Introduction to design of hardware systems using FPGA

Tutorial 03 on Dedicated systems

Teacher: Giuseppe Scollo

University of Catania
Department of Mathematics and Computer Science
Graduate Course in Computer Science, 2016-17

Table of Contents

tutorial outline

this tutorial deals with:

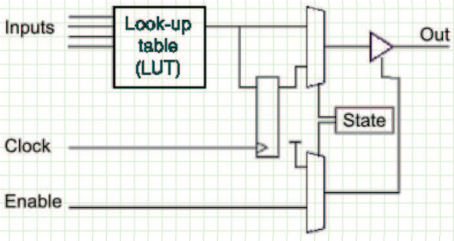
- FPGA structure and organization
- typical design and test workflow with FPGA
- automated synthesis of circuits on FPGA
- lab experience

FPGA structure

precursors: PLA, PAL, CPLD

typical FPGA constituents:

- logical elements (LE): specializable for user-defined logical functions
- complex logical blocks (CLB): built from LEs, multiplexers, registers etc., configurable to implement user-defined circuit components
- interconnection, routing, and I/O blocks



Wilson, Fig. 2.3 - FPGA complex logic block

configuration of the CLB in the figure:

logical function assignment to the look-up table (LUT)

combinational or synchronous operation

input to enable output driver

design workflow with FPGA

typical work sequence (not all steps are present in every design):

1. RTL specification (schematic or HDL design)
2. syntactic and static semantic analysis
   correction of any errors, analysis reiteration
3. RTL simulation
   correction of any semantic errors, reiteration of analysis and simulation
4. RTL synthesis
5. timing analysis and clock adjustment
6. physical synthesis

automated synthesis on FPGA

physical synthesis, which is automated by several analysis and optimization tools, is composed of various processes:

- mapping of RTL components to FPGA components (LE, CLB etc.)
- placement of components and signal routing over the FPGA
  using suitable optimization algorithms: recursive cut, simulated annealing
- FPGA netlist generation

lab experience

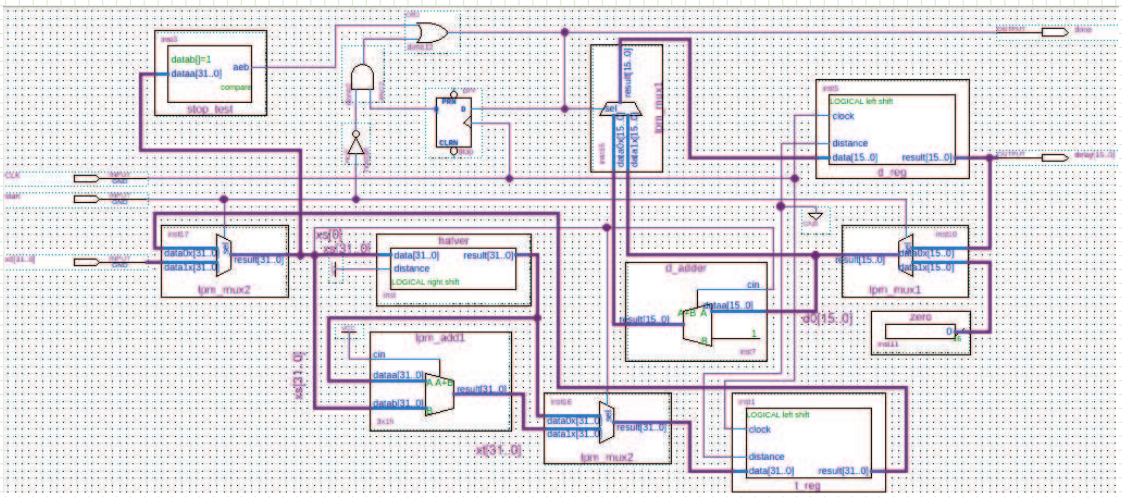
it is possible to describe, simulate and synthesize hardware circuit models also without making use of an HDL, when a graphical editor is available for schematic design, together with adequate software tools

- launch Quartus 13.1 (Web edition) and therein create a new project named schematic_delay_collatz
- draw the schematic of the hardware datapath for the delay of Collatz trajectories presented in the second lecture and worked out in the second lab experience

  a model using several library parameterized modules (*lpm*) is shown in the next figure
- compile and fix any detected errors
- compare usage of resources (n. of LEs and registers) and worst-case slack with those obtained from compilation and timing analysis completed in the second lab experience
- extend the schematic with a testbench circuit, to drive the datapath with the numbers from 1 up to 255 as trajectory start values

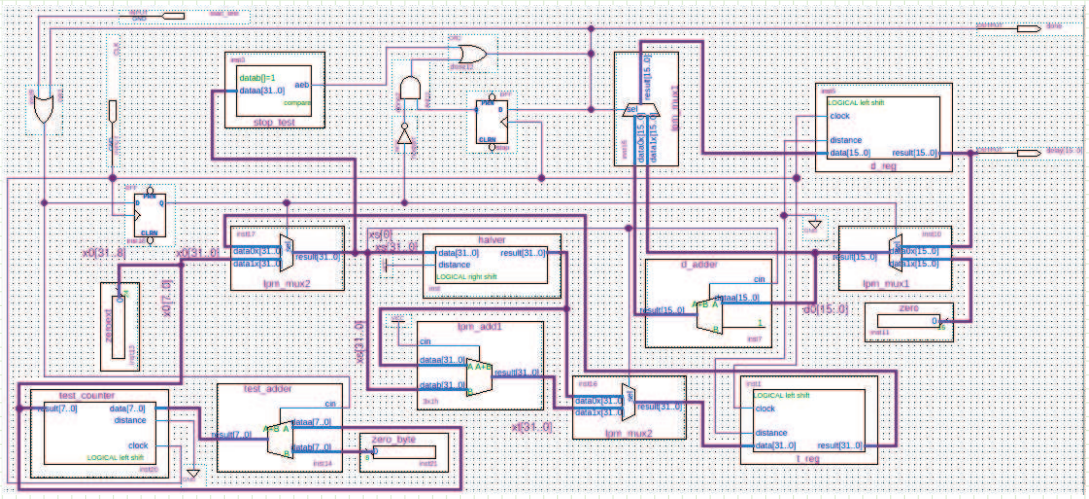
  the input for each trajectory following the first one must wait for the datapath signalling the end of the computation for the previous trajectory, see for example the subsequent figure
- run the functional simulation and check the correspondence of the outcome to expectations

a Quartus II schematic for the Collatz delay datapath



schematic using library parameterized modules (lpm)

a Quartus II testbench for the Collatz delay datapath



a testbench for the Collatz delay datapath

references

recommended readings:

Wilson (2015) Capp. 2, 5, 6

useful materials for the proposed lab experience (source: Altera University Program, 2014)

Quartus II Introduction Using Schematic Designs - For Quartus II 13.1

Using Library Modules in VHDL Designs - For Quartus II 13.1