Introduction to design of hardware systems using FPGA

Tutorial 02 on Dedicated systems

Teacher: Giuseppe Scollo

University of Catania
Department of Mathematics and Computer Science
Graduate Course in Computer Science, 2017-18

Table of Contents

1. Introduction to design of hardware systems using FPGA
2. tutorial outline
3. FPGA structure
4. design workflow with FPGA
5. automated synthesis on FPGA
6. lab experience
7. a Quartus Prime schematic for the Collatz delay datapath
8. a simulation testbench for the Collatz delay datapath
9. a simulation outcome
10. operational tips
11. references

tutorial outline

this tutorial deals with:

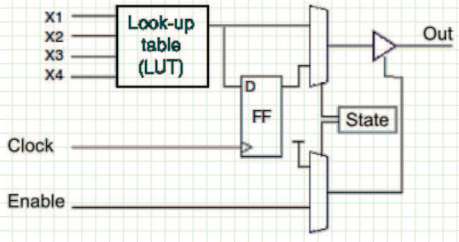
- FPGA structure and organization
- typical design and test workflow with FPGA
- automated synthesis of circuits on FPGA
- lab experience

FPGA structure

precursors: PLA, PAL, CPLD

typical FPGA constituents:

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



a model of a programmable logic block

configuration of the block 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 synthesis
4. RTL simulation
   correction of any semantic errors, reiteration of analysis and simulation
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's, programmable logic blocks, registers etc.
- placement of components and signal routing over the FPGA
  using suitable optimization algorithms: recursive cut, simulated annealing
- FPGA netlist generation

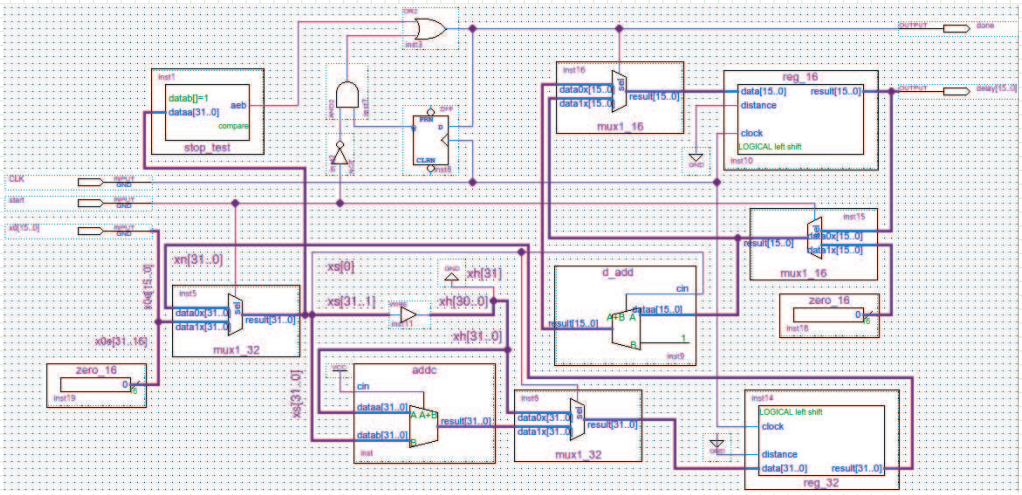
it is possible to describe, simulate and synthesize hardware circuit models also without making use of an HDL, when a graphical editor is available for schematic design, together with adequate software tools

- launch Quartus 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

  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 first lab experience
- create a simulation testbench with a sequence of two or three inputs as trajectory starting points

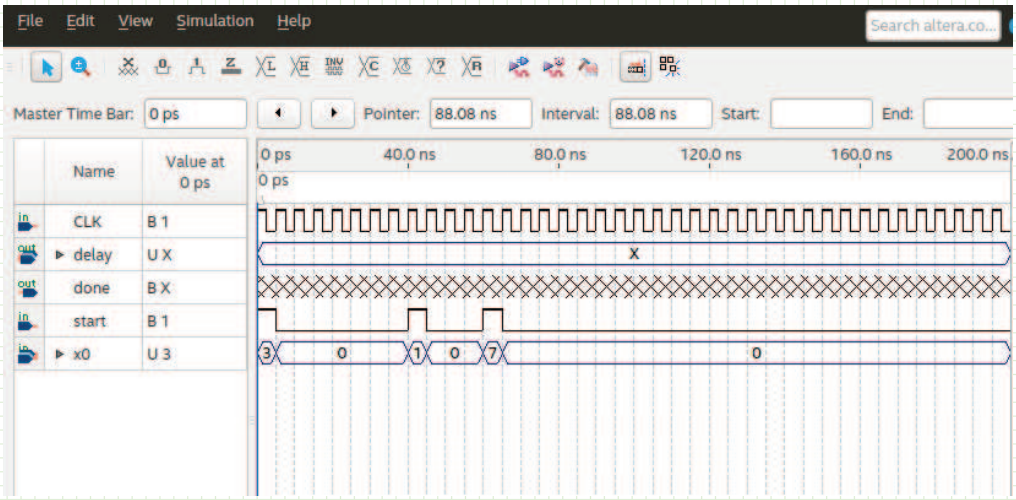
  the timing of the input for each trajectory after the first one should be calculated so as to follow the previous trajectory output signalling the end of its computation, see for example the subsequent figure
- run the functional simulation and check the correspondence of the outcome to expectations, see for example the last figure

a Quartus Prime schematic for the Collatz delay datapath



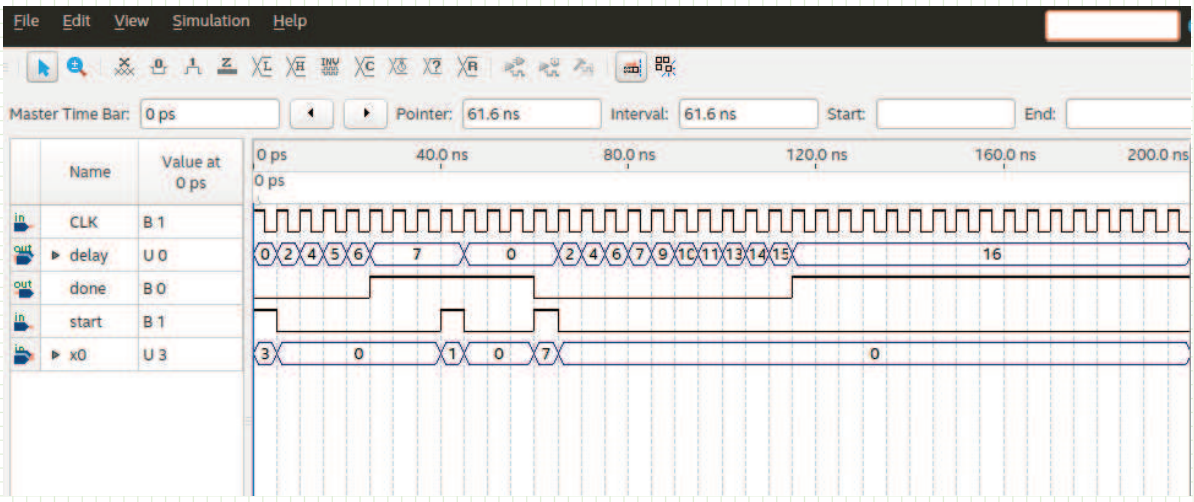
schematic using library parameterized modules (lpm)

a simulation testbench for the Collatz delay datapath



simulation testbench for the Collatz delay datapath

a simulation outcome



simulation outcome for the Collatz delay datapath

a few tips to perform the lab experience without unnecessary effort:

the following notes present a few workarounds to little troubles which may slow down or jeopardize the execution of the lab experience

- *you may download the ZIP archive of all tips*

1. Quartus Schematic tips
2. tips on using Quartus Library Parameterized Modules (LPM)

recommended readings:

Zwolinski Ch. 1, Sect. 1.3

useful materials for the proposed lab experience (source: Intel® FPGA University Program, November 2016)

Quartus Prime Introduction Using Schematic Designs - For Quartus Prime 16.1

Using Library Modules in VHDL Designs - For Quartus Prime 16.1