.3.5 5. complex

File: **complex.ice**

Show/Hide code

```
{
  "version": "1.2",
  "package": {
    "name": "Complex",
    "version": "1.0",
    "description": "Example including projects as blocks",
    "author": "Jesús Arroyo",
    "image": ""
  },
  "design": {
    "board": "icezum",
    "graph": {
```

(continues on next page)

(continued from previous page)

```

"blocks": [
  {
    "id": "0fbbb687-4a61-4b1d-a022-8884a20bef5c",
    "type": "basic.output",
    "data": {
      "name": "led",
      "pins": [
        {
          "index": "0",
          "name": "LED7",
          "value": "104"
        }
      ],
      "virtual": false
    },
    "position": {
      "x": 704,
      "y": 224
    }
  },
  {
    "id": "21e9e7f9-9b8a-4fca-904d-e266f1496454",
    "type": "basic.input",
    "data": {
      "name": "in",
      "range": "[1:0]",
      "pins": [
        {
          "index": "1",
          "name": "SW1",
          "value": "10"
        },
        {
          "index": "0",
          "name": "SW2",
          "value": "11"
        }
      ],
      "virtual": false
    },
    "position": {
      "x": 112,
      "y": 320
    }
  },
  {
    "id": "6c809d38-547d-4c70-92eb-2d5c389429e7",
    "type": "basic.output",
    "data": {
      "name": "debug",
      "range": "[1:0]",
      "pins": [
        {
          "index": "1",
          "name": "LED1",
          "value": "96"
        }
      ],

```

(continues on next page)

(continued from previous page)

```

        {
            "index": "0",
            "name": "LED0",
            "value": "95"
        }
    ],
    "virtual": false
},
"position": {
    "x": 704,
    "y": 320
}
},
{
    "id": "bfebb831-8b03-43e1-9b87-013f1b5a9cdf",
    "type": "basic.constant",
    "data": {
        "name": "C",
        "value": "4'b1010",
        "local": false
    },
    "position": {
        "x": 112,
        "y": 88
    }
},
{
    "id": "83038d5e-e513-4677-9621-86982a8451e2",
    "type": "020b28e4cf51f767c7a70ef39afda26bbc780652",
    "position": {
        "x": 368,
        "y": 224
    },
    "size": {
        "width": 96,
        "height": 64
    }
},
{
    "id": "f18340fe-86cb-4cb4-8d4c-38abcdf9ac5d",
    "type": "c6871b49e229a4923964e3e03a842f9d8dd4e640",
    "position": {
        "x": 536,
        "y": 224
    },
    "size": {
        "width": 96,
        "height": 64
    }
},
{
    "id": "2f51c3f2-24ea-4a05-b6ce-fba02ebd165b",
    "type": "ea66cd513caebe3f7faa727d2a9d7e49f8d4e73c",
    "position": {
        "x": 112,
        "y": 208
    }
},

```

(continues on next page)

(continued from previous page)

```

    "size": {
      "width": 96,
      "height": 64
    }
  },
],
"wires": [
  {
    "source": {
      "block": "21e9e7f9-9b8a-4fca-904d-e266f1496454",
      "port": "out"
    },
    "target": {
      "block": "6c809d38-547d-4c70-92eb-2d5c389429e7",
      "port": "in"
    },
    "size": 2
  },
  {
    "source": {
      "block": "f18340fe-86cb-4cb4-8d4c-38abcdf9ac5d",
      "port": "07895985-9d14-4a6f-8f2d-b2a6ddf61852"
    },
    "target": {
      "block": "0fbbb687-4a61-4b1d-a022-8884a20bef5c",
      "port": "in"
    }
  },
  {
    "source": {
      "block": "21e9e7f9-9b8a-4fca-904d-e266f1496454",
      "port": "out"
    },
    "target": {
      "block": "83038d5e-e513-4677-9621-86982a8451e2",
      "port": "f6528039-852b-41f9-aa41-268994b3f631"
    },
    "vertices": [
      {
        "x": 288,
        "y": 288
      }
    ],
    "size": 2
  },
  {
    "source": {
      "block": "83038d5e-e513-4677-9621-86982a8451e2",
      "port": "60d40fc8-3388-4066-8f0a-af17e179a9bd"
    },
    "target": {
      "block": "f18340fe-86cb-4cb4-8d4c-38abcdf9ac5d",
      "port": "a4058fa5-b66e-4e5e-b542-28d7c3e9d3cd"
    }
  },
  {
    "source": {

```

(continues on next page)

(continued from previous page)

```

        "block": "bfebb831-8b03-43e1-9b87-013f1b5a9cdf",
        "port": "constant-out"
    },
    "target": {
        "block": "2f51c3f2-24ea-4a05-b6ce-fba02ebd165b",
        "port": "909655b9-5ef0-4c45-9494-c0238d2e4732"
    }
},
{
    "source": {
        "block": "2f51c3f2-24ea-4a05-b6ce-fba02ebd165b",
        "port": "ef743d41-5941-4831-becd-0d930c4eed54"
    },
    "target": {
        "block": "83038d5e-e513-4677-9621-86982a8451e2",
        "port": "95f8c313-6e18-4ee3-b9cf-7266dec53c93"
    },
    "size": 4
}
]
}
},
"dependencies": {
    "020b28e4cf51f767c7a70ef39afda26bbc780652": {
        "package": {
            "name": "Mux4:1",
            "version": "1.1",
            "description": "Multiplexer 4 to 1",
            "author": "Jesús Arroyo",
            "image": "%3Csvg%20xmlns=%22http://www.w3.org/2000/svg%22%20viewBox=%22-252
↪%20400.9%2081%2040%22%20width=%2281%22%20height=%2240%22%3E%3Cpath%20d=%22M-191
↪%20419.9v-7.2l-41-11.8v40l41-11.7v-7.4zm-39%2018.5v-35l37%2010.8v13.5z%22/%3E%3C/svg
↪%3E"
        },
        "design": {
            "graph": {
                "blocks": [
                    {
                        "id": "95f8c313-6e18-4ee3-b9cf-7266dec53c93",
                        "type": "basic.input",
                        "data": {
                            "name": "d",
                            "range": "[3:0]",
                            "size": 4
                        },
                        "position": {
                            "x": 64,
                            "y": 160
                        }
                    },
                    {
                        "id": "60d40fc8-3388-4066-8f0a-af17e179a9bd",
                        "type": "basic.output",
                        "data": {
                            "name": "out"
                        },
                        "position": {

```

(continues on next page)

(continued from previous page)

```

        "x": 720,
        "y": 192
    },
    {
        "id": "f6528039-852b-41f9-aa41-268994b3f631",
        "type": "basic.input",
        "data": {
            "name": "s",
            "range": "[1:0]",
            "size": 2
        },
        "position": {
            "x": 64,
            "y": 232
        }
    },
    {
        "id": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
        "type": "basic.code",
        "data": {
            "code": "// Multiplexer 4 to 1\n\nassign out = data[sel];",
            "params": [],
            "ports": {
                "in": [
                    {
                        "name": "data",
                        "range": "[3:0]",
                        "size": 4
                    },
                    {
                        "name": "sel",
                        "range": "[1:0]",
                        "size": 2
                    }
                ],
                "out": [
                    {
                        "name": "out"
                    }
                ]
            }
        },
        "position": {
            "x": 312,
            "y": 152
        },
        "size": {
            "width": 272,
            "height": 144
        }
    }
],
"wires": [
    {
        "source": {
            "block": "95f8c313-6e18-4ee3-b9cf-7266dec53c93",

```

(continues on next page)

(continued from previous page)

```

        "port": "out"
      },
      "target": {
        "block": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
        "port": "data"
      },
      "vertices": [],
      "size": 4
    },
    {
      "source": {
        "block": "f6528039-852b-41f9-aa41-268994b3f631",
        "port": "out"
      },
      "target": {
        "block": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
        "port": "sel"
      },
      "size": 2
    },
    {
      "source": {
        "block": "5e1563d7-86de-4618-a9b0-2a08075af9ec",
        "port": "out"
      },
      "target": {
        "block": "60d40fc8-3388-4066-8f0a-af17e179a9bd",
        "port": "in"
      }
    }
  ]
}
},
"6871b49e229a4923964e3e03a842f9d8dd4e640": {
  "package": {
    "name": "Not",
    "version": "1.0",
    "description": "NOT logic gate",
    "author": "Jesús Arroyo",
    "image": "%3Csvg%20xmlns=%22http://www.w3.org/2000/svg%22%20width=%2291.33%22%20height=%2245.752%22%20version=%221%22%3E%3Cpath%20d=%22M0%2020.446h27v2H0zm70.322.001h15.3v2h-15.3z%22/%3E%3Cpath%20d=%22M66.05%2026.746c-2.9%200-5.3-2.4-5.3-5.3s2.4-5.3%205.3-5.3%205.3%202.4%205.3%205.3-2.4%205.3-5.3%205.3zm0-8.6c-1.8%200-3.3%201.5-3.3%203.3%200%201.8%201.5%203.3%203.3%203.3%201.8%200%203.3-1.5%203.3-3.3%200-1.8-1.5-3.3-3.3-3.3z%22/%3E%3Cpath%20d=%22M25.962%202.563l33.624%2018.883L25.962%2040.33v2.563z%22%20fill=%22none%22%20stroke=%22#000%22%20stroke-width=%223%22/%3E%3C/svg%3E"
  },
  "design": {
    "graph": {
      "blocks": [
        {
          "id": "a4058fa5-b66e-4e5e-b542-28d7c3e9d3cd",
          "type": "basic.input",
          "data": {
            "name": ""
          }
        }
      ]
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "position": {
      "x": 72,
      "y": 184
    }
  },
  {
    "id": "07895985-9d14-4a6f-8f2d-b2a6ddf61852",
    "type": "basic.output",
    "data": {
      "name": ""
    },
    "position": {
      "x": 728,
      "y": 184
    }
  },
  {
    "id": "364b95cc-e8ff-4c65-b332-d6125c5968ee",
    "type": "basic.code",
    "data": {
      "code": "// NOT logic gate\n\nassign b = ~a;",
      "params": [],
      "ports": {
        "in": [
          {
            "name": "a"
          }
        ],
        "out": [
          {
            "name": "b"
          }
        ]
      }
    },
    "position": {
      "x": 248,
      "y": 88
    },
    "size": {
      "width": 384,
      "height": 256
    }
  }
],
"wires": [
  {
    "source": {
      "block": "a4058fa5-b66e-4e5e-b542-28d7c3e9d3cd",
      "port": "out"
    },
    "target": {
      "block": "364b95cc-e8ff-4c65-b332-d6125c5968ee",
      "port": "a"
    }
  }
],

```

(continues on next page)

(continued from previous page)

```

        {
            "source": {
                "block": "364b95cc-e8ff-4c65-b332-d6125c5968ee",
                "port": "b"
            },
            "target": {
                "block": "07895985-9d14-4a6f-8f2d-b2a6ddf61852",
                "port": "in"
            }
        }
    ]
}
},
"ea66cd513caebe3f7faa727d2a9d7e49f8d4e73c": {
    "package": {
        "name": "Assign",
        "version": "1.1",
        "description": "Assign the value plus an offset to the 4bit output",
        "author": "Jesús Arroyo",
        "image": "%3Csvg%20xmlns=%22http://www.w3.org/2000/svg%22%20width=%22557.531
↪%22%20height=%22417.407%22%20viewBox=%220%200%20522.68539%20391.31919%22%3E%3Ctext
↪%20style=%22line-height:125%25;text-align:center%22%20x=%22388.929%22%20y=%22571.69
↪%22%20font-weight=%22400%22%20font-size=%22382.156%22%20font-family=%22sans-serif%22
↪%20letter-spacing=%220%22%20word-spacing=%220%22%20text-anchor=%22middle%22
↪%20transform=%22translate(-127.586%20-256.42)%22%3E%3Ctspan%20x=%22388.929%22%20y=
↪%22571.69%22%3E%3C/tspan%3E%3C/text%3E%3C/svg%3E"
    },
    "design": {
        "graph": {
            "blocks": [
                {
                    "id": "ef743d41-5941-4831-becd-0d930c4eed54",
                    "type": "basic.output",
                    "data": {
                        "name": "out",
                        "range": "[3:0]",
                        "size": 4
                    },
                    "position": {
                        "x": 616,
                        "y": 240
                    }
                },
                {
                    "id": "909655b9-5ef0-4c45-9494-c0238d2e4732",
                    "type": "basic.constant",
                    "data": {
                        "name": "value",
                        "value": "4'b1110",
                        "local": false
                    },
                    "position": {
                        "x": 192,
                        "y": 112
                    }
                }
            ]
        }
    }
},

```

(continues on next page)

(continued from previous page)

```

    {
      "id": "7e351e09-634d-407c-ab7e-452519468292",
      "type": "basic.constant",
      "data": {
        "name": "offset",
        "value": "0",
        "local": true
      },
      "position": {
        "x": 328,
        "y": 112
      }
    },
    {
      "id": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
      "type": "basic.code",
      "data": {
        "code": "assign out = C + D;",
        "params": [
          {
            "name": "C"
          },
          {
            "name": "D"
          }
        ],
        "ports": {
          "in": [],
          "out": [
            {
              "name": "out",
              "range": "[3:0]",
              "size": 4
            }
          ]
        },
        "position": {
          "x": 168,
          "y": 232
        },
        "size": {
          "width": 272,
          "height": 80
        }
      }
    },
    "wires": [
      {
        "source": {
          "block": "909655b9-5ef0-4c45-9494-c0238d2e4732",
          "port": "constant-out"
        },
        "target": {
          "block": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
          "port": "C"
        }
      }
    ]
  }

```

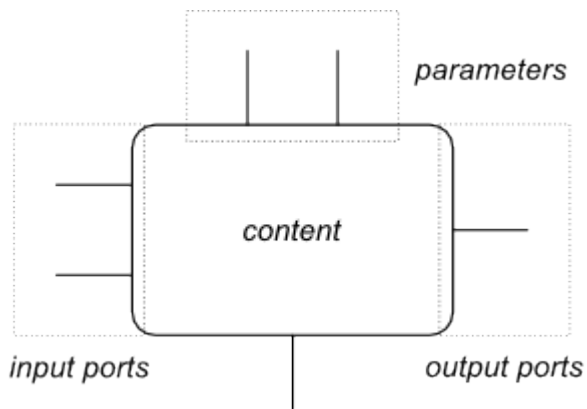
(continues on next page)

(continued from previous page)

```
    },
    {
      "source": {
        "block": "7e351e09-634d-407c-ab7e-452519468292",
        "port": "constant-out"
      },
      "target": {
        "block": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
        "port": "D"
      }
    },
    {
      "source": {
        "block": "b6bc7556-6362-45ca-80e5-6db7a3100c7d",
        "port": "out"
      },
      "target": {
        "block": "ef743d41-5941-4831-becd-0d930c4eed54",
        "port": "in"
      },
      "size": 4
    }
  ]
}
}
```

7.1 Definition

A block is an entity with *input* and *output* ports, parameters and some content.



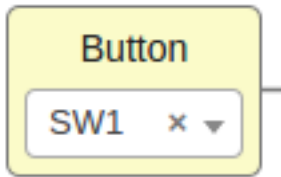
7.2 Basic blocks

7.2.1 Input block

- Type: `basic.input`
- States:
 - FPGA pin: *yellow*
 - Virtual: *green*
- Show clock

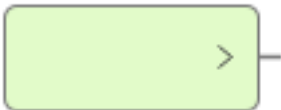
Wire

E.g.: basic FPGA input block with name *Button*.



```
{
  "data": {
    "name": "Button",
    "pins": [
      {
        "index": "0",
        "name": "SW1",
        "value": "10"
      }
    ],
    "virtual": false,
    "clock": false
  }
}
```

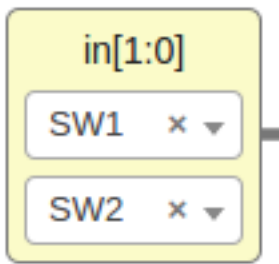
E.g.: basic Virtual input block with no name and the clock symbol.



```
{
  "data": {
    "name": "",
    "pins": [
      {
        "index": "0",
        "name": "",
        "value": ""
      }
    ],
    "virtual": true,
    "clock": true
  }
}
```

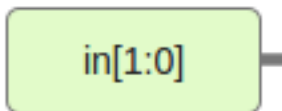
Bus

E.g.: basic FPGA input block with name *in[1:0]*.



```
{
  "data": {
    "name": "in",
    "range": "[1:0]",
    "pins": [
      {
        "index": "1",
        "name": "SW1",
        "value": "10"
      },
      {
        "index": "0",
        "name": "SW2",
        "value": "11"
      }
    ]
  },
  "virtual": false,
  "clock": false
}
```

E.g.: basic Virtual input block with name *in[1:0]*.



```
{
  "data": {
    "name": "in",
    "range": "[1:0]",
    "pins": [
      {
        "index": "1",
        "name": "",
        "value": ""
      },
      {
        "index": "0",
        "name": "",
        "value": ""
      }
    ]
  },
  "virtual": false,
  "clock": false
}
```

(continues on next page)

(continued from previous page)

```

    "virtual": true,
    "clock": false
  }
}

```

7.2.2 Output block

- Type: `basic.output`
- States:
 - FPGA pin: *yellow*
 - Virtual: *green*

Wire

E.g.: basic FPGA output block with no name.

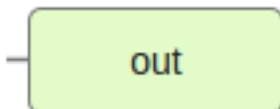


```

{
  "data": {
    "name": "",
    "pins": [
      {
        "index": "0",
        "name": "LED0",
        "value": "95"
      }
    ],
    "virtual": false
  }
}

```

E.g.: basic Virtual output block with name “out”.



```

{
  "data": {
    "name": "out",
    "pins": [
      {
        "index": "0",

```

(continues on next page)

(continued from previous page)

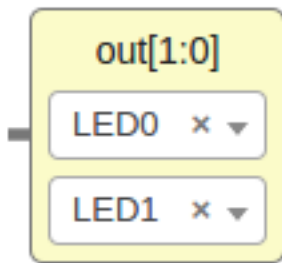
```

        "name": "",
        "value": ""
    }
],
    "virtual": true
}
}

```

Bus

E.g.: basic FPGA output block with name *out[1:0]*.

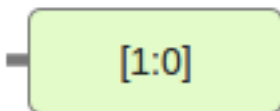


```

{
  "data": {
    "name": "out",
    "range": "[1:0]",
    "pins": [
      {
        "index": "1",
        "name": "LED0",
        "value": "95"
      },
      {
        "index": "0",
        "name": "LED1",
        "value": "96"
      }
    ]
  },
  "virtual": false
}

```

E.g.: basic Virtual output block with name *[1:0]*.

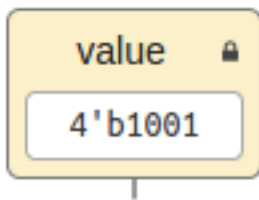


```
{
  "data": {
    "name": "",
    "range": "[1:0]",
    "pins": [
      {
        "index": "1",
        "name": "",
        "value": ""
      },
      {
        "index": "0",
        "name": "",
        "value": ""
      }
    ]
  },
  "virtual": true
}
```

7.2.3 Constant block

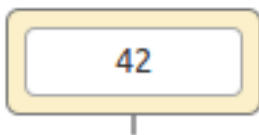
- Type: `basic.constant`
- States:
 - Local parameter (lock)

E.g.: basic constant block with name *value* and the local flag.



```
{
  "data": {
    "name": "value",
    "value": "4'b1001",
    "local": true
  }
}
```

E.g.: basic constant block with no name and no local flag.

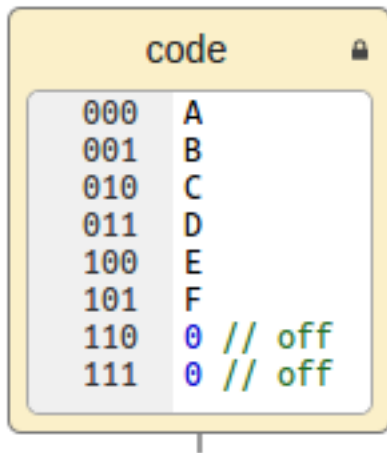


```
{
  "data": {
    "name": "",
    "value": "42",
    "local": false
  }
}
```

7.2.4 Memory block

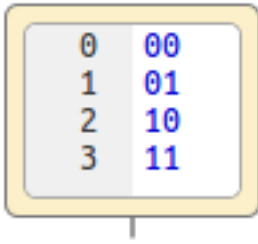
- Type: `basic.memory`
- Address format:
 - Binary
 - Decimal
 - Hexadecimal
- States:
 - Local parameter (lock)

E.g.: basic memory block with name *code*, binary address format and the local flag.



```
{
  "data": {
    "name": "code",
    "list": "A \nB\nC\nD\nE\nF\n0 // off\n0 // off",
    "local": true,
    "format": 2
  }
}
```

E.g.: basic memory block with no name, decimal address format and no local flag.

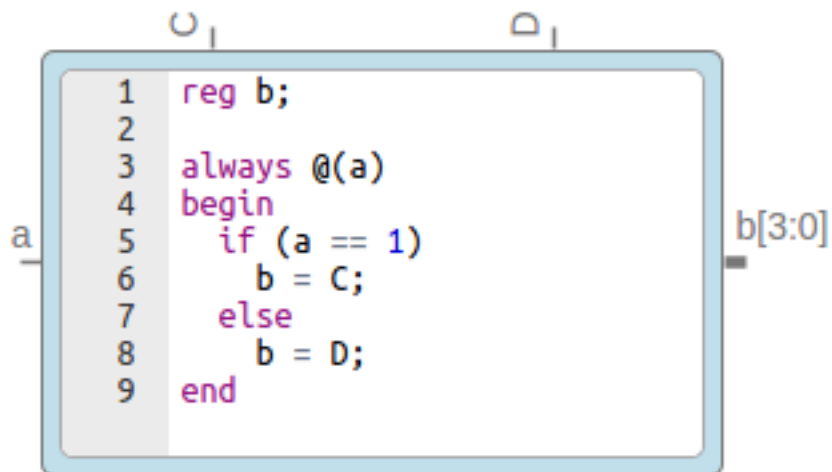


```
{
  "data": {
    "name": "",
    "list": "00\n01\n10\n11",
    "local": false,
    "format": 10
  }
}
```

7.2.5 Code block

- Type: `basic.code`

E.g.: basic code block with input port *a*, output port *b[3:0]* and parameters *C* and *D*.



```
{
  "data": {
    "code": "reg [3:0] b_aux;\n\nalways @(a)\nbegin\n  if (a == 1)\n    b_aux = C;\n  else\n    b_aux = D;\nend\n\nassign b = b_aux;\n",
    "params": [
      {
        "name": "C"
      },
      {
        "name": "D"
      }
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```

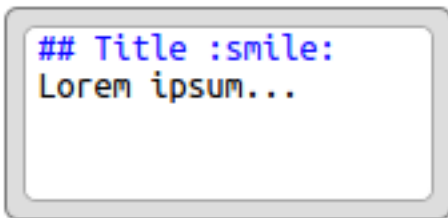
    }
  ],
  "ports": {
    "in": [
      {
        "name": "a"
      }
    ],
    "out": [
      {
        "name": "b",
        "range": "[3:0]",
        "size": 4
      }
    ]
  }
}

```

7.2.6 Information block

- Type: `basic.info`
- States:
 - Readonly

E.g.: basic information block in editor mode.



```

{
  "data": {
    "info": "## Title :smile:\nLorem ipsum...\n",
    "readonly": false
  }
}

```

E.g.: basic information block in render mode.

Title 😊

Lorem ipsum...

```
{
  "data": {
    "info": "## Title :smile:\nLorem ipsum...\n",
    "readonly": true
  }
}
```

7.3 Generic blocks

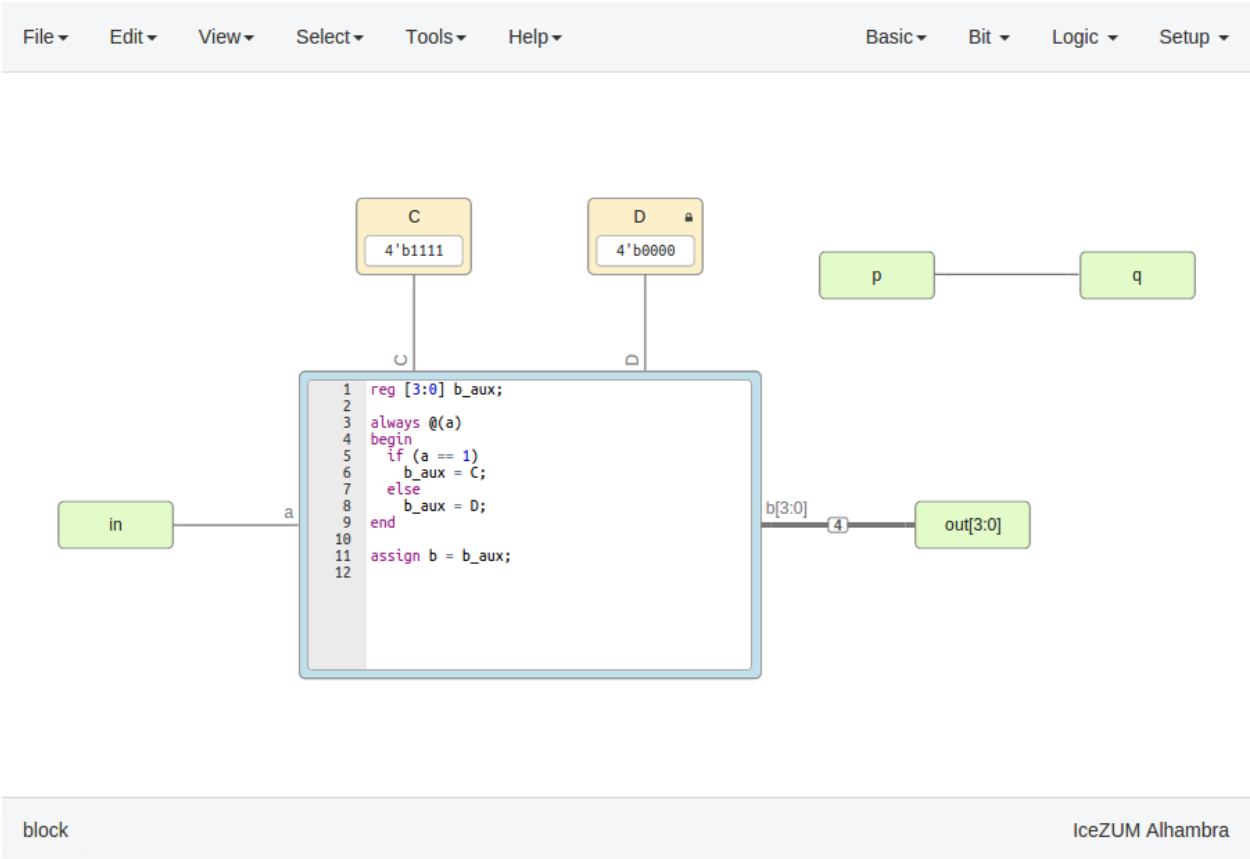
Any project can be added as a read-only **generic block**:

- The *input blocks* become *input ports*.
- The *output blocks* become *output ports*.
- The *constant blocks* become *parameters*.

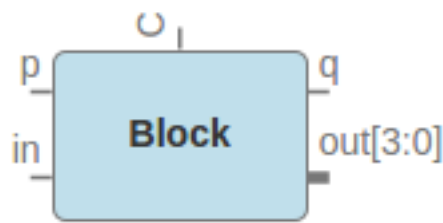
The block information is stored in **dependencies**, without the unnecessary information:

- The version number is removed.
- The FPGA *board* is removed.
- The FPGA *data.pins* are removed.
- An additional field *data.size* with the *pins.length* is created if greater than 1.
- The *data.virtual* flag is removed.

E.g.: this project *block.ice*.



becomes this block:



Show/Hide code

CHAPTER 8

Board Rules

Icestudio *board rules* allow to automate tasks such as connecting all input *CLK* wires in a sequential circuit or initializing certain output ports to 1 or 0. The result is an easier and cleaner design. In Icestudio, rules can be enabled or disabled in “Edit > Preferences > Board rules”.

The board rules are implemented in the *rules.json* file in each board directory. Each rule file is attached to a specific board. For example, this could be a rules file for the *IceZUM Alhambra*:

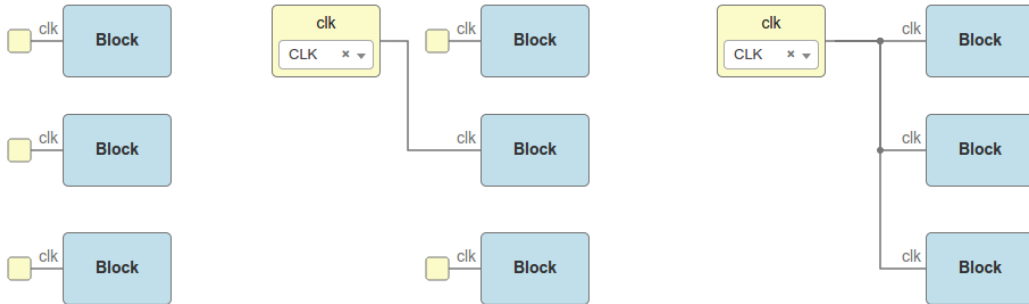
File: **rules.json**

```
{
  "input" : [
    {
      "port" : "clk",
      "pin": "21"
    }
  ],
  "output": [
    {
      "pin": "95",
      "bit": "0"
    },
    {
      "pin": "96",
      "bit": "0"
    },
    {
      "pin": "97",
      "bit": "0"
    },
    {
      "pin": "98",
      "bit": "0"
    }
  ]
}
```

8.1 Input rules

Input rules define default connections between not connected ports with the name **port** and the specified FPGA I/O **pin**. In the example above, all not connected input ports named “clk” are automatically connected to the pin “21”.

With the rules enabled, these three designs are equivalent:

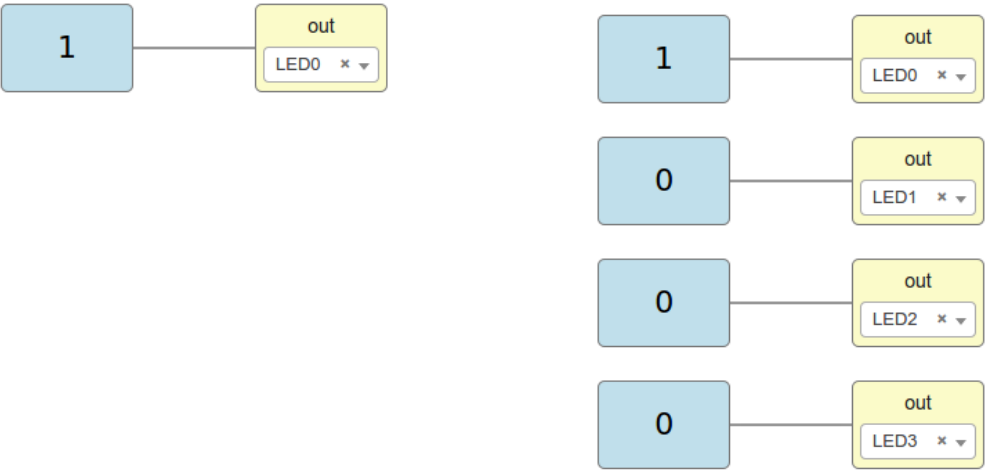


Note: If an input rule is set for ports named “clk”, it also applies to input ports with “Show clock” enabled.

8.2 Output rules

Output rules define default **bit** values for unused output FPGA I/O **pins**. In the example above, output pins “95”, “96”, “97” and “98” will be initialized to “0” if they do not appear in the circuit.

With the rules enabled, these two designs are equivalent:



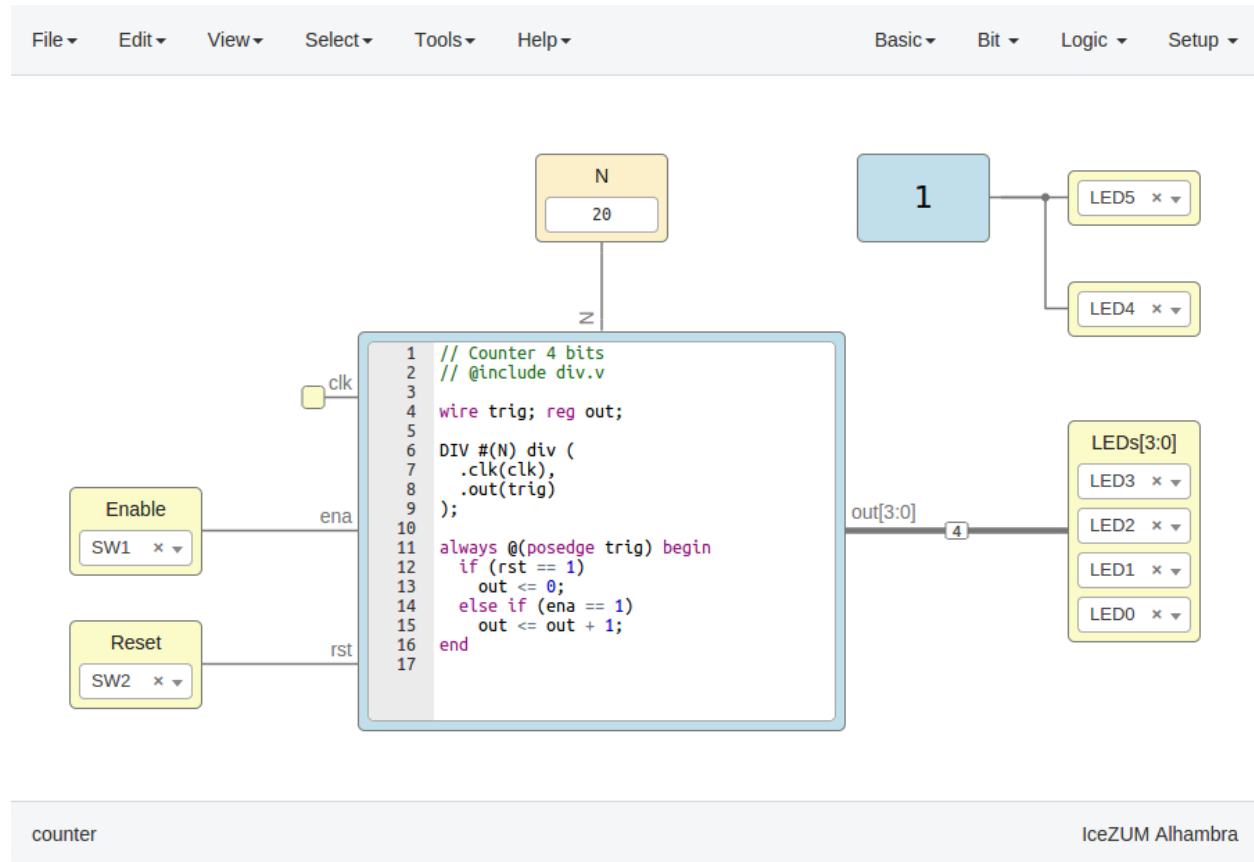
1. Verilog
2. PCF
3. Testbench
4. GTKWave

9.1 Implementation

Compiler [source code](#).

9.2 Sample

9.2.1 Counter



counter.ice

```

{
  "version": "1.2",
  "package": {
    "name": "counter",
    "version": "1.0",
    "description": "4-bit counter. N is the number of bits to count.",
    "author": "Jesús Arroyo",
    "image": ""
  },
  "design": {
    "board": "icezum",
    "graph": {
      "blocks": [
        {
          "id": "1a49c635-92d6-4641-bd3b-dbd7604a76bf",

```

(continues on next page)

(continued from previous page)

```

    "type": "basic.output",
    "data": {
      "name": "",
      "pins": [
        {
          "index": "0",
          "name": "LED5",
          "value": "101"
        }
      ],
      "virtual": false
    },
    "position": {
      "x": 760,
      "y": 56
    }
  },
  {
    "id": "1f3764d6-7db2-4e5a-912d-a25aad6459e2",
    "type": "basic.output",
    "data": {
      "name": "",
      "pins": [
        {
          "index": "0",
          "name": "LED4",
          "value": "99"
        }
      ],
      "virtual": false
    },
    "position": {
      "x": 760,
      "y": 136
    }
  },
  {
    "id": "4c30cdb0-f1af-4af1-bb4b-12e443b84a17",
    "type": "basic.output",
    "data": {
      "name": "LEDs",
      "range": "[3:0]",
      "pins": [
        {
          "index": "3",
          "name": "LED3",
          "value": "98"
        },
        {
          "index": "2",
          "name": "LED2",
          "value": "97"
        },
        {
          "index": "1",
          "name": "LED1",
          "value": "96"
        }
      ]
    }
  }
}

```

(continues on next page)

(continued from previous page)

```

        },
        {
            "index": "0",
            "name": "LED0",
            "value": "95"
        }
    ],
    "virtual": false
},
"position": {
    "x": 760,
    "y": 248
}
},
{
    "id": "9803de82-f844-48f0-9f6a-b428395073b4",
    "type": "basic.input",
    "data": {
        "name": "Enable",
        "pins": [
            {
                "index": "0",
                "name": "SW1",
                "value": "10"
            }
        ],
        "virtual": false
    },
    "position": {
        "x": 40,
        "y": 296
    }
},
{
    "id": "4caf869e-5202-4aa0-acbf-14fac565eaf1",
    "type": "basic.input",
    "data": {
        "name": "Reset",
        "pins": [
            {
                "index": "0",
                "name": "SW2",
                "value": "11"
            }
        ],
        "virtual": false
    },
    "position": {
        "x": 40,
        "y": 392
    }
},
{
    "id": "02460189-14a0-48d0-ad87-74faf9a1177e",
    "type": "basic.constant",
    "data": {
        "name": "N",

```

(continues on next page)

(continued from previous page)

```

        "value": "20",
        "local": false
    },
    "position": {
        "x": 376,
        "y": 56
    }
},
{
    "id": "976c6f41-7ed1-41b5-953b-cd4a5709c701",
    "type": "3e6c249e205080168c1bf4ee8dbc33b50653d5f4",
    "position": {
        "x": 608,
        "y": 56
    },
    "size": {
        "width": 96,
        "height": 64
    }
},
{
    "id": "e38831b6-fd92-4e35-9fea-17b439002721",
    "type": "basic.code",
    "data": {
        "code": "// Counter 4 bits\n// @include div.v\n\nwire trig; reg out;\n→\n\nDIV #(N) div (\n    .clk(clk),\n    .out(trig)\n);\n\nalways @(posedge trig)\n→begin\n    if (rst == 1)\n        out <= 0;\n    else if (ena == 1)\n        out <= out + 1;\n→\nend\n",
        "params": [
            {
                "name": "N"
            }
        ],
        "ports": {
            "in": [
                {
                    "name": "clk"
                },
                {
                    "name": "ena"
                },
                {
                    "name": "rst"
                }
            ],
            "out": [
                {
                    "name": "out",
                    "range": "[3:0]",
                    "size": 4
                }
            ]
        }
    },
    "position": {
        "x": 248,
        "y": 184
    }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "size": {
      "width": 352,
      "height": 288
    }
  },
  ],
  "wires": [
    {
      "source": {
        "block": "02460189-14a0-48d0-ad87-74faf9a1177e",
        "port": "constant-out"
      },
      "target": {
        "block": "e38831b6-fd92-4e35-9fea-17b439002721",
        "port": "N"
      }
    },
    {
      "source": {
        "block": "9803de82-f844-48f0-9f6a-b428395073b4",
        "port": "out"
      },
      "target": {
        "block": "e38831b6-fd92-4e35-9fea-17b439002721",
        "port": "ena"
      }
    },
    {
      "source": {
        "block": "4caf869e-5202-4aa0-acbf-14fac565eaf1",
        "port": "out"
      },
      "target": {
        "block": "e38831b6-fd92-4e35-9fea-17b439002721",
        "port": "rst"
      }
    },
    {
      "source": {
        "block": "976c6f41-7ed1-41b5-953b-cd4a5709c701",
        "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
      },
      "target": {
        "block": "1a49c635-92d6-4641-bd3b-dbd7604a76bf",
        "port": "in"
      }
    },
    {
      "source": {
        "block": "976c6f41-7ed1-41b5-953b-cd4a5709c701",
        "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
      },
      "target": {
        "block": "1f3764d6-7db2-4e5a-912d-a25aad6459e2",
        "port": "in"
      }
    }
  ]
}

```

(continues on next page)

(continued from previous page)

```

    },
    {
      "source": {
        "block": "e38831b6-fd92-4e35-9fea-17b439002721",
        "port": "out"
      },
      "target": {
        "block": "4c30cdb0-flaf-4af1-bb4b-12e443b84a17",
        "port": "in"
      },
      "size": 4
    }
  ]
}
},
"dependencies": {
  "3e6c249e205080168c1bf4ee8dbc33b50653d5f4": {
    "package": {
      "name": "Bit 1",
      "version": "1.0.0",
      "description": "Assign 1 to the output wire",
      "author": "Jesús Arroyo",
      "image": "%3Csvg%20xmlns=%22http://www.w3.org/2000/svg%22%20width=%2247.303%22%20height=%2227.648%22%20viewBox=%220%200%2044.346456%2025.919999%22%3E%3Ctext%20style=%22line-height:125%25%22%20x=%22325.218%22%20y=%22315.455%22%20font-weight=%22400%22%20font-size=%2212.669%22%20font-family=%22sans-serif%22%20letter-spacing=%220%22%20word-spacing=%220%22%20transform=%22translate(-307.01%20-298.51)%22%3E%3Ctspan%20x=%22325.218%22%20y=%22315.455%22%20style=%22-inkscape-font-specification:'Courier%2010%20Pitch'%22%20font-family=%22Courier%2010%20Pitch%22%3E1%3C/tspan%3E%3C/text%3E%3C/svg%3E"
    },
    "design": {
      "graph": {
        "blocks": [
          {
            "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
            "type": "basic.code",
            "data": {
              "code": "// Bit 1\n\nassign v = 1'b1;",
              "params": [],
              "ports": {
                "in": [],
                "out": [
                  {
                    "name": "v"
                  }
                ]
              }
            }
          }
        ],
        "position": {
          "x": 96,
          "y": 96
        }
      },
      {
        "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
        "type": "basic.output",

```

(continues on next page)

(continued from previous page)

```

        "data": {
            "name": ""
        },
        "position": {
            "x": 608,
            "y": 192
        }
    }
],
"wires": [
    {
        "source": {
            "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
            "port": "v"
        },
        "target": {
            "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
            "port": "in"
        }
    }
]
}
}
}
}
```

div.v

```
module DIV (input clk, output out);

    parameter N = 22;
    localparam M = $pow(2, N); // 2**N

    wire clk_temp;
    reg [N - 1:0] c = 0;
    reg dout;

    assign out = dout;

    always @(posedge clk)
        if (M == 0)
            c <= 0;
        else if (c == M - 1)
            c <= 0;
        else
            c <= c + 1;

endmodule
```

(continues on next page)

(continued from previous page)

```

assign clk_temp = (c == 0) ? 1 : 0;

always @(posedge clk)
  if (N == 0)
    out <= 0;
  else if (clk_temp == 1)

dout <= ~dout;

endmodule

```

Generates

counter.v

```

// Code generated by Icestudio 0.3.3-rc
// Tue, 24 Jul 2018 16:04:48 GMT

`default_nettype none

module main #(
  parameter v30f40a = 20
) (
  input vb8a806,
  input ve0befb,
  input vclk,
  output v32232e,
  output v1dlaf8,
  output [3:0] v6a65cd,
  output [0:1] vinit
);
  localparam p0 = v30f40a;
  wire w1;
  wire w2;
  wire w3;
  wire w4;
  wire [0:3] w5;
  wire w6;
  assign w1 = vb8a806;
  assign w2 = ve0befb;
  assign v32232e = w3;
  assign v1dlaf8 = w4;
  assign v6a65cd = w5;
  assign w6 = vclk;
  assign w4 = w3;

```

(continues on next page)

(continued from previous page)

```

v3e6c24 vc4b1b9 (
    .v608bd9(w3)
);
main_v4549a6 #(
    .N(p0)
) v4549a6 (
    .ena(w1),
    .rst(w2),
    .out(w5),
    .clk(w6)
);
assign vinit = 2'b00;
endmodule

module v3e6c24 (
    output v608bd9
);
    wire w0;
    assign v608bd9 = w0;
    v3e6c24_v68c173 v68c173 (
        .v(w0)
    );
endmodule

module v3e6c24_v68c173 (
    output v
);
    // Bit 1

    assign v = 1'b1;
endmodule

module main_v4549a6 #(
    parameter N = 0
) (
    input clk,
    input ena,
    input rst,
    output [3:0] out
);
    // Counter 4 bits
    // @include div.v

    wire trig; reg out;

    DIV #(N) div (
        .clk(clk),
        .out(trig)
    );

    always @(posedge trig) begin
        if (rst == 1)
            out <= 0;
        else if (ena == 1)
            out <= out + 1;
    end
end

```

(continues on next page)

(continued from previous page)

endmodule**counter.pcf**

```

# Code generated by Icestudio 0.3.3-rc
# Tue, 24 Jul 2018 16:04:52 GMT

set_io v32232e 101
set_io v1d1af8 99
set_io v6a65cd[3] 98
set_io v6a65cd[2] 97
set_io v6a65cd[1] 96
set_io v6a65cd[0] 95
set_io vb8a806 10
set_io ve0befb 11
set_io vclk 21
set_io vinit[0] 102
set_io vinit[1] 104

```

counter_tb.v

```

// Code generated by Icestudio 0.3.3-rc
// Tue, 24 Jul 2018 16:04:57 GMT

// Testbench template

`default_nettype none
`define DUMPSTR(x) `".x.vcd`"
`timescale 10 ns / 1 ns

module main_tb
;

    // Simulation time: 100ns (10 * 10ns)
    parameter DURATION = 10;

```

(continues on next page)

(continued from previous page)

```
// TODO: edit the module parameters here
// e.g. localparam constant_value = 1;
localparam constant_N = 20;

// Input/Output
reg Enable;
reg Reset;
wire 0;
wire 1;
wire [3:0] LEDs;

// Module instance
main #(
    .v30f40a(constant_N)
) MAIN (
    .vb8a806(Enable),
    .ve0befb(Reset),
    .v32232e(0),
    .vldlaf8(1),
    .v6a65cd(LEDs)
);

initial begin
    // File were to store the simulation results
    $dumpfile(`DUMPSTR(`VCD_OUTPUT));
    $dumpvars(0, main_tb);

    // TODO: initialize the registers here
    // e.g. value = 1;
    // e.g. #2 value = 0;
    Enable = 0;
    Reset = 0;

    #(DURATION) $display("End of simulation");
    $finish;
end

endmodule
```

counter_tb.gtkw

```
[*] Code generated by Icestudio 0.3.3-rc
[*] Tue, 24 Jul 2018 16:05:02 GMT

main_tb.Enable
main_tb.Reset
main_tb.0
```

(continues on next page)

(continued from previous page)

```
main_tb.1  
main_tb.LEDs[3:0]
```