Test Bench for CCounterLevel

Core description

The core ***CCounterLevel*** is a counter clocked at each step, i.e. at FPGA clock. The ***level*** of the inputs will control the counter operation, Ex: if ***iUp*** is 1 then the counter counts up.

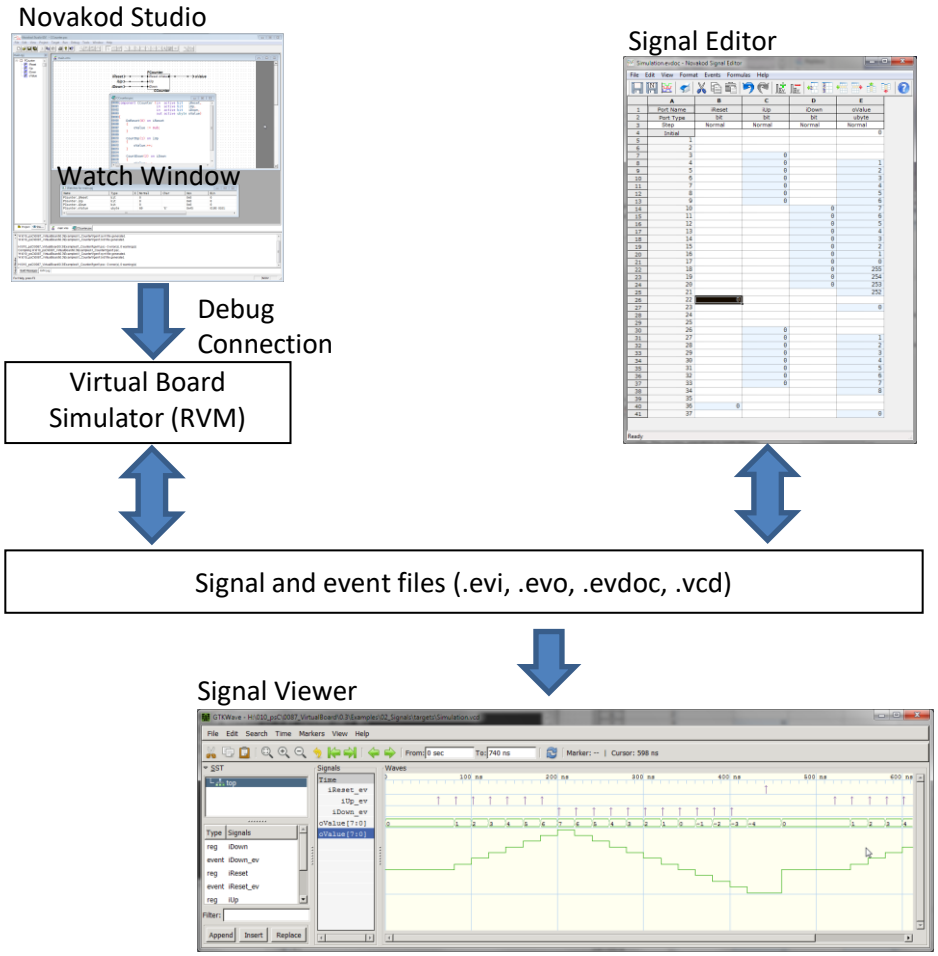
Test bench

This test bench uses the ***signal editor*** and the ***signal viewer*** to test the core. The test will be done in single step simulation. Follow the instructions to:

- Create signals in the “Signal Editor”
- Execute with the “Single Step” function (F8)
- Use the “Watch Window” in single step mode
- View the result in the “Signal Viewer”

Description

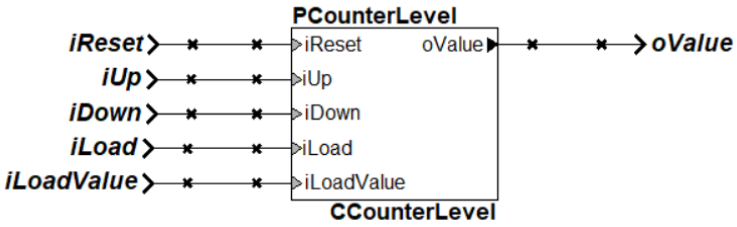
The ***Signal Editor*** is an Excel like spreadsheet to edit the input signals and view the output signals, including the events. The signal editor saves its information in a ***.evdoc*** file. During simulation, the signals are read from an input event file (***.evi***) and written to an output event file (***.evo***). After simulation, the ***Signal Editor*** is used to view the resulting output signals.



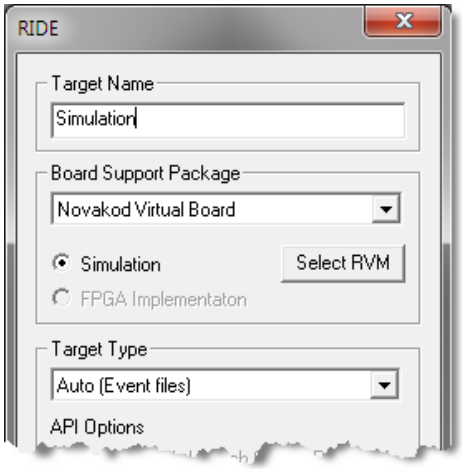
Then, the ***Signal Viewer*** will display the signal in the more common temporal view, with digital or analog representation. The signal viewer is ***GTK Wave*** and it uses ***.vcd*** (value change dump) files.

The psC test program

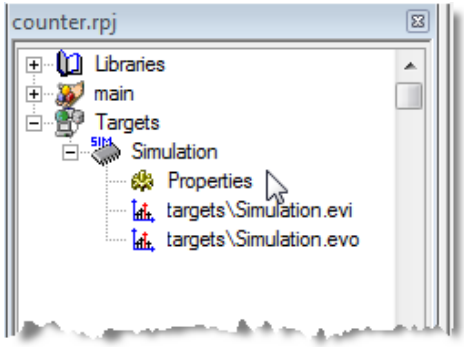
The test program consists of a single component, the core to be tested, and IO connections. The counter operation is controlled by the level of the inputs.



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.
4) Right-click on ***Targets*** to view the target. The ***Target Type*** is ***Auto (Event files)*** indicating that simulation will use ***.evi*** and ***.evo*** event files.



5) Expand ***Targets*** then ***Simulation***, you should see the two event files:

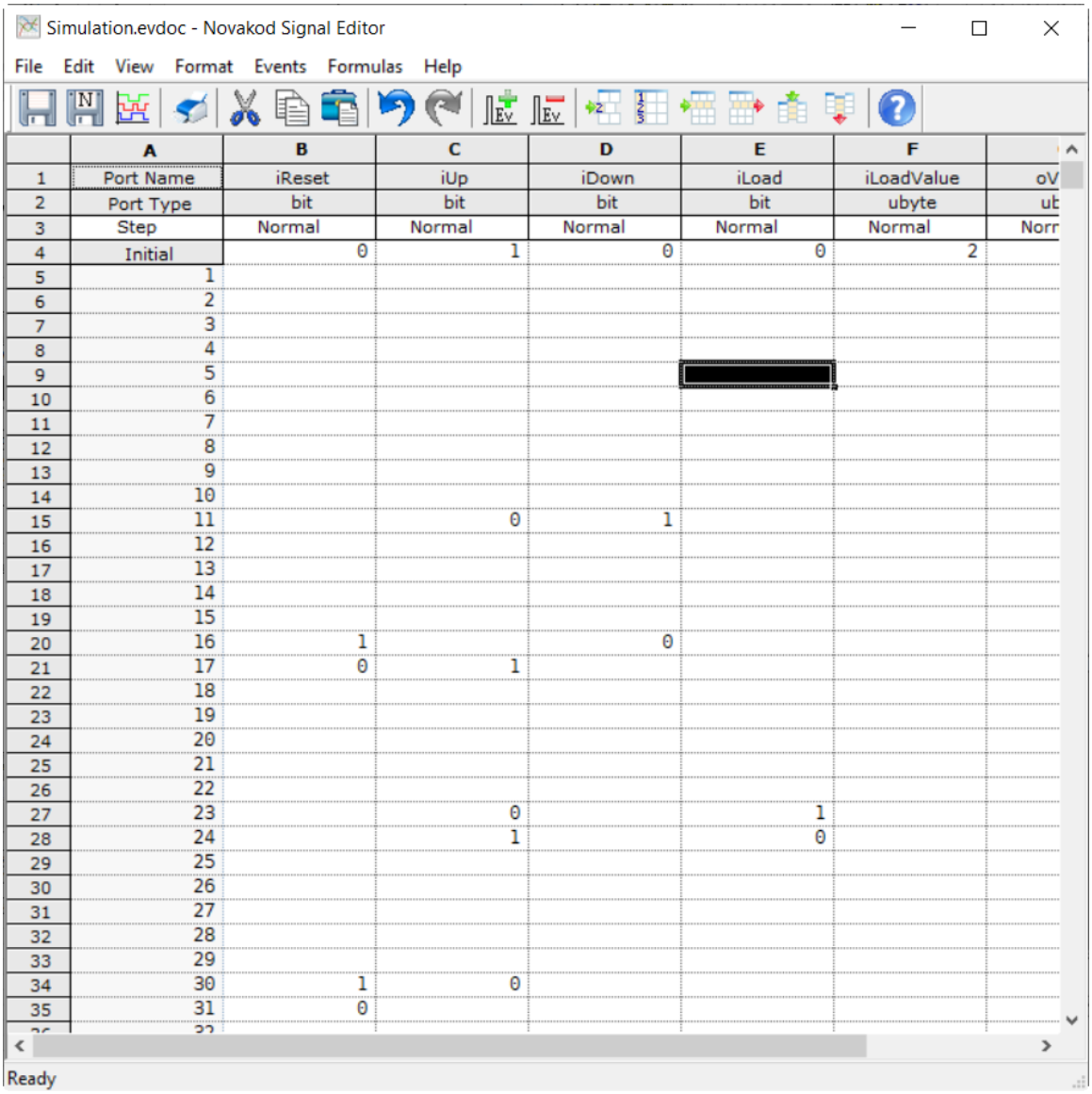


Viewing and editing input signals

The signals are already set, but you can change them as you wish.

6) Double-click on ***targets\Simulation.evi*** to start the ***Signal Editor***.

As you can see, the signal iUp is 1 from step 0 to step 10, therefore the counter should count up to 11. You can easily interpret the other signals to predict the counter value. If you change the signals, don't forget to save.

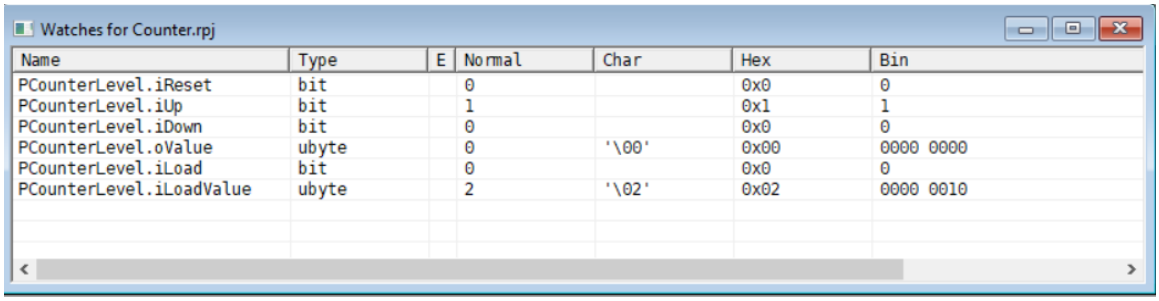


	A	B	C	D	E	F	
1	Port Name	iReset	iUp	iDown	iLoad	iLoadValue	oV
2	Port Type	bit	bit	bit	bit	ubyte	ub
3	Step	Normal	Normal	Normal	Normal	Normal	Norr
4	Initial	0	1	0	0	2	
5	1						
6	2						
7	3						
8	4						
9	5						
10	6						
11	7						
12	8						
13	9						
14	10						
15	11		0	1			
16	12						
17	13						
18	14						
19	15						
20	16	1		0			
21	17	0	1				
22	18						
23	19						
24	20						
25	21						
26	22						
27	23		0		1		
28	24		1		0		
29	25						
30	26						
31	27						
32	28						
33	29						
34	30	1	0				
35	31	0					

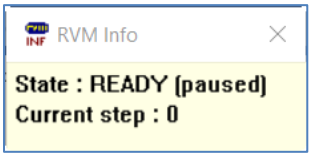
Compiling and running the test

You are now ready to run and use the ***Single Step*** execution mode.

7) Select the menu ***Run → Start Paused*** to begin simulation, accept the build confirmation.
8) Right click on each port in the project ***Inspect*** tab to add them to the ***Watch window***.
9) You should see the initial values.



10) The RVM is in pause, and you can execute step by step.

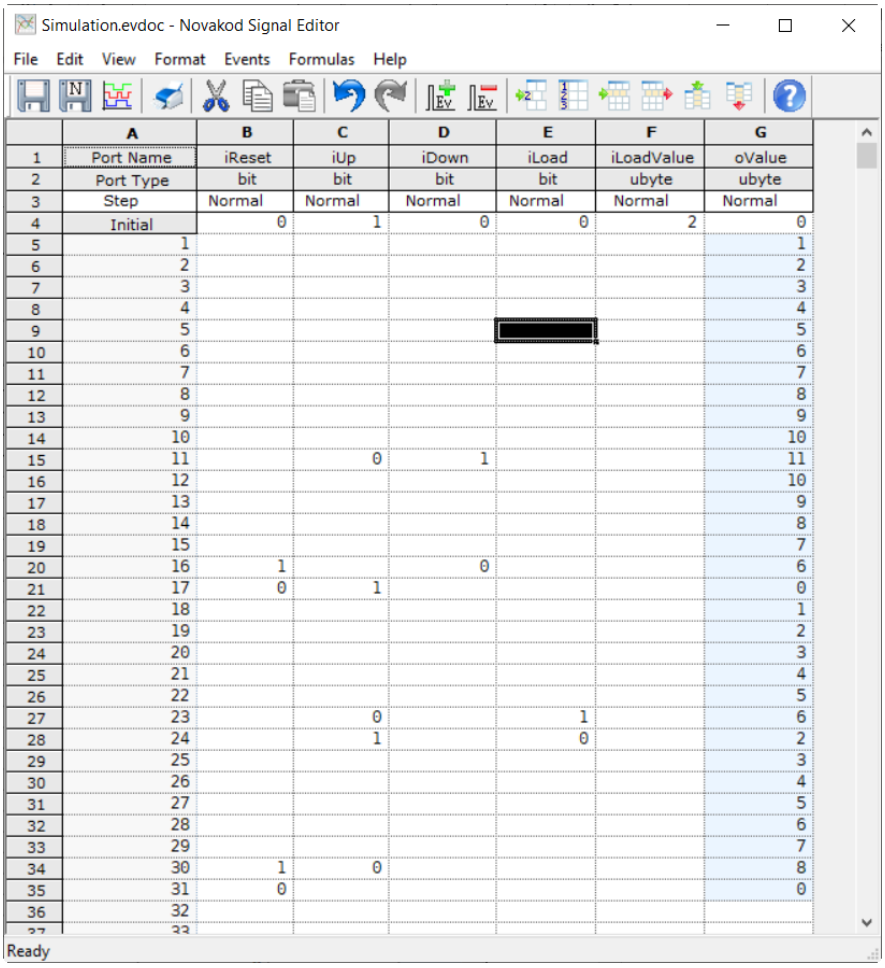


11) ***For single step, simply type F8.*** Repeat until you reach step 30. You will see all the input signals being applied at each step.
12) After 30 or more steps, stop the simulation by selecting menu, ***Run → Stop***.

View results in signal editor

You are now ready to view the results.

13) Double-click on ***targets\Simulation.evo***, the ***Signal Editor*** window starts with the values in ***oValue*** column.
14) Verify the counter operation.



	A	B	C	D	E	F	G
1	Port Name	iReset	iUp	iDown	iLoad	iLoadValue	oValue
2	Port Type	bit	bit	bit	bit	ubyte	ubyte
3	Step	Normal	Normal	Normal	Normal	Normal	Normal
4	Initial	0	1	0	0	2	0
5	1						1
6	2						2
7	3						3
8	4						4
9	5						5
10	6						6
11	7						7
12	8						8
13	9						9
14	10						10
15	11		0	1			11
16	12						10
17	13						9
18	14						8
19	15						7
20	16	1		0			6
21	17	0	1				0
22	18						1
23	19						2
24	20						3
25	21						4
26	22						5
27	23		0		1		6
28	24		1		0		2
29	25						3
30	26						4
31	27						5
32	28						6
33	29						7
34	30	1	0				8
35	31	0					0
36	32						

View results in signal viewer

Novakod Studio integrates the well-known signal viewer called ***GTK Wave***.

15) Select menu ***File→ View Signals*** or click the shortcut  . The GTK Wave windows appears. GTK Wave has been pre-configured to show the desired signals. You can see the inputs and the output in hexadecimal or as an analog signal.