Hello Digital World
# Test Bench for CCounterEvent

Core description

The core **CCounterEvent** is a counter controlled by the events at the inputs. 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. for one clock cycle. Events might be consecutive at each step.

Test bench

This test bench uses the **Control Panel** to test the core. It 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 value. The test bench also shows how to use the **Watch Window** for debugging.

Novakod Studio

Watch Window

Control Panel

Debug
Connection

Novakod API

Virtual
Board

Simulator (RVM)
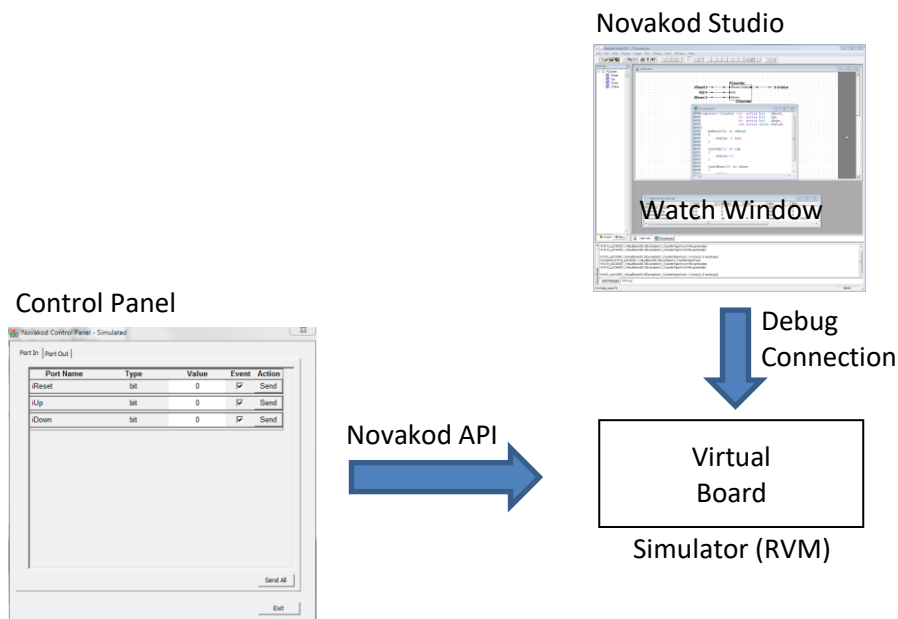
The psC test program

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.
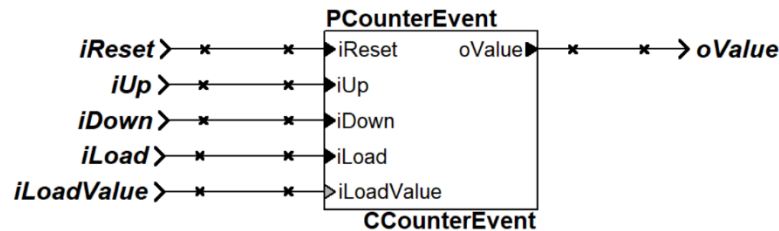

1) Double-click on **Counter.rpj** to start Novakod Studio.
2) Double-click on the **main** component to view the schematic.
3) Double-click on the **CCounterEvent** component to view the counter code.

Compiling and running the test

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

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.
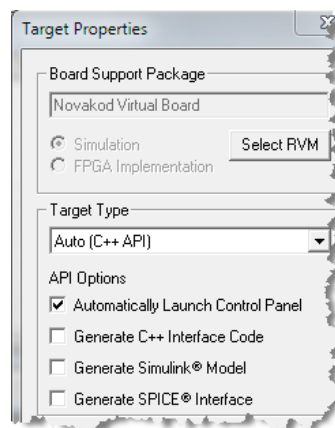
5) Now select the menu **Run → Start** to begin simulation. Click OK to the build confirmation dialog. The **Control Panel** should appear. The simulation is running, waiting for events.
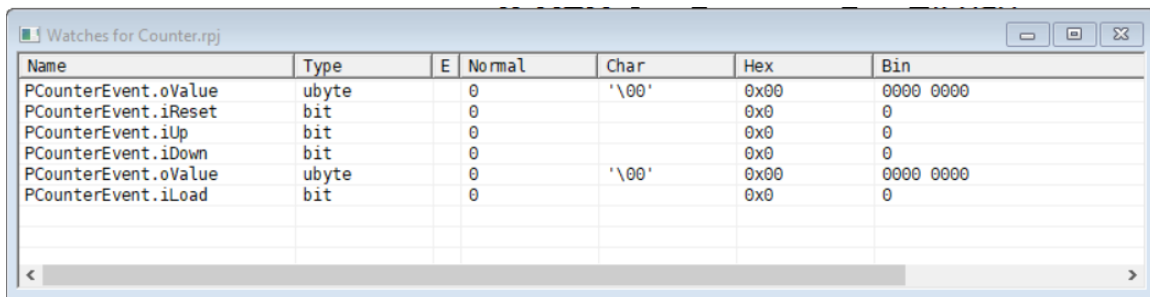
6) Using the *Control Panel*, check all event boxes and click **Send** to send the event for the desired operation.
7) Select the *Port Out* tab to verify the result on the **oValue** port.
8) Try all operations, click multiple times, and observe the result.
9) Write a value for **iLoadValue** click the corresponding *Send*, then the *Send* for *iLoad.*

Note that there is a 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*.
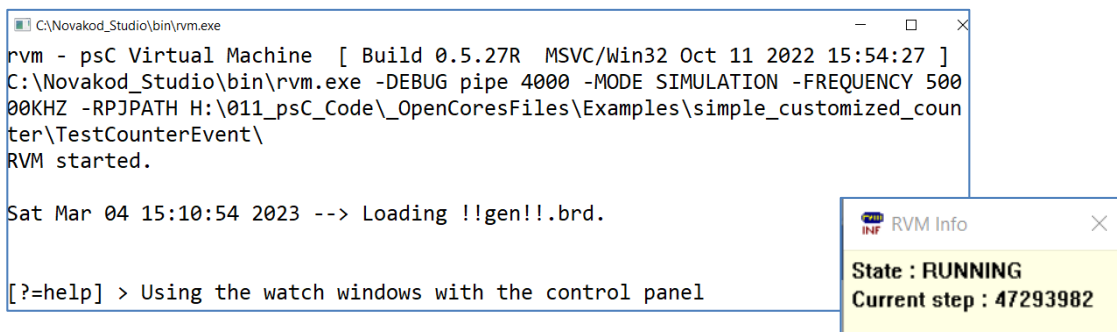2) For each port, right-click and select *Add to watches* to insert the port in the watch window. You should see:

Name	Type	E	Normal	Char	Hex	Bin
PCounterEvent.oValue	ubyte		0	'\00'	0x00	0000 0000
PCounterEvent.iReset	bit		0		0x0	0
PCounterEvent.iUp	bit		0		0x0	0
PCounterEvent.iDown	bit		0		0x0	0
PCounterEvent.oValue	ubyte		0	'\00'	0x00	0000 0000
PCounterEvent.iLoad	bit		0		0x0	0

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

Simulation windows

During 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.


```
C:\Novakod_Studio\bin\rvm.exe                                      —   □   ×
rvm - psC Virtual Machine  [ Build 0.5.27R  MSVC/Win32 Oct 11 2022 15:54:27 ]
C:\Novakod_Studio\bin\rvm.exe -DEBUG pipe 4000 -MODE SIMULATION -FREQUENCY 500
00KHZ -RPJPATH H:\011_psC_Code\_OpenCoresFiles\Examples\simple_customized_coun
ter\TestCounterEvent\
RVM started.

Sat Mar 04 15:10:54 2023 --> Loading !!gen!!.brd.


[?=help] > Using the watch windows with the control panel
```

RVM Info   ×

State : RUNNING
Current step : 47293982