



[Home](#) » [Tutorials](#) » [Alchitry Constraints Reference](#)

# Alchitry Constraints Reference

2025-02-04 · 8 min · Justin Rajewski | [Suggest Changes](#)

► [Table of Contents](#)

This page is a reference for Alchitry Constraints V2.

## Alchitry Constraints File Contents

An Alchitry Constraints File (.acf extension) contains a list of [pin constraints](#).

### Comments

Comments take the same form as in [Lucid](#).

Use `//` for a single line comment and `/* */` for a multi-line comment.

### Pin Constraints

The main purpose of a pin constraint is to tell the tools which physical pin on the FPGA corresponds to a port of your design's top-level module.



A pin constraint takes the following form.

```
pin port_name PIN_NAME OPTIONAL_ATTRIBUTE_LIST
```

Here, `port_name` corresponds to a port of your top-level module. If the port is an array, then it should also include **array selectors** to select only a single bit.

The `PIN_NAME` is the name of the physical pin to connect to `port_name`. This name is the Alchitry pin name and not the pin name of the FPGA. See the **pinout section** for the various pin names.

Finally, the `OPTIONAL_ATTRIBUTE_LIST` is a list of **attributes** that apply to only this pin.

## Attribute Block

Specifying attributes for every single pin constraint would often become very tedious as many pins will share the same attribute values. This is where attribute blocks come into play. They take a very similar form to the **connection blocks** in Lucid.

```
ATTRIBUTES_LIST {  
    ...  
}
```

The `ATTRIBUTES_LIST` is a comma-separated list of **attributes**.

The attributes in the list are applied to every **pin constraint** inside the brackets.

Attribute blocks can also be nested to allow for fine-grained assignments.

For example, the Hd constraints uses two `STANDARD` blocks along with a single `PINOUT` block.



```
pin hdmi_data_1_n[1] A58
pin hdmi_data_1_p[2] A66
pin hdmi_data_1_n[2] A64

pin hdmi_clk_2_p A47
pin hdmi_clk_2_n A45
pin hdmi_data_2_p[0] A53
pin hdmi_data_2_n[0] A51
pin hdmi_data_2_p[1] A59
pin hdmi_data_2_n[1] A57
pin hdmi_data_2_p[2] A65
pin hdmi_data_2_n[2] A63
}
```

## Attributes

An attribute takes the following form.

```
ATTRIBUTE_NAME(attribute_value)
```

This assigns the value, `attribute_value`, to the attribute named `ATTRIBUTE_NAME`.

There are a handful of different attributes currently supported.

### PINOUT

This attribute is either `V2`, `V1`, or not specified.

When it is specified, it says that the pinout used for that pin should use that version's pin mapping.

When it isn't specified, the project's board's version is used. For example, an Au V1 will use `V1`.



Most of the time, if specified, it should match the board's version. However, if a **V1** pinout is specified when using a **V2** board, then the tools assume you are using a **V2->V1 adapter**. In this case, the tools will translate the pinout to the adapter's pinout.

If a **V2** version is used with a **V1** board an error is thrown.

## STANDARD

This specifies the IO standard to apply to the pin.

Most of the time, it will be set to **LVCMS33**. This is the basic 3.3V IO standard. It is also the only standard supported by the **Cu**.

The **Au** and **Pt** both support the same IO standards. There are too many standards to list here, but page 98 of **UG471** has a nice table summarizing the requirements for each standard.

Each IO standard has various requirements that must be met in order for it to be used. The biggest one is the **Vcco** voltage. All the pins on the Cu and most of the Au and Pt have **Vcco** fixed at 3.3V.

The Au and Pt have one group of pins that can have **Vcco** switched between 3.3V, 2.5V, and 1.8V. This drastically opens up the IO standards available on these pins.

For example, by setting **Vcco** to 2.5V, the tri-voltage pins can be used as **LVDS\_25** outputs or enable termination when used as inputs. Any pins on the Au/Pt can be used as **LVDS\_25** inputs as long as they don't use internal termination.

The other common requirement is the **Vref** voltage. Many standards don't require a specific **Vref** and for the ones that do, an internal reference can typically be used. However, only one **Vref** can be set for a bank of pins. Here, *bank* refers not to the connector's bank but rather the FPGA's internal bank.

In the **Au's schematic**, the bank number is the first number of the FPGA's IO signal names. On page 2, you can see the header's pinout as well as the associated pin banks for each signal. The banks are numbers like **14**, **34**, and **35**.



All the IO standards used in a bank must have at most one **Vref** requirement.

## FREQUENCY

This attribute marks the pin as a clock input with the given frequency.

The frequency value takes the form of a number followed by a frequency unit. The unit can be **Hz** , **KHz** , **MHz** , or **GHz** . For example, **100MHz** .

Clock inputs should also be typically placed on a clock capable input pin. Xilinx (for the Au/Pt) calls these **MRCC** or **SRCC** pins. Lattice (for the Cu) calls these **GBIN** . You can look at each board's schematic to see what pins are clock capable inputs.

For V2 boards, pins **A41** , **A42** , **A47** , **A48** , **B41** , **B42** , and **B47** are clock capable inputs.

The Au also includes **B48** .

The Pt also includes **B24** , **B29** , **B30** , **B48** , **C41** , **C42** , **C47** , **C48** , **D41** , **D42** , **D47** , and **D48** .

## PULL

This attribute is used to enable a pin's pull-up or pull-down resistors.

For the Cu, **UP** is the only valid value.

For the Au/Pt, **UP** , **DOWN** , or **KEEP** are valid values.

**UP** enables a weak pull-up. **DOWN** enables a weak pull-down. **KEEP** enables a weak pull to the same value the pin currently is (0 -> pull-down, 1 -> pull-up).

## DRIVE

The attribute specifies the drive strength of the pin in mA.

It is not supported by the Cu.



It is only valid when paired with specific **IO standards**. For the valid values of each standard, see table 1-56 on page 101 of **UG471**.

For **LVCMS33** on the Au/Pt, these are **4** , **8** , **12** , or **16** .

This number specifies the drive strength in mA and defaults to **12** .

## SLEW

This attribute specifies the slew rate of the pin and is only supported by the Au/Pt, not the Cu.

The slew rate is how fast the edge transitions.

It is only valid when paired with specific **IO standards**. For the valid values of each standard, see table 1-56 on page 101 of **UG471**.

For standards that support a slew rate, the valid values are **FAST** or **SLOW** with **SLOW** being the default.

Using a **FAST** slew rate may help with high speed signals but may also cause more power consumption and noise if not used carefully.

## Pinouts

The available pins vary depending on your board, but they follow a standard format.

The pins broken out on the connectors are named by the bank letter followed by the pin number. For example, pin 2 on bank A is **A2** .

The V2 boards have two banks on top, A and B. The **Pt** has two more on the bottom, C and D.

The V1 boards have four banks, A, B, C, and D on top.



Check the schematic for the **Cu V1**, **Au V1**, **Cu V2** or **Au V2** for what pins are populated on the connectors.

In addition to the pins on the connectors, there are some special internal pin names.

All boards have the pins **LED0 - LED7** , **RESET** , **CLOCK** , **USB\_RX** , and **USB\_TX** .

The Cu (V1 and V2) also have **SPI\_MOSI** , **SPI\_MISO** , **SPI\_SCK** , and **SPI\_SS** that connect to the configuration flash.

The Au (V1, V1+, and V2) and Pt also have **SPI\_D0 - SPI\_D3** , **SPI\_SCK** , and **SPI\_SS** that connect to the configuration flash. There is also **VP** and **VN** which are special analog inputs.

Finally, there are **DDR\_DQ0 - DDR\_DQ15** , **DDR\_DQS0\_P** , **DDR\_DQS0\_N** , **DDR\_DQS1\_P** , **DDR\_DQS1\_N** , **DDR\_DM0** , **DDR\_DM1** , **DDR\_ODT** , **DDR\_RESET** , **DDR\_BA0 - DDR\_BA2** , **DDR\_CK\_P** , **DDR\_CK\_N** , **DDR\_CKE** , **DDR\_CS** , **DDR\_RAS** , **DDR\_CAS** , **DDR\_WE** , **DDR\_A0 - DDR\_A13** for interfacing with the DDR3. You generally shouldn't specify these directly as the **MIG core** does it for you.

[« PREV](#)[NEXT »](#)[Lucid Reference](#)[Your First FPGA Project](#)

© 2025 [Alchitry](#)

