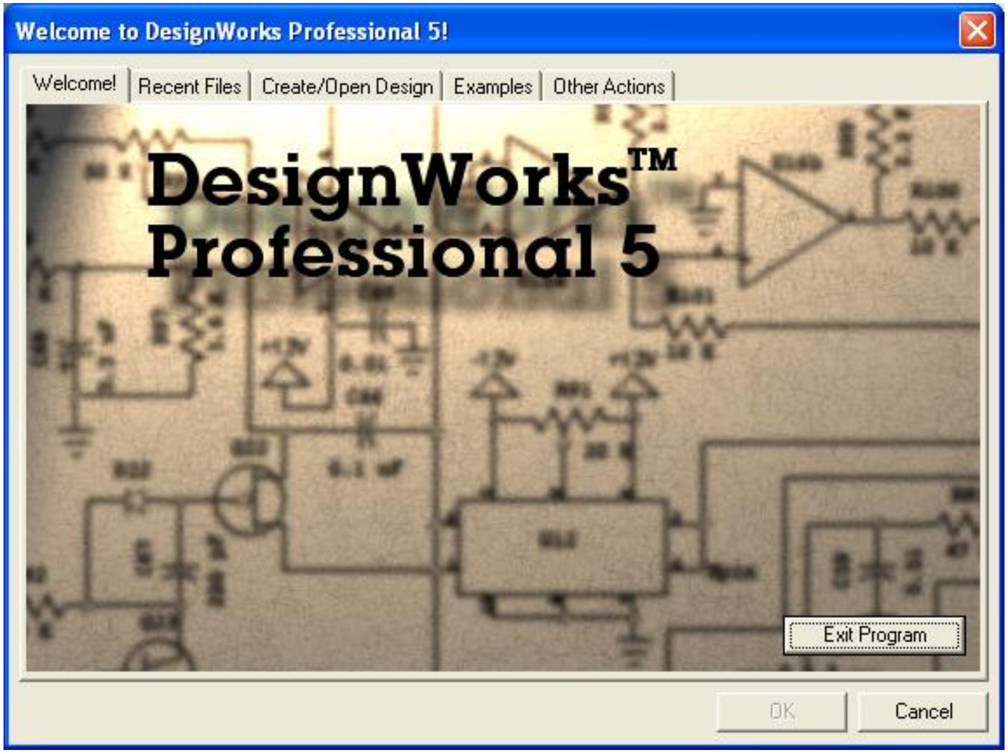
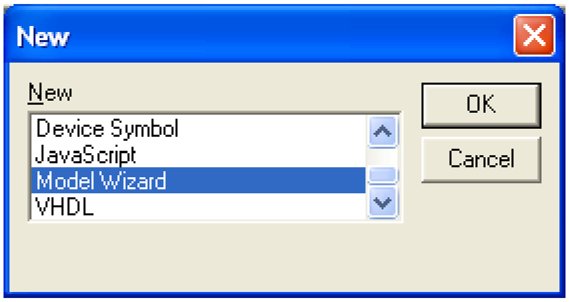
CSC 355 Lab 4 Pre-Lab Tutorial: Using DesignWorks[1]-VHDL

DesignWorks can be used to design circuits using VHDL. This tutorial gives instructions to create a specialized 2-input AND gate that has 1 of its inputs inverted. In particular it creates a gate that realizes the function: $F = \bar{A}B$.

1. Locate DesignWorks on the **`Start Menu`** and open **`DesignWorks`**. When the first screen appears as usual (see below), click on the "**`Cancel`**" button.



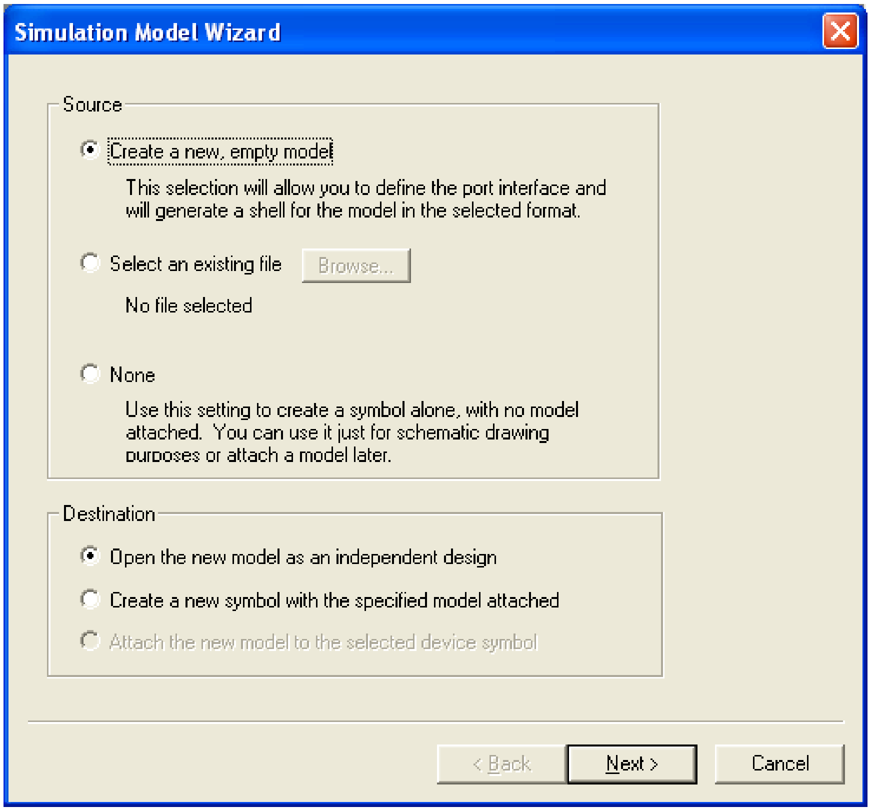
2. Go to the "**`File`**" menu and choose "**`New`**". In the list of available document types, select "**`Model Wizard`**", and click "**`OK`**".



[1] Credits to Victoria Li and the Design Works documentation

3. A dialogue window appears (as below) called *Simulation Model Wizard*. The Wizard helps to start a VHDL source code file and it is by no means necessary. (VHDL code can be typed in its entirety directly into a new file, but the Wizard makes it convenient to set up the Entity model.)

   In the "`Source`" selection, choose "`Create a new, empty model`". In the "`Destination`" selection, choose "Open the new model as an independent design". Click "`Next`".



Simulation Model Wizard

Source

● Create a new, empty model

This selection will allow you to define the port interface and will generate a shell for the model in the selected format.

○ Select an existing file

No file selected

○ None

Use this setting to create a symbol alone, with no model attached. You can use it just for schematic drawing purposes or attach a model later.

Destination

● Open the new model as an independent design

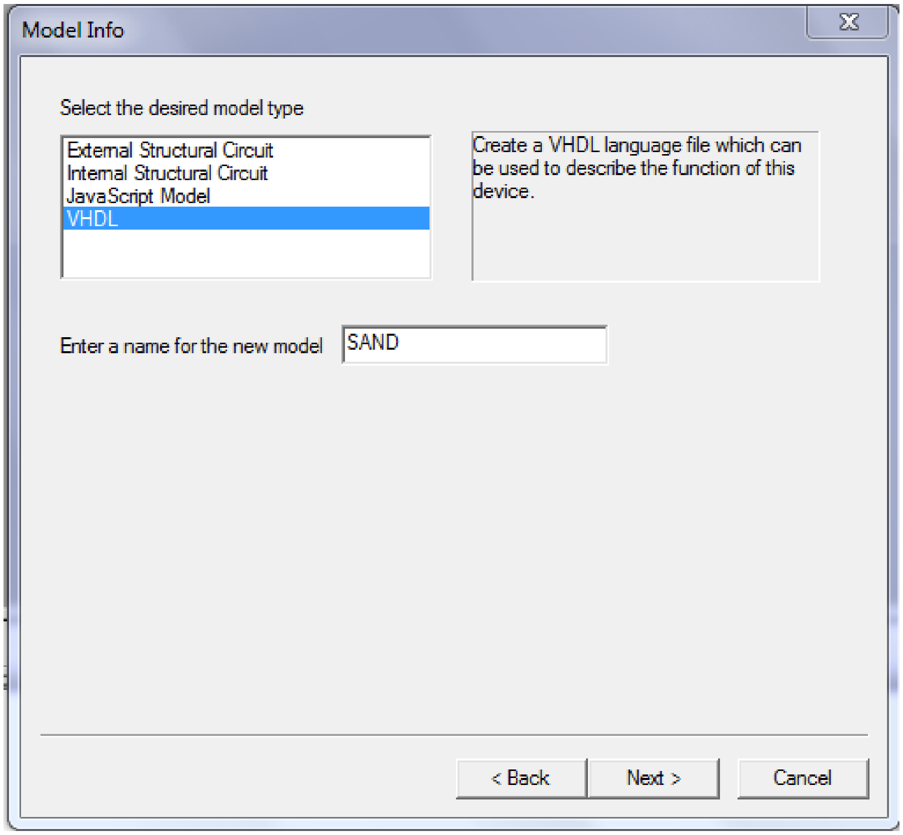
○ Create a new symbol with the specified model attached

○ Attach the new model to the selected device symbol

A new dialogue window called "`Model Info`" appears.

4. In "`Select the desired model type`" choose "`VHDL`". In this example, a specialized AND gate will be created: it realizes the function: $F = \bar{A}B$. Use the word "S**AND**" to represent the new model; thus, type "S**AND**" for the new model name. (Note that *AND* itself is a reserved word, therefore, it should not be used as a model name.)

   Click "`Next`".



Select the desired model type

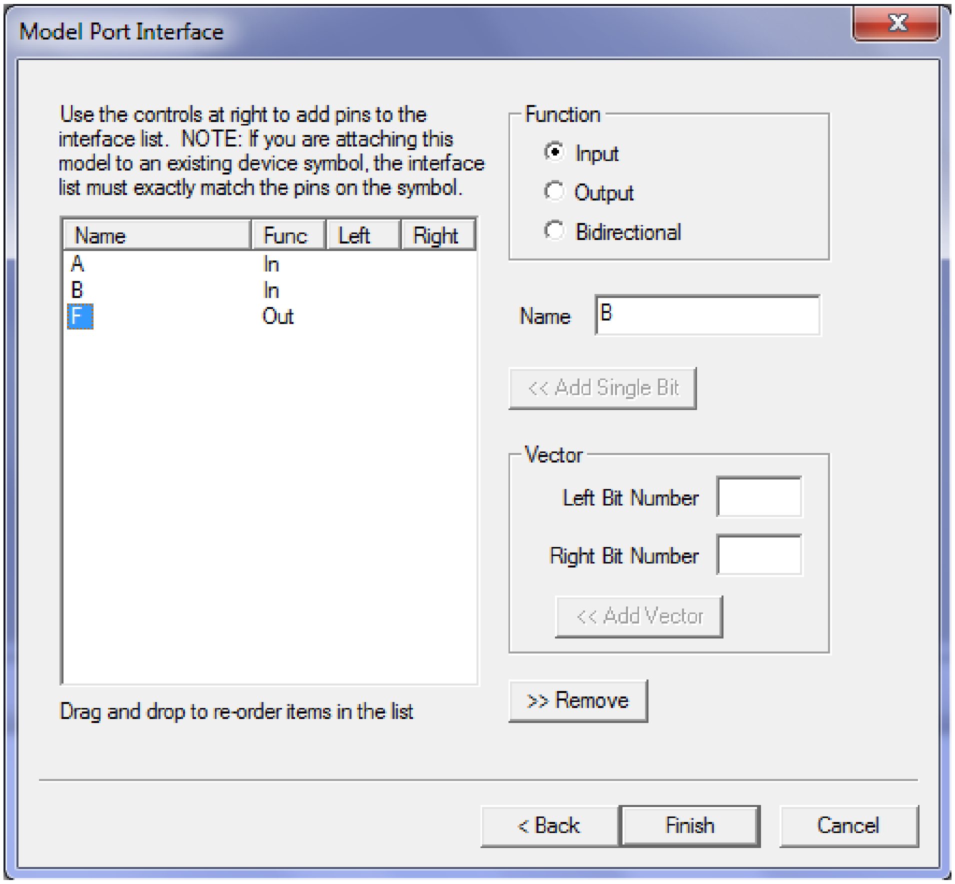
- External Structural Circuit
- Internal Structural Circuit
- JavaScript Model
- VHDL

Create a VHDL language file which can be used to describe the function of this device.

Enter a name for the new model: SAND

A new dialogue window called "**`Model Port Interface`**" appears. This is where the input and outputs for the entity interface are set through the dialogue window, avoiding having to type the actual VHDL code, which will be produced automatically.

5. The circuit to be designed is to implement $F = \bar{A}B$. One needs to enter **`X`** and **`Y`** as the inputs and **`F`** as the output. Type the name "**`X`**" in the small window for "**`Name`**", click the radio button for "**`input`**" above and then click on "**`Add Single Bit`**" button to add that port to the design. Repeat the process similarly for **`Y`** and **`F`**, changing the radio button selection to "**`Output`**" for **`F`**. At the end the screen should look like this:



Model Port Interface

Use the controls at right to add pins to the interface list. NOTE: If you are attaching this model to an existing device symbol, the interface list must exactly match the pins on the symbol.

Name	Func	Left	Right
A	In		
B	In		
F	Out		

Drag and drop to re-order items in the list

Function

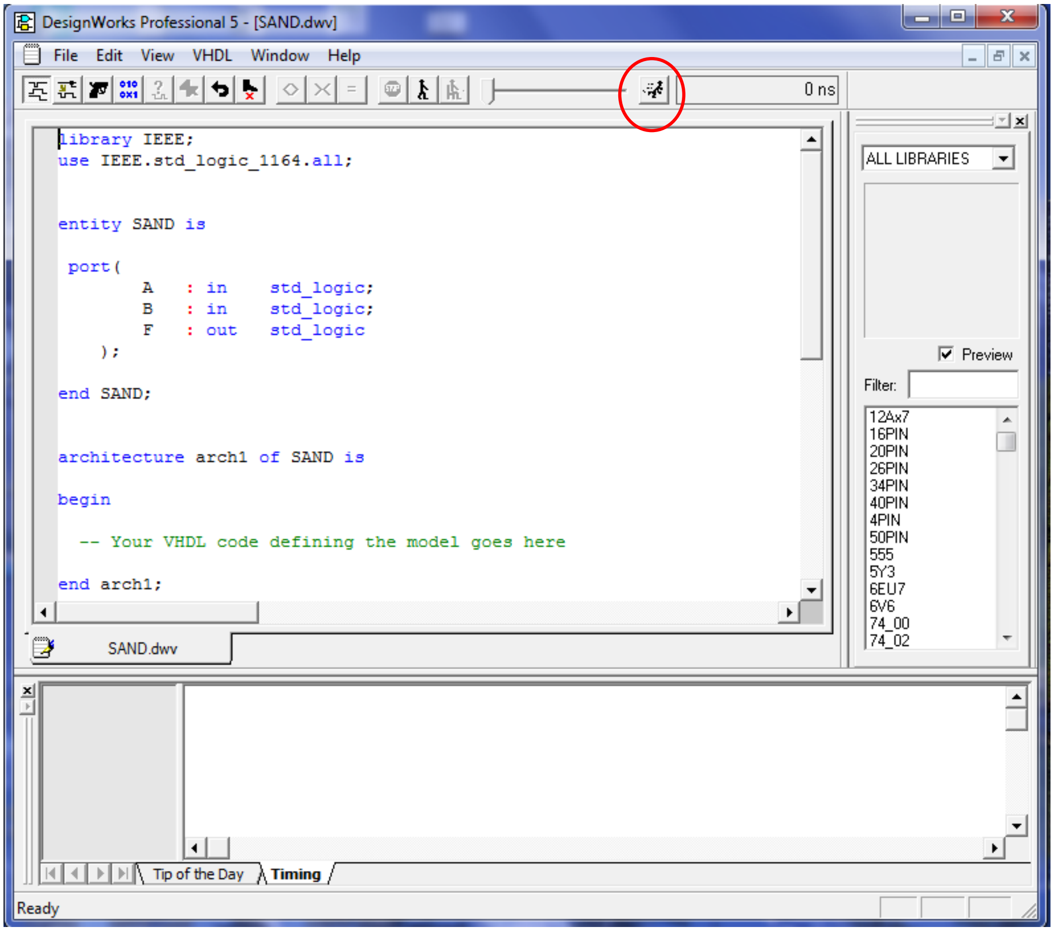
● Input

○ Output

○ Bidirectional

Name: B

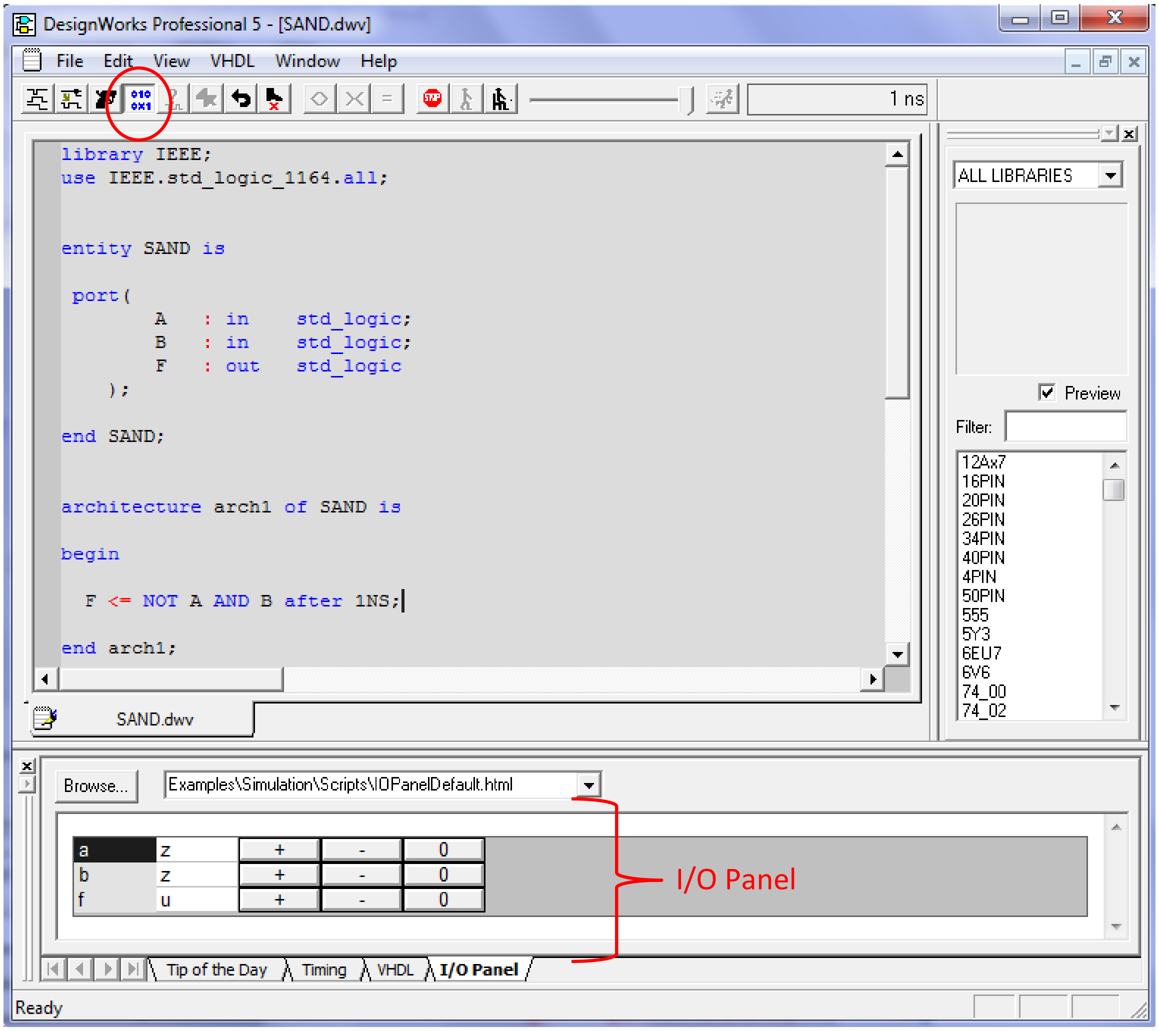
6. After clicking on "**Finish**", the VHDL code for the entity interface is generated and the source code file is opened automatically. The screen should look like the picture below here.



```
library IEEE;
use IEEE.std_logic_1164.all;


entity SAND is

 port(
        A   : in    std_logic;
        B   : in    std_logic;
        F   : out   std_logic
    );

end SAND;


architecture arch1 of SAND is

begin

  -- Your VHDL code defining the model goes here

end arch1;
```

Note how the **ENTITY** declaration has been filled in, the library references added, and a template for an **ARCHITECTURE** description started. The palette of built-in circuits available from the libraries is also shown on the right hand side, just as in the schematic entry design process.

7. Replace the green comment "**-- Your VHDL code**…" with the new code for the architecture. In this example, type "**F <= NOT A AND B after 1NS;**" Save the file as "S**AND.dwv**" (the default - **.vhd** also works).
8. Click the "**RUN**" button to start the simulator (top right in the toolbar, see it circled in the picture above). The VHDL source code turns gray. If there are no compilation errors, continue; else correct the syntax errors and repeat. Similarly you could click on the menu at "**VHDL → Compile**".
9. Click the "I/O Panel" button (top left). A new panel at the bottom of the screen is displayed. Similarly you could click on the menu at "**VHDL → Run Simulation**".



```
library IEEE;
use IEEE.std_logic_1164.all;


entity SAND is

 port(
        A   : in    std_logic;
        B   : in    std_logic;
        F   : out   std_logic
    );

end SAND;


architecture arch1 of SAND is

begin

  F <= NOT A AND B after 1NS;

end arch1;
```

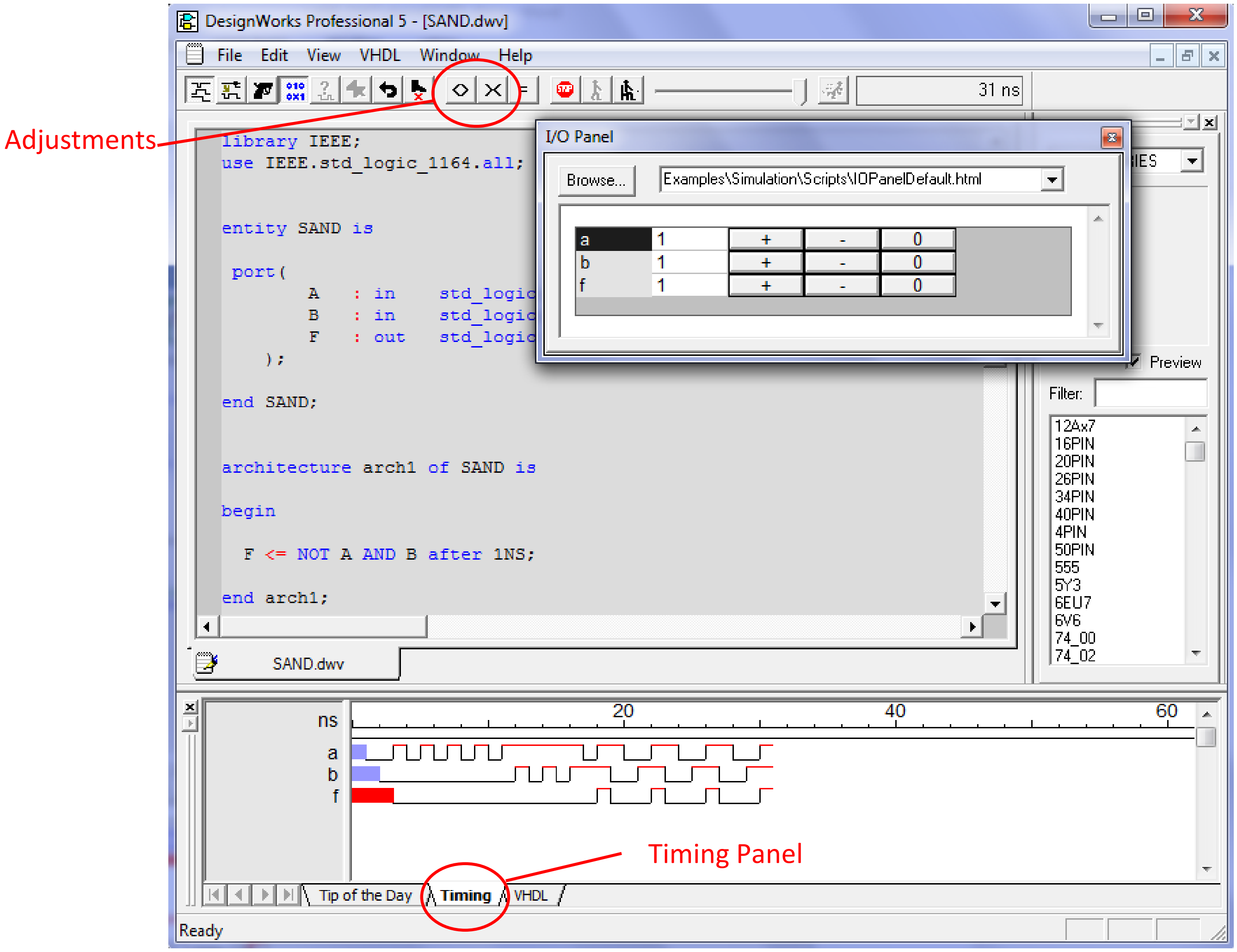
Examples\Simulation\Scripts\IOPanelDefault.html

a	z	+	-	0
b	z	+	-	0
f	u	+	-	0

The behaviour of the circuit is simulated by changing the input values corresponding to the signals available. The original state as shown is “`Z`” for inputs and “`U`” for the output. Change the input values by clicking in the little boxes of the I/O panel. The corresponding waveform is shown in the “`Timing`” panel at the bottom by selecting its corresponding tab (look at the bottom of the window).

Both panels do not show at the same time, so it is easier to disengage one of them form the bottom of the window and make it “float” elsewhere. To do this, right-click on the top of the I/O panel and select “`Float Current Tab`”. This will use the docking feature and make the panel an independent moving small window. By selecting “Dock” you can place the tab back with the others.

Use the “**<>**” and “**><**” tools from the toolbar to reshape the width of the timing panel so that you can look at the waveform in details more easily.



Adjustments

Timing Panel

```
library IEEE;
use IEEE.std_logic_1164.all;

entity SAND is

 port(
        A   : in    std_logic
        B   : in    std_logic
        F   : out   std_logic
    );

end SAND;

architecture arch1 of SAND is

begin

  F <= NOT A AND B after 1NS;

end arch1;
```

I/O Panel

Examples\Simulation\Scripts\IOPanelDefault.html

a	1	+	-	0
b	1	+	-	0
f	1	+	-	0