Hello Digital World

Test Bench for CCounterEvent

# Core description

The core ***CCounterEvent\_T*** is a counter controlled by the events at the inputs. There is a template parameter to define the type of the counter, for example if TYPE = ubyte, an 8 bits unsigned counter is created. The values at the inputs are not considered. Ex: If there is an event on ***iUp*** then the counter counts up. An event is a pulse signal for a single step, i.e., one FPGA clock cycle. There may be events at each step.

# Test bench

This test bench uses the ***Control Panel*** to test the core. The ***Control Panel*** is an automatically generated dialog box that allows manual entry of input values and events and observation of output values and events. The test bench also makes use of the ***Watch Window***.

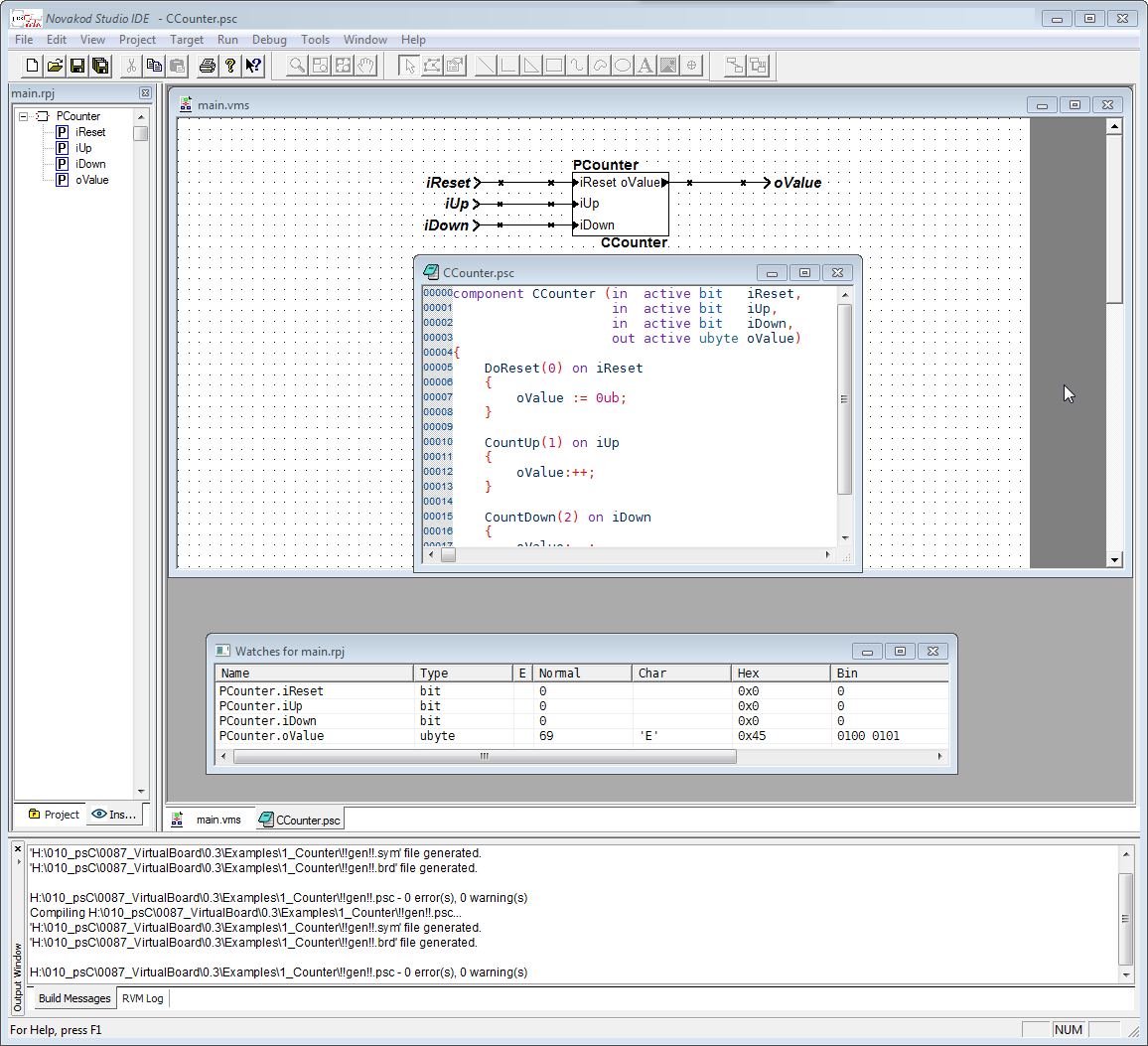
# Description

As shown below, the ***Control Panel*** communicates using the ***Novakod API*** with the simulator (RVM). With the **Control Panel**, you can send events to the counter and observe the output values. The test bench also shows how to use the ***Watch Window*** for debugging.

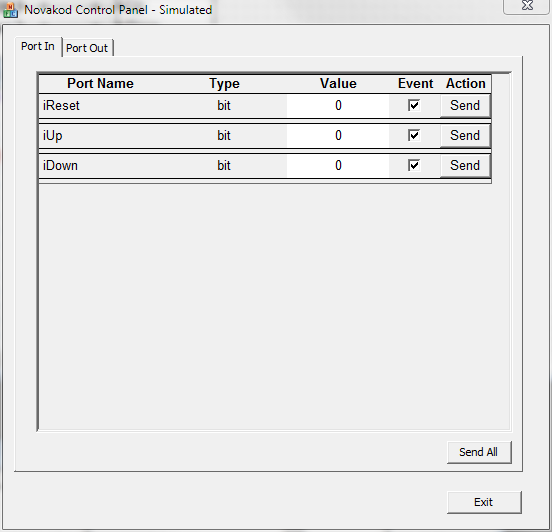
Novakod API

Virtual

Board



Debug Connection



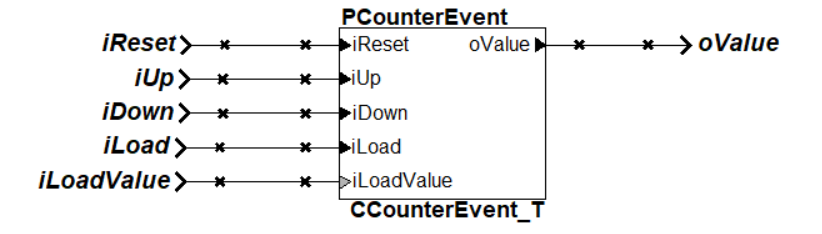
Control Panel

Novakod Studio

Simulator (RVM)

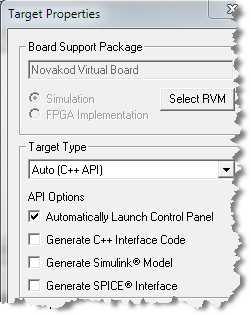
Watch Window

# The psC test program

The test program consists of a single component, the core to be tested, and IO connections. The counter operation is controlled by sending an event on one of the inputs.

# Compiling and running the test program

You will now compile the program, start execution and use the control panel to test it.

1. Double-click on ***Counter.rpj*** to start Novakod Studio.
2. Double-click on the ***main*** component to view the schematic.
3. Double-click on the ***CCounterEvent\_T*** component to view the counter code.
4. Under ***Targets*** open the ***Manual*** properties and verify that the target is the ***Novakod Virtual Board***. Also notice that the API Options ***Automatically Launch Control Panel*** is checked.

1. Now select the menu ***Run🡪 Start*** to begin simulation. Click OK to the build confirmation dialog. The ***Control Panel*** should appear. The simulation is running, waiting for events.

Table

Description automatically generated

1. Using the ***Control Panel***, check all event boxes and click **Send** to send the event for the desired operation.
2. Select the ***Port Out*** tab to verify the result on the **oValue** port.
3. Try all operations, click multiple times, and observe the result.
4. Write a value for **iLoadValue** click the corresponding ***Send***, then the ***Send*** for ***iLoad.***

Note the priority on inputs. Ex: A reset event will have priority on all other events.

# Using the watch windows with the control panel

The watch window allows you to observe the values and the events as they are generated during simulation. It displays the values into four formats: Normal, Character, Hexadecimal and Binary. You can use the watch window and the control panel.

1. In the ***Inspect*** tab, click the plus sign to open the port list for ***PCounterEvent***.
2. For each port, right-click and select ***Add to watches*** to insert the port in the watch window. You should see:

Graphical user interface, table

Description automatically generated

1. Now keep using the ***Control Panel*** to send events and observe the results in the watch window. You will not see the events since their duration is only one step and the RVM is running continuously.

# Virtual machine windows

Text

Description automatically generatedDuring the simulation, a ***Virtual Machine*** (RVM) runs. There are two windows associated with the RVM: The ***RVM console*** allows minimal interface with the ***Virtual Machine*** and the ***RVM Info*** shows the state and the current step.

Graphical user interface, text, application, chat or text message

Description automatically generated