

# Test bench for CCounterOprEvent with a simulated or the real DE1SoC board

## Core description

The core **CCounterOprEvent** is almost identical to the core **CCounterEvent\_T**. 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 counter is clocked at each step, i.e. at FPGA clock. Instead of individual signals, the input **iOpr** controls the counter operation. Its type is:

```
enum Opr_t { cOprNone, cOprReset, cOprUp, cOprDown, cOprLoad };
```

Therefore, if **iOpr** equals **cOprUp** then the counter counts up.

## Test bench

This test bench uses either a simulated board or the real board as target. **The psC code runs without changes on both.** Here is a picture of the simulated DE1SoC board during simulation, the real board will look and work as the simulated one.



## Description

The test bench uses the **Leds**, the **Switches**, the **Keys** and the **7-segments** on the board.

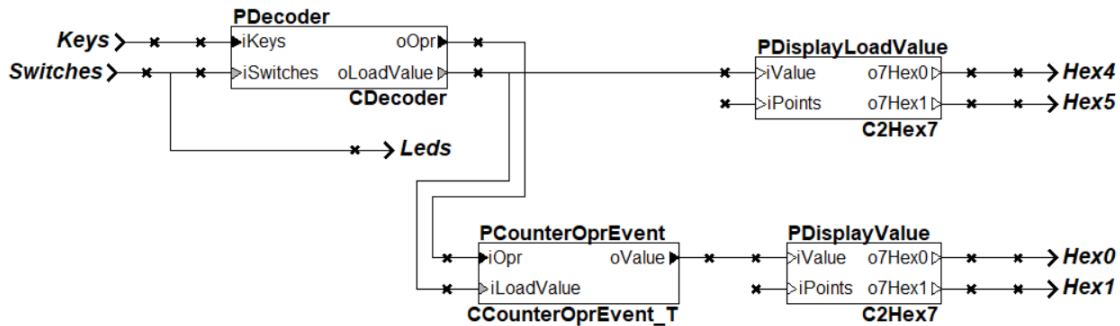
- **Keys** are used to control the counter:
  - Key0 for reset
  - Key1 for up
  - Key2 for down
  - Key3 to load the value on the Switches.
- The **Switches** are used to set the load value.
- The load value is displayed on **Hex5-4**.
- The counter value is displayed on **Hex1-0**.

## The psC test program

As shown on the next page, the test program consists of the core to be tested, **CCounterOprEvent\_T**, and components to interface it with the DE1SoC peripherals:

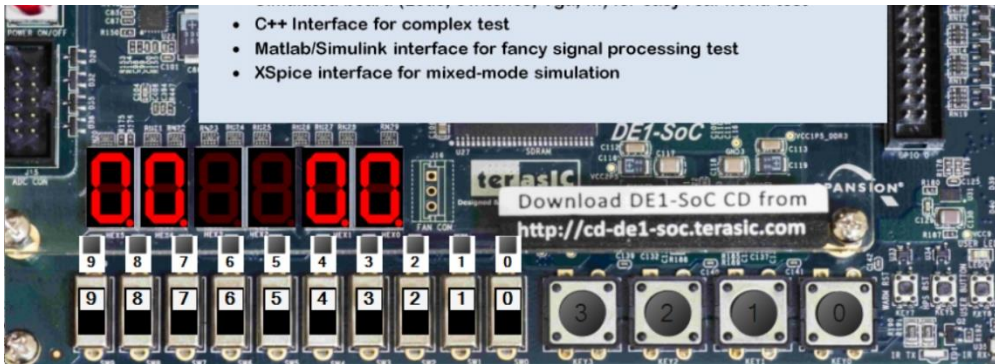
- The component **CDecoder** generates the signals **iOpr** and **iLoadValue** from the inputs **Keys** and **Switches**.

- The Processes *PDisplayValue* and *PDisplayLoadValue* are used to drive the **7-segments**. They decode a byte value and control individual LEDs.



## Execution on the simulated board

- 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 **CCounterOprEvent\_T** component to view the counter code.
- 4) Double-click on the other components to view their code.
- 5) Under **Targets**, select the target **LedSwDriver**.
- 6) Run the program, menu: **Run-> Start**
- 7) A working simulated board will appear.
- 8) Use the **Keys** and the **Switches** to test the counter.



## Execution on the FPGA board

If you have the DE1SoC board, connect the power, turn it on and connect the USB programming cable. You need to install the Intel Quartus tools. **Refer to the DE1SoC BSP manuals for installation instructions.**

- 9) Select the target **DE1SoC**.
- 10) Select the **main** component and **right click->Build project**. The psC code is compiled, the Intel Quartus tools are used to generate the programming file, then the FPGA is automatically programmed.
- 11) As in simulation, use the **Keys** and the **Switches** to test the counter.