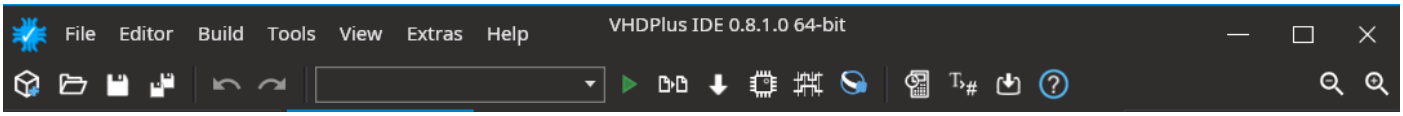
VHDP IDE Overview

Inhalt

Layout Overview

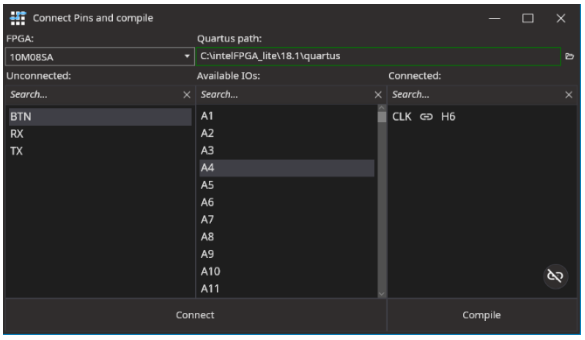
Toolbar

File | Test & Program | Tools & Help | |Editor|



1. New Project
2. Open Project
3. Save current file
4. Save all open files

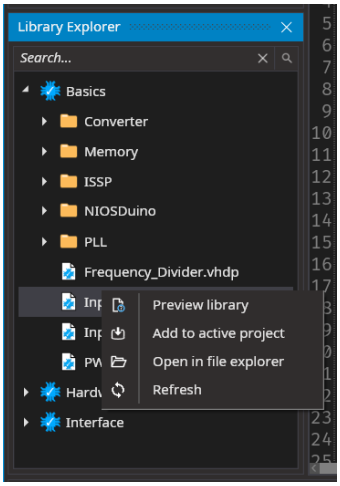
1. Undo/Redo for current file
2. Select the Project to compile
3. Converts VHDP files to VHDL files, allows to connect the I/Os and starts compiling.



1. Select the used FPGA
2. Select a signal, search for the I/O to connect and press Enter
3. Make sure that the path of the Quartus folder is correct
4. Press Compile to start compiling

4. Converts VHDP files to VHDL files.
5. Opens the programmer window to program the FPGA with the created program files.
6. Opens the NIOS II programming environment (https://vhdplus.com/docs/guide_nios2/)
7. Opens the window for In-System Sources and Probes debugging (First add the ISSP library to your project and add a NewComponent in the file to debug)
8. Opens Quartus with the current project

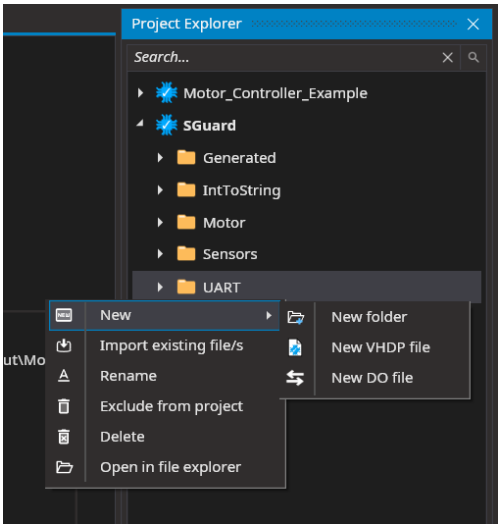
Library Explorer



Libraries:

1. Right click a file or folder you want to use
2. Select “Add to project”
3. Read the instruction in the .vhdp file to use it

Project Explorer and Project Overview



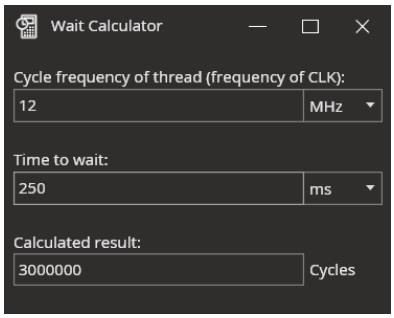
Right click a folder

1. New Folder, VHDP or Simulation file
2. Import VHDP, DO, VHDL, QSYS, QIP or other files (VHDL files can be used, but are not converted to VHDP files)
3. Rename folder
4. Exclude folder form the project
5. Delete the folder with content
6. Open the folder in the file explorer

- More options if clicked on File or Project
- Drang & Drop to move a file into a folder

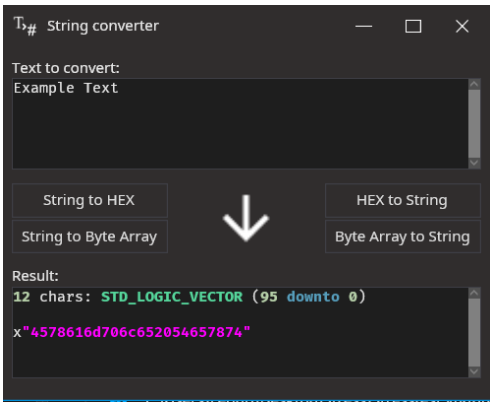
Tools

Wait Calculator



1. Enter the frequency of the external oscillator that is connected to CLK
2. Enter the time you want to wait
3. Copy the number of cycles to wait
4. Write e.g. “Wait(6000000);”

String Converter



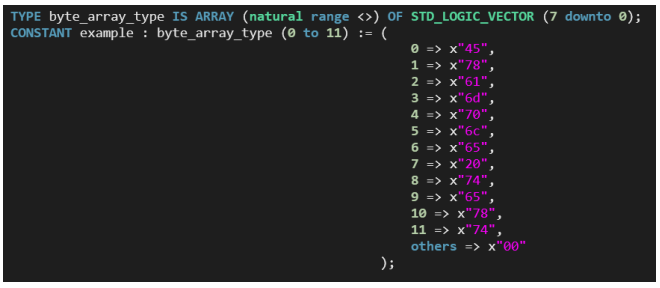
1. Enter the text to convert
2. Press “String to HEX” to save the text in one STD_LOGIC_VECTOR or “String to Byte Array” to save the text in an array of bytes (STD_LOGIC_VECTOR(7 downto 0))

String to Hex:

3.1. Copy the generated HEX value

3.2. Set the value like this:
my_vector(95 downto 0) <= x"4578616d706c652074657874";

3.3. If you have to set the length:
my_bytes <= 12;



```
TYPE byte_array_type IS ARRAY (natural range <>) OF STD_LOGIC_VECTOR (7 downto 0);
CONSTANT example : byte_array_type (0 to 11) := (
                                                  0 => x"45",
                                                  1 => x"78",
                                                  2 => x"61",
                                                  3 => x"6d",
                                                  4 => x"70",
                                                  5 => x"6c",
                                                  6 => x"65",
                                                  7 => x"20",
                                                  8 => x"74",
                                                  9 => x"65",
                                                  10 => x"78",
                                                  11 => x"74",
                                                  others => x"00"
                                                );
```

String to Byte Array:

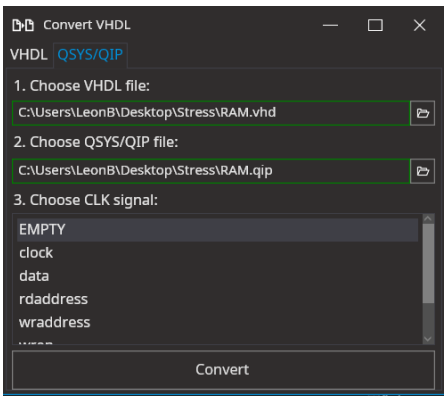
4.1. Copy the generated text (CTRL + A)

4.2. Save it e.g. with a constant or set a signal to to the bytes

Import VHDL/QSYS/QIP

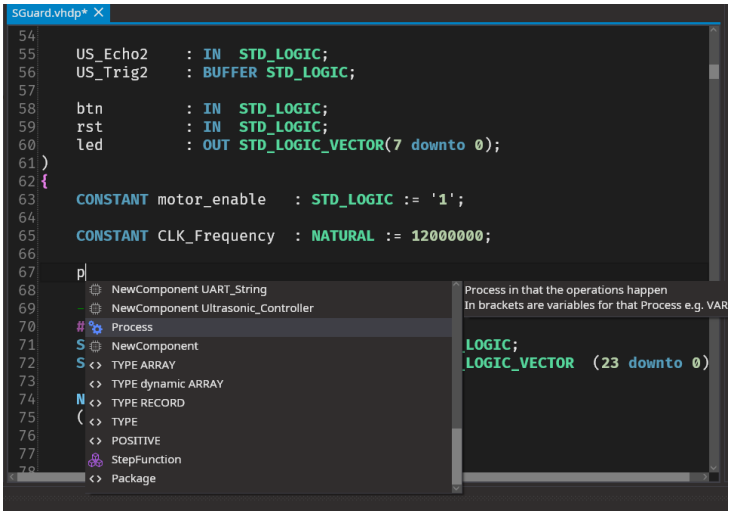
1. Import a VHDL file to use it like a VHDP file
2. Import a QSYS or QIP file that is generated by Quartus.

If one input should connect to the main CLK signal, select a signal or select the first “EMPTY” line to keep all inputs.



Code Editor

Code Completion



When you write your VHDP code, the VHDP IDE tries to give suggestions of possible things to add at this position in your code. For example if you want to set a value, it wouldn't show outputs, because they can't be read. But don't worry if the thing you want to write isn't in the list. At the moment the completion can be wrong sometimes.

Press enter to insert the code part or write along to ignore the completion. The completion also includes components in your project, so e.g. if you add "NewComponent Frequency_Divider", the NewComponent with all I/Os to connect will be inserted.

Autocorrection

The syntax of VHDL for the signals and the operators can be confusing for people that worked with other programming languages. So we included an autocorrection to make the shift easier. Here a list of things that will be replaced if used:

Operators:

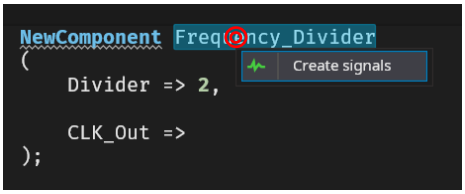
String to replace	Correct string	Example
= or :=/<= if used wrong	:= for variables and <= for signals	Example_var = … → Example_var :=
!=	/= (unequal)	If (a != b) → If (a /= b)
==	= (equal)	If (a == b) → If (a = b)
&&	AND	If (a && b) → If (a AND b)
||	OR	If (a || b) → If (a OR b)
i ++	i := i + 1	
i +=	i := i +	
i -=	i := i -	
100s/ms/ns	Cycles at 12MHz	Wait(100ms); → Wait(1200000);

Signal declaration:

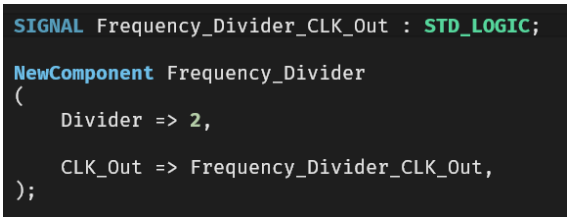
Signal: INTEGER i = 0; → SIGNAL i : INTEGER := 0;
Variable: NATURAL range 0 to 255 count = 0; → VARIABLE count : NATURAL range 0 to 255 := 0;
I/Os: OUT STD_LOGIC_VECTOR (7 downto 0) LED = (others => '0'); →
LED : OUT STD_LOGIC_VECTOR(7 downto 0) := (others => '0');
Arrays: INTEGER range 0 to 255 [5] [7] arr = (others => (others => 0)); →
TYPE arr_type IS ARRAY (0 to 4, 0 to 6) OF INTEGER range 0 to 255;
SIGNAL arr : arr_type := (others => (others => 0));

The text will be replaced with pressing the last char of the string to replace. If you paste a text, you e.g. have to delete and write the ';' again to replace the string.

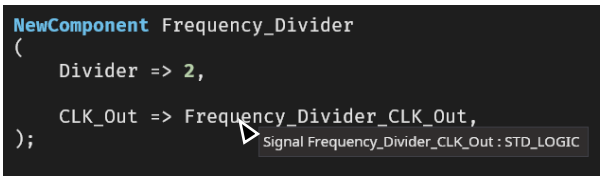
Signal Creation



Right click the name of the component and select “Create signals”. This will create a signal for all I/Os and will connect them.



Hover



If you want to know the type of a signal or information on some other code part, hover above the name and the information will be shown.

Errors and Warnings

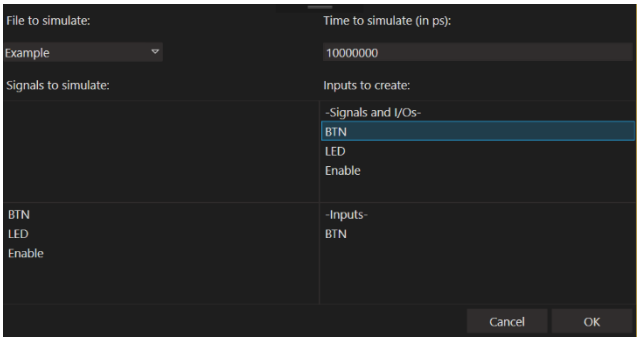
While writing your code, the code is analyzed, and some problems will be shown in the console. Warnings don’t have to be fixed but can lead to problems.

All other problems will be shown after clicking on create. You can double click an error and the file will open. If you double click again, the cursor will go to the line.

Important: Don’t fix the problem in the VHDL code. In this example the output “i” was set by more than one Process, so go back to the VHDP file where “i” is used and use a different signal for the second Process or combine the code in one Process.

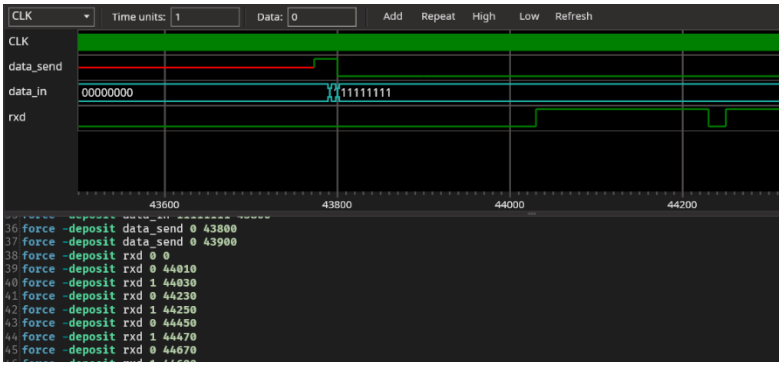
Simulation Editor (incomplete)

Simulation Assistant



First create a new .do file. The assistant will open, and you first have to select the file to simulate. First you can select the signals and I/Os you want to see after simulation. After that you can select the inputs that you want to simulate and assign values to. Also the time to simulate is needed.

Simulation Editor



1. Select the input
2. Select the time units to set a value
3. Select the data to set the input to (or press High or Low for ‘1’ and ‘0’)
4. For CLK press Low, High and then repeat
5. Or press Add for different inputs that shouldn’t repeat