

---

# **icestudio Documentation**

***Release 0.2.2***

**Jesús Arroyo Torrens**

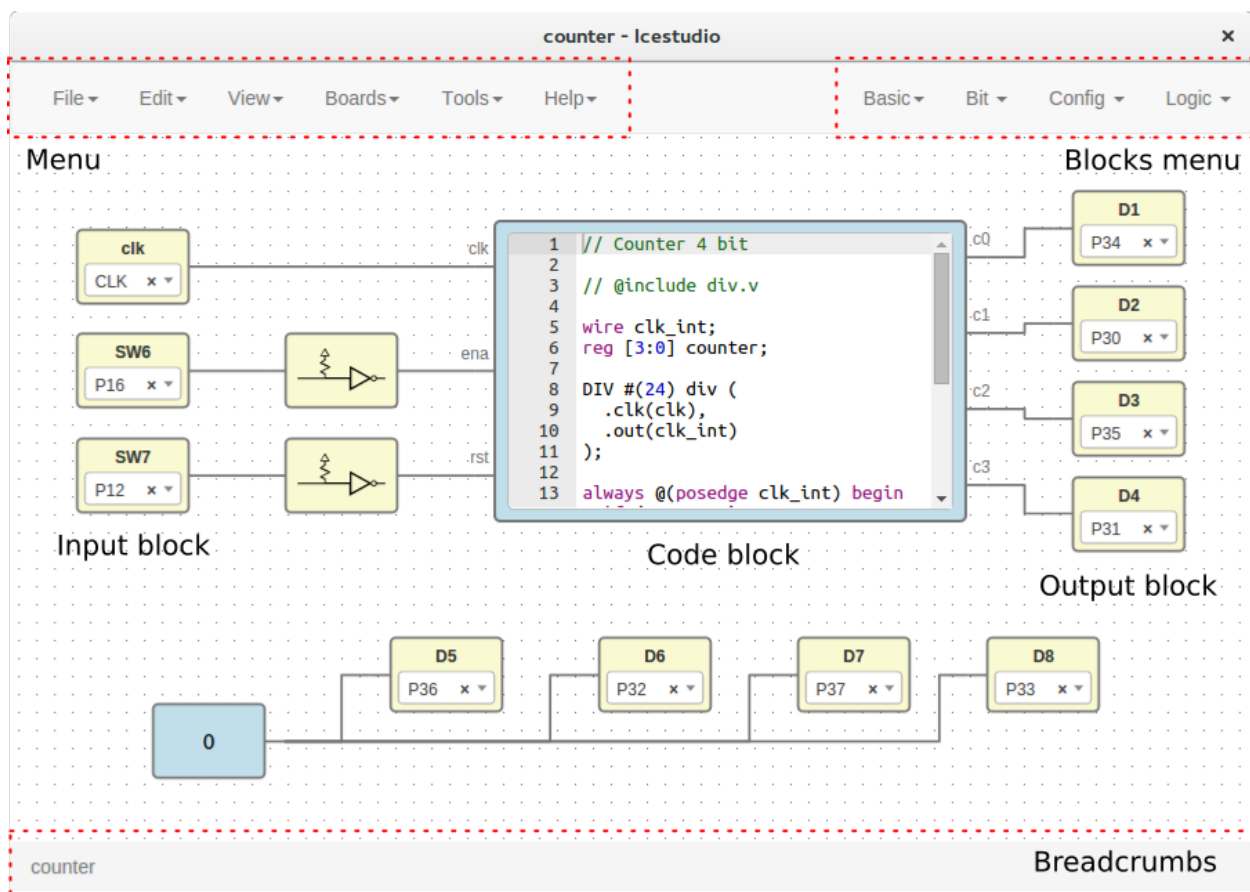
November 10, 2016



<b>1</b>	<b>Contents</b>	<b>1</b>
1.1	GUI . . . . .	1
1.2	How to... . . . .	6
1.3	Blocks . . . . .	29
1.4	Project . . . . .	55
1.5	Compiler . . . . .	79



## 1.1 GUI



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

### 1.1.1 Menu

#### File

- **New project:** create a new project.

- **Open project:** show a file dialog to open a project (.ice).
- **Examples:** contains all stored examples. A example is loaded as a project.
- **Templates:** contains all stored templates. A template is loaded as a project.
- **Save:** save the current project (.ice).
- **Save as:** show a save file dialog to save the current project (.ice).
- **Import block:** load a block file (.iceb) into the current project.
- **Export as block:** show a save file dialog to export the current project as a block file (.iceb).
- **Export verilog:** show a save file dialog to export the current verilog code file (.v).
- **Export PCF:** show a save file dialog to export the current pcf file (.pcf).
- **Export testbench:** show a save file dialog to export a auto-generated testbench (.v).
- **Export GTKWave:** show a save file dialog to export a GTKWave file with all signals showed (.gtkw).

---

**Note:** When a project is exported as a block, all FPGA I/O information is removed.

---

---

**Hint:** Examples and templates are stored in *app/resources/examples* and *app/resources/templates* respectively. To create a new examples/templates category just create a directory there. To create a new example/template copy and paste an .ice file.

---

## Edit

- **Clear all:** remove all blocks and wires from the graph.
- **Clone selected:** clone the selected block. It can also be done with *Ctrl + c* key.
- **Remove selected:** remove the selected block. It can also be done with *Ctrl + x* and *Supr* keys.
- **Image path:** set the project's relative image path. This image will be shown in the exported block. For example, a valid value can be: 'resources/images/and.svg'.
- **Remote hostname:** set the hostame of a remote device with a 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.
- **Language:** select the application language: English, Spanish, Galician, Basque. This selection is stored in the app profile.

## View

- **Reset view:** reset pan and zoom to its default values.

## Boards

It contains the supported boards: **Icezum**, **Go board**, **iCEstick**, **ice40-HX8K**, **icoBOARD 1.0**. When a board is selected all I/O block combos are updated and its current values removed.

---

**Hint:** This information is stored in the *app/resources/boards* directory. To support a new board just create a new JSON file with its information. Also, a *generator.py* script has been created to generate the board JSON file from a PCF file.

---

## Tools

- **Verify:** check the generated verilog source code.
- **Build:** generate the bitstream from the graphic source.
- **Upload:** generate and upload the bitstream to the FPGA board.
- **Install/Upgrade toolchain:** install a python virtualenv in *.icestudio/venv*, apio and icestorm toolchain. It requires Python 2.7 installed and Internet connection.
- **Remove toolchain:** remove the directories *.icestudio* and *.apio*.
- **Enable drivers:** launch the FTDI drivers configuration. Each OS has a different process.
- **Disable drivers:** revert the FTDI drivers configuration. Each OS has a different process.

---

**Hint:** Generated files are stored in the *\_build* directory.

---

## Help

- **View license:** open the Icestudio's license in a web browser.
- **Version:** show the current version.
- **Documentation:** open the Icestudio's documentation in a web browser.
- **Source code:** open the Icestudio's source code in a web browser.
- **Community forum:** open the FPGAwars forum in a web browser.
- **About Icestudio:** information about the application.

### 1.1.2 Blocks menu

#### Basic

It contains the basic blocks:

- **Code:** code block. Ports are asked in a prompt dialog.
- **Info:** info block. Text box for comments and notes.
- **Input:** input block. Block name is asked in a prompt dialog.
- **Output:** output block. Block name is asked in a prompt dialog.

---

**Note:** Multiple **input** and **output** blocks can be created using the *space* 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** and **output** blocks allows to modify the block name. In **code** block ports definition, *input* and *output* ports are separated by a space. Port names are separated by *,*.

For example:  $a, b$ : input  $a$  and  $b$ ,  $a, b \rightarrow c$ : input  $a$  and  $b$ , output  $c$ . Double click over **code** blocks allows to modify its *input* and *output* ports.

## Stored blocks

It contains all stored blocks sorted by categories. These menu is generated when the application starts.

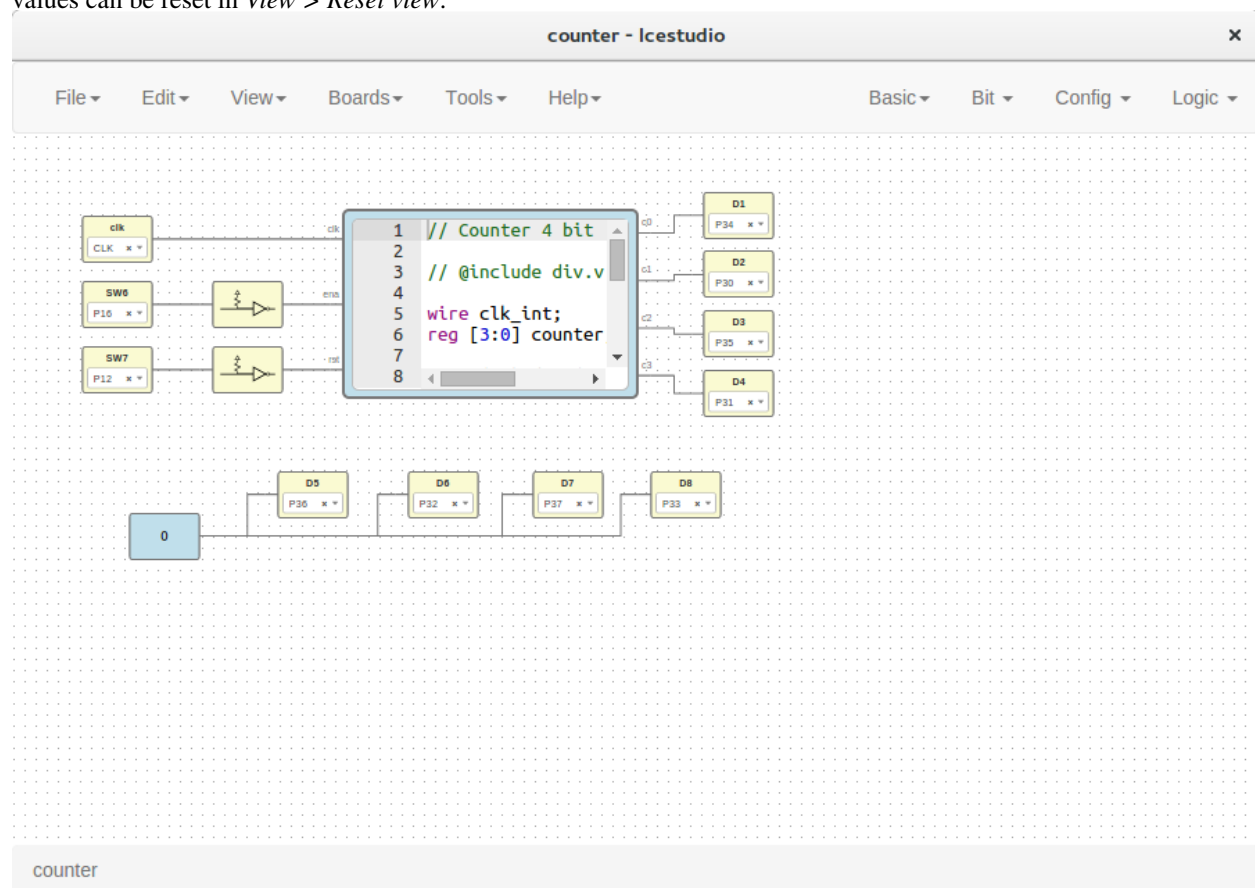
**Hint:** Blocks are stored in *app/resources/blocks*. To create a new block category just create a directory there. To create a new block copy and paste an **.iceb** file.

### 1.1.3 Graph

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

## Pan & Zoom

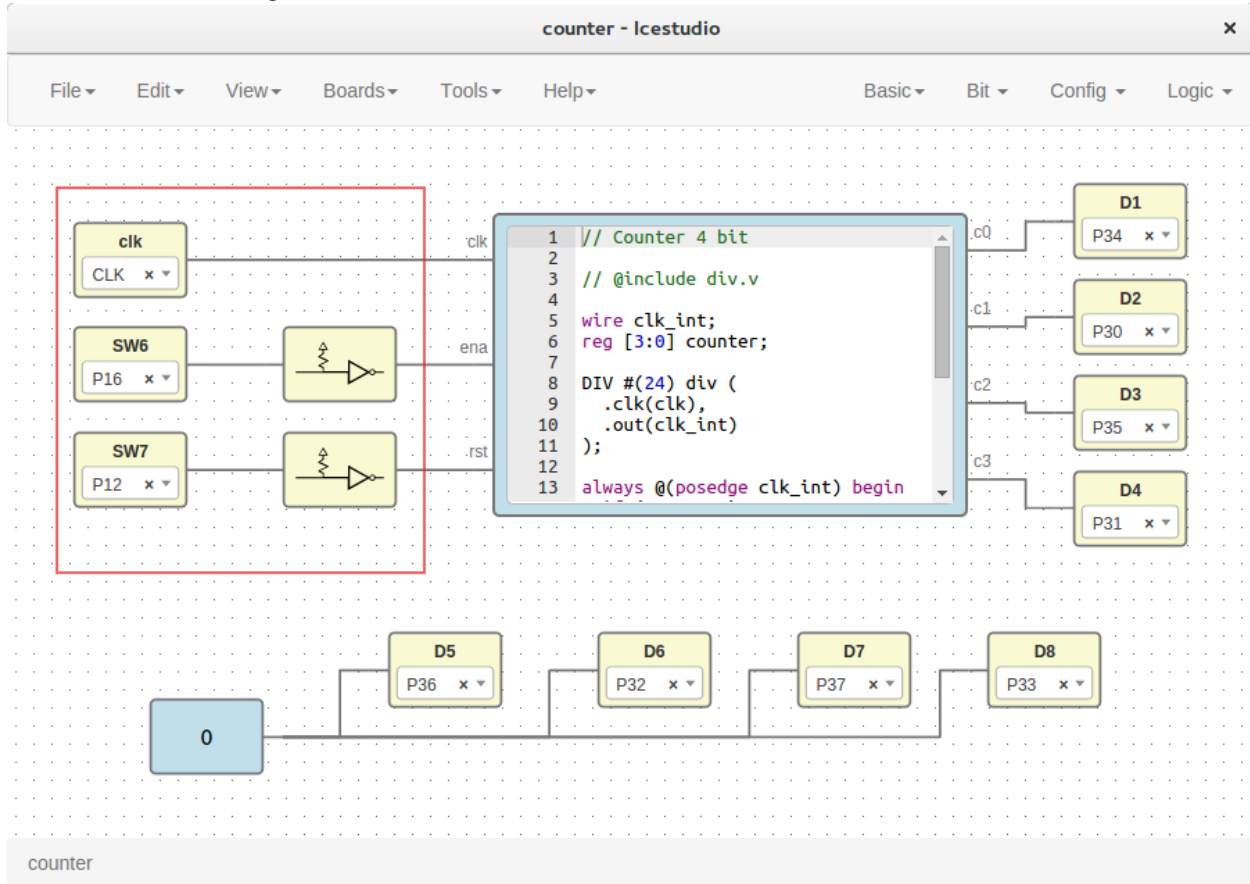
Pan is performed using the **mouse left button** over the background. Zoom is performed using **mouse wheel**. Both values can be reset in *View > Reset view*.





## Select

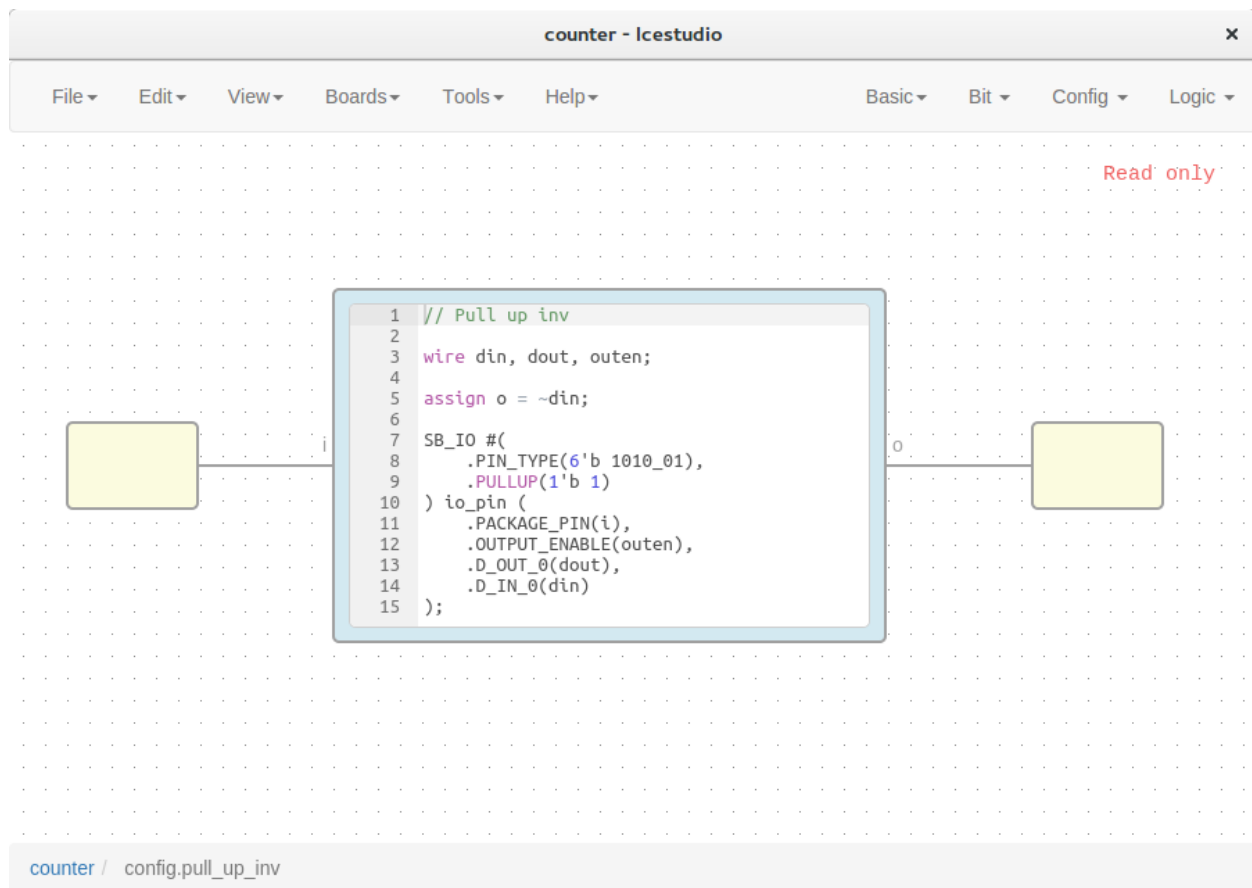
Block selection is performed using the **mouse right button**. Blocks can be selected/unselected individually using right-click/Ctrl+right-click, respectively. In addition, several blocks can be selected by a selection box. Selection is cancelled when the background is clicked.



## Blocks examination

Non-basic blocks can be read only examined by **double clicking** the block using the **mouse left button**. This is a recursive action.

During the examination, pan, zoom and code navigation are enabled.



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

## 1.2 How to...

### 1.2.1 Install the toolchain

#### 1. Install Python 2.7

**Warning:** Windows users: DON'T FORGET to select Add python.exe to Path feature on the "Customize" stage.

#### 2. Connect to the internet

#### 3. Launch the toolchain installation process

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

**Note:** When the toolchain is installed, the menu option appears as **Tools > Upgrade toolchain**

## 1.2.2 Install the drivers

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

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

---

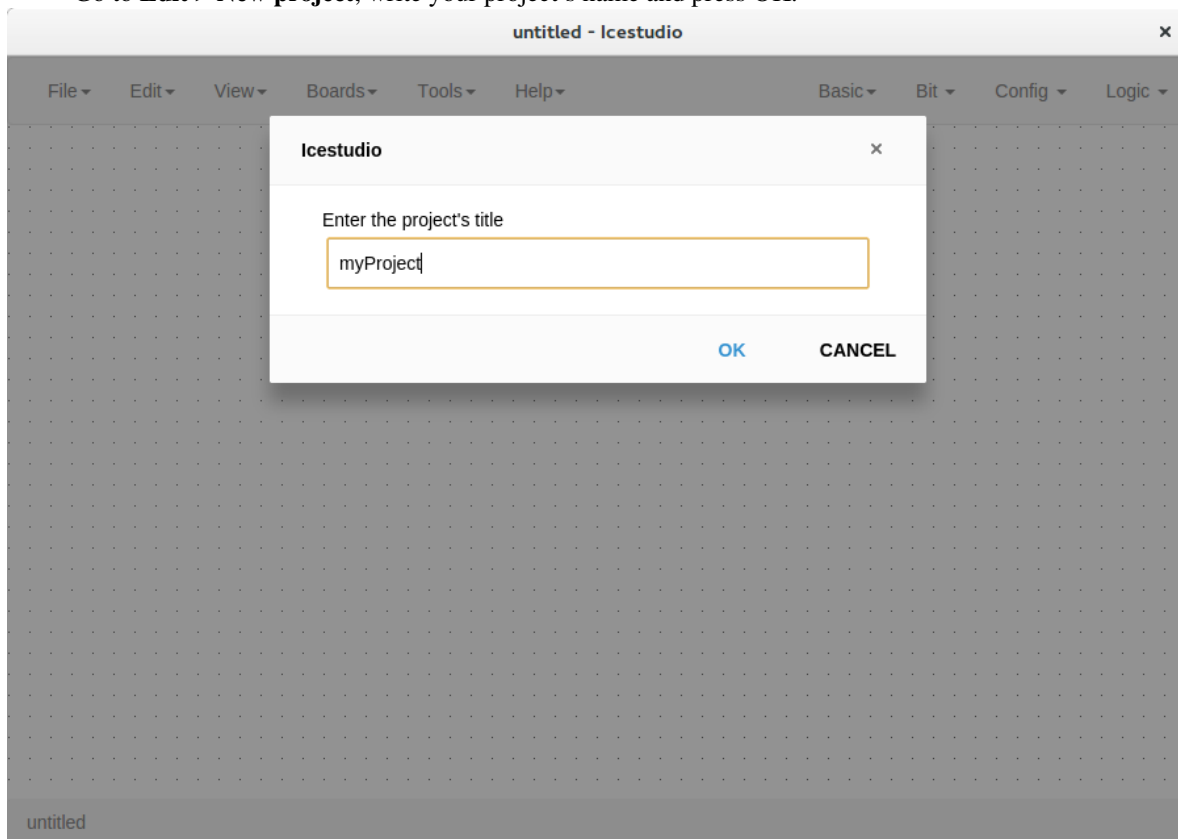
**Note:** To revert the drivers configuration go to **Tools > Disable drivers**

---

## 1.2.3 Create a project

1. **Create a new project**

Go to **Edit > New project**, write your project's name and press OK.

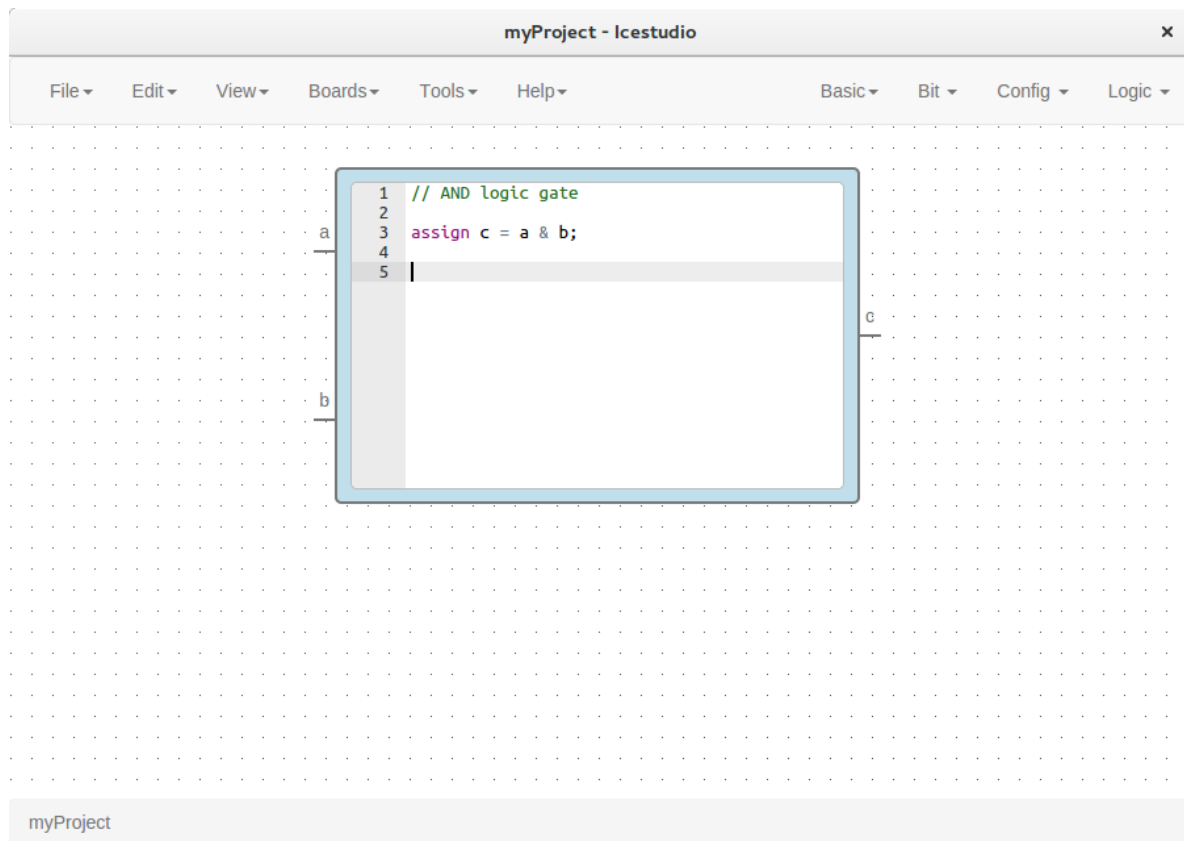


2. **Add your blocks**

1. *Code blocks*

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

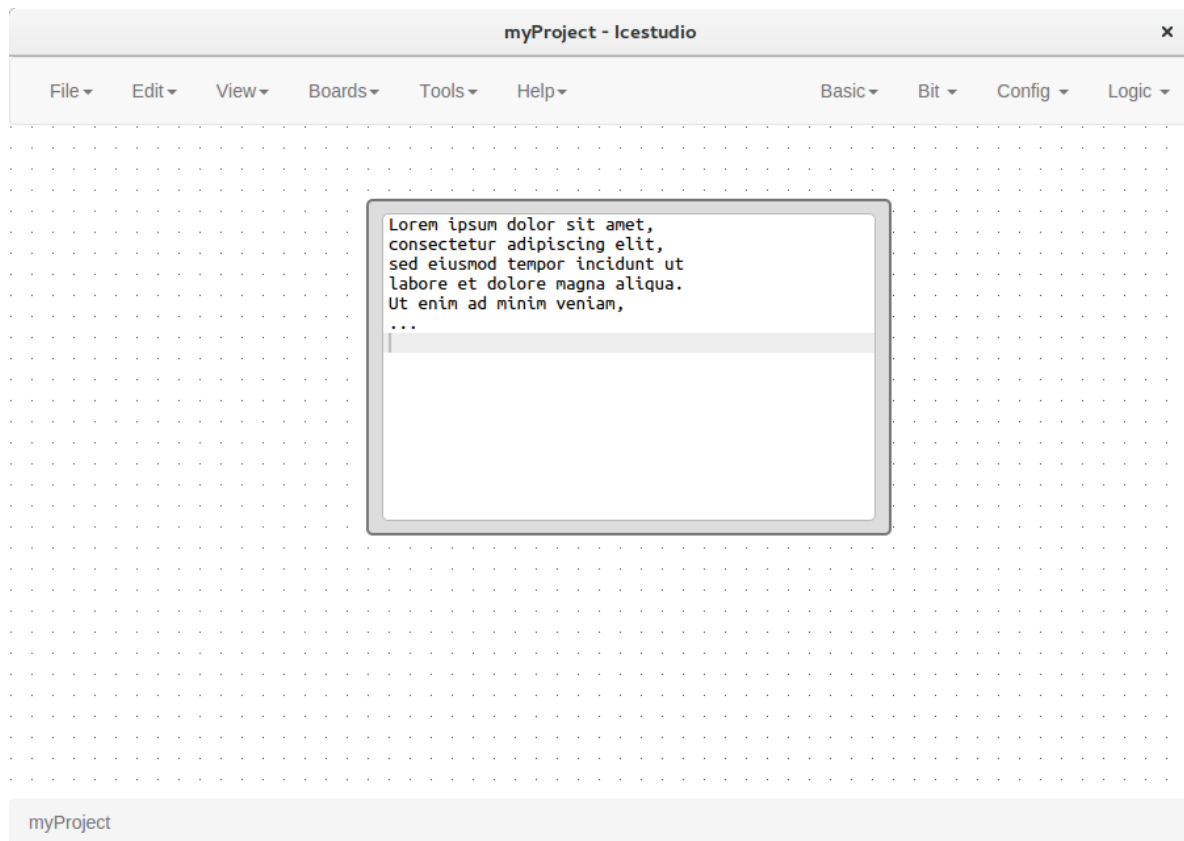
This block contains a text editor to write your module verilog code. Module header and footer are not required.



## 2. Info blocks

Click on **Basic > Info**.

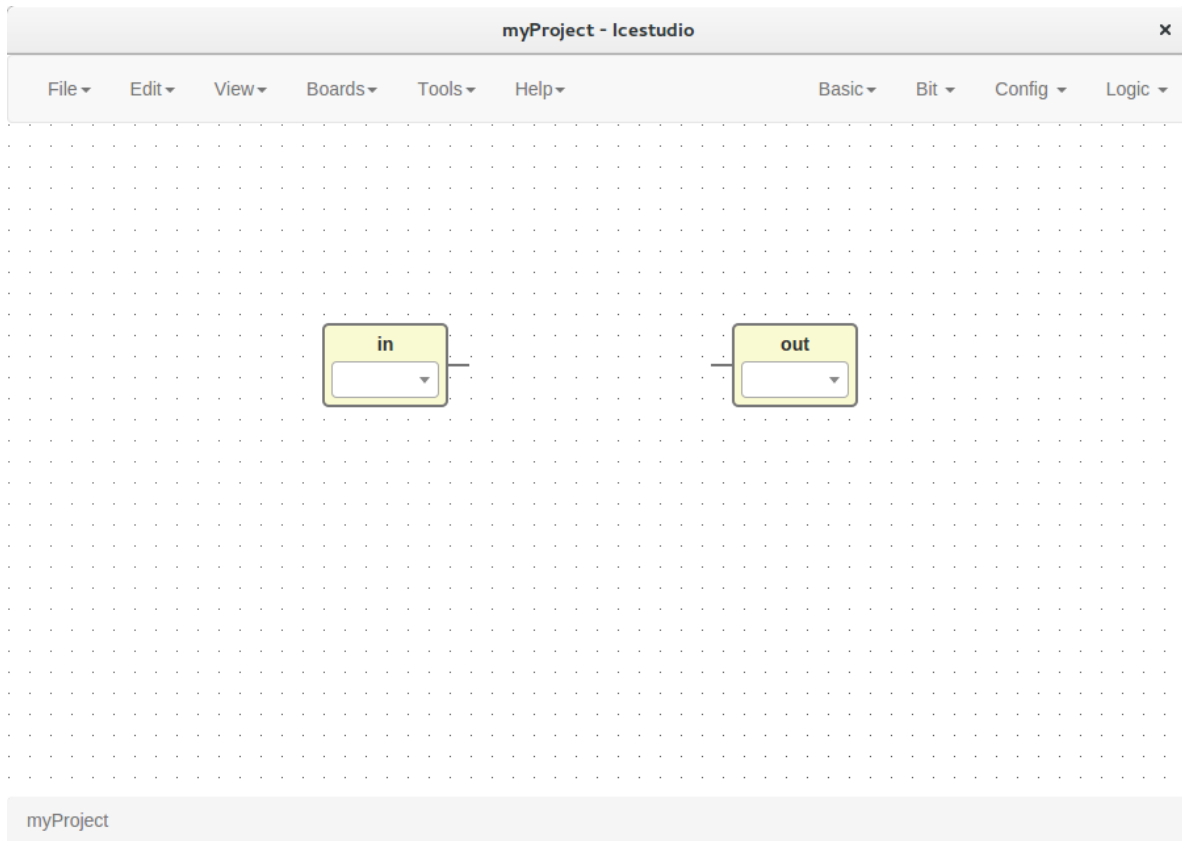
This block contains a text editor to add comments about the project.



### 3. *Input/Output blocks*

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

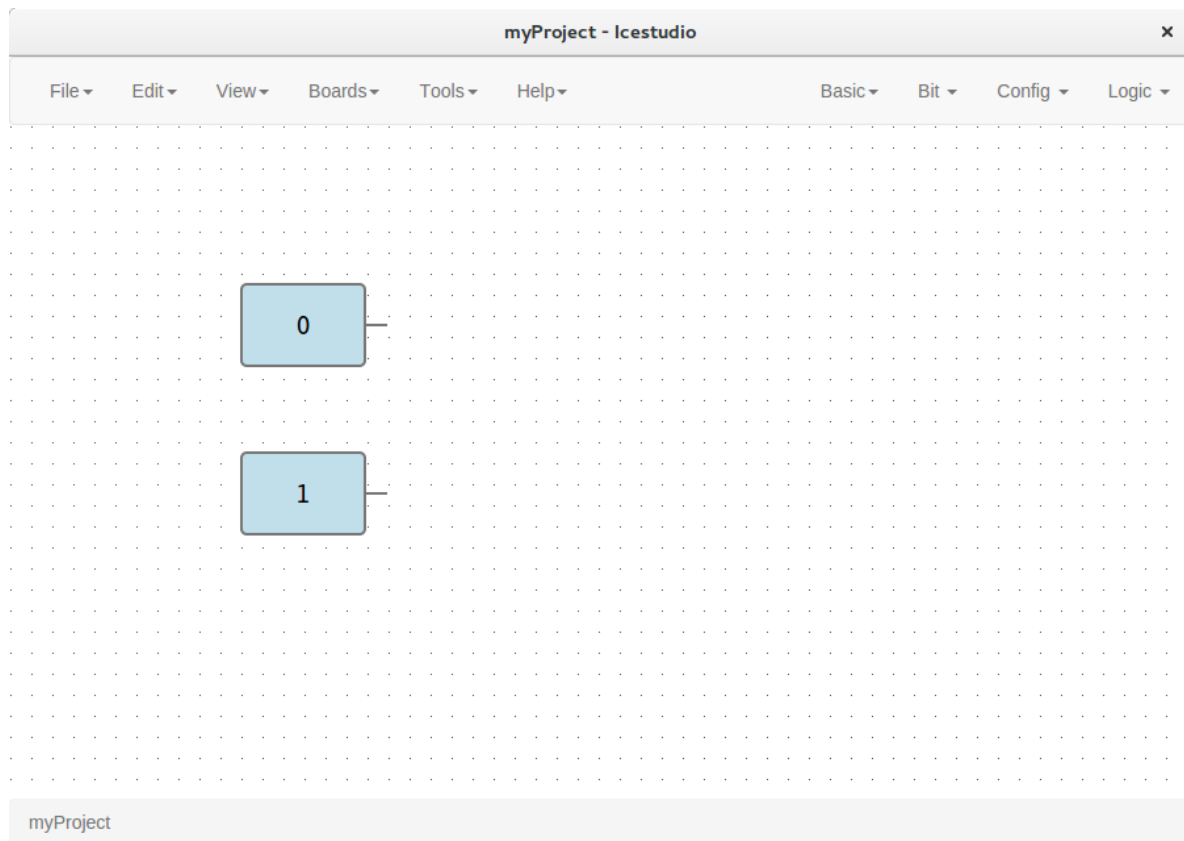
These blocks contain a FPGA pin selector depending on the selected board.



#### 4. *Bit blocks*

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

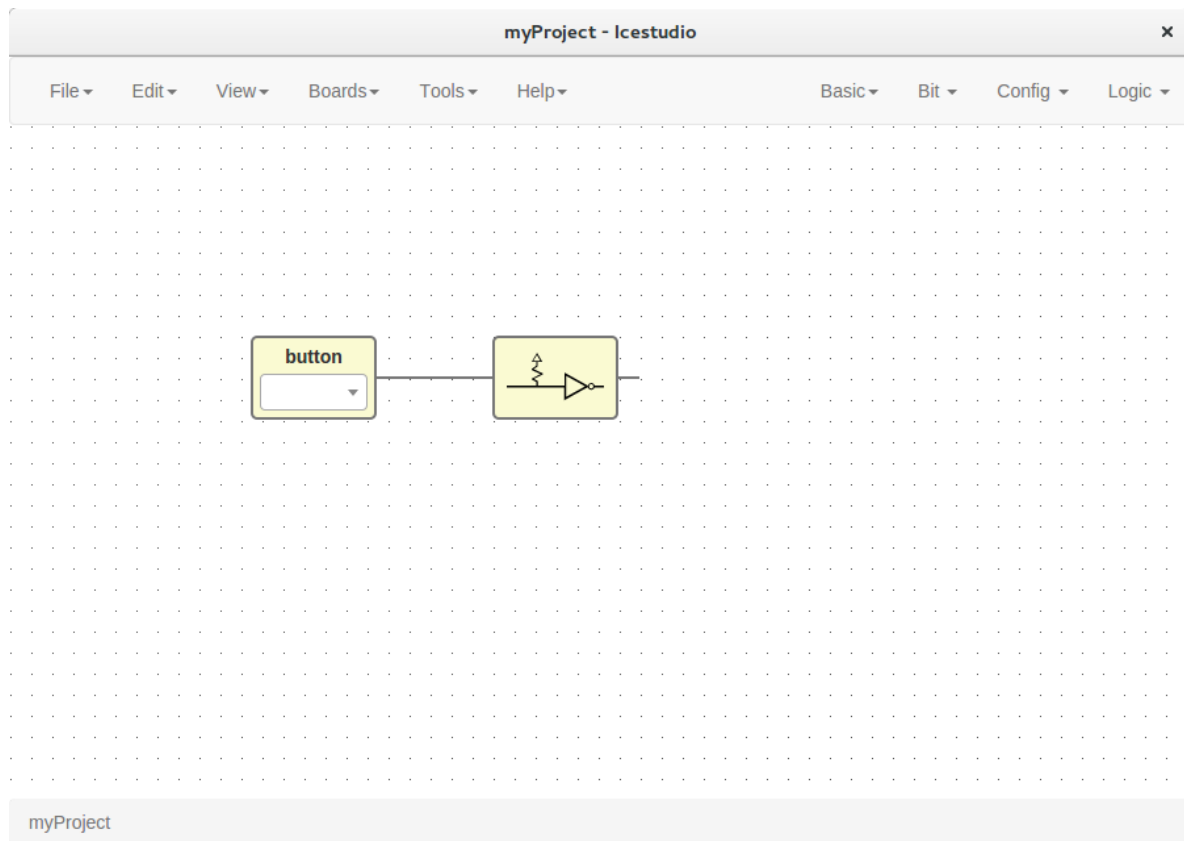
These blocks are low and high logic drivers.



### 5. Config block

Click on **Config > Pull up** or **Config > Pull up inv.**

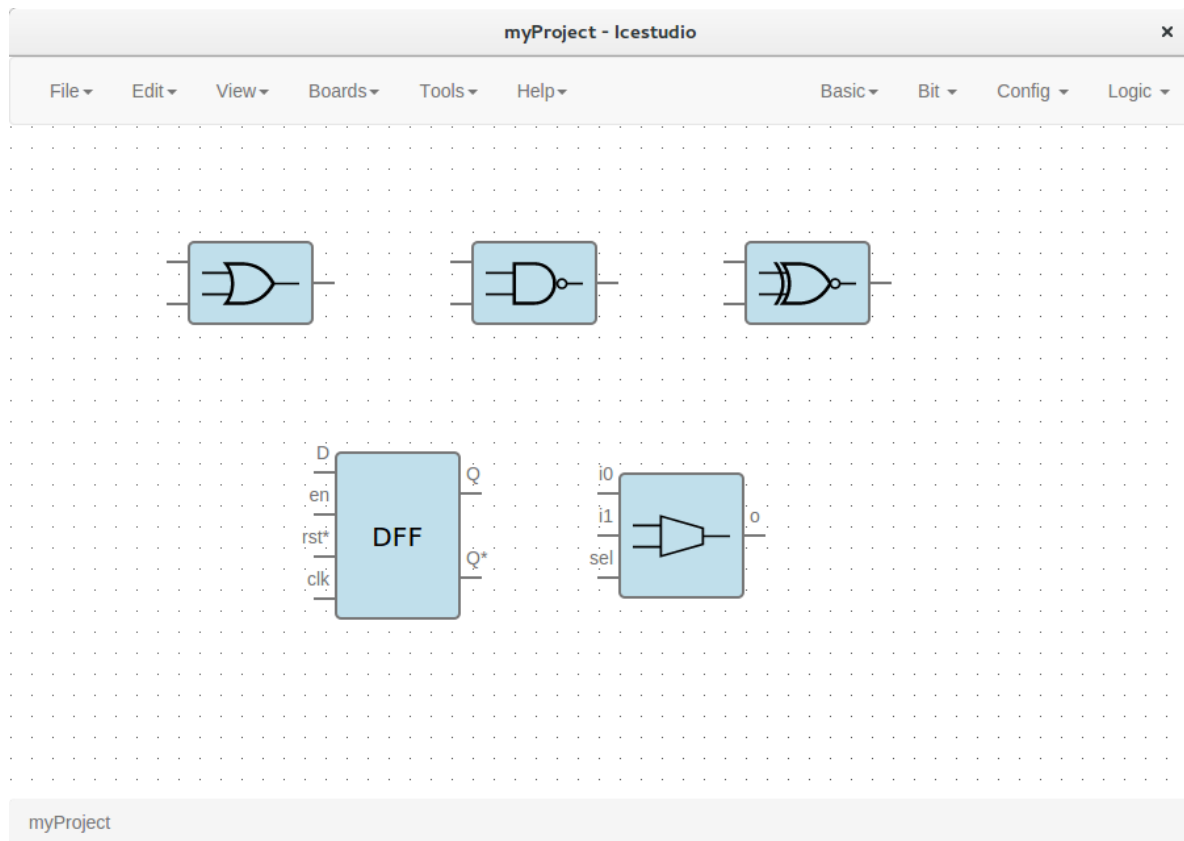
This block must be connected to input ports in order to configure a pull up.



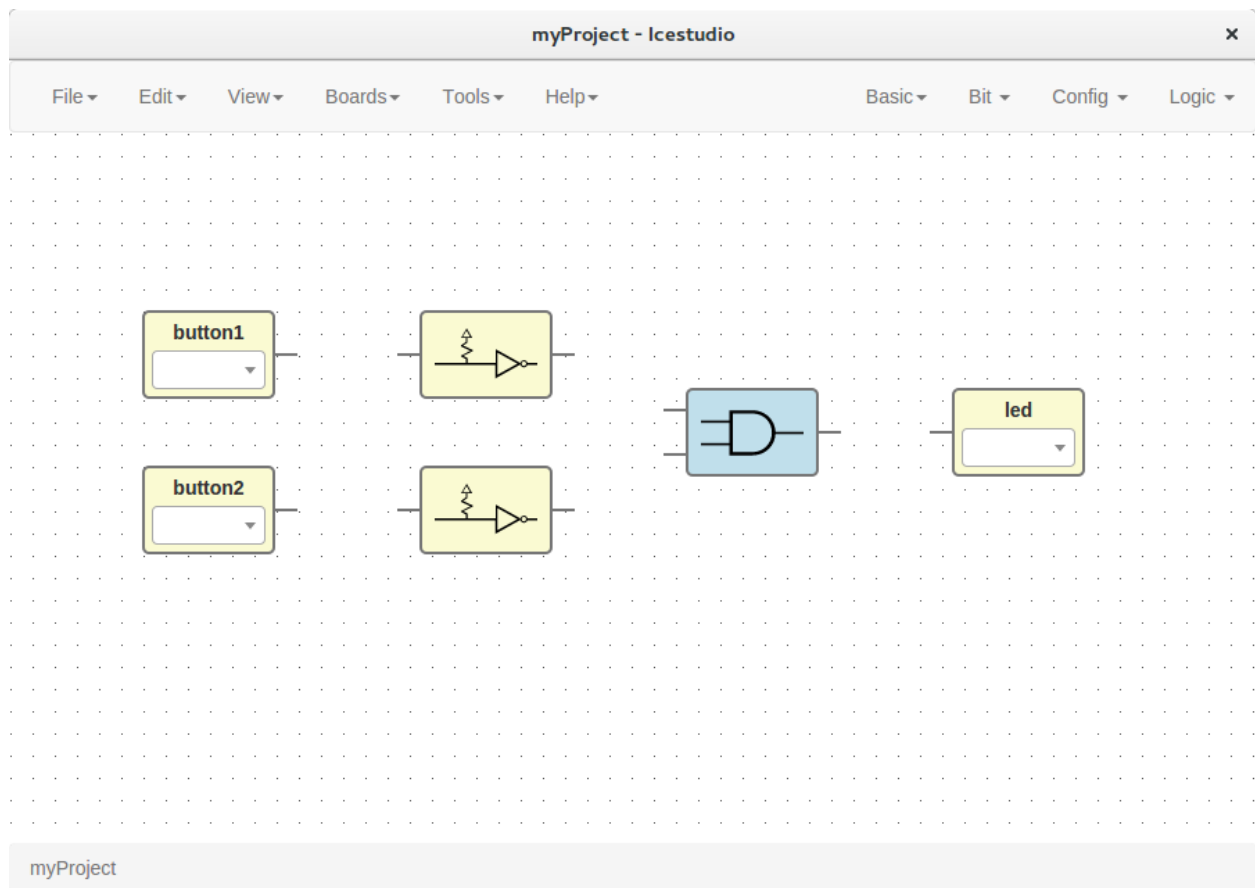
## 6. Logic blocks

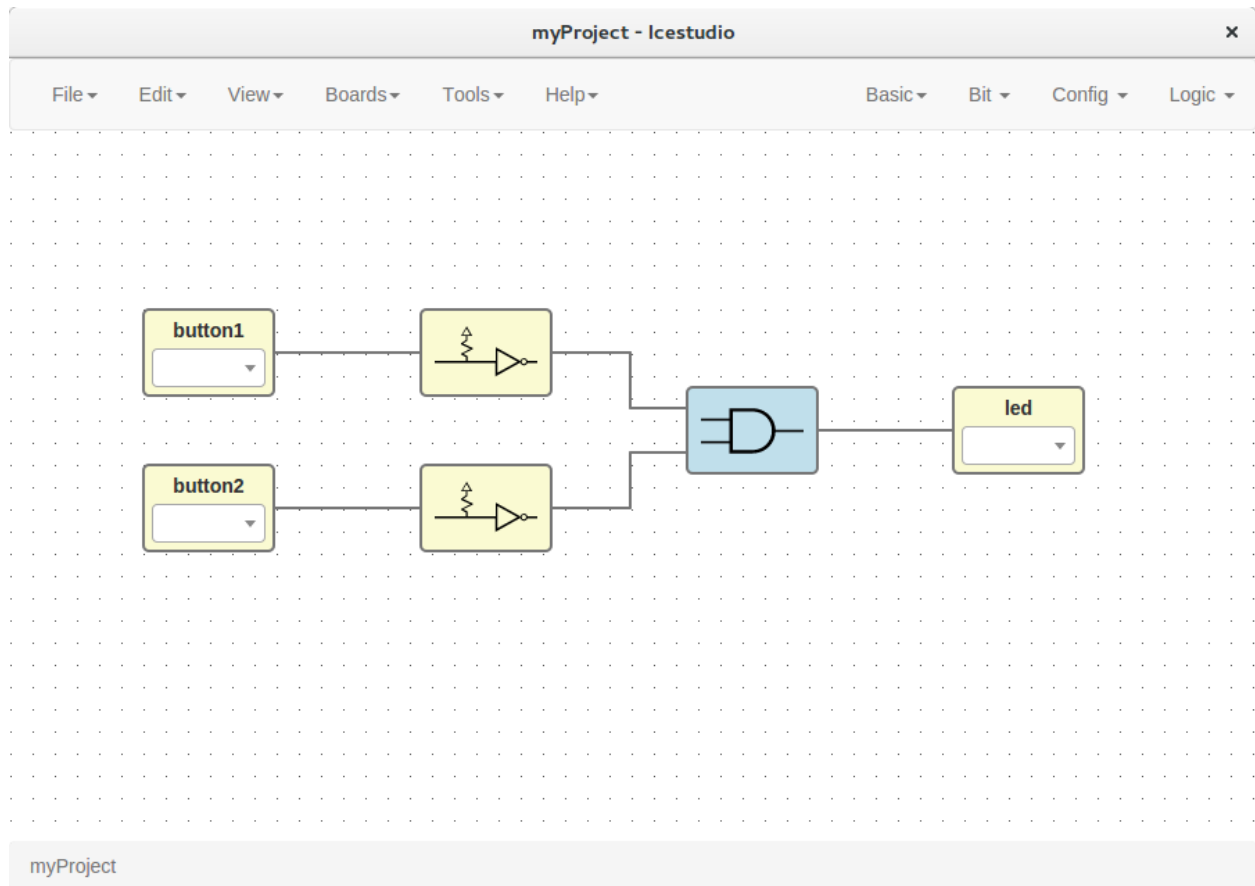
Go to the **Logic** menu and select. This menu contains **Logic Gates**, **Combinational blocks** and **Sequential flip-flops**.





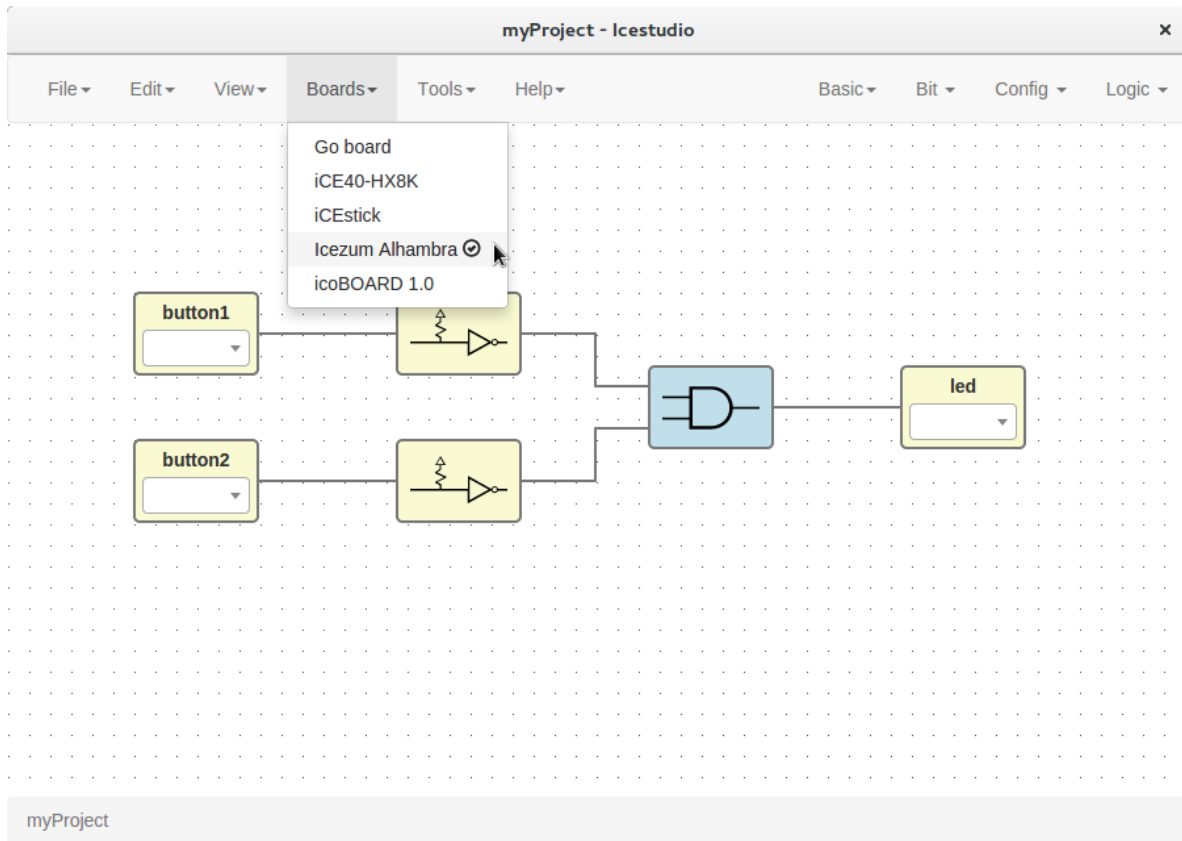
### 3. Connect your blocks





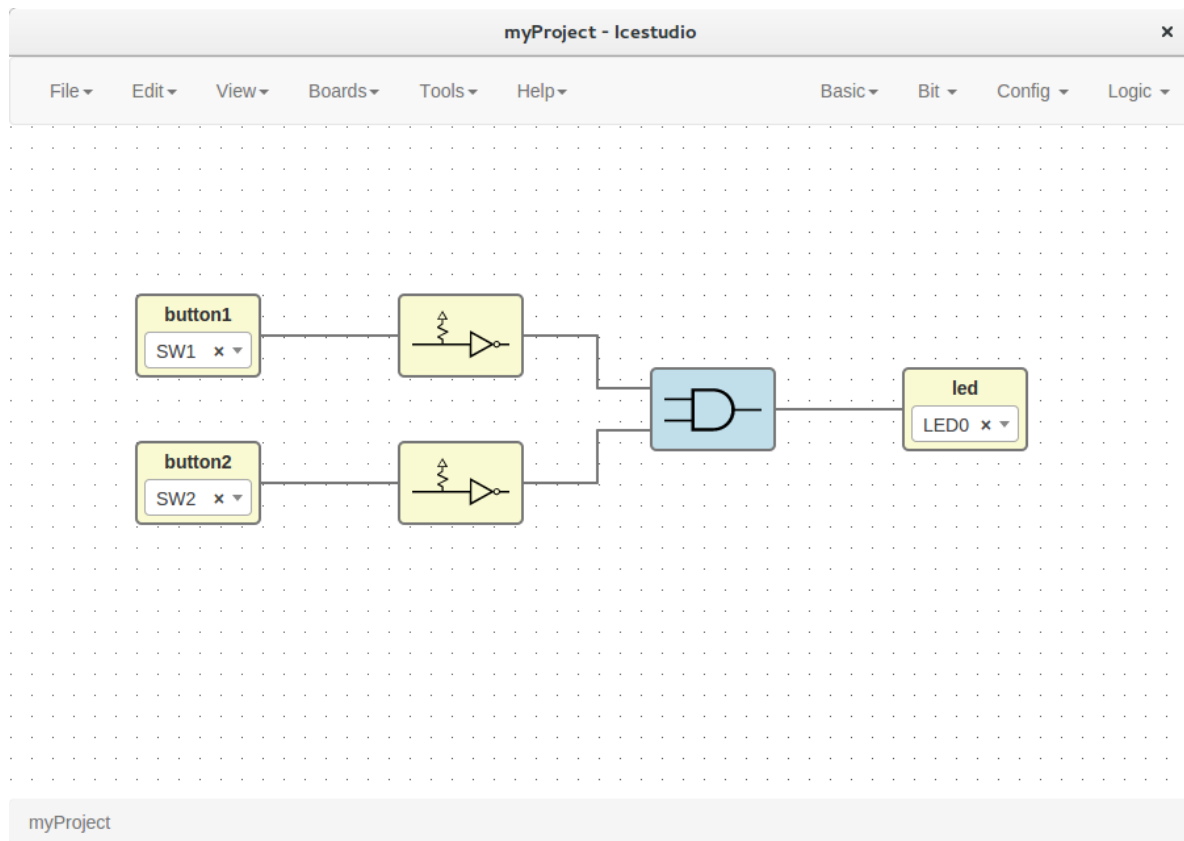
#### 4. Select your board

Go to **Boards** menu and select **Go board, iCE40-HX8K, iCEstick, Icezum Alhambra** or **icoBOARD 1.0**.



## 5. Set FPGA I/O pins

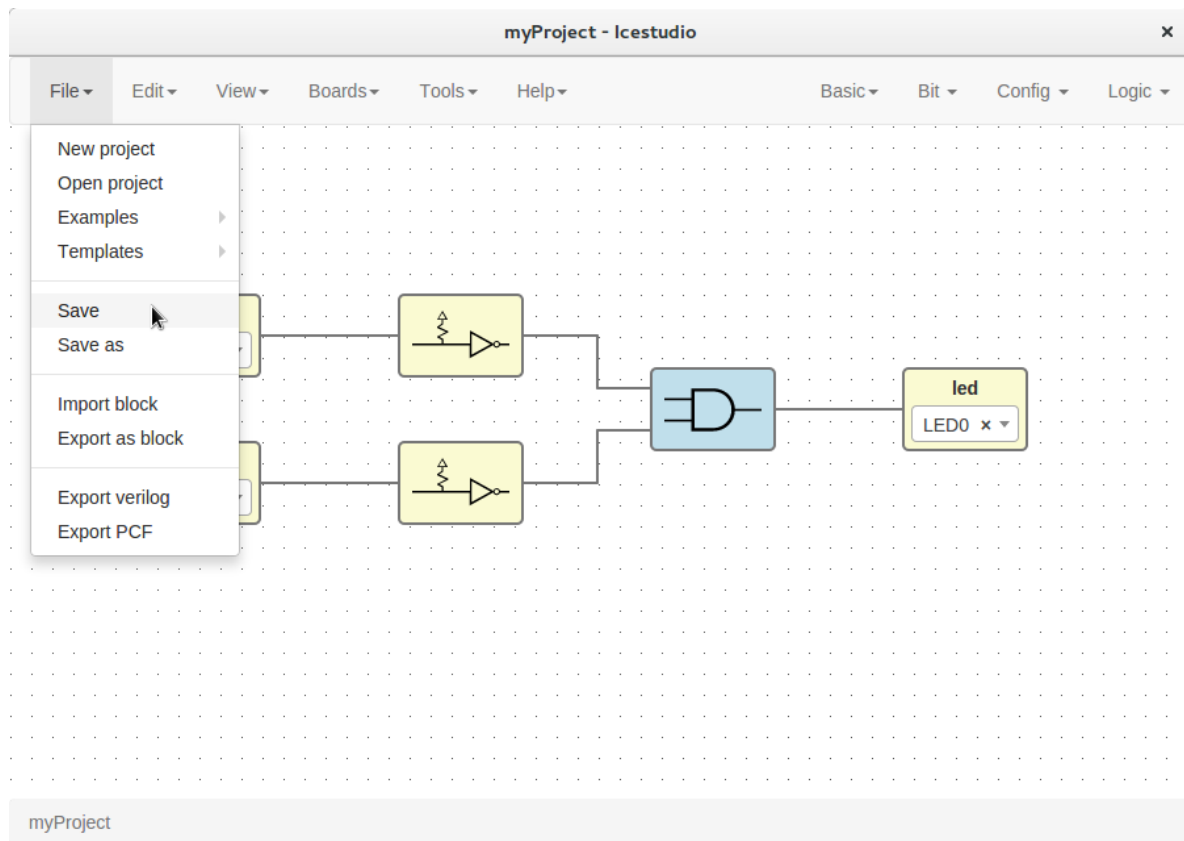
Select all Input/Output blocks' pins.



## 6. Save the project

Go to **Edit > Save**:

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



## 1.2.4 Upload a bitstream

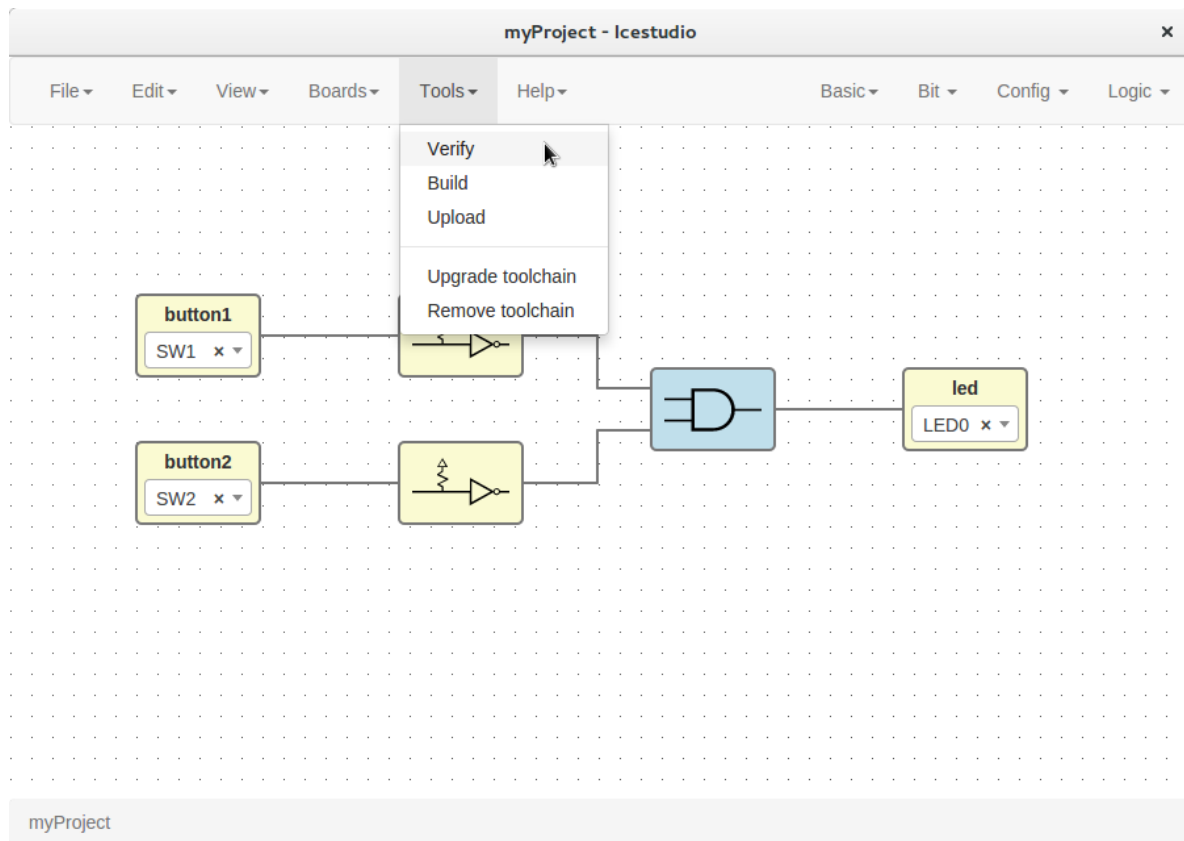
### 1. Open a project

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

### 2. Verify the project

Go to **Tools > Verify**.

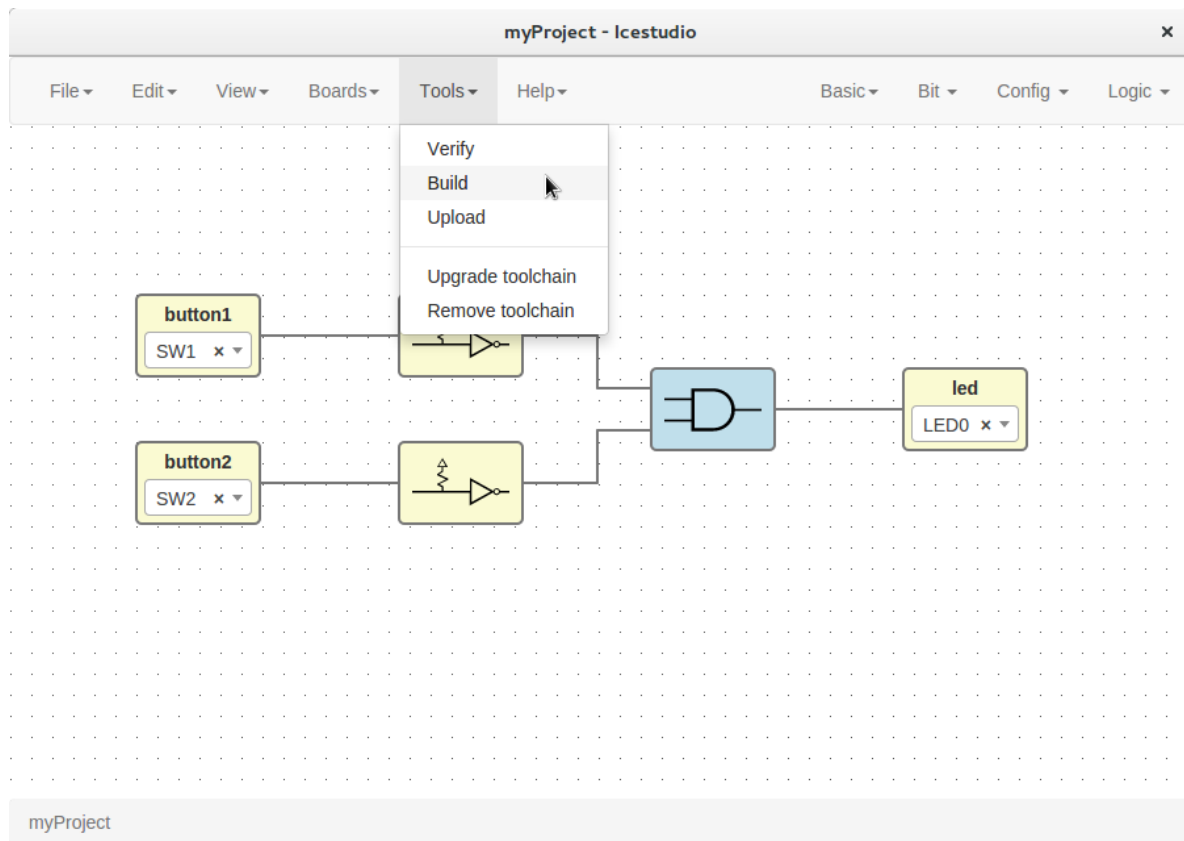
This option checks the generated verilog code using `apio verify`.



### 3. Build the project

Go to **Tools > Build**.

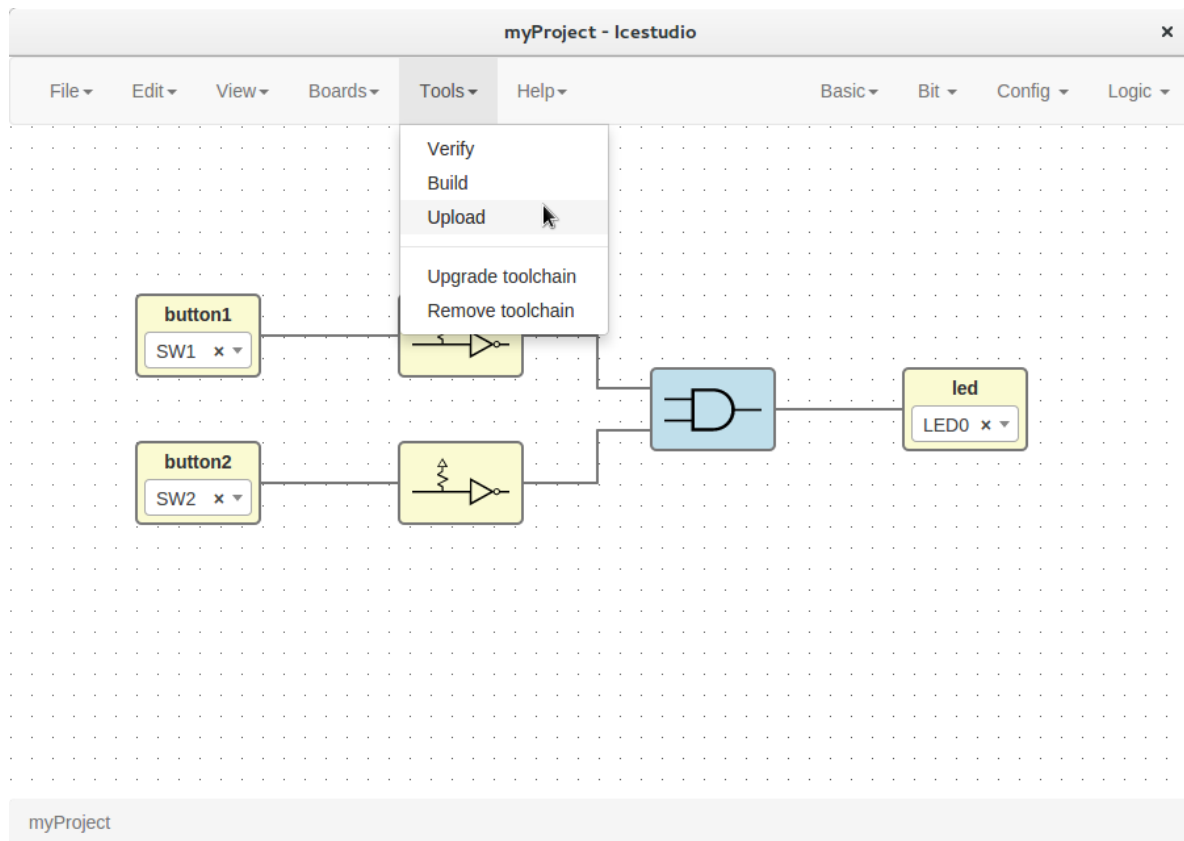
This option generates a bitstream using `apio build`.



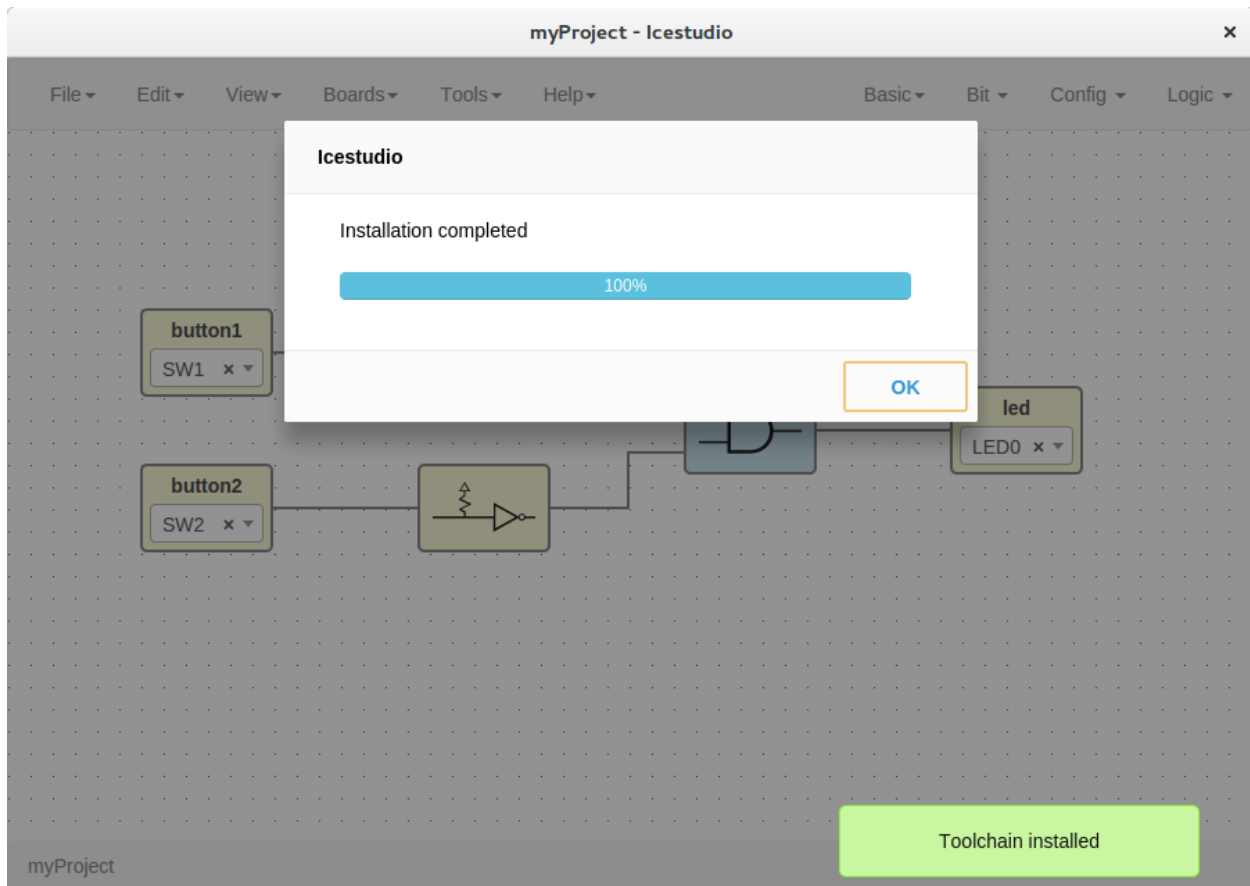
#### 4. Upload the project

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





**Note:** If the FPGA toolchain is not installed, it will be installed automatically when any tool is pressed. It can also be installed or removed in the menu **Tools** section.

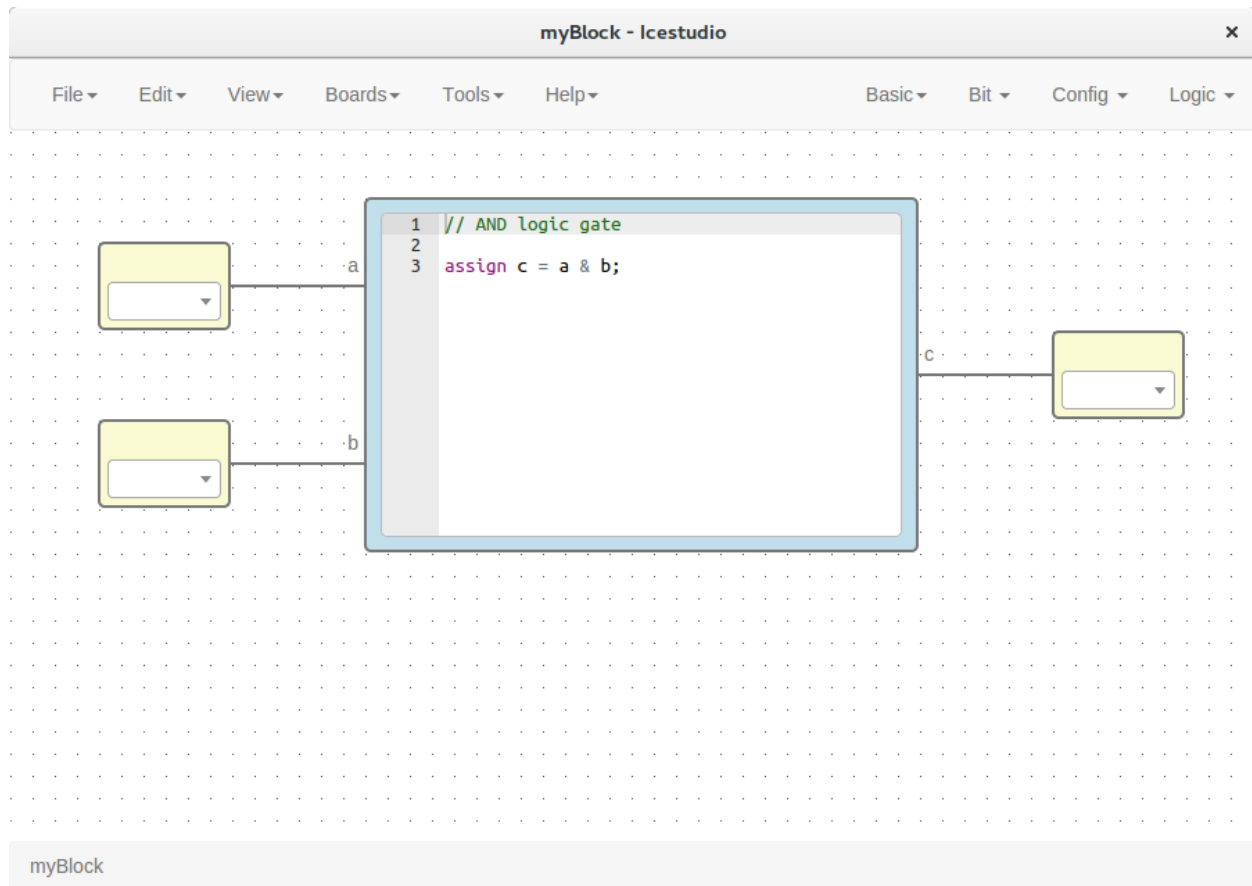


---

## 1.2.5 Create a block

### 1. Open a project

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



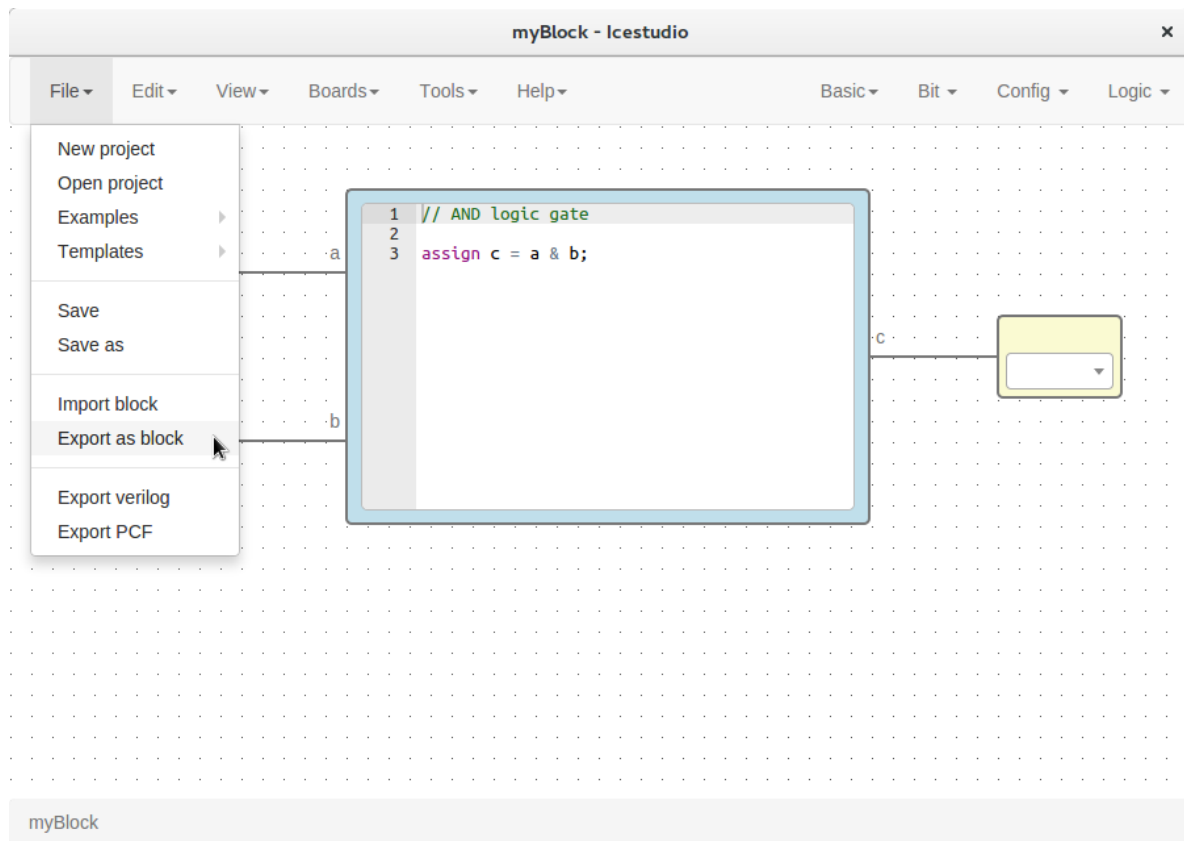
## 2. Verify the project

Go to **Tools > Verify**.

## 3. Export the project as a block

Go to **Edit > Export as block**.

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



---

**Note:** Input/Output blocks will become new Block I/O pins.

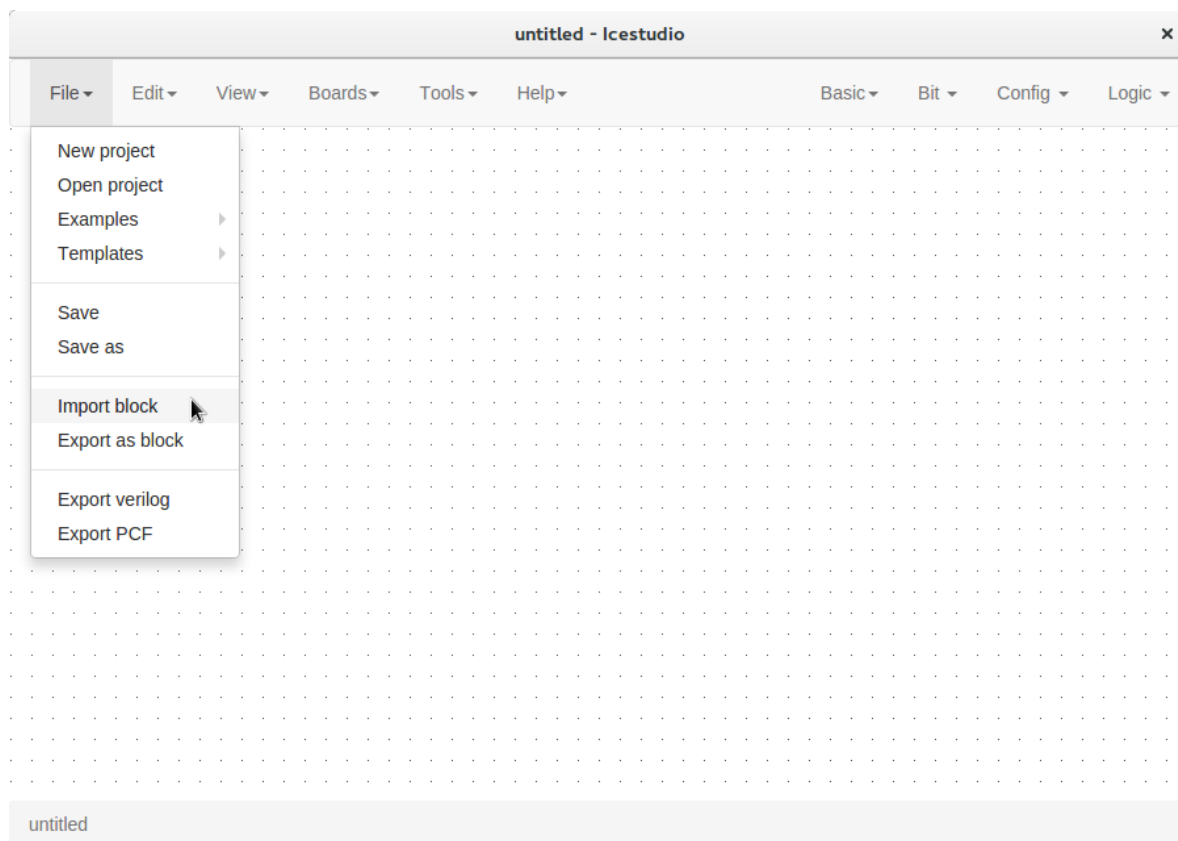
---

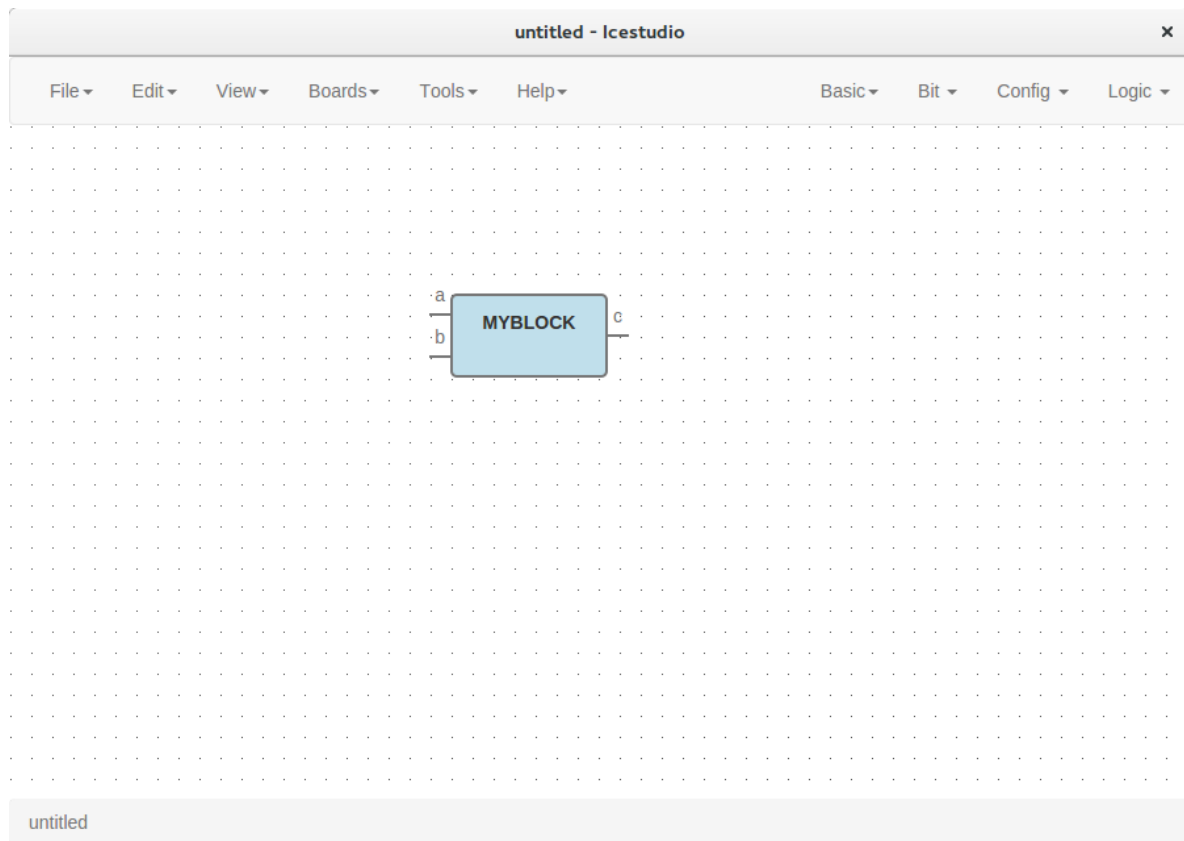
## 1.2.6 Use a custom block

### 1. Open or create a new project

### 2. Import the custom block

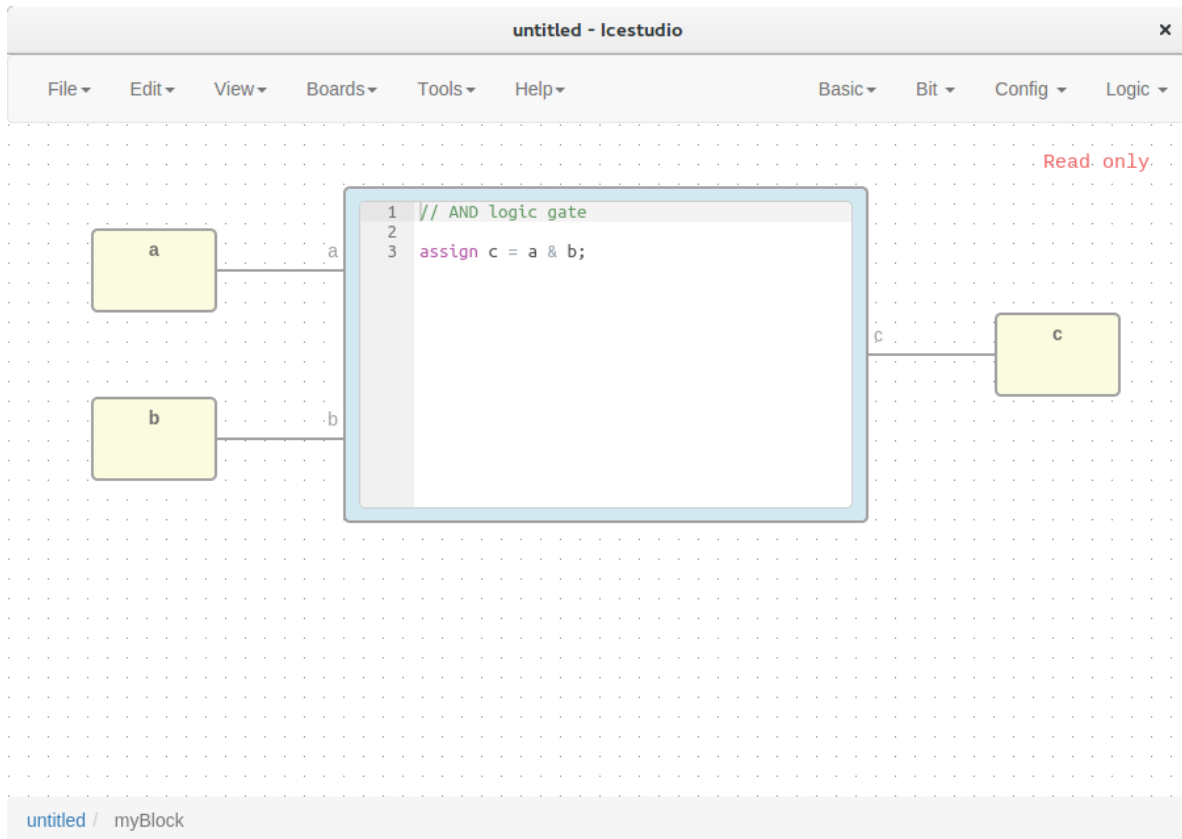
Go to **Edit > Import block** and select an **.iceb** file.





### 3. Examine the custom block

Complex blocks can be examined by double clicking the block.



### 1.2.7 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**

### 1.2.8 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

## 1.2.9 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:

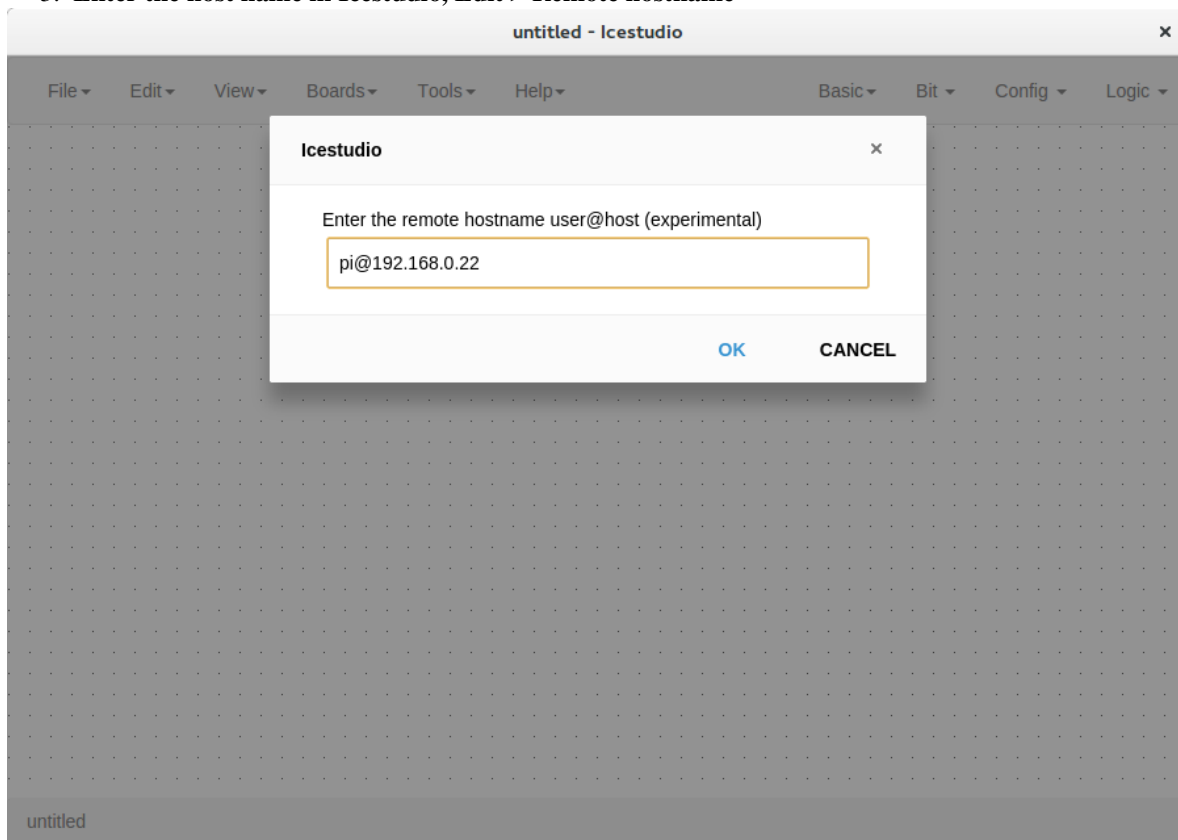
#### 1. Copy your SSH public key into the server

```
$ 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



## 1.3 Blocks

### 1.3.1 Definitions

#### Block

A block is an entity with *input* and *output* ports composed by blocks.

Its *input* and *output* ports are defined from its *input* and *output* block instances.

Extension: **iceb**

```
{
  "graph": {
    "blocks" : [],
    "wires": []
  },
  "deps" : {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
}
```

#### Block instances

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

#### Wire instances

```
{
  "source": {
    "block": "",
    "port": ""
  },
  "target": {
    "block": "",
    "port": ""
  },
  "vertices": [
    {
```

```
    "x": 0,  
    "y": 0  
  }  
]  
}
```

### 1.3.2 Basic blocks

#### Input instance

This special block is used to define input blocks in a project. It has one output port named ‘out’.

```
{  
  "id": "",  
  "type": "basic.input",  
  "data": {  
    "label": "a"  
  },  
  "position": {  
    "x": 0,  
    "y": 0  
  }  
}
```

#### Output instance

This special block is used to define output blocks in a projects. It has one input port named ‘in’.

```
{  
  "id": "",  
  "type": "basic.output",  
  "data": {  
    "label": "o"  
  },  
  "position": {  
    "x": 0,  
    "y": 0  
  }  
}
```

#### Code instance

This special block is used to define verilog code in a block. It has input and output ports defined in *value.ports* field.

```
{  
  "id": "",  
  "type": "basic.code",  
  "data": {  
    "code": "// And gate\n\nassign o = a & b;\n",  
    "ports": {  
      "in": [  
        "a",  
        "b"  
      ],  
    },  
  },  
}
```

```

    "out": [
      "o"
    ]
  },
  "position": {
    "x": 0,
    "y": 0
  }
}

```

### 1.3.3 Simple blocks

Simple blocks contain **only** basic blocks. It has no dependencies.

#### Low block



File: low.iceb

Show/Hide code

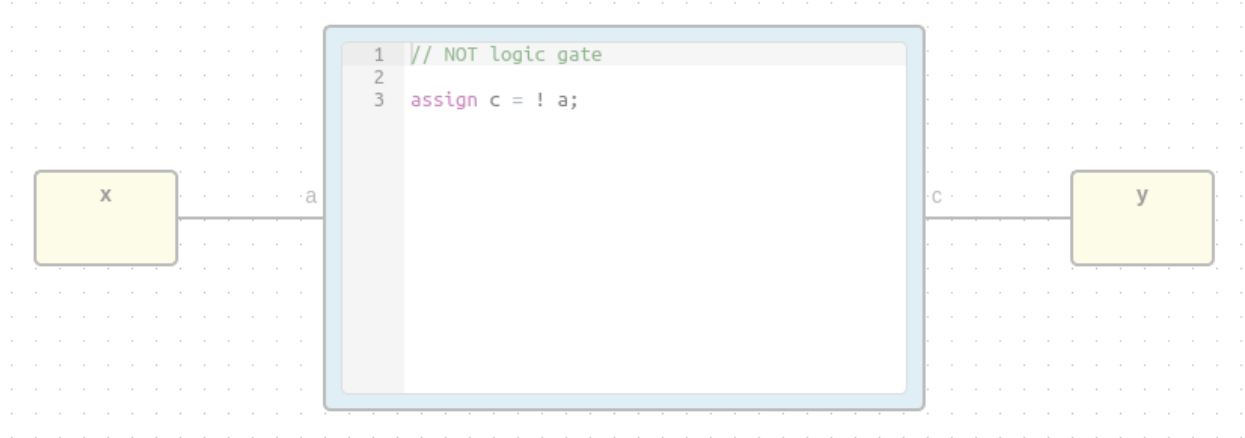
```

{
  "graph": {
    "blocks": [
      {
        "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
        "type": "basic.code",
        "data": {
          "code": "// Bit 0\n\nassign v = 1'b0;",
          "ports": {
            "in": [],

```

```
        "out": [
            "v"
        ]
    },
    "position": {
        "x": 96,
        "y": 96
    }
},
{
    "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
    "type": "basic.output",
    "data": {
        "label": "o"
    },
    "position": {
        "x": 608,
        "y": 192
    }
}
],
"wires": [
    {
        "source": {
            "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
            "port": "v"
        },
        "target": {
            "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
            "port": "in"
        }
    }
]
},
"deps": {},
"image": "",
"state": {
    "pan": {
        "x": 0,
        "y": 0
    },
    "zoom": 1
}
}
```

## Not block



File: not.iceb

Show/Hide code

```

{
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x"
        },
        "position": {
          "x": 64,
          "y": 144
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
          "label": "y"
        },
        "position": {
          "x": 752,
          "y": 144
        }
      },
      {
        "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "type": "basic.code",
        "data": {
          "code": "// NOT logic gate\nassign c = ! a;",
          "ports": {
            "in": [

```

```
        "a"
      ],
      "out": [
        "c"
      ]
    }
  },
  "position": {
    "x": 256,
    "y": 48
  }
},
],
"wires": [
  {
    "source": {
      "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
      "port": "out"
    },
    "target": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "a"
    }
  },
  {
    "source": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "c"
    },
    "target": {
      "block": "664caf9e-5f40-4df4-800a-b626af702e62",
      "port": "in"
    }
  }
],
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
```

## Or block



File: or.iceb

Show/Hide code

```

{
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x"
        },
        "position": {
          "x": 64,
          "y": 80
        }
      },
      {
        "id": "97b51945-d716-4b6c-9db9-970d08541249",
        "type": "basic.input",
        "data": {
          "label": "y"
        },
        "position": {
          "x": 64,
          "y": 208
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
          "label": "o"
        },
        "position": {

```

```
        "x": 752,
        "y": 144
    },
    {
        "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "type": "basic.code",
        "data": {
            "code": "// OR logic gate\n\nassign c = a | b;",
            "ports": {
                "in": [
                    "a",
                    "b"
                ],
                "out": [
                    "c"
                ]
            }
        },
        "position": {
            "x": 256,
            "y": 48
        }
    },
    ],
    "wires": [
        {
            "source": {
                "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
                "port": "out"
            },
            "target": {
                "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
                "port": "a"
            }
        },
        {
            "source": {
                "block": "97b51945-d716-4b6c-9db9-970d08541249",
                "port": "out"
            },
            "target": {
                "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
                "port": "b"
            }
        },
        {
            "source": {
                "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
                "port": "c"
            },
            "target": {
                "block": "664caf9e-5f40-4df4-800a-b626af702e62",
                "port": "in"
            }
        }
    ]
},
```



```

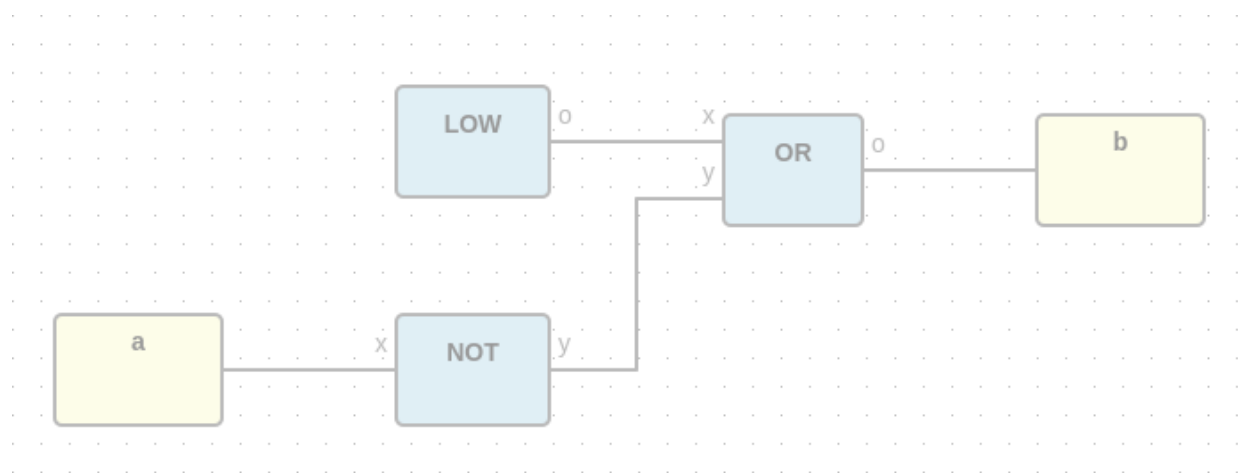
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
}

```

### 1.3.4 Complex blocks

Complex blocks contain **not only** basic blocks.

#### Cnot block



File: **cnot.iceb**

Show/Hide code

```

{
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "graph": {

```

```
"blocks": [
  {
    "id": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
    "type": "not",
    "data": {},
    "position": {
      "x": 280,
      "y": 248
    }
  },
  {
    "id": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
    "type": "or",
    "data": {},
    "position": {
      "x": 464,
      "y": 136
    }
  },
  {
    "id": "55c6c056-3630-4482-ad47-f4d9ee83b835",
    "type": "basic.input",
    "data": {
      "label": "a"
    },
    "position": {
      "x": 88,
      "y": 248
    }
  },
  {
    "id": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "type": "basic.output",
    "data": {
      "label": "b"
    },
    "position": {
      "x": 640,
      "y": 136
    }
  },
  {
    "id": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
    "type": "low",
    "data": {},
    "position": {
      "x": 280,
      "y": 120
    }
  }
],
"wires": [
  {
    "source": {
      "block": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
      "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
    },
    "target": {
```

```

        "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
        "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    },
    {
        "source": {
            "block": "55c6c056-3630-4482-ad47-f4d9ee83b835",
            "port": "out"
        },
        "target": {
            "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
            "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
        }
    },
    {
        "source": {
            "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
            "port": "664caf9e-5f40-4df4-800a-b626af702e62"
        },
        "target": {
            "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
            "port": "97b51945-d716-4b6c-9db9-970d08541249"
        }
    },
    {
        "source": {
            "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
            "port": "664caf9e-5f40-4df4-800a-b626af702e62"
        },
        "target": {
            "block": "c8c6eed3-548c-49c7-a162-282179d427b1",
            "port": "in"
        }
    }
],
"deps": {
    "or": {
        "graph": {
            "blocks": [
                {
                    "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
                    "type": "basic.input",
                    "data": {
                        "label": "x"
                    },
                    "position": {
                        "x": 64,
                        "y": 80
                    }
                },
                {
                    "id": "97b51945-d716-4b6c-9db9-970d08541249",
                    "type": "basic.input",
                    "data": {
                        "label": "y"
                    },
                    "position": {

```

```
        "x": 64,
        "y": 208
    },
    {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
            "label": "o"
        },
        "position": {
            "x": 752,
            "y": 144
        }
    },
    {
        "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "type": "basic.code",
        "data": {
            "code": "// OR logic gate\n\nassign c = a | b;",
            "ports": {
                "in": [
                    "a",
                    "b"
                ],
                "out": [
                    "c"
                ]
            }
        },
        "position": {
            "x": 256,
            "y": 48
        }
    }
],
"wires": [
    {
        "source": {
            "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
            "port": "out"
        },
        "target": {
            "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
            "port": "a"
        }
    },
    {
        "source": {
            "block": "97b51945-d716-4b6c-9db9-970d08541249",
            "port": "out"
        },
        "target": {
            "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
            "port": "b"
        }
    }
],
{
```

```

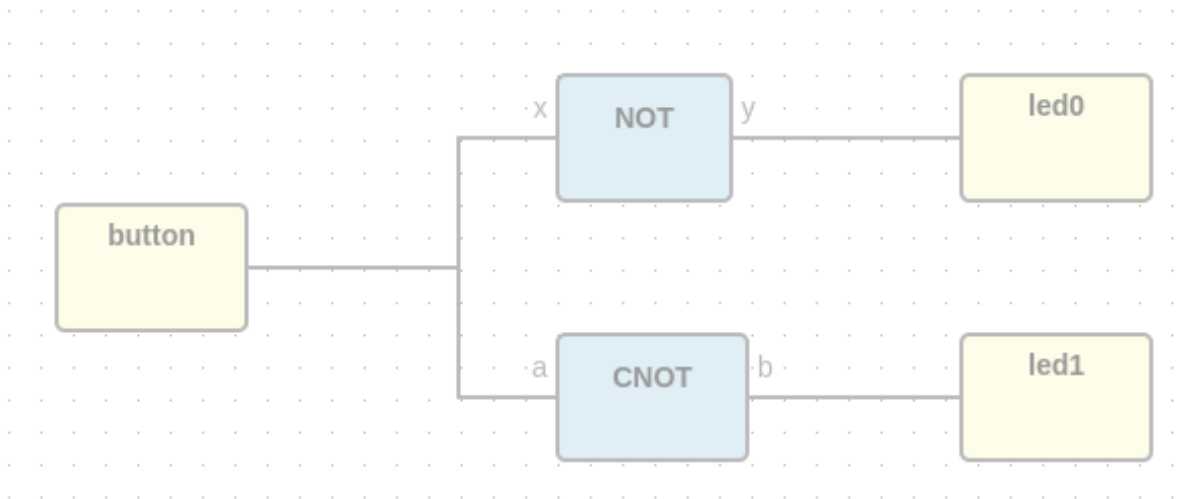
    "source": {
      "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
      "port": "c"
    },
    "target": {
      "block": "664caf9e-5f40-4df4-800a-b626af702e62",
      "port": "in"
    }
  }
]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
},
"not": {
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x"
        },
        "position": {
          "x": 64,
          "y": 144
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
          "label": "y"
        },
        "position": {
          "x": 752,
          "y": 144
        }
      },
      {
        "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "type": "basic.code",
        "data": {
          "code": "// NOT logic gate\n\nassign c = ! a;",
          "ports": {
            "in": [
              "a"
            ],
            "out": [
              "c"
            ]
          }
        }
      }
    ]
  }
}

```

```
    }
  },
  "position": {
    "x": 256,
    "y": 48
  }
},
"wires": [
  {
    "source": {
      "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
      "port": "out"
    },
    "target": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "a"
    }
  },
  {
    "source": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "c"
    },
    "target": {
      "block": "664caf9e-5f40-4df4-800a-b626af702e62",
      "port": "in"
    }
  }
]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
},
"low": {
  "graph": {
    "blocks": [
      {
        "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
        "type": "basic.code",
        "data": {
          "code": "// Bit 0\n\nassign v = 1'b0;",
          "ports": {
            "in": [],
            "out": [
              "v"
            ]
          }
        }
      },
      {
        "position": {
          "x": 96,
```

```
        "y": 96
      },
    ],
    {
      "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
      "type": "basic.output",
      "data": {
        "label": "o"
      },
      "position": {
        "x": 608,
        "y": 192
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
        "port": "v"
      },
      "target": {
        "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
}
```

## Dnot block



File: dnot.iceb

Show/Hide code

```
{
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "graph": {
    "blocks": [
      {
        "id": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
        "type": "basic.input",
        "data": {
          "label": "button"
        },
        "position": {
          "x": 104,
          "y": 176
        }
      },
      {
        "id": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
        "type": "cnot",
        "data": {},
        "position": {
          "x": 352,
          "y": 240
        }
      }
    ]
  }
}
```



```

    },
    {
      "id": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
      "type": "not",
      "data": {},
      "position": {
        "x": 352,
        "y": 112
      }
    },
    {
      "id": "4c4d2ddd-a97d-4fcb-9c68-ba1149f25082",
      "type": "basic.output",
      "data": {
        "label": "led0"
      },
      "position": {
        "x": 552,
        "y": 112
      }
    },
    {
      "id": "0e777320-de37-4dca-a077-51fbf10a6565",
      "type": "basic.output",
      "data": {
        "label": "led1"
      },
      "position": {
        "x": 552,
        "y": 240
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
        "port": "out"
      },
      "target": {
        "block": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
        "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
      }
    },
    {
      "source": {
        "block": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
        "port": "out"
      },
      "target": {
        "block": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
        "port": "55c6c056-3630-4482-ad47-f4d9ee83b835"
      }
    },
    {
      "source": {
        "block": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",

```

```
    "port": "664caf9e-5f40-4df4-800a-b626af702e62"
  },
  "target": {
    "block": "4c4d2ddd-a97d-4fcb-9c68-ba1149f25082",
    "port": "in"
  }
},
{
  "source": {
    "block": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
    "port": "c8c6eed3-548c-49c7-a162-282179d427b1"
  },
  "target": {
    "block": "0e777320-de37-4dca-a077-51fbf10a6565",
    "port": "in"
  }
}
]
},
"deps": {
  "logic.not": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": ""
          },
          "position": {
            "x": 64,
            "y": 144
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": ""
          },
          "position": {
            "x": 752,
            "y": 144
          }
        }
      ],
      "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "type": "basic.code",
      "data": {
        "code": "// NOT logic gate\n\nassign c = ~ a;",
        "ports": {
          "in": [
            "a"
          ],
          "out": [
            "c"
          ]
        }
      }
    }
  }
}
```

```

    },
    "position": {
      "x": 256,
      "y": 48
    }
  },
  ],
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ],
  ],
  "deps": {},
  "image": "resources/images/not.svg",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  },
  "cnot": {
    "image": "",
    "state": {
      "pan": {
        "x": 0,
        "y": 0
      },
      "zoom": 1
    },
  },
  "graph": {
    "blocks": [
      {
        "id": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
        "type": "not",
        "data": {},
        "position": {
          "x": 280,
          "y": 248
        }
      }
    ]
  }
}

```

```
    }
  },
  {
    "id": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
    "type": "or",
    "data": {},
    "position": {
      "x": 464,
      "y": 136
    }
  },
  {
    "id": "55c6c056-3630-4482-ad47-f4d9ee83b835",
    "type": "basic.input",
    "data": {
      "label": "a"
    },
    "position": {
      "x": 88,
      "y": 248
    }
  },
  {
    "id": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "type": "basic.output",
    "data": {
      "label": "b"
    },
    "position": {
      "x": 640,
      "y": 136
    }
  },
  {
    "id": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
    "type": "low",
    "data": {},
    "position": {
      "x": 280,
      "y": 120
    }
  }
],
"wires": [
  {
    "source": {
      "block": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
      "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "55c6c056-3630-4482-ad47-f4d9ee83b835",
      "port": "out"
```

```

    },
    "target": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "664caf9e-5f40-4df4-800a-b626af702e62"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "97b51945-d716-4b6c-9db9-970d08541249"
    }
  },
  {
    "source": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "664caf9e-5f40-4df4-800a-b626af702e62"
    },
    "target": {
      "block": "c8c6eed3-548c-49c7-a162-282179d427b1",
      "port": "in"
    }
  }
]
},
"deps": {
  "or": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": "x"
          },
          "position": {
            "x": 64,
            "y": 80
          }
        },
        {
          "id": "97b51945-d716-4b6c-9db9-970d08541249",
          "type": "basic.input",
          "data": {
            "label": "y"
          },
          "position": {
            "x": 64,
            "y": 208
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {

```

```
        "label": "o"
      },
      "position": {
        "x": 752,
        "y": 144
      }
    },
    {
      "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
      "type": "basic.code",
      "data": {
        "code": "// OR logic gate\n\nassign c = a | b;",
        "ports": {
          "in": [
            "a",
            "b"
          ],
          "out": [
            "c"
          ]
        }
      },
      "position": {
        "x": 256,
        "y": 48
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "97b51945-d716-4b6c-9db9-970d08541249",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "b"
      }
    },
    {
      "source": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
}
```

```

    }
  ],
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
},
"not": {
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x"
        },
        "position": {
          "x": 64,
          "y": 144
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
          "label": "y"
        },
        "position": {
          "x": 752,
          "y": 144
        }
      }
    ],
    {
      "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "type": "basic.code",
      "data": {
        "code": "// NOT logic gate\n\nassign c = ! a;",
        "ports": {
          "in": [
            "a"
          ],
          "out": [
            "c"
          ]
        }
      },
      "position": {
        "x": 256,
        "y": 48
      }
    }
  ]
},

```

```
    "wires": [
      {
        "source": {
          "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "port": "out"
        },
        "target": {
          "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
          "port": "a"
        }
      },
      {
        "source": {
          "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
          "port": "c"
        },
        "target": {
          "block": "664caf9e-5f40-4df4-800a-b626af702e62",
          "port": "in"
        }
      }
    ]
  },
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
},
"low": {
  "graph": {
    "blocks": [
      {
        "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
        "type": "basic.code",
        "data": {
          "code": "// Bit 0\n\nassign v = 1'b0;",
          "ports": {
            "in": [],
            "out": [
              "v"
            ]
          }
        }
      },
      {
        "position": {
          "x": 96,
          "y": 96
        }
      }
    ],
    {
      "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
      "type": "basic.output",
      "data": {
        "label": "o"
      }
    }
  ]
}
```



```

        },
        "position": {
            "x": 608,
            "y": 192
        }
    },
    ],
    "wires": [
        {
            "source": {
                "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
                "port": "v"
            },
            "target": {
                "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
                "port": "in"
            }
        }
    ]
},
"deps": {},
"image": "",
"state": {
    "pan": {
        "x": 0,
        "y": 0
    },
    "zoom": 1
}
}
}
},
"not": {
    "graph": {
        "blocks": [
            {
                "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
                "type": "basic.input",
                "data": {
                    "label": "x"
                },
                "position": {
                    "x": 64,
                    "y": 144
                }
            },
            {
                "id": "664caf9e-5f40-4df4-800a-b626af702e62",
                "type": "basic.output",
                "data": {
                    "label": "y"
                },
                "position": {
                    "x": 752,
                    "y": 144
                }
            }
        ],
        {

```

```
    "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
    "type": "basic.code",
    "data": {
      "code": "// NOT logic gate\n\nassign c = ! a;",
      "ports": {
        "in": [
          "a"
        ],
        "out": [
          "c"
        ]
      }
    },
    "position": {
      "x": 256,
      "y": 48
    }
  },
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ],
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
}
```

## 1.4 Project

### 1.4.1 Definition

A project is a composition of blocks. It includes the FPGA board information.

Its *input* and *output* block instances have also the FPGA I/O values to allow the synthesis.

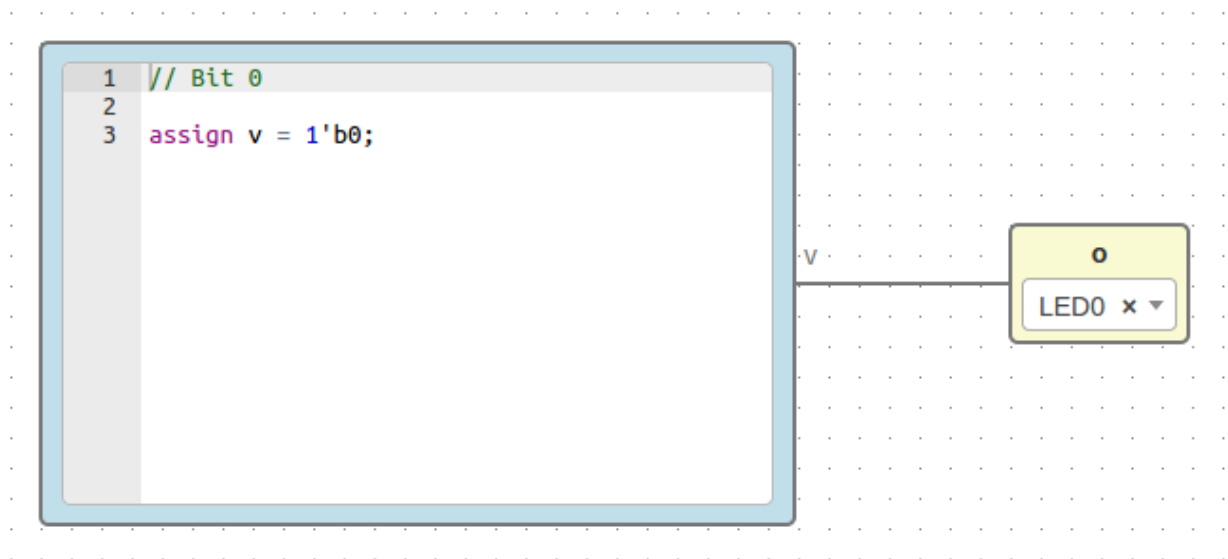
It can be exported as a block, by removing the FPGA board and I/O data.

Extension: **.ice**

```
{
  "board": "",
  "graph": {
    "blocks": [],
    "wires": []
  },
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
}
```

### 1.4.2 Examples

#### Low project



File: **low.ice**

Show/Hide code

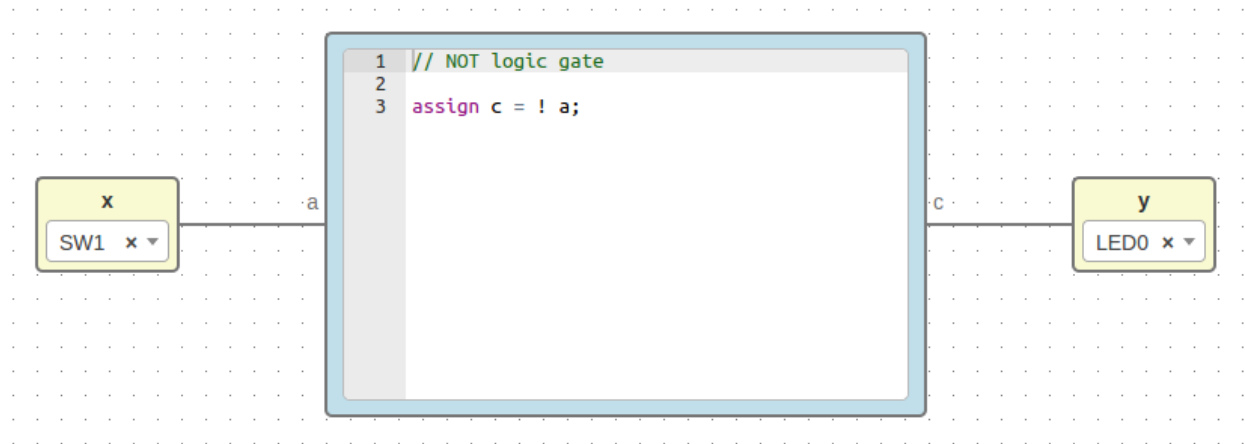
```
{
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
        "type": "basic.code",
        "data": {
          "code": "// Bit 0\n\nassign v = 1'b0;",
          "ports": {
            "in": [],
            "out": [
              "v"
            ]
          }
        },
        "position": {
          "x": 96,
          "y": 96
        }
      },
      {
        "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
        "type": "basic.output",
        "data": {
          "label": "o",
          "pin": {
            "name": "LED0",
            "value": "95"
          }
        },
        "position": {
          "x": 608,
          "y": 192
        }
      }
    ],
    "wires": [
      {
        "source": {
          "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
          "port": "v"
        },
        "target": {
          "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
          "port": "in"
        }
      }
    ],
    "deps": {},
    "image": "",
    "state": {
      "pan": {
        "x": 0,
```

```

    "y": 0
  },
  "zoom": 1
}
}

```

## Not project



File: not.ice

Show/Hide code

```

{
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x",
          "pin": {
            "name": "SW1",
            "value": "10"
          }
        },
        "position": {
          "x": 64,
          "y": 144
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",

```

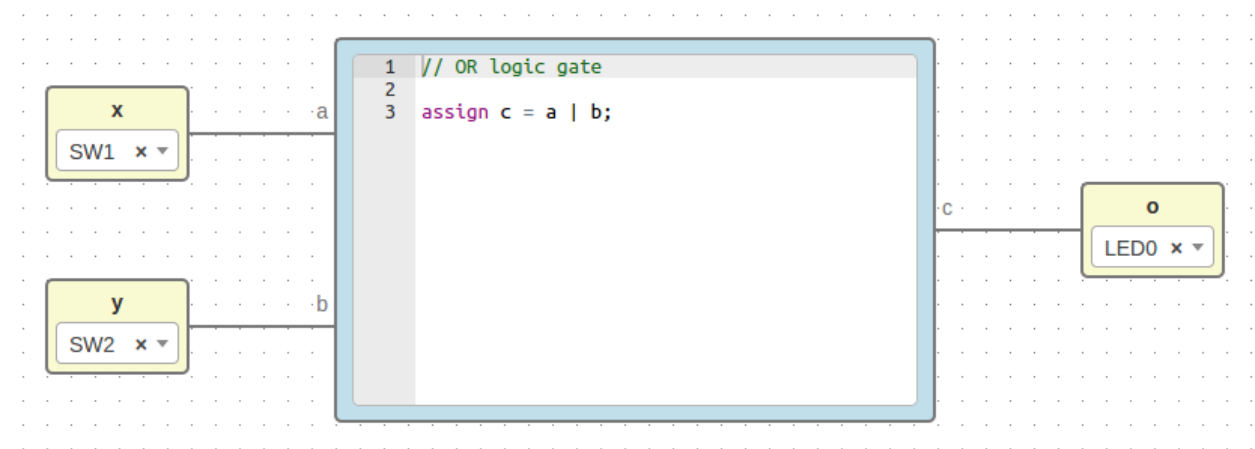
```
    "type": "basic.output",
    "data": {
      "label": "y",
      "pin": {
        "name": "LED0",
        "value": "95"
      }
    },
    "position": {
      "x": 752,
      "y": 144
    }
  },
  {
    "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
    "type": "basic.code",
    "data": {
      "code": "// NOT logic gate\n\nassign c = ! a;",
      "ports": {
        "in": [
          "a"
        ],
        "out": [
          "c"
        ]
      }
    },
    "position": {
      "x": 256,
      "y": 48
    }
  }
],
"wires": [
  {
    "source": {
      "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
      "port": "out"
    },
    "target": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "a"
    }
  },
  {
    "source": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "c"
    },
    "target": {
      "block": "664caf9e-5f40-4df4-800a-b626af702e62",
      "port": "in"
    }
  }
]
},
"deps": {},
"image": "",
```

```

"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
}

```

## Or project



File: or.ice

Show/Hide code

```

{
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x",
          "pin": {
            "name": "SW1",
            "value": "10"
          }
        },
        "position": {
          "x": 64,
          "y": 80
        }
      }
    ]
  }
}

```

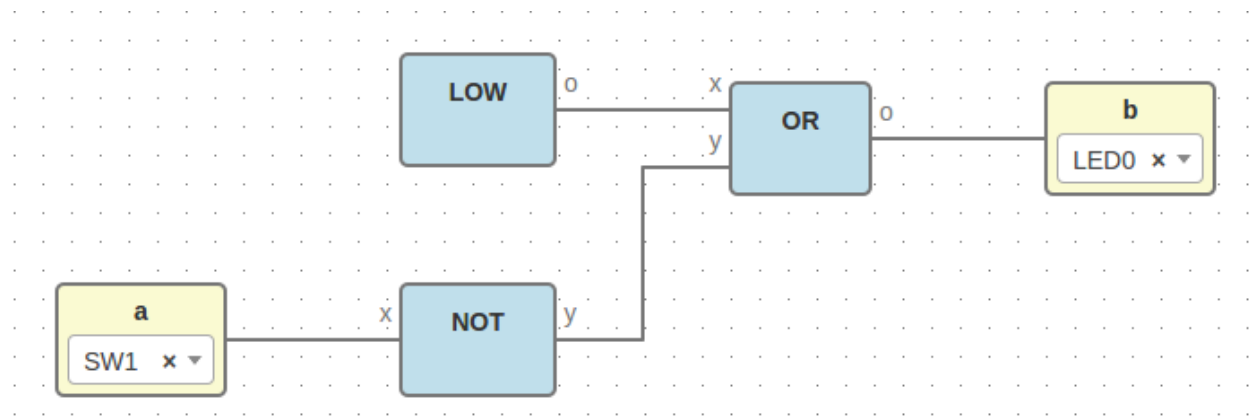
```
,
{
  "id": "97b51945-d716-4b6c-9db9-970d08541249",
  "type": "basic.input",
  "data": {
    "label": "y",
    "pin": {
      "name": "SW2",
      "value": "11"
    }
  },
  "position": {
    "x": 64,
    "y": 208
  }
},
{
  "id": "664caf9e-5f40-4df4-800a-b626af702e62",
  "type": "basic.output",
  "data": {
    "label": "o",
    "pin": {
      "name": "LED0",
      "value": "95"
    }
  },
  "position": {
    "x": 752,
    "y": 144
  }
},
{
  "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
  "type": "basic.code",
  "data": {
    "code": "// OR logic gate\n\nassign c = a | b;",
    "ports": {
      "in": [
        "a",
        "b"
      ],
      "out": [
        "c"
      ]
    }
  },
  "position": {
    "x": 256,
    "y": 48
  }
}
],
"wires": [
  {
    "source": {
      "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
      "port": "out"
    },
  },

```



```
    "target": {
      "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
      "port": "a"
    },
    {
      "source": {
        "block": "97b51945-d716-4b6c-9db9-970d08541249",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "b"
      }
    },
    {
      "source": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
```

## Cnot project



File: cnot.ice

Show/Hide code

```

{
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
        "type": "not",
        "data": {},
        "position": {
          "x": 280,
          "y": 248
        }
      },
      {
        "id": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
        "type": "or",
        "data": {},
        "position": {
          "x": 464,
          "y": 136
        }
      },
      {
        "id": "55c6c056-3630-4482-ad47-f4d9ee83b835",
        "type": "basic.input",

```

```

    "data": {
      "label": "a",
      "pin": {
        "name": "SW1",
        "value": "10"
      }
    },
    "position": {
      "x": 88,
      "y": 248
    }
  },
  {
    "id": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "type": "basic.output",
    "data": {
      "label": "b",
      "pin": {
        "name": "LED0",
        "value": "95"
      }
    },
    "position": {
      "x": 640,
      "y": 136
    }
  },
  {
    "id": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
    "type": "low",
    "data": {},
    "position": {
      "x": 280,
      "y": 120
    }
  }
],
"wires": [
  {
    "source": {
      "block": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
      "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "55c6c056-3630-4482-ad47-f4d9ee83b835",
      "port": "out"
    },
    "target": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  }
],

```

```
{
  "source": {
    "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
    "port": "664caf9e-5f40-4df4-800a-b626af702e62"
  },
  "target": {
    "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
    "port": "97b51945-d716-4b6c-9db9-970d08541249"
  }
},
{
  "source": {
    "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
    "port": "664caf9e-5f40-4df4-800a-b626af702e62"
  },
  "target": {
    "block": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "port": "in"
  }
}
]
},
"deps": {
  "or": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": "x"
          },
          "position": {
            "x": 64,
            "y": 80
          }
        },
        {
          "id": "97b51945-d716-4b6c-9db9-970d08541249",
          "type": "basic.input",
          "data": {
            "label": "y"
          },
          "position": {
            "x": 64,
            "y": 208
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": "o"
          },
          "position": {
            "x": 752,
            "y": 144
          }
        }
      ]
    }
  }
}
```

```

    },
    {
      "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
      "type": "basic.code",
      "data": {
        "code": "// OR logic gate\n\nassign c = a | b;",
        "ports": {
          "in": [
            "a",
            "b"
          ],
          "out": [
            "c"
          ]
        }
      },
      "position": {
        "x": 256,
        "y": 48
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "97b51945-d716-4b6c-9db9-970d08541249",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "b"
      }
    },
    {
      "source": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {

```

```
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "not": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": "x"
          },
          "position": {
            "x": 64,
            "y": 144
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": "y"
          },
          "position": {
            "x": 752,
            "y": 144
          }
        },
        {
          "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
          "type": "basic.code",
          "data": {
            "code": "// NOT logic gate\n\nassign c = ! a;",
            "ports": {
              "in": [
                "a"
              ],
              "out": [
                "c"
              ]
            }
          },
          "position": {
            "x": 256,
            "y": 48
          }
        }
      ],
      "wires": [
        {
          "source": {
            "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
            "port": "out"
          },
```

```

        "target": {
            "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
            "port": "a"
        }
    },
    {
        "source": {
            "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
            "port": "c"
        },
        "target": {
            "block": "664caf9e-5f40-4df4-800a-b626af702e62",
            "port": "in"
        }
    }
]
},
"deps": {},
"image": "",
"state": {
    "pan": {
        "x": 0,
        "y": 0
    },
    "zoom": 1
},
"low": {
    "graph": {
        "blocks": [
            {
                "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
                "type": "basic.code",
                "data": {
                    "code": "// Bit 0\n\nassign v = 1'b0;",
                    "ports": {
                        "in": [],
                        "out": [
                            "v"
                        ]
                    }
                },
                "position": {
                    "x": 96,
                    "y": 96
                }
            },
            {
                "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
                "type": "basic.output",
                "data": {
                    "label": "o"
                },
                "position": {
                    "x": 608,
                    "y": 192
                }
            }
        ]
    }
}

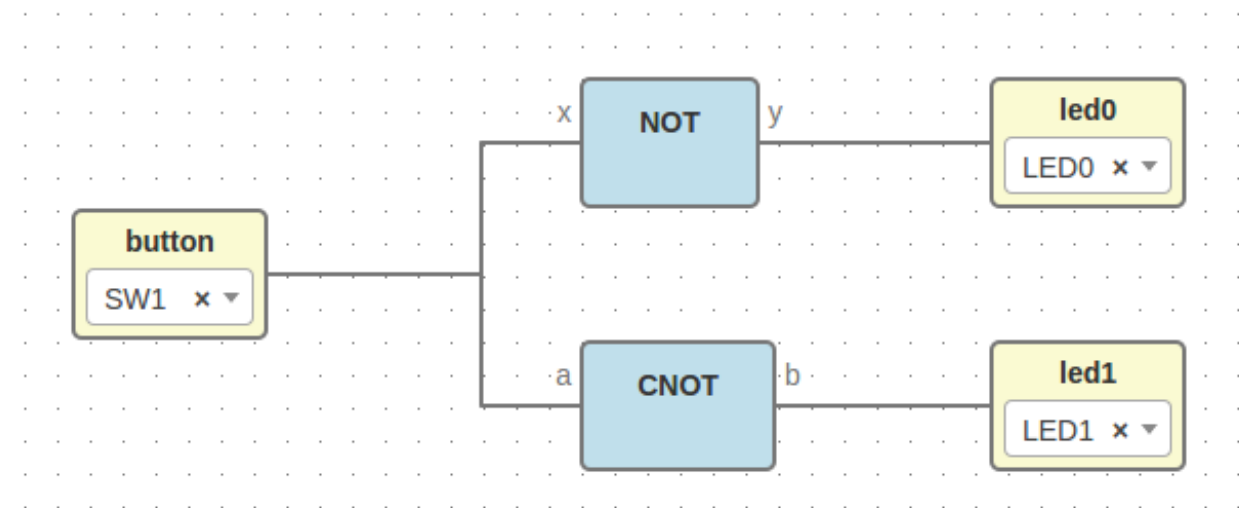
```

```

    ],
    "wires": [
      {
        "source": {
          "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
          "port": "v"
        },
        "target": {
          "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
          "port": "in"
        }
      }
    ]
  },
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
}

```

### Dnot project



File: **dnot.ice**

Show/Hide code



```

{
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
        "type": "basic.input",
        "data": {
          "label": "button",
          "pin": {
            "name": "SW1",
            "value": "10"
          }
        },
        "position": {
          "x": 104,
          "y": 176
        }
      },
      {
        "id": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
        "type": "cnot",
        "data": {},
        "position": {
          "x": 352,
          "y": 240
        }
      },
      {
        "id": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
        "type": "not",
        "data": {},
        "position": {
          "x": 352,
          "y": 112
        }
      },
      {
        "id": "4c4d2ddd-a97d-4fcb-9c68-ba1149f25082",
        "type": "basic.output",
        "data": {
          "label": "led0",
          "pin": {
            "name": "LED0",
            "value": "95"
          }
        },
        "position": {
          "x": 552,
          "y": 112
        }
      }
    ]
  }
}

```

```
    },
  },
  {
    "id": "0e777320-de37-4dca-a077-51fbf10a6565",
    "type": "basic.output",
    "data": {
      "label": "led1",
      "pin": {
        "name": "LED1",
        "value": "96"
      }
    },
    "position": {
      "x": 552,
      "y": 240
    }
  }
],
"wires": [
  {
    "source": {
      "block": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
      "port": "out"
    },
    "target": {
      "block": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
      "port": "out"
    },
    "target": {
      "block": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
      "port": "55c6c056-3630-4482-ad47-f4d9ee83b835"
    }
  },
  {
    "source": {
      "block": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
      "port": "664caf9e-5f40-4df4-800a-b626af702e62"
    },
    "target": {
      "block": "4c4d2ddd-a97d-4fcb-9c68-ba1149f25082",
      "port": "in"
    }
  },
  {
    "source": {
      "block": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
      "port": "c8c6eed3-548c-49c7-a162-282179d427b1"
    },
    "target": {
      "block": "0e777320-de37-4dca-a077-51fbf10a6565",
      "port": "in"
    }
  }
]
```

```

    }
  ]
},
"deps": {
  "logic.not": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": ""
          },
          "position": {
            "x": 64,
            "y": 144
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": ""
          },
          "position": {
            "x": 752,
            "y": 144
          }
        },
        {
          "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
          "type": "basic.code",
          "data": {
            "code": "// NOT logic gate\n\nassign c = ~ a;",
            "ports": {
              "in": [
                "a"
              ],
              "out": [
                "c"
              ]
            }
          },
          "position": {
            "x": 256,
            "y": 48
          }
        }
      ],
      "wires": [
        {
          "source": {
            "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
            "port": "out"
          },
          "target": {
            "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
            "port": "a"
          }
        }
      ]
    }
  }
}

```

```
    }
  },
  {
    "source": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "c"
    },
    "target": {
      "block": "664caf9e-5f40-4df4-800a-b626af702e62",
      "port": "in"
    }
  }
]
},
"deps": {},
"image": "resources/images/not.svg",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
},
},
"cnot": {
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
},
"graph": {
  "blocks": [
    {
      "id": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "type": "not",
      "data": {},
      "position": {
        "x": 280,
        "y": 248
      }
    },
    {
      "id": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "type": "or",
      "data": {},
      "position": {
        "x": 464,
        "y": 136
      }
    },
    {
      "id": "55c6c056-3630-4482-ad47-f4d9ee83b835",
      "type": "basic.input",
      "data": {
        "label": "a"
      }
    }
  ]
}
```

```

    },
    "position": {
      "x": 88,
      "y": 248
    }
  },
  {
    "id": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "type": "basic.output",
    "data": {
      "label": "b"
    },
    "position": {
      "x": 640,
      "y": 136
    }
  },
  {
    "id": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
    "type": "low",
    "data": {},
    "position": {
      "x": 280,
      "y": 120
    }
  }
],
"wires": [
  {
    "source": {
      "block": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
      "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "55c6c056-3630-4482-ad47-f4d9ee83b835",
      "port": "out"
    },
    "target": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "664caf9e-5f40-4df4-800a-b626af702e62"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "97b51945-d716-4b6c-9db9-970d08541249"
    }
  }
],

```

```
{
  "source": {
    "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
    "port": "664caf9e-5f40-4df4-800a-b626af702e62"
  },
  "target": {
    "block": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "port": "in"
  }
}
]
},
"deps": {
  "or": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": "x"
          },
          "position": {
            "x": 64,
            "y": 80
          }
        },
        {
          "id": "97b51945-d716-4b6c-9db9-970d08541249",
          "type": "basic.input",
          "data": {
            "label": "y"
          },
          "position": {
            "x": 64,
            "y": 208
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": "o"
          },
          "position": {
            "x": 752,
            "y": 144
          }
        },
        {
          "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
          "type": "basic.code",
          "data": {
            "code": "// OR logic gate\n\nassign c = a | b;",
            "ports": {
              "in": [
                "a",
                "b"
              ]
            }
          }
        }
      ]
    }
  }
}
```

```

        ],
        "out": [
            "c"
        ]
    },
    "position": {
        "x": 256,
        "y": 48
    }
}
],
"wires": [
    {
        "source": {
            "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
            "port": "out"
        },
        "target": {
            "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
            "port": "a"
        }
    },
    {
        "source": {
            "block": "97b51945-d716-4b6c-9db9-970d08541249",
            "port": "out"
        },
        "target": {
            "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
            "port": "b"
        }
    },
    {
        "source": {
            "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
            "port": "c"
        },
        "target": {
            "block": "664caf9e-5f40-4df4-800a-b626af702e62",
            "port": "in"
        }
    }
]
},
"deps": {},
"image": "",
"state": {
    "pan": {
        "x": 0,
        "y": 0
    },
    "zoom": 1
}
},
"not": {
    "graph": {
        "blocks": [

```

```
{
  "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
  "type": "basic.input",
  "data": {
    "label": "x"
  },
  "position": {
    "x": 64,
    "y": 144
  }
},
{
  "id": "664caf9e-5f40-4df4-800a-b626af702e62",
  "type": "basic.output",
  "data": {
    "label": "y"
  },
  "position": {
    "x": 752,
    "y": 144
  }
},
{
  "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
  "type": "basic.code",
  "data": {
    "code": "// NOT logic gate\n\nassign c = ! a;",
    "ports": {
      "in": [
        "a"
      ],
      "out": [
        "c"
      ]
    }
  },
  "position": {
    "x": 256,
    "y": 48
  }
},
],
"wires": [
  {
    "source": {
      "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
      "port": "out"
    },
    "target": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "a"
    }
  },
  {
    "source": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "c"
    }
  },

```



```

        "target": {
            "block": "664caf9e-5f40-4df4-800a-b626af702e62",
            "port": "in"
        }
    }
}
],
},
"deps": {},
"image": "",
"state": {
    "pan": {
        "x": 0,
        "y": 0
    },
    "zoom": 1
}
},
"low": {
    "graph": {
        "blocks": [
            {
                "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
                "type": "basic.code",
                "data": {
                    "code": "// Bit 0\n\nassign v = 1'b0;",
                    "ports": {
                        "in": [],
                        "out": [
                            "v"
                        ]
                    }
                },
                "position": {
                    "x": 96,
                    "y": 96
                }
            },
            {
                "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
                "type": "basic.output",
                "data": {
                    "label": "o"
                },
                "position": {
                    "x": 608,
                    "y": 192
                }
            }
        ]
    },
    "wires": [
        {
            "source": {
                "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
                "port": "v"
            },
            "target": {
                "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
                "port": "in"
            }
        }
    ]
}

```

```
    }
  }
]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
}
},
"not": {
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x"
        },
        "position": {
          "x": 64,
          "y": 144
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
          "label": "y"
        },
        "position": {
          "x": 752,
          "y": 144
        }
      },
      {
        "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "type": "basic.code",
        "data": {
          "code": "// NOT logic gate\n\nassign c = ! a;",
          "ports": {
            "in": [
              "a"
            ],
            "out": [
              "c"
            ]
          }
        },
        "position": {
          "x": 256,
          "y": 48
        }
      }
    ]
  }
}
```

```

    }
  },
  ],
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ],
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
}

```

## 1.5 Compiler

The JSON structure of a project is a block definition.

Output verilog structure:

1. Modules
2. Main module
  - (a) Wires definition
  - (b) Wires connections
  - (c) Blocks instances

## 1.5.1 Implementation

Show/Hide code

```
1  /**
2   * @author Jesús Arroyo Torrens <jesus.jkhl@gmail.com>
3   *
4   * June 2016
5   */
6
7   'use strict';
8
9   var fs = require('fs');
10  var sha1 = require('sha1');
11
12
13  function digestId(id, force) {
14    if (id.indexOf('-') !== -1) {
15      return 'v' + sha1(id).toString().substring(0, 6);
16    }
17    else {
18      return id.replace('.', '_');
19    }
20  }
21
22  function module(data) {
23    var code = '';
24
25    if (data &&
26        data.name &&
27        data.ports &&
28        data.content) {
29
30      // Header
31
32      code += 'module ';
33      code += data.name;
34      code += ' (';
35
36      var params = [];
37      var paramsSpace = 10 + data.name.length;
38
39      for (var i in data.ports.in) {
40        params.push('input ' + data.ports.in[i]);
41      }
42      for (var o in data.ports.out) {
43        params.push('output ' + data.ports.out[o]);
44      }
45
46      code += params.join(', \n' + new Array(paramsSpace).join(' '));
47
48      code += '); \n';
49
50      // Content
```

```

51     var content = data.content.split('\n');
52
53     content.forEach(function (element, index, array) {
54         array[index] = ' ' + element;
55     });
56
57     code += content.join('\n');
58
59     // Footer
60
61     code += '\nendmodule\n\n';
62 }
63
64 return code;
65 }
66
67 function getPorts(project) {
68     var ports = {
69         in: [],
70         out: []
71     };
72     var graph = project.graph;
73
74     for (var i in graph.blocks) {
75         var block = graph.blocks[i];
76         if (block.type == 'basic.input') {
77             ports.in.push(digestId(block.id));
78         }
79         else if (block.type == 'basic.output') {
80             ports.out.push(digestId(block.id));
81         }
82     }
83 }
84
85 return ports;
86 }
87
88 function getContent(name, project) {
89     var content = '';
90     var graph = project.graph;
91
92     // Wires
93
94     for (var w in graph.wires) {
95         content += 'wire w' + w + ';\n'
96     }
97
98     // I/O connections
99
100    for (var w in graph.wires) {
101        var wire = graph.wires[w];
102        for (var i in graph.blocks) {
103            var block = graph.blocks[i];
104            if (block.type == 'basic.input') {
105                if (wire.source.block == block.id) {
106                    content += 'assign w' + w + ' = ' + digestId(block.id) + ';\n';
107                }
108            }
109        }
110    }

```

```
109     else if (block.type == 'basic.output') {
110         if (wire.target.block == block.id) {
111             content += 'assign ' + digestId(block.id) + ' = w' + w + ';\n';
112         }
113     }
114 }
115 }
116
117 // Wires Connections
118
119 var numWires = graph.wires.length;
120 for (var i = 1; i < numWires; i++) {
121     for (var j = 0; j < i; j++) {
122         var wi = graph.wires[i];
123         var wj = graph.wires[j];
124         if (wi.source.block == wj.source.block &&
125             wi.source.port == wj.source.port) {
126             content += 'assign w' + i + ' = w' + j + ';\n';
127         }
128     }
129 }
130
131 // Block instances
132
133 var instances = []
134 for (var b in graph.blocks) {
135     var block = graph.blocks[b];
136     if (block.type != 'basic.input' &&
137         block.type != 'basic.output' &&
138         block.type != 'basic.info') {
139
140         var id = digestId(block.type, true);
141         if (block.type == 'basic.code') {
142             id += '_' + digestId(block.id);
143         }
144         instances.push(name + '_' + digestId(id) + ' ' + digestId(block.id) + ' (');
145
146         // Parameters
147
148         var params = [];
149         var paramsNames = [];
150         for (var w in graph.wires) {
151             var param = '';
152             var paramName = '';
153             var wire = graph.wires[w];
154             if (block.id == wire.source.block) {
155                 paramName = digestId(wire.source.port);
156             }
157             else if (block.id == wire.target.block) {
158                 paramName = digestId(wire.target.port);
159             }
160             if (paramName && paramsNames.indexOf(paramName) == -1) {
161                 paramsNames.push(paramName);
162                 param += ' ' + paramName;
163                 param += '(w' + w + ')';
164                 params.push(param);
165             }
166         }
```

```

167     instances.push(params.join(',\n') + '\n;');
168   }
169 }
170 }
171 content += instances.join('\n');
172
173 return content;
174 }
175
176 function verilogCompiler(name, project) {
177   var code = '';
178
179   if (project &&
180       project.graph) {
181
182     // Scape dot in name
183
184     name = digestId(name);
185
186     // Main module
187
188     if (name) {
189       var data = {
190         name: name,
191         ports: getPorts(project),
192         content: getContent(name, project)
193       };
194       code += module(data);
195     }
196
197     // Dependencies modules
198
199     for (var d in project.deps) {
200       code += verilogCompiler(name + '_' + digestId(d, true), project.deps[d]);
201     }
202
203     // Code modules
204
205     for (var i in project.graph.blocks) {
206       var block = project.graph.blocks[i];
207       if (block) {
208         if (block.type == 'basic.code') {
209           var data = {
210             name: name + '_' + digestId(block.type, true) + '_' + digestId(block.id),
211             ports: block.data.ports,
212             content: block.data.code
213           };
214           code += module(data);
215         }
216       }
217     }
218   }
219
220   return code;
221 }
222
223 function pcfCompiler(project) {
224   var code = '';

```

```

225
226   for (var i in project.graph.blocks) {
227       var block = project.graph.blocks[i];
228       if (block.type == 'basic.input' ||
229           block.type == 'basic.output') {
230           code += 'set_io ';
231           code += digestId(block.id);
232           code += ' ';
233           code += block.data.pin.value;
234           code += '\n';
235       }
236   }
237
238   return code;
239 }
240
241 // Examples
242
243 var fs = require('fs');
244
245 function compare_string(s1, s2) {
246     var diff = [];
247     var string1 = s1.split(" ");
248     var string2 = s2.split(" ");
249     var size = Math.max(s1.length, s2.length);
250
251     for(var x = 0; x < size; x++) {
252         if(string1[x] != string2[x]) {
253             diff.push(string1[x]);
254         }
255     }
256
257     return diff.join(' ');
258 }
259
260 function test_example(name, extension) {
261     var filename = ['. ', 'resources', 'examples', name, name].join('/');
262     fs.readFile(filename + '.' + extension, 'utf8', function (err, data) {
263         if (err) throw err;
264
265         var example = JSON.parse(fs.readFileSync(filename + '.ice'));
266         if (extension == 'v') {
267             var s1 = verilogCompiler('main', example).replace(/\r\n/g, "");
268         }
269         else {
270             var s1 = pcfCompiler(example).replace(/\r\n/g, "");
271         }
272         var s2 = data.replace(/\r\n/g, "");
273
274         if (extension == 'v') {
275             process.stdout.write('Testing ' + name + ' v ...');
276         }
277         else {
278             process.stdout.write('Testing ' + name + ' pcf ...');
279         }
280         if (s1 == s2) {
281             process.stdout.write(' [OK]\n');
282         }

```



```

283     else {
284         process.stdout.write(' [Fail]\n');
285         process.stdout.write(compare_string(s1, s2) + '\n');
286     }
287 });
288 }
289
290 // Test examples
291
292 test_example('low', 'v');
293 test_example('low', 'pcf');
294 test_example('not', 'v');
295 test_example('not', 'pcf');
296 test_example('or', 'v');
297 test_example('or', 'pcf');
298 test_example('cnot', 'v');
299 test_example('cnot', 'pcf');
300 test_example('dnot', 'v');
301 test_example('dnot', 'pcf');
302
303 //console.log(verilogCompiler('main', JSON.parse(fs.readFileSync('../resources/examples/dnot/dnot.ice
304 //console.log(pcfCompiler(JSON.parse(fs.readFileSync('../resources/examples/dnot/dnot.ice'))));

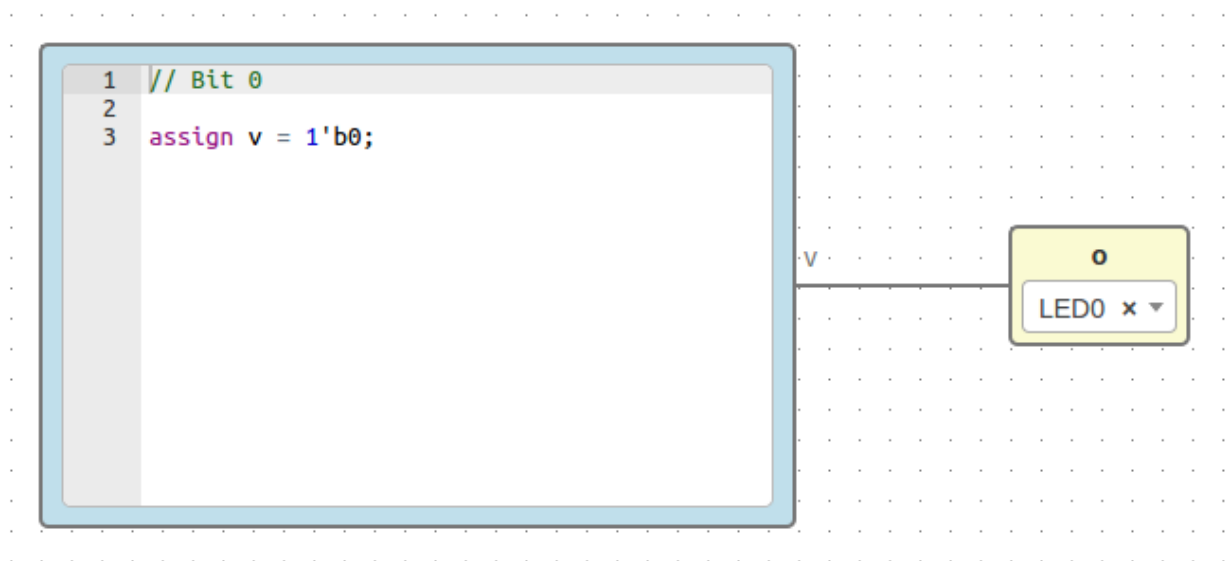
```

```
npm install fs sha1
```

```
node compiler.js
```

## 1.5.2 Examples

### Low project



File: low.ice

Show/Hide code

```
{
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
        "type": "basic.code",
        "data": {
          "code": "// Bit 0\n\nassign v = 1'b0;",
          "ports": {
            "in": [],
            "out": [
              "v"
            ]
          }
        },
        "position": {
          "x": 96,
          "y": 96
        }
      },
      {
        "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
        "type": "basic.output",
        "data": {
          "label": "o",
          "pin": {
            "name": "LED0",
            "value": "95"
          }
        },
        "position": {
          "x": 608,
          "y": 192
        }
      }
    ],
    "wires": [
      {
        "source": {
          "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
          "port": "v"
        },
        "target": {
          "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
          "port": "in"
        }
      }
    ]
  },
}
```

```

"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
}

```

Generates

```

module main (output v608bd9);
  wire w0;
  assign v608bd9 = w0;
  main_basic_code_v68c173 v68c173 (
    .v(w0)
  );
endmodule

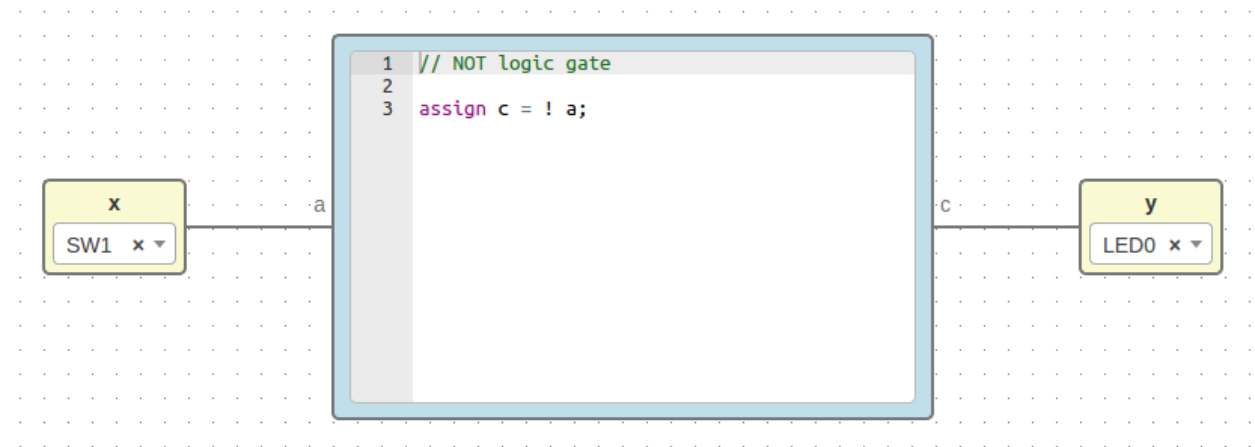
module main_basic_code_v68c173 (output v);
  // Bit 0

  assign v = 1'b0;
endmodule

```

```
set_io v608bd9 95
```

## Not project



File: not.ice

Show/Hide code

```
{
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x",
          "pin": {
            "name": "SW1",
            "value": "10"
          }
        },
        "position": {
          "x": 64,
          "y": 144
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
          "label": "y",
          "pin": {
            "name": "LED0",
            "value": "95"
          }
        },
        "position": {
          "x": 752,
          "y": 144
        }
      },
      {
        "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "type": "basic.code",
        "data": {
          "code": "// NOT logic gate\n\nassign c = ! a;",
          "ports": {
            "in": [
              "a"
            ],
            "out": [
              "c"
            ]
          }
        },
        "position": {
          "x": 256,
          "y": 48
        }
      }
    ],
    "wires": [
      {
        "source": {
          "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
```

```

        "port": "out"
      },
      "target": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
}

```

Generates

```

module main (input v0e28cb,
              output vcbab45);
  wire w0;
  wire w1;
  assign w0 = v0e28cb;
  assign vcbab45 = w1;
  main_basic_code_vd54ca1 vd54ca1 (
    .a(w0),
    .c(w1)
  );
endmodule

module main_basic_code_vd54ca1 (input a,
                              output c);
  // NOT logic gate

  assign c = ! a;
endmodule

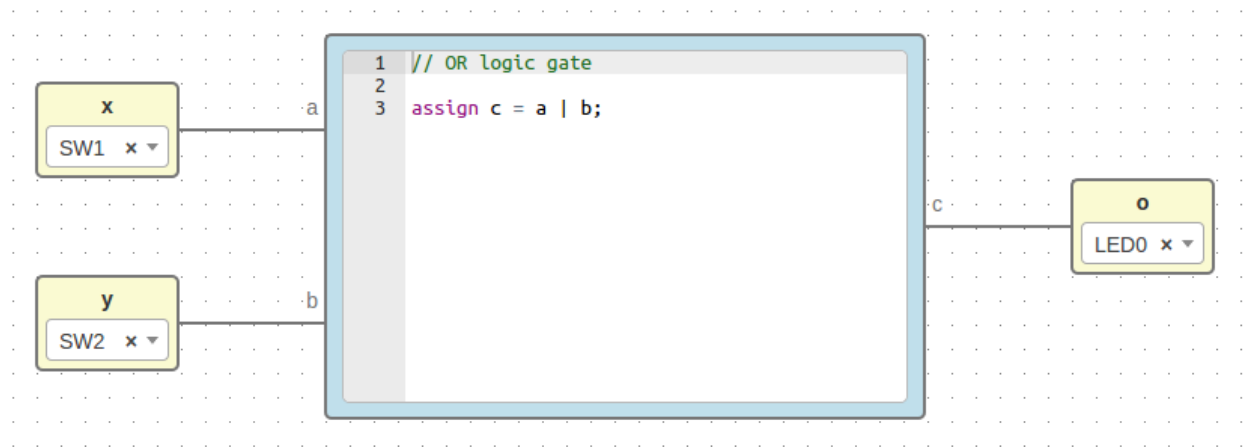
```

```

set_io v0e28cb 10
set_io vcbab45 95

```

## Or project



File: or.ice

Show/Hide code

```
{
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x",
          "pin": {
            "name": "SW1",
            "value": "10"
          }
        },
        "position": {
          "x": 64,
          "y": 80
        }
      },
      {
        "id": "97b51945-d716-4b6c-9db9-970d08541249",
        "type": "basic.input",
        "data": {
          "label": "y",
          "pin": {
            "name": "SW2",
            "value": "11"
          }
        },
        "position": {
          "x": 64,
          "y": 208
        }
      }
    ]
  }
}
```

```

    },
    {
      "id": "664caf9e-5f40-4df4-800a-b626af702e62",
      "type": "basic.output",
      "data": {
        "label": "o",
        "pin": {
          "name": "LED0",
          "value": "95"
        }
      },
      "position": {
        "x": 752,
        "y": 144
      }
    },
    {
      "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
      "type": "basic.code",
      "data": {
        "code": "// OR logic gate\n\nassign c = a | b;",
        "ports": {
          "in": [
            "a",
            "b"
          ],
          "out": [
            "c"
          ]
        }
      },
      "position": {
        "x": 256,
        "y": 48
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "97b51945-d716-4b6c-9db9-970d08541249",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "b"
      }
    }
  ]
}

```

```
    },
    {
      "source": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
}
}
```

Generates

```
module main (input v0e28cb,
              input v3ca442,
              output vcbab45);

  wire w0;
  wire w1;
  wire w2;
  assign w0 = v0e28cb;
  assign w1 = v3ca442;
  assign vcbab45 = w2;
  main_basic_code_vf4938a vf4938a (
    .a(w0),
    .b(w1),
    .c(w2)
  );
endmodule

module main_basic_code_vf4938a (input a,
                              input b,
                              output c);

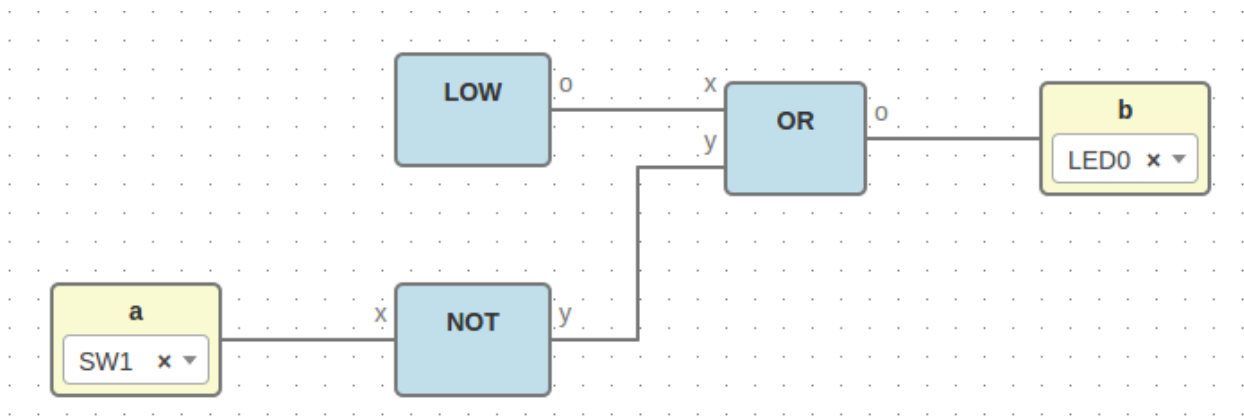
  // OR logic gate

  assign c = a | b;
endmodule
```

```
set_io v0e28cb 10
set_io v3ca442 11
set_io vcbab45 95
```



## Cnot project



File: cnot.ice

Show/Hide code

```

{
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
        "type": "not",
        "data": {},
        "position": {
          "x": 280,
          "y": 248
        }
      },
      {
        "id": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
        "type": "or",
        "data": {},
        "position": {
          "x": 464,
          "y": 136
        }
      },
      {
        "id": "55c6c056-3630-4482-ad47-f4d9ee83b835",
        "type": "basic.input",

```

```
    "data": {
      "label": "a",
      "pin": {
        "name": "SW1",
        "value": "10"
      }
    },
    "position": {
      "x": 88,
      "y": 248
    }
  },
  {
    "id": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "type": "basic.output",
    "data": {
      "label": "b",
      "pin": {
        "name": "LED0",
        "value": "95"
      }
    },
    "position": {
      "x": 640,
      "y": 136
    }
  },
  {
    "id": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
    "type": "low",
    "data": {},
    "position": {
      "x": 280,
      "y": 120
    }
  }
],
"wires": [
  {
    "source": {
      "block": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
      "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "55c6c056-3630-4482-ad47-f4d9ee83b835",
      "port": "out"
    },
    "target": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  }
],
```

```

{
  "source": {
    "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
    "port": "664caf9e-5f40-4df4-800a-b626af702e62"
  },
  "target": {
    "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
    "port": "97b51945-d716-4b6c-9db9-970d08541249"
  }
},
{
  "source": {
    "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
    "port": "664caf9e-5f40-4df4-800a-b626af702e62"
  },
  "target": {
    "block": "c8c6eed3-548c-49c7-a162-282179d427b1",
    "port": "in"
  }
}
]
},
"deps": {
  "or": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": "x"
          },
          "position": {
            "x": 64,
            "y": 80
          }
        },
        {
          "id": "97b51945-d716-4b6c-9db9-970d08541249",
          "type": "basic.input",
          "data": {
            "label": "y"
          },
          "position": {
            "x": 64,
            "y": 208
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": "o"
          },
          "position": {
            "x": 752,
            "y": 144
          }
        }
      ]
    }
  }
}

```

```
    },
    {
      "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
      "type": "basic.code",
      "data": {
        "code": "// OR logic gate\n\nassign c = a | b;",
        "ports": {
          "in": [
            "a",
            "b"
          ],
          "out": [
            "c"
          ]
        }
      },
      "position": {
        "x": 256,
        "y": 48
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "97b51945-d716-4b6c-9db9-970d08541249",
        "port": "out"
      },
      "target": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "b"
      }
    },
    {
      "source": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {
```

```

    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "not": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": "x"
          },
          "position": {
            "x": 64,
            "y": 144
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": "y"
          },
          "position": {
            "x": 752,
            "y": 144
          }
        },
        {
          "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
          "type": "basic.code",
          "data": {
            "code": "// NOT logic gate\n\nassign c = ! a;",
            "ports": {
              "in": [
                "a"
              ],
              "out": [
                "c"
              ]
            }
          },
          "position": {
            "x": 256,
            "y": 48
          }
        }
      ],
      "wires": [
        {
          "source": {
            "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
            "port": "out"
          },

```

```
    "target": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "a"
    }
  },
  {
    "source": {
      "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "port": "c"
    },
    "target": {
      "block": "664caf9e-5f40-4df4-800a-b626af702e62",
      "port": "in"
    }
  }
]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
},
"low": {
  "graph": {
    "blocks": [
      {
        "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
        "type": "basic.code",
        "data": {
          "code": "// Bit 0\n\nassign v = 1'b0;",
          "ports": {
            "in": [],
            "out": [
              "v"
            ]
          }
        },
        "position": {
          "x": 96,
          "y": 96
        }
      },
      {
        "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
        "type": "basic.output",
        "data": {
          "label": "o"
        },
        "position": {
          "x": 608,
          "y": 192
        }
      }
    ]
  }
}
```

```

    ],
    "wires": [
      {
        "source": {
          "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
          "port": "v"
        },
        "target": {
          "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
          "port": "in"
        }
      }
    ]
  },
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
}
}
}

```

Generates

```

module main (input vald1bb,
              output vecf2e3);

  wire w0;
  wire w1;
  wire w2;
  wire w3;
  assign w1 = vald1bb;
  assign vecf2e3 = w3;
  main_not va44cd3 (
    .v0e28cb(w1),
    .vcbab45(w2)
  );
  main_or v0b7a71 (
    .v0e28cb(w0),
    .v3ca442(w2),
    .vcbab45(w3)
  );
  main_low v2d7478 (
    .v608bd9(w0)
  );
endmodule

module main_or (input v0e28cb,
                input v3ca442,
                output vcbab45);

  wire w0;
  wire w1;
  wire w2;
  assign w0 = v0e28cb;

```

```
assign w1 = v3ca442;
assign vcbab45 = w2;
main_or_basic_code_vf4938a vf4938a (
    .a(w0),
    .b(w1),
    .c(w2)
);
endmodule

module main_or_basic_code_vf4938a (input a,
                                input b,
                                output c);

    // OR logic gate

    assign c = a | b;
endmodule

module main_not (input v0e28cb,
                 output vcbab45);

    wire w0;
    wire w1;
    assign w0 = v0e28cb;
    assign vcbab45 = w1;
    main_not_basic_code_vd54ca1 vd54ca1 (
        .a(w0),
        .c(w1)
    );
endmodule

module main_not_basic_code_vd54ca1 (input a,
                                   output c);

    // NOT logic gate

    assign c = ! a;
endmodule

module main_low (output v608bd9);
    wire w0;
    assign v608bd9 = w0;
    main_low_basic_code_v68c173 v68c173 (
        .v(w0)
    );
endmodule

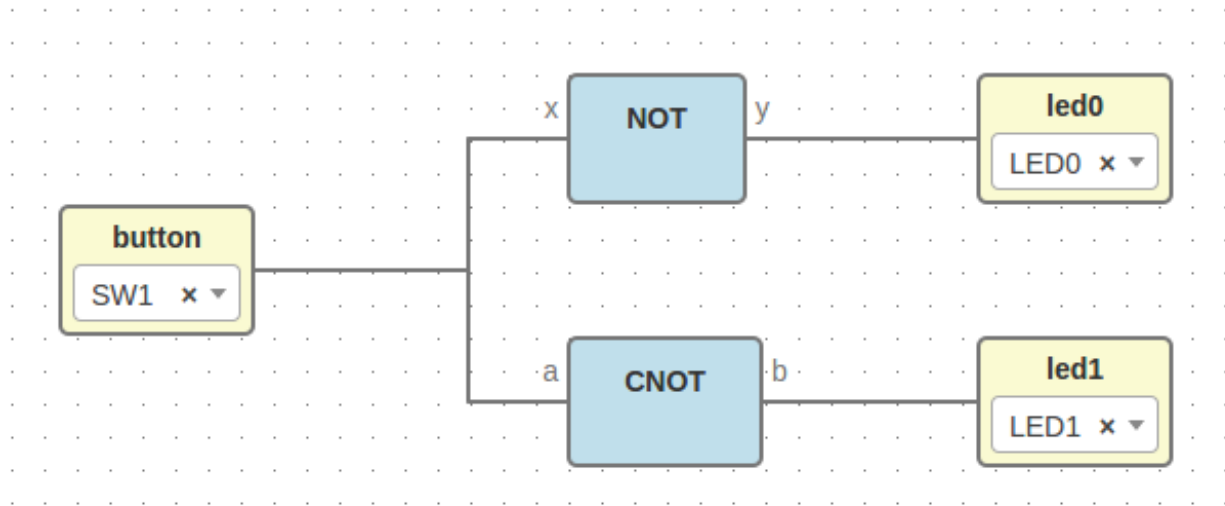
module main_low_basic_code_v68c173 (output v);
    // Bit 0

    assign v = 1'b0;
endmodule
```

```
set_io vald1bb 10
set_io vecf2e3 95
```



## Dnot project



File: dnot.ice

Show/Hide code

```

{
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "board": "icezum",
  "graph": {
    "blocks": [
      {
        "id": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
        "type": "basic.input",
        "data": {
          "label": "button",
          "pin": {
            "name": "SW1",
            "value": "10"
          }
        },
        "position": {
          "x": 104,
          "y": 176
        }
      },
      {
        "id": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
        "type": "cnot",

```

```
    "data": {},
    "position": {
      "x": 352,
      "y": 240
    }
  },
  {
    "id": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
    "type": "not",
    "data": {},
    "position": {
      "x": 352,
      "y": 112
    }
  },
  {
    "id": "4c4d2ddd-a97d-4fcb-9c68-ba1149f25082",
    "type": "basic.output",
    "data": {
      "label": "led0",
      "pin": {
        "name": "LED0",
        "value": "95"
      }
    },
    "position": {
      "x": 552,
      "y": 112
    }
  },
  {
    "id": "0e777320-de37-4dca-a077-51fbf10a6565",
    "type": "basic.output",
    "data": {
      "label": "led1",
      "pin": {
        "name": "LED1",
        "value": "96"
      }
    },
    "position": {
      "x": 552,
      "y": 240
    }
  }
],
"wires": [
  {
    "source": {
      "block": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
      "port": "out"
    },
    "target": {
      "block": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
```

```

    "source": {
      "block": "327f1a9e-ba42-4d25-adcd-f7f16ac8f451",
      "port": "out"
    },
    "target": {
      "block": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
      "port": "55c6c056-3630-4482-ad47-f4d9ee83b835"
    }
  },
  {
    "source": {
      "block": "88b3c210-c6f5-4cd3-a578-2e5ab8aa1562",
      "port": "664caf9e-5f40-4df4-800a-b626af702e62"
    },
    "target": {
      "block": "4c4d2ddd-a97d-4fcb-9c68-ba1149f25082",
      "port": "in"
    }
  },
  {
    "source": {
      "block": "58c892ba-89a3-4da7-9d0a-56f2523bfd98",
      "port": "c8c6eed3-548c-49c7-a162-282179d427b1"
    },
    "target": {
      "block": "0e777320-de37-4dca-a077-51fbf10a6565",
      "port": "in"
    }
  }
]
},
"deps": {
  "logic.not": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": ""
          },
          "position": {
            "x": 64,
            "y": 144
          }
        },
        {
          "id": "664caf9e-5f40-4df4-800a-b626af702e62",
          "type": "basic.output",
          "data": {
            "label": ""
          },
          "position": {
            "x": 752,
            "y": 144
          }
        }
      ],
    }
  },
  {

```

```
    "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
    "type": "basic.code",
    "data": {
      "code": "// NOT logic gate\n\nassign c = ~ a;",
      "ports": {
        "in": [
          "a"
        ],
        "out": [
          "c"
        ]
      }
    },
    "position": {
      "x": 256,
      "y": 48
    }
  },
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ],
  "deps": {},
  "image": "resources/images/not.svg",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  },
  "cnot": {
    "image": "",
    "state": {
      "pan": {
        "x": 0,
        "y": 0
      }
    }
  }
}
```

```

    },
    "zoom": 1
  },
  "graph": {
    "blocks": [
      {
        "id": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
        "type": "not",
        "data": {},
        "position": {
          "x": 280,
          "y": 248
        }
      },
      {
        "id": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
        "type": "or",
        "data": {},
        "position": {
          "x": 464,
          "y": 136
        }
      },
      {
        "id": "55c6c056-3630-4482-ad47-f4d9ee83b835",
        "type": "basic.input",
        "data": {
          "label": "a"
        },
        "position": {
          "x": 88,
          "y": 248
        }
      },
      {
        "id": "c8c6eed3-548c-49c7-a162-282179d427b1",
        "type": "basic.output",
        "data": {
          "label": "b"
        },
        "position": {
          "x": 640,
          "y": 136
        }
      },
      {
        "id": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
        "type": "low",
        "data": {},
        "position": {
          "x": 280,
          "y": 120
        }
      }
    ],
    "wires": [
      {
        "source": {

```

```
      "block": "d2a2eac1-f8b0-4e5b-a693-626f6d14b8e5",
      "port": "19c8f68d-5022-487f-9ab0-f0a3cd58bead"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "55c6c056-3630-4482-ad47-f4d9ee83b835",
      "port": "out"
    },
    "target": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "18c2ebc7-5152-439c-9b3f-851c59bac834"
    }
  },
  {
    "source": {
      "block": "db6b84db-bc29-46d6-86a4-f48cc50c8076",
      "port": "664caf9e-5f40-4df4-800a-b626af702e62"
    },
    "target": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "97b51945-d716-4b6c-9db9-970d08541249"
    }
  },
  {
    "source": {
      "block": "ba7c5fb1-172d-4fa0-8a59-1905c4a71332",
      "port": "664caf9e-5f40-4df4-800a-b626af702e62"
    },
    "target": {
      "block": "c8c6eed3-548c-49c7-a162-282179d427b1",
      "port": "in"
    }
  }
]
},
"deps": {
  "or": {
    "graph": {
      "blocks": [
        {
          "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
          "type": "basic.input",
          "data": {
            "label": "x"
          },
          "position": {
            "x": 64,
            "y": 80
          }
        },
        {
          "id": "97b51945-d716-4b6c-9db9-970d08541249",
          "type": "basic.input",
```

```

        "data": {
            "label": "y"
        },
        "position": {
            "x": 64,
            "y": 208
        }
    },
    {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
            "label": "o"
        },
        "position": {
            "x": 752,
            "y": 144
        }
    },
    {
        "id": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "type": "basic.code",
        "data": {
            "code": "// OR logic gate\n\nassign c = a | b;",
            "ports": {
                "in": [
                    "a",
                    "b"
                ],
                "out": [
                    "c"
                ]
            }
        },
        "position": {
            "x": 256,
            "y": 48
        }
    }
],
"wires": [
    {
        "source": {
            "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
            "port": "out"
        },
        "target": {
            "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
            "port": "a"
        }
    },
    {
        "source": {
            "block": "97b51945-d716-4b6c-9db9-970d08541249",
            "port": "out"
        },
        "target": {
            "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",

```

```
        "port": "b"
      }
    },
    {
      "source": {
        "block": "00925b04-5004-4307-a737-fa4e97c8b6ab",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {
  "pan": {
    "x": 0,
    "y": 0
  },
  "zoom": 1
},
"not": {
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x"
        },
        "position": {
          "x": 64,
          "y": 144
        }
      },
      {
        "id": "664caf9e-5f40-4df4-800a-b626af702e62",
        "type": "basic.output",
        "data": {
          "label": "y"
        },
        "position": {
          "x": 752,
          "y": 144
        }
      }
    ],
    {
      "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "type": "basic.code",
      "data": {
        "code": "// NOT logic gate\n\nassign c = ! a;",
        "ports": {
          "in": [
            "a"
          ]
        }
      }
    }
  ]
}
```



```

        ],
        "out": [
            "c"
        ]
    },
    "position": {
        "x": 256,
        "y": 48
    }
},
],
"wires": [
    {
        "source": {
            "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
            "port": "out"
        },
        "target": {
            "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
            "port": "a"
        }
    },
    {
        "source": {
            "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
            "port": "c"
        },
        "target": {
            "block": "664caf9e-5f40-4df4-800a-b626af702e62",
            "port": "in"
        }
    }
],
},
"deps": {},
"image": "",
"state": {
    "pan": {
        "x": 0,
        "y": 0
    },
    "zoom": 1
},
},
"low": {
    "graph": {
        "blocks": [
            {
                "id": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
                "type": "basic.code",
                "data": {
                    "code": "// Bit 0\n\nassign v = 1'b0;",
                    "ports": {
                        "in": [],
                        "out": [
                            "v"
                        ]
                    }
                }
            }
        ]
    }
}

```

```
        },
        "position": {
          "x": 96,
          "y": 96
        }
      },
      {
        "id": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
        "type": "basic.output",
        "data": {
          "label": "o"
        },
        "position": {
          "x": 608,
          "y": 192
        }
      }
    ],
    "wires": [
      {
        "source": {
          "block": "b959fb96-ac67-4aea-90b3-ed35a4c17bf5",
          "port": "v"
        },
        "target": {
          "block": "19c8f68d-5022-487f-9ab0-f0a3cd58bead",
          "port": "in"
        }
      }
    ]
  },
  "deps": {},
  "image": "",
  "state": {
    "pan": {
      "x": 0,
      "y": 0
    },
    "zoom": 1
  }
}
},
"not": {
  "graph": {
    "blocks": [
      {
        "id": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "type": "basic.input",
        "data": {
          "label": "x"
        },
        "position": {
          "x": 64,
          "y": 144
        }
      }
    ]
  },
}
```

```

    {
      "id": "664caf9e-5f40-4df4-800a-b626af702e62",
      "type": "basic.output",
      "data": {
        "label": "Y"
      },
      "position": {
        "x": 752,
        "y": 144
      }
    },
    {
      "id": "5365ed8c-e5db-4445-938f-8d689830ea5c",
      "type": "basic.code",
      "data": {
        "code": "// NOT logic gate\n\nassign c = ! a;",
        "ports": {
          "in": [
            "a"
          ],
          "out": [
            "c"
          ]
        }
      },
      "position": {
        "x": 256,
        "y": 48
      }
    }
  ],
  "wires": [
    {
      "source": {
        "block": "18c2ebc7-5152-439c-9b3f-851c59bac834",
        "port": "out"
      },
      "target": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "a"
      }
    },
    {
      "source": {
        "block": "5365ed8c-e5db-4445-938f-8d689830ea5c",
        "port": "c"
      },
      "target": {
        "block": "664caf9e-5f40-4df4-800a-b626af702e62",
        "port": "in"
      }
    }
  ]
},
"deps": {},
"image": "",
"state": {
  "pan": {

```

```
        "x": 0,  
        "y": 0  
    },  
    "zoom": 1  
}  
}  
}  
}
```

Generates

```
module main (input v121a14,  
             output v31c150,  
             output v71e6a9);  
  
    wire w0;  
    wire w1;  
    wire w2;  
    wire w3;  
    assign w0 = v121a14;  
    assign w1 = v121a14;  
    assign v31c150 = w2;  
    assign v71e6a9 = w3;  
    assign w1 = w0;  
    main_cnot vc6f497 (  
        .vald1bb(w1),  
        .vecf2e3(w3)  
    );  
    main_not v59fef8 (  
        .v0e28cb(w0),  
        .vcbab45(w2)  
    );  
endmodule  
  
module main_logic_not (input v0e28cb,  
                      output vcbab45);  
  
    wire w0;  
    wire w1;  
    assign w0 = v0e28cb;  
    assign vcbab45 = w1;  
    main_logic_not_basic_code_vd54ca1 vd54ca1 (  
        .a(w0),  
        .c(w1)  
    );  
endmodule  
  
module main_logic_not_basic_code_vd54ca1 (input a,  
                                         output c);  
  
    // NOT logic gate  
  
    assign c = ~ a;  
endmodule  
  
module main_cnot (input vald1bb,  
                 output vecf2e3);  
  
    wire w0;  
    wire w1;  
    wire w2;  
    wire w3;
```

```

assign w1 = vald1bb;
assign vecf2e3 = w3;
main_cnot_not va44cd3 (
    .v0e28cb(w1),
    .vcbab45(w2)
);
main_cnot_or v0b7a71 (
    .v0e28cb(w0),
    .v3ca442(w2),
    .vcbab45(w3)
);
main_cnot_low v2d7478 (
    .v608bd9(w0)
);
endmodule

module main_cnot_or (input v0e28cb,
                    input v3ca442,
                    output vcbab45);

    wire w0;
    wire w1;
    wire w2;
    assign w0 = v0e28cb;
    assign w1 = v3ca442;
    assign vcbab45 = w2;
    main_cnot_or_basic_code_vf4938a vf4938a (
        .a(w0),
        .b(w1),
        .c(w2)
    );
endmodule

module main_cnot_or_basic_code_vf4938a (input a,
                                       input b,
                                       output c);

    // OR logic gate

    assign c = a | b;
endmodule

module main_cnot_not (input v0e28cb,
                     output vcbab45);

    wire w0;
    wire w1;
    assign w0 = v0e28cb;
    assign vcbab45 = w1;
    main_cnot_not_basic_code_vd54ca1 vd54ca1 (
        .a(w0),
        .c(w1)
    );
endmodule

module main_cnot_not_basic_code_vd54ca1 (input a,
                                       output c);

    // NOT logic gate

    assign c = ! a;
endmodule

```

```
module main_cnot_low (output v608bd9);
  wire w0;
  assign v608bd9 = w0;
  main_cnot_low_basic_code_v68c173 v68c173 (
    .v(w0)
  );
endmodule

module main_cnot_low_basic_code_v68c173 (output v);
  // Bit 0

  assign v = 1'b0;
endmodule

module main_not (input v0e28cb,
                  output vcbab45);

  wire w0;
  wire w1;
  assign w0 = v0e28cb;
  assign vcbab45 = w1;
  main_not_basic_code_vd54ca1 vd54ca1 (
    .a(w0),
    .c(w1)
  );
endmodule

module main_not_basic_code_vd54ca1 (input a,
                                   output c);

  // NOT logic gate

  assign c = ! a;
endmodule
```

```
set_io v121a14 10
set_io v31c150 95
set_io v71e6a9 96
```