How to validate your counter design with DSch

by Łukasz Starzak, version 2007

1. Create a T flip-flop schematic and symbol.
   (a) Take a D flip-flop from the Symbol library.
   (b) Add connections and/or elements so as to get a T flip-flop schematic.
   (c) Connect a *Button* to the logical input T. Connect a *Clock* to the clock input. Connect a *Light* to each of the outputs. Name inputs and outputs (double click the button/clock/light element).
   (d) Save your schematic.
   (e) Run simulation (*Simulation* ▷ *Start Simulation*) and test your flip-flop.

   Note: As the T-type flip-flop contains a feedback loop, it will be necessary to set outputs to a determined state at the beginning of the simulation; you achieve this through applying a reset signal.

   (f) Convert the schematic into a user-defined symbol (*File* ▷ *Schema to new symbol*). Place the inputs and outputs (L=left, R=right, T=top, B=bottom). Enter an output path in the *Save in* field (don’t use the “...” button; it won’t work). Click OK.
   (g) Check if a SYM and a TXT file have been created in the folder you specified.
2. Enter the schematic of your counter.
   (a) Place flip-flops, gates and connect them. Use *Insert* ▷ *User Symbol* to insert the T-type flip-flop you created.
   (b) Add a *Clock* and connect it to the clock inputs of the flip-flops.
   (c) Add a *Button* and connect it to the reset inputs of all the flip-flops.
   (d) Place a hexadecimal LED display (*Hexa display*) and connect the true outputs of the flip-flops to it (the most right-hand pin is for the least significant bit).
   (e) Don’t forget to save your schematic.
3. Test your counter design.
   (a) Run simulation. Reset the counter at start. Watch the LED display and check if the state sequence is correct. You may stop simulation with the red button and examine circuit waveforms (all inputs—*buttons* and *clocks*—and all outputs—*lights* and *hexa displays*) with *View* ▷ *Timing Diagrams*.
   (b) Debug your design if the output was wrong. Start with connecting a light to each of the T inputs, restarting simulation and checking which of these signals have incorrect values. This will give you an idea of where to search for errors.