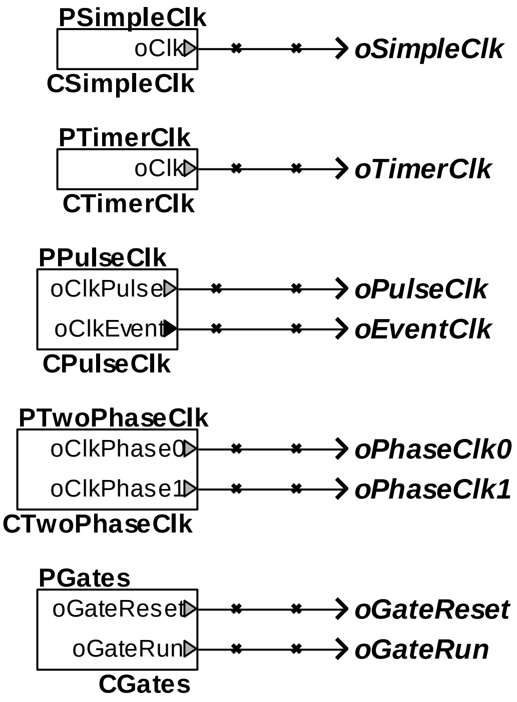
Examples 12 Clock

This example illustrates how to generate clocks.

Description

Clocks are often required in FPGA application to synchronize actions with the real time or with events, or to enable operations. The psC program consists of five components corresponding to the various clocks. The example includes 5 clock generators:



The psC code

The code for each component is self-explanatory, here is a description of each generator. The sample code is part of the component code and shows how each clock is generated.

Component	Description	Sample code

CSimpleClk	Output is FPGA clock divided by 2	`always()`
		`{`
		`    oClk = !oClk;`
		`}`

CTimerClk	Uses the timer to generate a clock, output is FPGA clock divided by 50	`timerEnd()`
		`{`
		`    oClk = !oClk;`
		`    startTimer(24l);`
		`}`

CPulseClk	A free running Counter (0 to 31), first output is one if Counter = 31 and second output sends an event if Counter = 15	`always()`
		`{`
		`    Counter++;`
		`    oClkPulse = Counter == 31;`
		`    if(Counter == 15)`
		`    {`
		`         oClkEvent :;`
		`    }`
		`}`

CTwoPhaseClk	A free running Counter (0 to 3), phase 0 is one if Counter = 1 and phase 1 is one if Counter = 3	`always()`
		`{`
		`    Counter++;`
		`    oClkPhase0 = Counter == 1;`
		`    oClkPhase1 = Counter == 3;`
		`}`

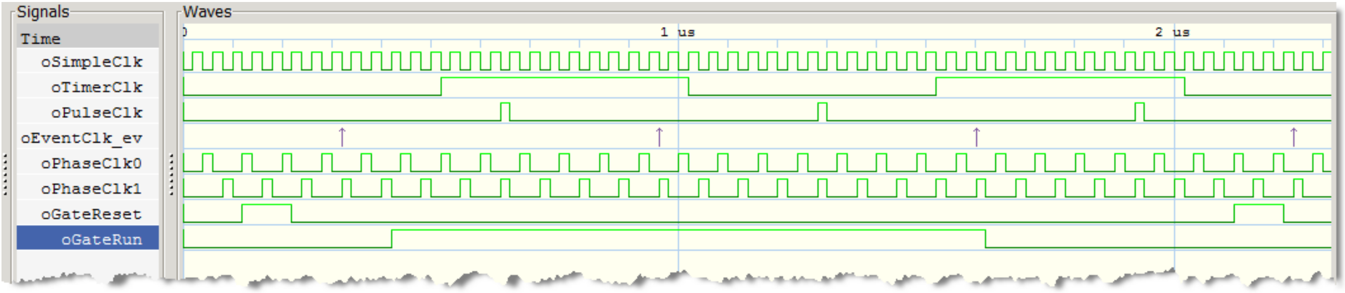
CGates	A free running Counter (0 to 99), generates controlled signals, reset for 5 <= Counter < 10 and run for 20 <= Counter < 80	`always()`
		`{`
		`    Counter = (Counter >= 99)`
		`             ?(0):(Counter + 1);`
		`    oGateReset = Counter >= 5`
		`               && Counter < 10;`
		`    oGateRun   = Counter >= 20`
		`               && Counter < 80;`
		`}`

Compiling and running the application

You can compile and execute the program, then use the signal viewer to look at the clock signals.

1) Start Novakod Studio with a double-click on the “main.prj.
2) Select the menu Run→Run N Steps to simulate.
3) Double click on “Simulation.evo”.
4) Click on the signal viewer icon.

The various clocks are shown in simulation:


Signals
Waves
Time
0
1 us
2 us
oSimpleClk
oTimerClk
oPulseClk
oEventClk_ev
oPhaseClk0
oPhaseClk1
oGateReset
oGateRun