



Approaching Field Programmable Gate Arrays with Deeds

Giuliano Donzellini

giuliano.donzellini@unige.it

DITEN – University of Genoa, Italy



Thank you for the invitation!

The software tools presented are available, free of charge, at:
<https://www.digitalelectronicsdeeds.com>



Introduction



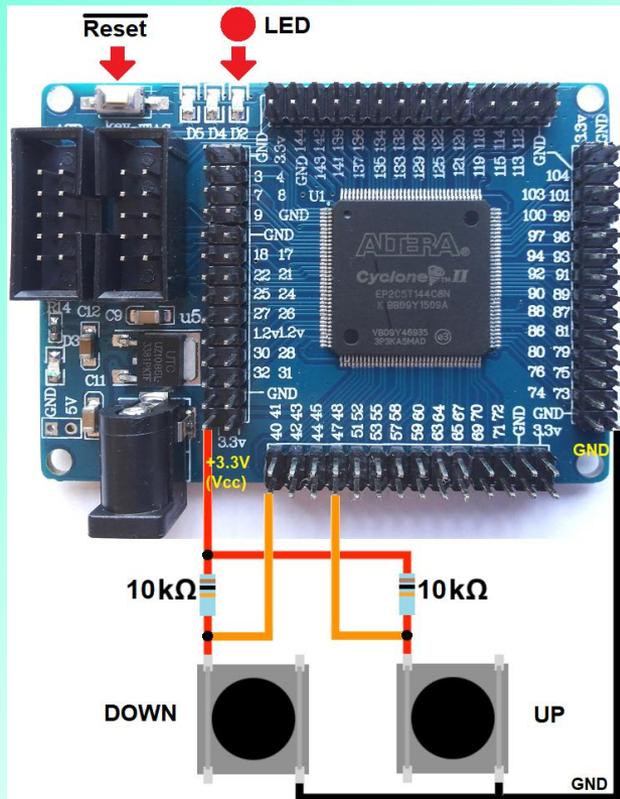
- **Deeds** (Digital Electronics Education and Design Suite)
- It is developed at DITEN (ex DIBE), University of Genoa.
- The suite is composed by three simulators and a wide collection of associated **learning material** to learn-by-doing and practice with:
 - **Combinational and sequential logic networks**
 - **Finite state machine design**
 - **Embedded microcomputer interfacing and programming**
 - **FPGA programming (exporting projects to EDA tools)**



<https://www.digitalelectronicsdeeds.com>

Deeds: the simulation tools

- Deeds-DcS Digital Circuit Simulator
- Deeds-FsM Finite State Machine Simulator
- Deeds-McE Microcomputer Emulator



- The three simulators are *fully integrated*, to design and simulate *digital systems* with *standard logic*, *finite state machines* and *microcomputers*.
- Extensive *Learning Materials* are available on the web site
- Projects can *be exported in VHDL*.
- Projects can *be tested on FPGA*.





- Home
- Deeds Simulator
- Learning Materials
- Discussion Group
- Books & Digital Contents

Welcome to Digital Electronics Deeds

(by Giuliano Donzellini)

In this web site you'll find *digital circuits, ideas, projects, tools for simulation* and *testing on FPGA*, and more. A complete learning path to understanding and designing digital systems, supported step-by-step by [Deeds simulator](#). We tried to do our best but... is up to you to judge if our "deeds" (literal meaning of the word!) are good or bad...

News

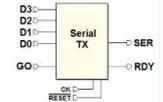
Introduction to Microprocessor-based Systems Design



Ideas & Projects

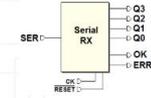
Synchronous Serial Transmitter (4 bits)

In this example, we'll design a simplified 4-bits synchronous serial transmitter... ([read more](#)).



Synchronous Serial Receiver (4 bits)

Let's design a 4-bits synchronous serial receiver. The unit will receive serial sequences on... ([read more](#)).



Synchronous Serial Communication System (4 bits)

In this example, starting from the



<https://www.digitalelectronicsdeeds.com>



Deeds website: Learning Materials



Learning Materials

To use the learning materials, please install the latest version of [Deeds](#).

- Combinational
- Sequential
- FSM
- μ P
- FPGA Tutorials
- FPGA Labs
- Demos

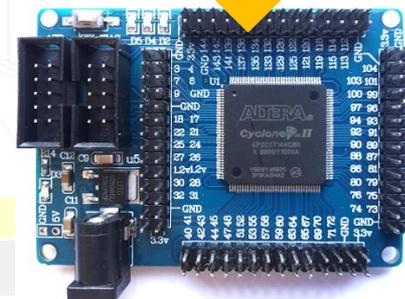
Labs		
Topic	Combinational Networks	
1	Introduction to digital electronics	Download
1.1	Introduction to the Digital Circuit Simulator	001001
1.2	Analysis of simple logic gates	001002
2	Multiplexers and Demultiplexers	Download
2.1	Analysis of a multiplexer (2 to 1)	005030
2.2	Analysis of a demultiplexer (1 to 2)	005040
2.3	Analysis of a simplified shared-line communication channel	005050
3	Applications of Boolean Algebra	Download
3.1	Analysis of a multi-level logic network	015060
3.2	Synthesis of a simple boolean function	015065
3.3	Design of a programmable logic gate	015070
3.4	Synthesis of a boolean function	015080
3.5	Functional analysis of a two-level combinational network	015090
3.6	Design of a simple combinational network	015095
3.7	Analysis and design of multiplexer-based combinational networks	015100



Deeds website: FPGA Tutorials and Labs

[Link](#)

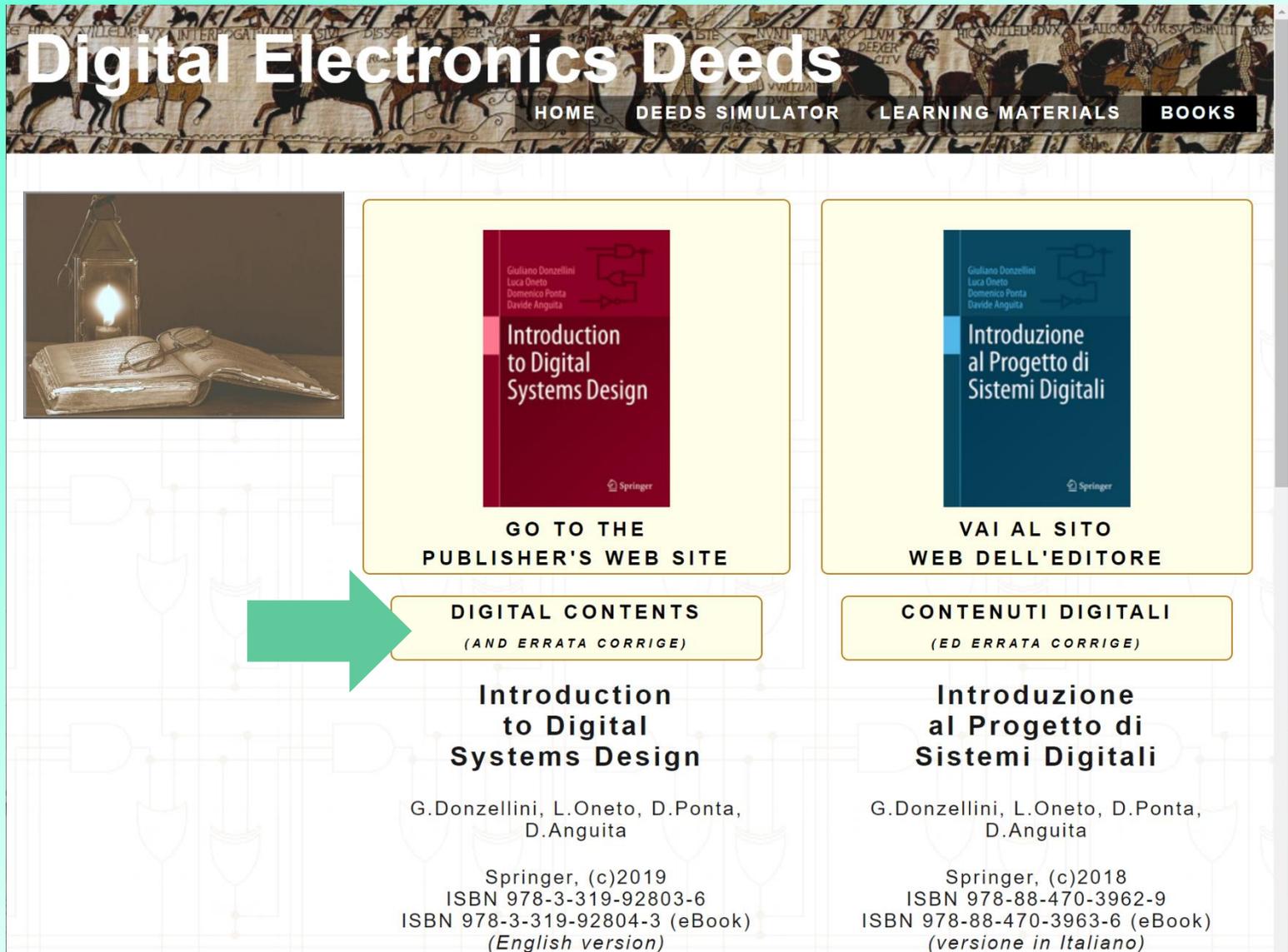
Topic	FPGA Tutorials
	Introduction to FPGA Programming with Deeds
1	Terasic/Altera DE2
	Circuit Prototyping on Terasic/Altera DE2 Board
	Sequential Circuit Testing on Terasic/Altera DE2 Board
	Microcomputer Testing on Terasic/Altera DE2 Board
2	Terasic/Altera DE0-CV
	Circuit Prototyping on Terasic/Altera DE0-CV Board
	Sequential Circuit Testing on Terasic/Altera DE0-CV Board
Topic	FPGA Labs
	Main Index <i>(all items are already listed above)</i>
FPGA	Analysis and implementation on FPGA of a 12-bits counter
	A light controller on FPGA
	Serial Transmitter and Receiver on FPGA
	Asynchronous Serial Transmitter and Receiver, on FPGA
	Push-button controlled byte generator on FPGA
	Emulation of a Register/Counter on FPGA
	Emulation of a Universal Shift Register on FPGA
	Asynchronous Serial Transmitter and Receiver, interrupt based, on FPGA
	Digital Waveform Generator, on FPGA
	Asynchronous Serial Line Data Processor, on FPGA



TOP



Books 'digital contents': FPGA exercises



The screenshot shows the website 'Digital Electronics Deeds' with a navigation bar containing 'HOME', 'DEEDS SIMULATOR', 'LEARNING MATERIALS', and 'BOOKS'. Below the navigation bar, there is a section for digital content. On the left, there is a small image of an open book with a lit lantern. A large green arrow points from the left towards the English version of the book. The English version is titled 'Introduction to Digital Systems Design' by Giuliano Donzellini, Luca Oneto, Domenico Ponta, and Davide Anguita, published by Springer in 2019. The Italian version is titled 'Introduzione al Progetto di Sistemi Digitali' by the same authors, published by Springer in 2018. Both versions have ISBNs for the print and eBook editions.

Digital Electronics Deeds
HOME DEEDS SIMULATOR LEARNING MATERIALS BOOKS

Introduction to Digital Systems Design
Giuliano Donzellini
Luca Oneto
Domenico Ponta
Davide Anguita
Springer

GO TO THE PUBLISHER'S WEB SITE

DIGITAL CONTENTS
(AND ERRATA CORRIGE)

Introduction to Digital Systems Design
G.Donzellini, L.Oneto, D.Ponta,
D.Anguita
Springer, (c)2019
ISBN 978-3-319-92803-6
ISBN 978-3-319-92804-3 (eBook)
(English version)

Introduzione al Progetto di Sistemi Digitali
Giuliano Donzellini
Luca Oneto
Domenico Ponta
Davide Anguita
Springer

VAI AL SITO WEB DELL'EDITORE

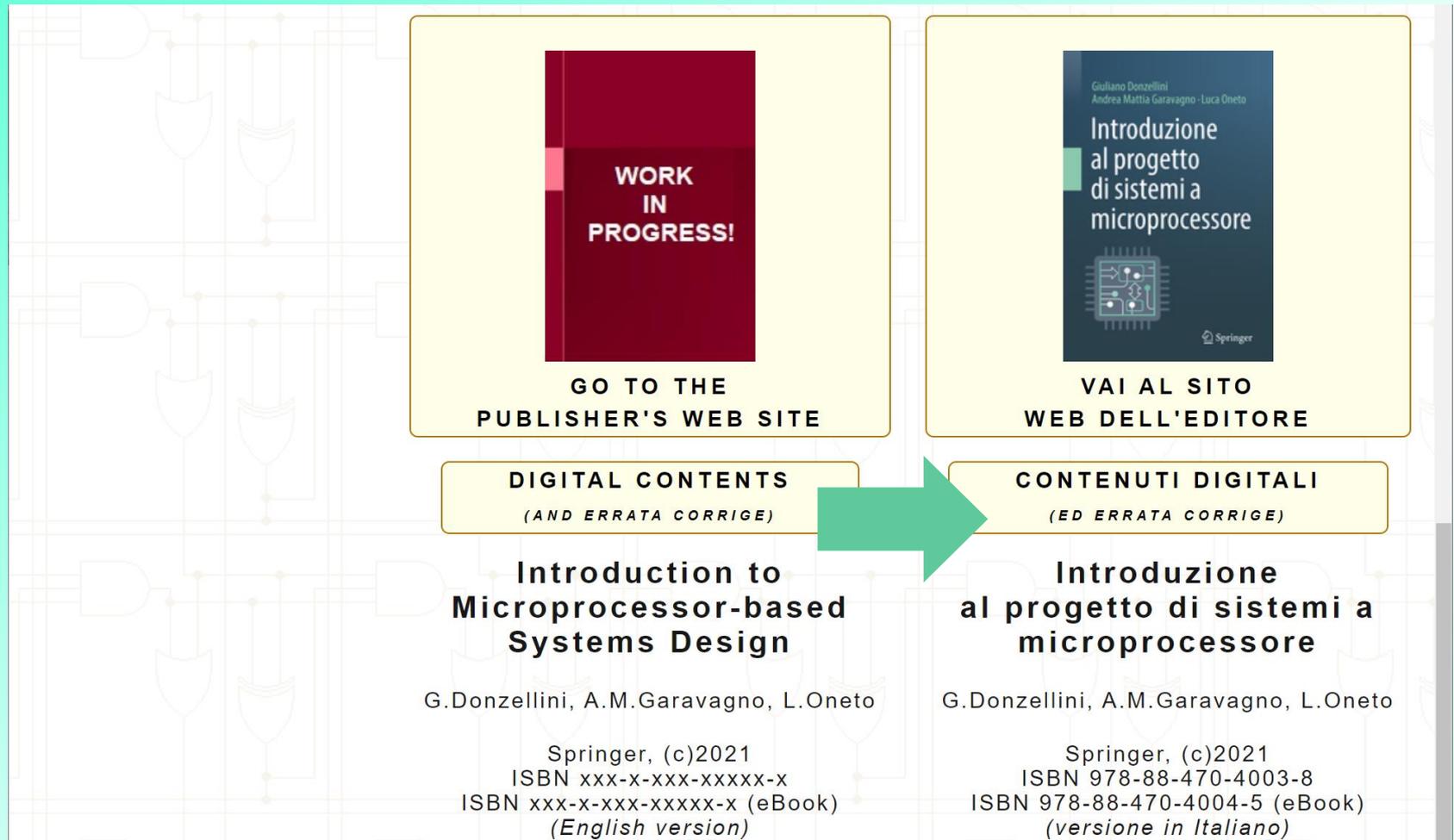
CONTENUTI DIGITALI
(ED ERRATA CORRIGE)

Introduzione al Progetto di Sistemi Digitali
G.Donzellini, L.Oneto, D.Ponta,
D.Anguita
Springer, (c)2018
ISBN 978-88-470-3962-9
ISBN 978-88-470-3963-6 (eBook)
(versione in Italiano)

Link:



Deeds web site: *FPGA in the books* (2)

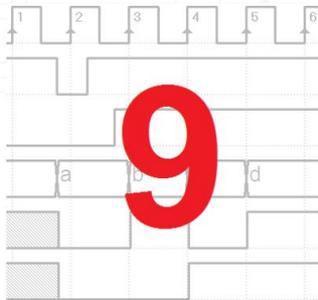


Link:



Deeds: books online support (an example, 1)

[Link](#)



Introduction to FPGA and HDL Design

Examples

Exercises

Solutions

Errata corrigere

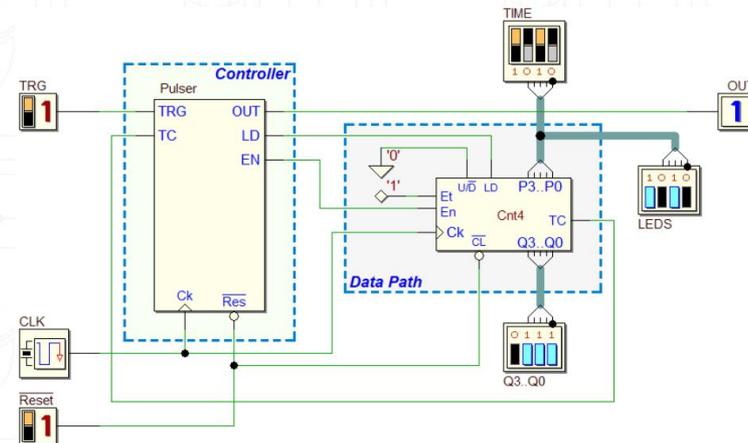
9.1 Field-Programmable Gate Arrays

Digital Contents

1. [Boolean Algebra and Combinational Logic](#)
2. [Combinational Network Design](#)
3. [Numeral Systems and Binary Arithmetic](#)
4. [Complements in Combinational Network Design](#)
5. [Introduction to Sequential Networks](#)
6. [Flip-Flop-Based Synchronous Networks](#)
7. [Sequential Networks as Finite State Machines](#)
8. [The Finite State Machine as System](#)
9. [Introduction to FPGA and HDL Design](#)

9.1.6 Deeds Support for FPGA

Pulse generator schematic (without FPGA configuration)



Deeds: books online support (an example, 2)

A Musical Box on FPGA (it uses a DMC8 microcomputer).

[Link](#)

(Sorry, this web page is in Italian.
The English version will be available soon)



5

5.6.4 Carillon musicale

Contenuti digitali (ed errata corrige)

1. [Introduzione alle reti di calcolo programmabili](#)
2. [Un sistema basato sul microprocessore DMC8](#)
3. [La programmazione del DMC8](#)
5. [Sistemi a microprocessore su FPGA](#)

Appendici

A. [Le memorie](#)

Sequenza musicale eseguita (tratta dalla Bourrée in Mi minore [BWV 996] di J.S.Bach):

Collegamenti (scheda DE2):

OCT1 ↑ OCT0 ↑ GLIDE ↑ PLAY ↑

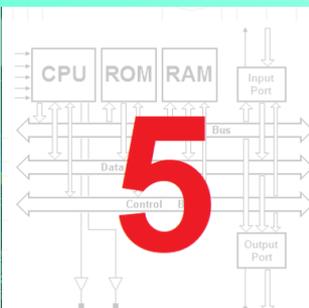
TOP

Deeds: books online support (an example, 3)

A Stepper-motor controller (it uses a DMC8 microcomputer).

[Link](#)

(Sorry, this web page is in Italian.
The English version will be available soon)



Contenuti digitali
(ed errata corrige)

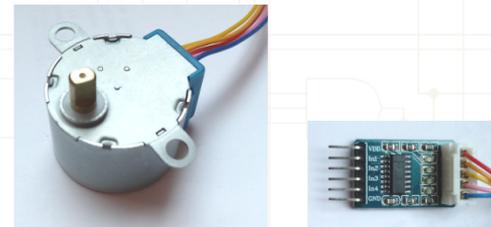
1. [Introduzione alle reti di calcolo programmabili](#)
2. [Un sistema basato sul microprocessore DMC8](#)
3. [La programmazione del DMC8](#)
4. [L'interfacciamento di dispositivi](#)
5. [Sistemi a microprocessore su FPGA](#)

Appendici

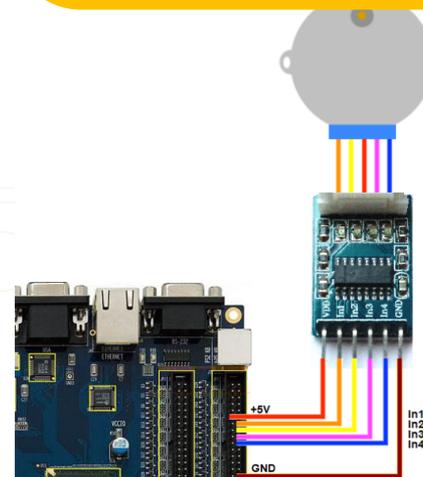
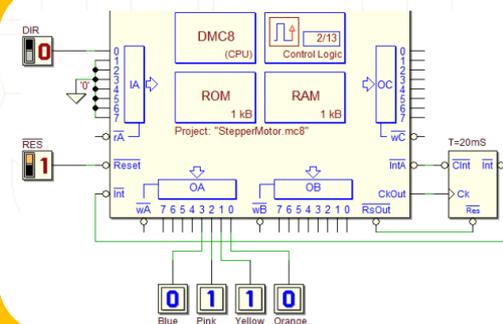
A. [Le memorie](#)

5.6.5 Controllo di motore passo-passo

Motore passo-passo e sua scheda controller:

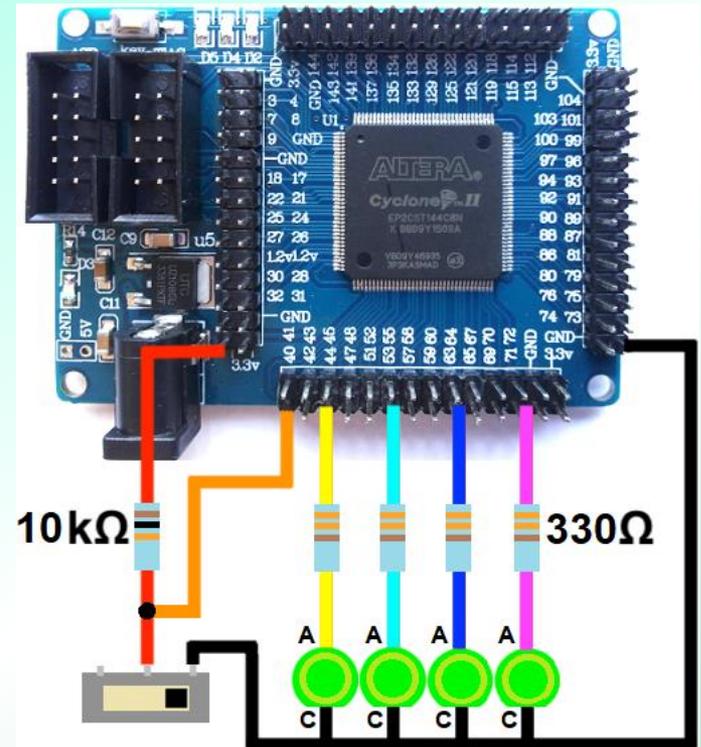


Il sistema:

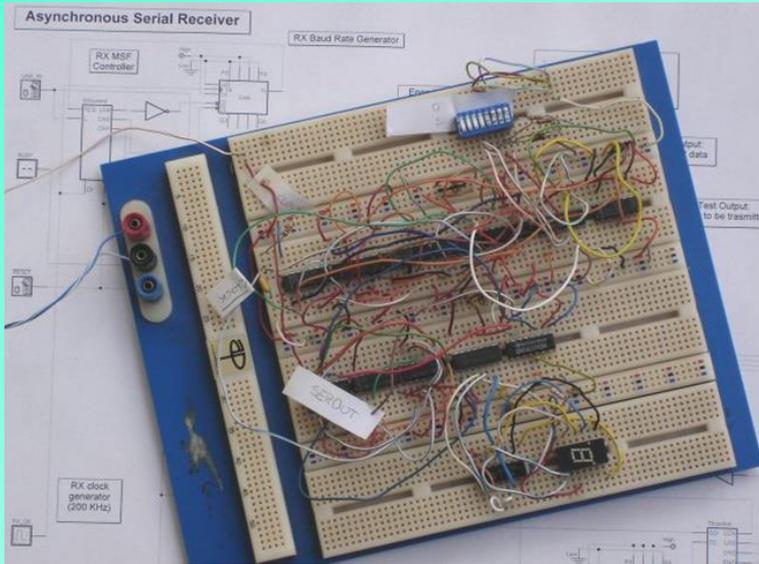


Approaching FPGA programming

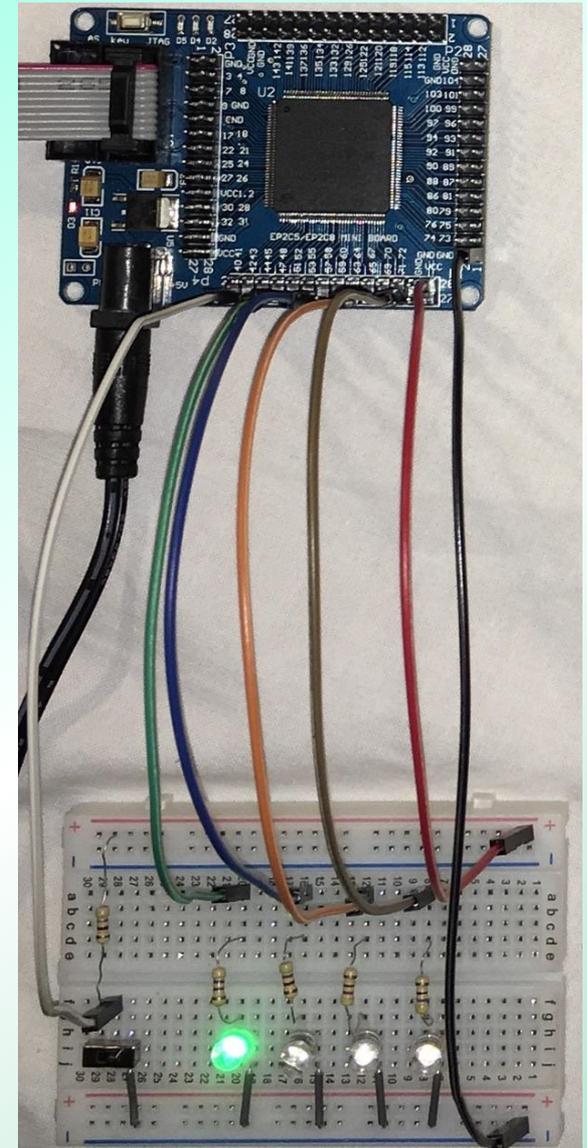
- Our experience in teaching a first year course of digital design shows that the introduction of **Field Programmable Gate Arrays** (FPGA) is advisable and very attractive.
- Students demonstrate a better interest for the topics, if they can really verify the circuits they study and design.



Bread-board and FPGA



- Traditional bread-board based prototyping is replaced by FPGA programming.
- Bread-board returns on the scene as a complement for those cheap FPGA boards that presents less I/O devices.



Introducing digital systems with HDL?

```
SerialSystem.vhd Components.vhd
130 ARCHITECTURE behavioral OF SiPoE8 IS
131 BEGIN
132   RegSiPoE8: PROCESS( Ck, nCL )
133     variable aReg: std_logic_vector( 7 downto 0 );
134     BEGIN
135       if (nCL = '0') then aReg := (others =>'0');
136       elsif (nCL = '1') then
137         if (Ck'event) AND (Ck='1') THEN -- Positive Edge -----
138           if (E = '1') then
139             aReg := (I & aReg(7) & aReg(6) & aReg(5) & aReg(4) & aReg(3) & aReg(2) & aReg(1));
140           elsif not(E = '0') then
141             aReg := (others =>'X');
142           END IF;
143         END IF;
144       else
145         aReg := (others =>'X');
146       END IF;
147       --
148       Q7 <= aReg(7);
149       Q6 <= aReg(6);
150       Q5 <= aReg(5);
151       Q4 <= aReg(4);
152       Q3 <= aReg(3);
153       Q2 <= aReg(2);
154       Q1 <= aReg(1);
155       Q0 <= aReg(0);
156     END PROCESS;
157 END behavioral;
```



- At professional level, FPGA programming is usually performed using Hardware Description Languages (HDL).
- We are somehow critical of the current trend of introducing digital systems with HDL in a first course.



HDL may hide basic issues from beginners

- It is not easy to build good foundations on logic design just by completely migrating the traditional schematic, simulation and bread-board based prototyping to VHDL based FPGA design flow.
- Such approach has many advantages, especially for complex systems.
- However, we think it may hide important basic issues and concepts from beginners.

```
PULSE: process( nAutoReset, iMClk )
    constant IsTime: integer := 100;
    constant IsLedEnd: integer := 25;
    variable Stepper: integer range 0 to IsTime;
    --
    constant TimeCycle: integer := 25;
    variable Pulser: integer range 0 to TimeCycle;
    --
    variable Level: integer range 0 to 1;
    variable Pulsing: boolean;
    --
begin
    if (nAutoReset = '0') then
        Stepper:= 0;
        Pulsing:= false;
        Pulser:= 0;
        StepPulse <= '0';
        StepLED <= '0';
        --
    elsif rising_edge( iMClk ) then

        if (ManualClkMode = '0') then
            Stepper:= 0;
            Pulsing:= false;
            Pulser:= 0;
            StepPulse <= '0';
            StepLED <= '0';
            -----

        else --(ManualClkMode = '1')
            ----- Button Pulsed Mode -----
            if (aTick = '1') then -- every 10 mS
                --
                if (aButton = '1') then
                    if (not Pulsing) then
                        if (Stepper < IsTime) then
                            -----
                            StepPulse <= '1';
                            StepLED <= '1';
                            Stepper:= Stepper + 1;
                        else
                            -----
                        end if
                    end if
                end if
            end if
        end if
    end if
end process;
```



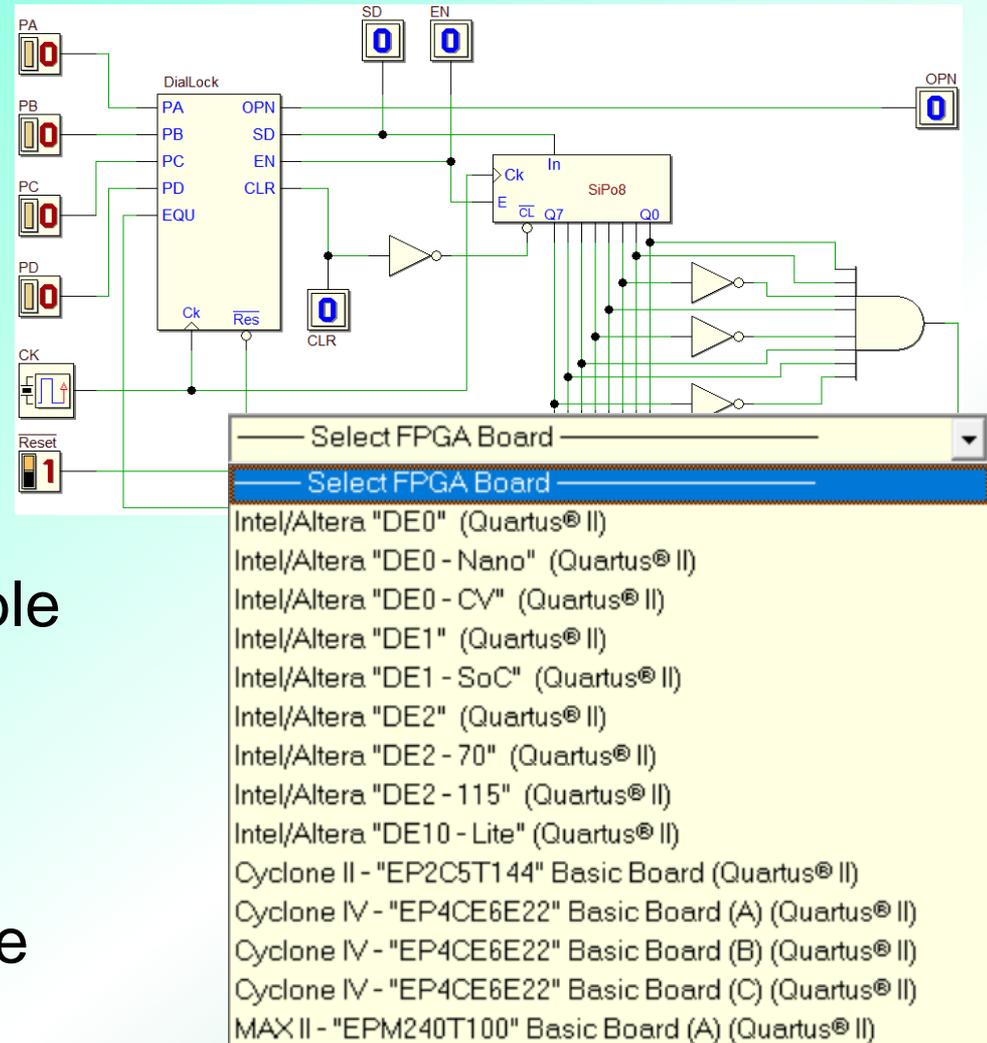
Making FPGA available to beginners...

- **Deeds** makes the process of FPGA configuration straightforward and compatible with the beginners' skills.
- It overcomes the prerequisite of some proficiency in high-level programming languages.
- It allows to configure FPGA boards for testing starting with traditional schematics-based entry, generally appreciated by beginners.



FPGA configuration with Deeds

- **Deeds** integrates FPGA configuration and testing into its design and simulation flow.
- This feature makes digital design, including microprocessor programming, demonstrable through FPGA boards.
- At the present moment a dozen of commercially available FPGA boards are supported.

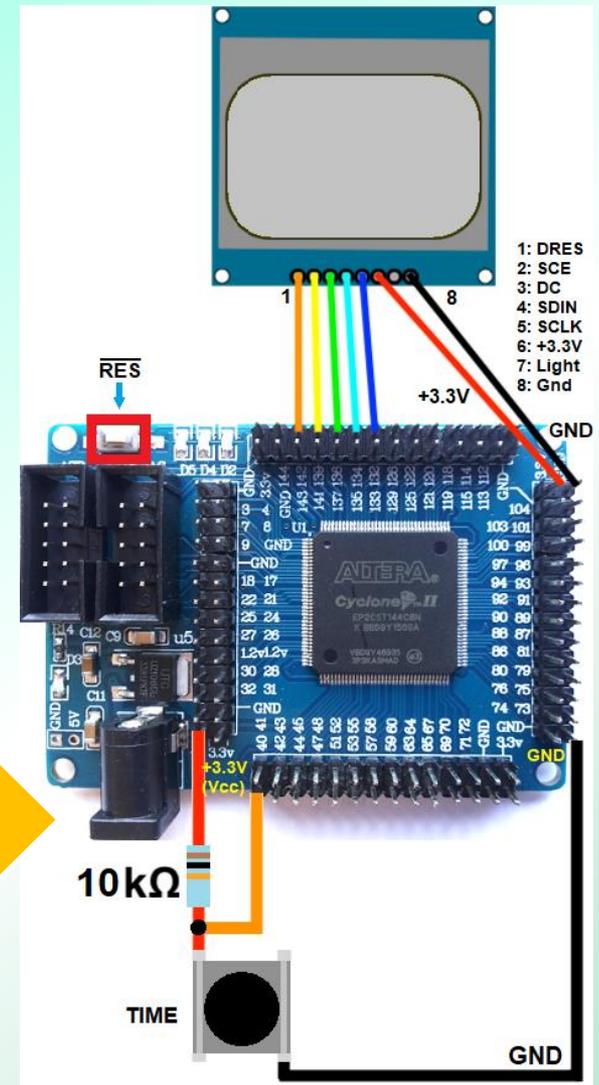
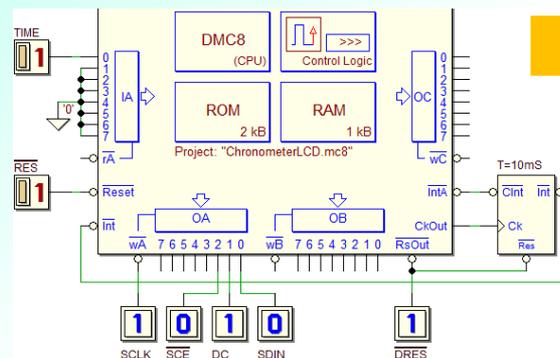


At the moment, only Intel (ex Altera) FPGA chip are supported.

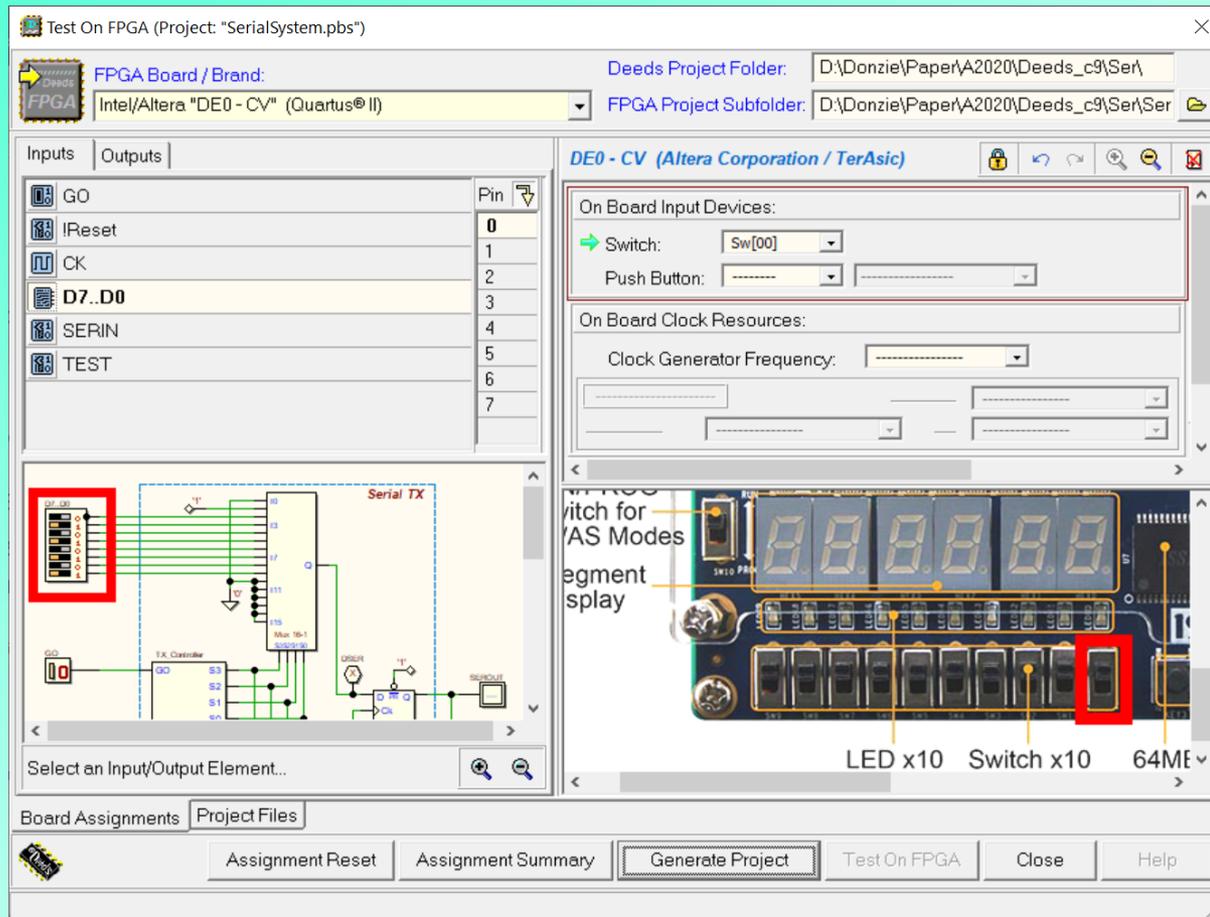


Programming FPGA in a transparent way

- Starting from Deeds, students can compile their project into an FPGA, *leaving in the background* the operations performed by the *FPGA-specific development software*.
- The intention is not to exemplify FPGA programming via high-level synthesis languages, but to allow a direct implementation on FPGA of Deeds projects.



The 'Test on FPGA' window (1)



The *Test on FPGA* window allow to specialize VHDL conversion for a particular FPGA board, generating all the specific *Project Files*, starting from the Deeds project.

Components, FSM and Microcomputers are exported in behavioral VHDL code, while the top level schematic is converted in structural VHDL.

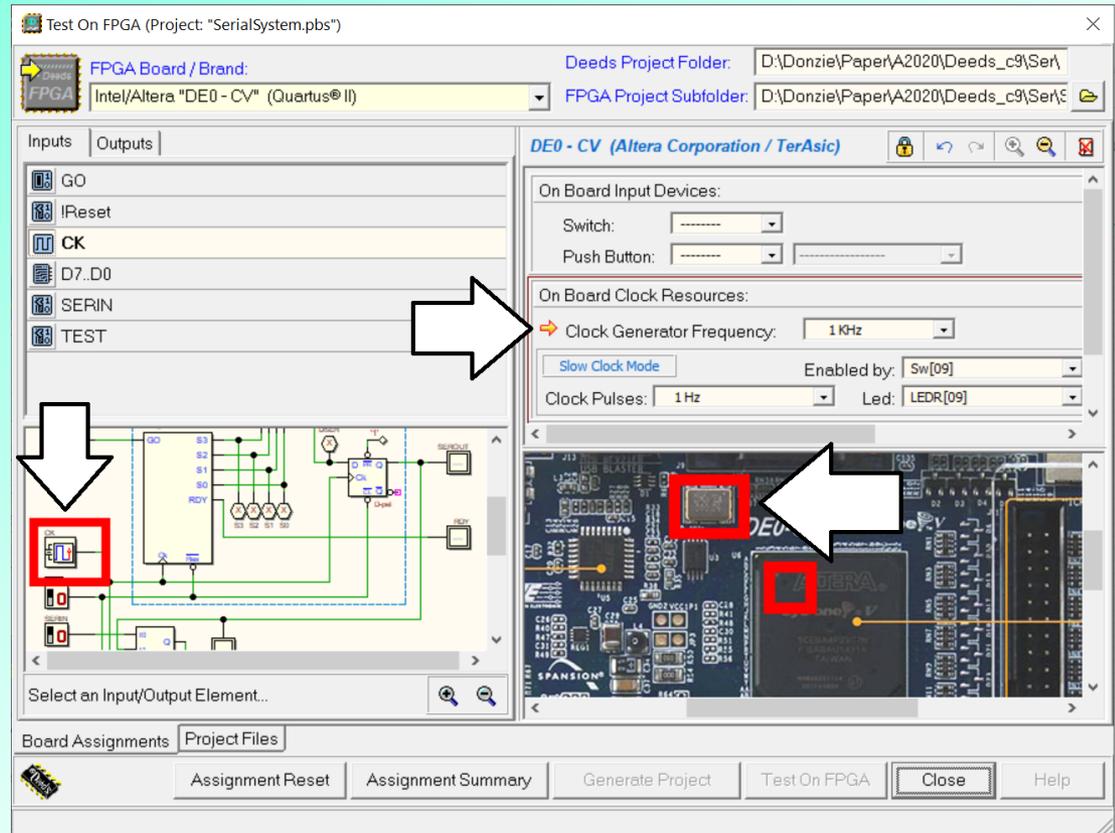


The 'Test on FPGA' window (2)

- The user selects each I/O termination in the Deeds schematic (see on the left), with the purpose of associating it to a device or resource available on board (on the right).

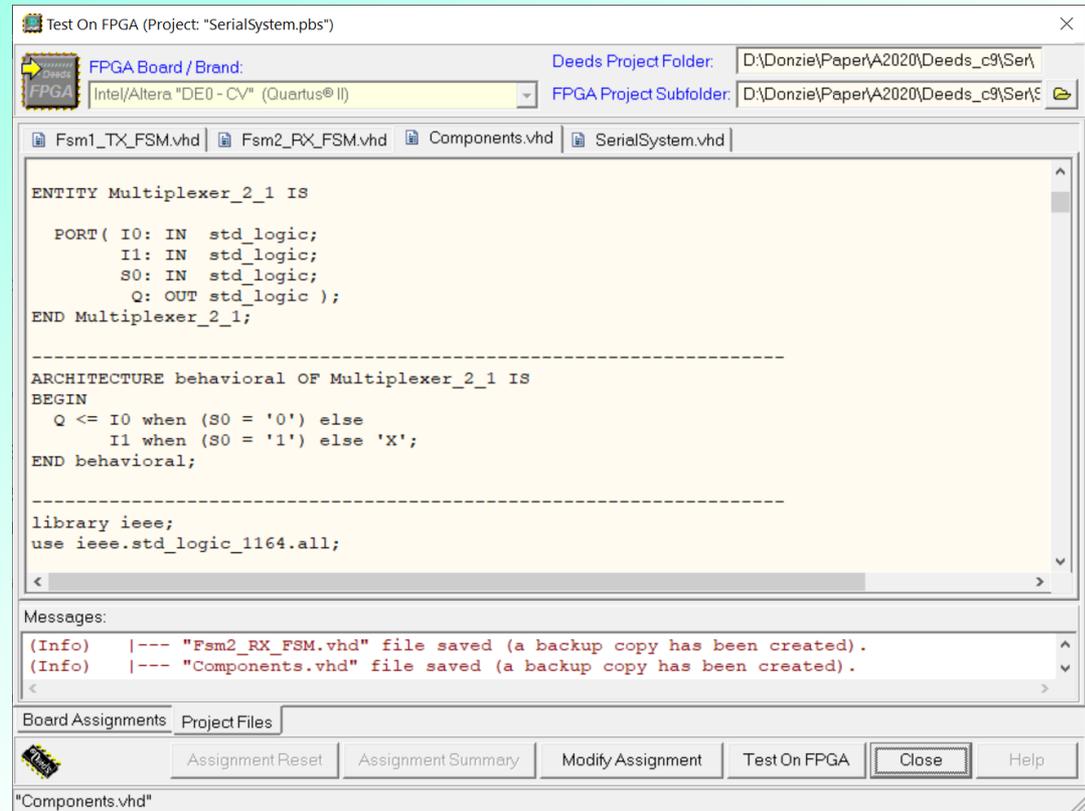
- Associations can be done *in advance by teachers*.

- In the present example, a clock resource on the FPGA board has been associated to the clock component in the Deeds schematic (assigning a clock frequency of 1KHz).



The generated VHDL code

- When the conversion ends, the VHDL files are presented to the user through a dedicated window.
- Users with a good VHDL familiarity can export, edit and re-use the generated code in a professional design tool of their choice.
- The VHDL generator builds also the project files required by the FPGA specific tool.

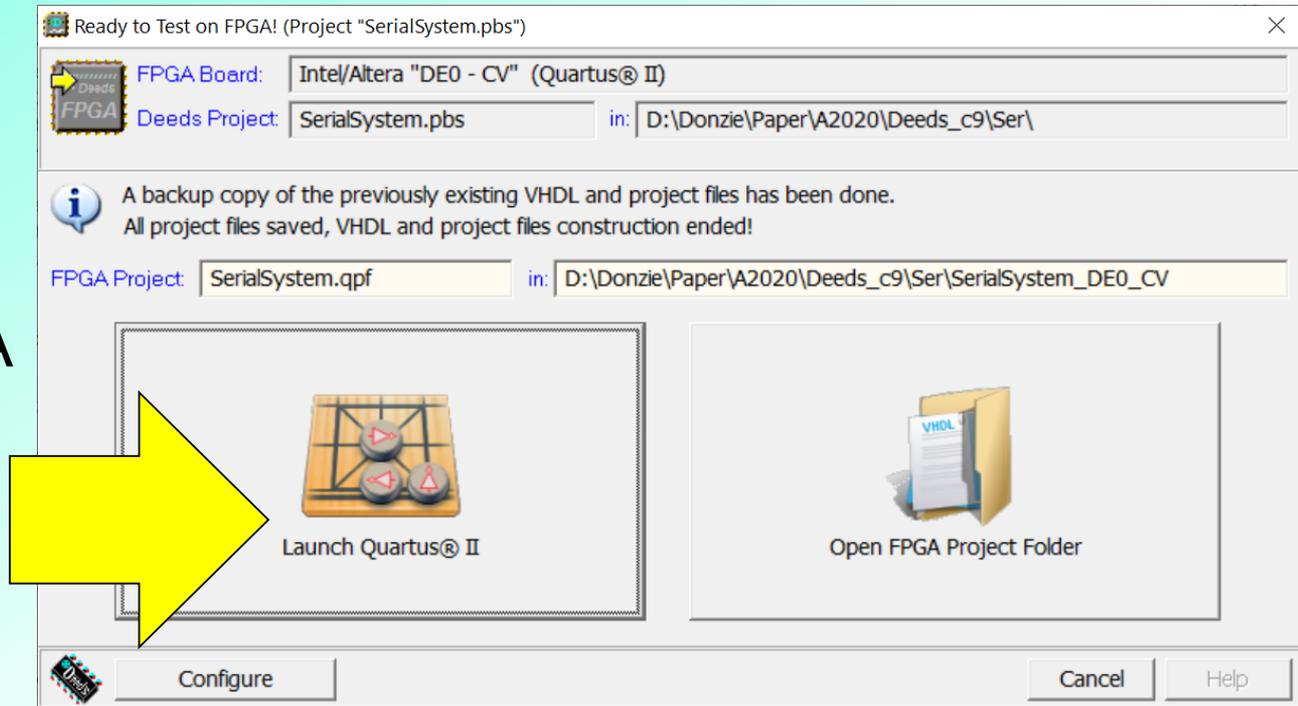


```
Test On FPGA (Project: "SerialSystem.pbs")
FPGA Board / Brand: Deeds Project Folder: D:\Donzie\Paper\A2020\Deeds_c9\Ser\
Intel/Altera "DE0 - CV" (Quartus® II) FPGA Project Subfolder: D:\Donzie\Paper\A2020\Deeds_c9\Ser\
Fsm1_TX_FSM.vhd | Fsm2_RX_FSM.vhd | Components.vhd | SerialSystem.vhd
ENTITY Multiplexer_2_1 IS
    PORT( I0: IN std_logic;
          I1: IN std_logic;
          S0: IN std_logic;
          Q: OUT std_logic );
END Multiplexer_2_1;
-----
ARCHITECTURE behavioral OF Multiplexer_2_1 IS
BEGIN
    Q <= I0 when (S0 = '0') else
        I1 when (S0 = '1') else 'X';
END behavioral;
-----
library ieee;
use ieee.std_logic_1164.all;
Messages:
(Info) |--- "Fsm2_RX_FSM.vhd" file saved (a backup copy has been created) .
(Info) |--- "Components.vhd" file saved (a backup copy has been created) .
Board Assignments | Project Files
Assignment Reset | Assignment Summary | Modify Assignment | Test On FPGA | Close | Help
"Components.vhd"
```



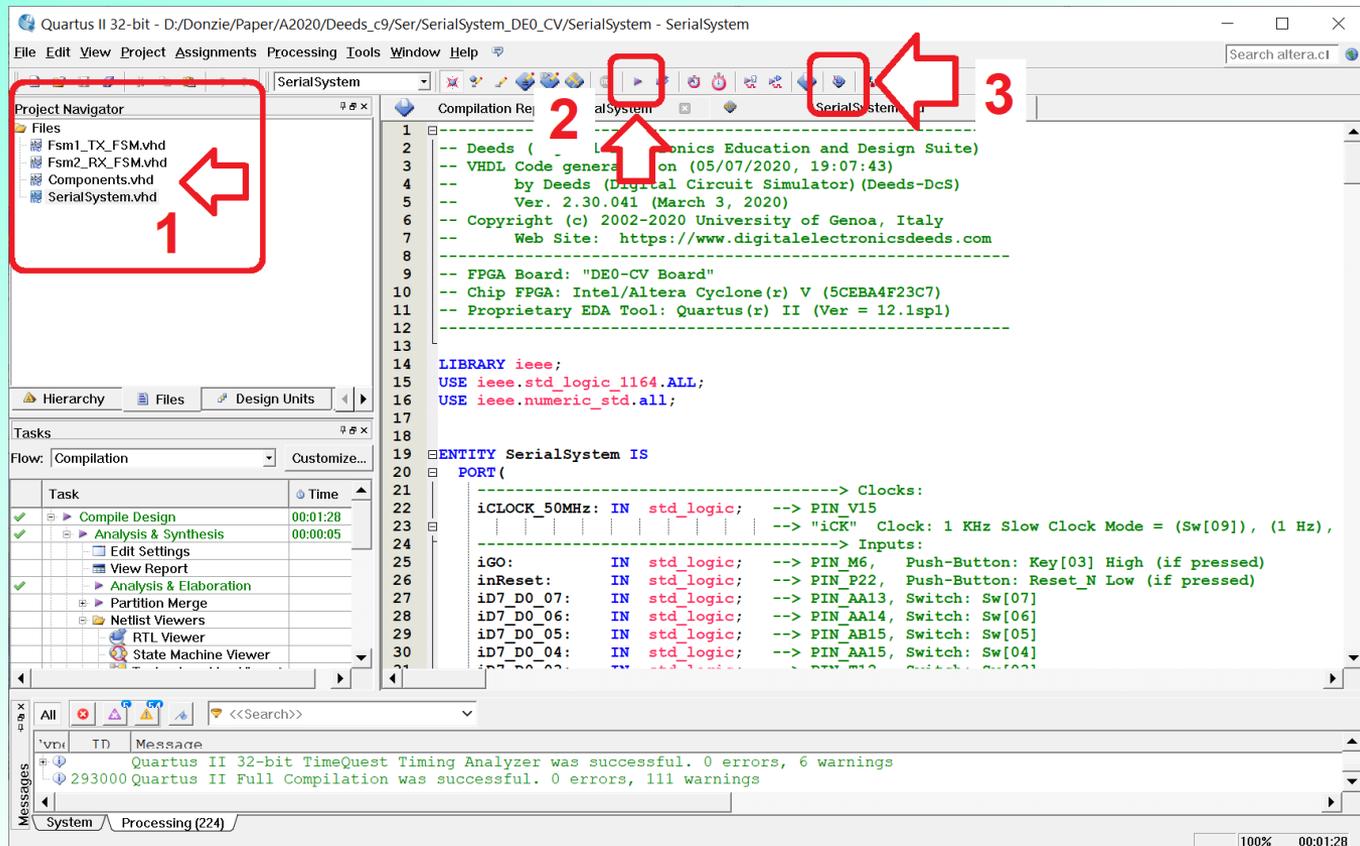
Now we can launch the FPGA tool

- According to the associations between the schematic and the board devices, Deeds defines all FPGA pins and board connections (in the generated project files).
- Once terminated the project generation, the user is invited to launch the FPGA software specific for the chip (Quartus® II).



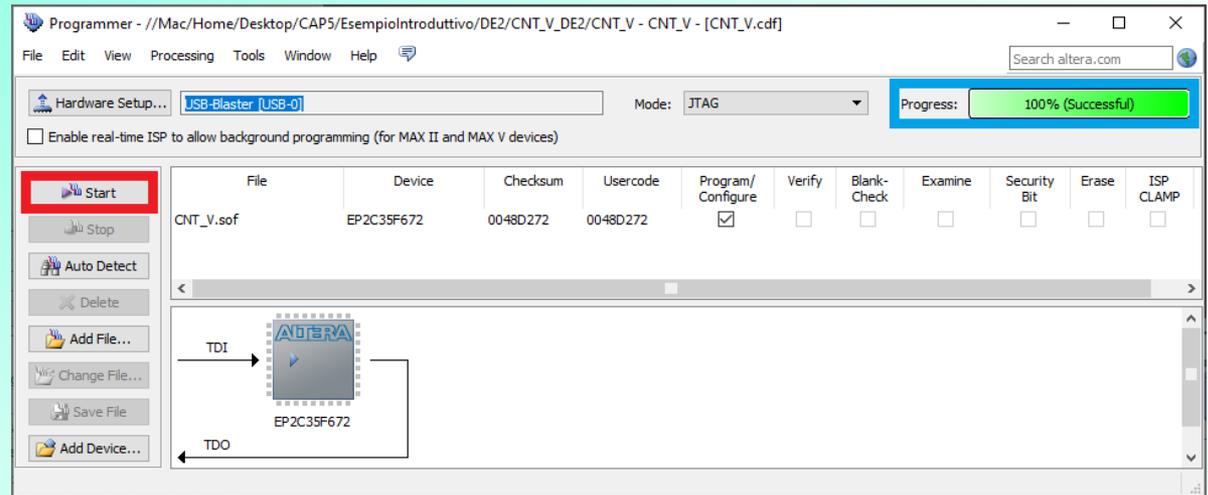
The role of the specific FPGA tool

- Now we have the project files opened in Quartus® II, ready to be processed (1).
- The student needs only to compile them (2); *no operation on the VHDL code* is needed by the beginner.
- Finally, the student will load the project binaries into the FPGA board, using the programmer command (3).



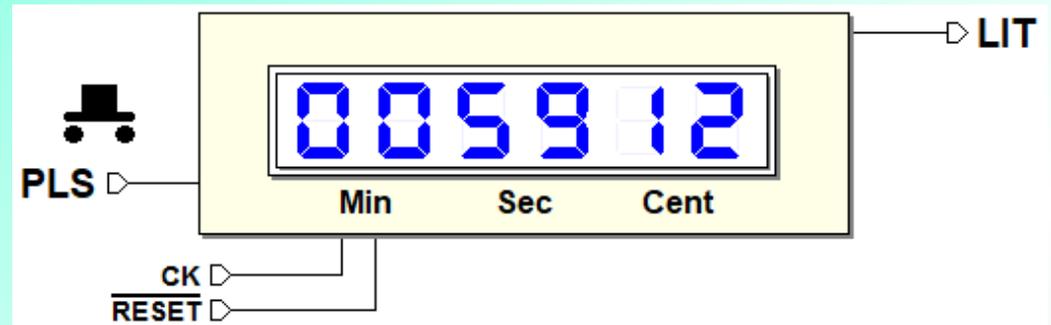
FPGA programmer module

- The *Programmer* software module allow to load the binaries into the FPGA chip, through its JTAG interface.
- Normally, a USB serial port is used to program the FPGA boards.
- The programming hardware can be already on board, or external.



A laboratory session: an example

- We propose to the students to design and test on FPGA a *digital chronometer* (*).
- Requested resolution is one hundredth of a second, maximum time measurable about one hour.
- Display will show six decimal digits, two for the minutes, two for the seconds and two for the hundredths of second.
- When the pushbutton PLS is pushed and then released, it start counting.
- On the second pressure it stops and the time will be readable on the displays; the third pressure resets all.
- The lamp LIT is activated for all the time PLS is pushed.
- The frequency of the clock is 100 Hz.
- We implement it on the DE0-CV FPGA board (Terasic/Altera).

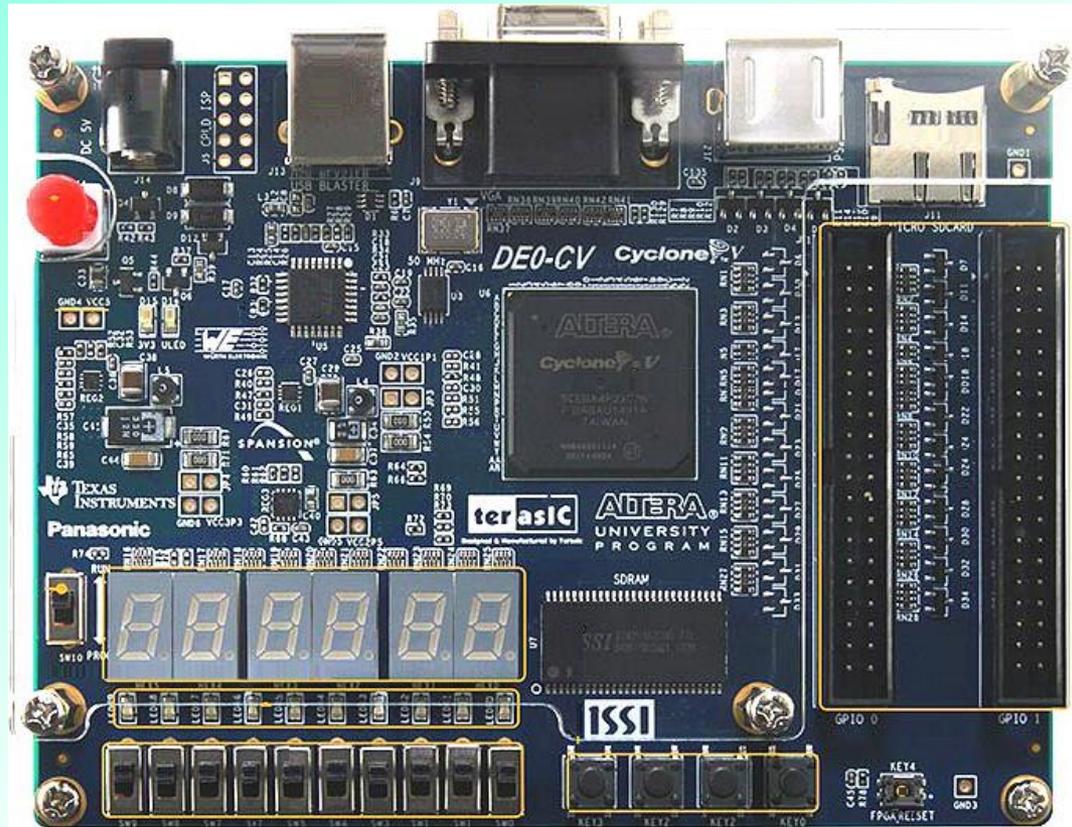


(* *This exercise is proposed in the book “Introduction to Digital System Design”:
[click here to download the files from the online support](#)*



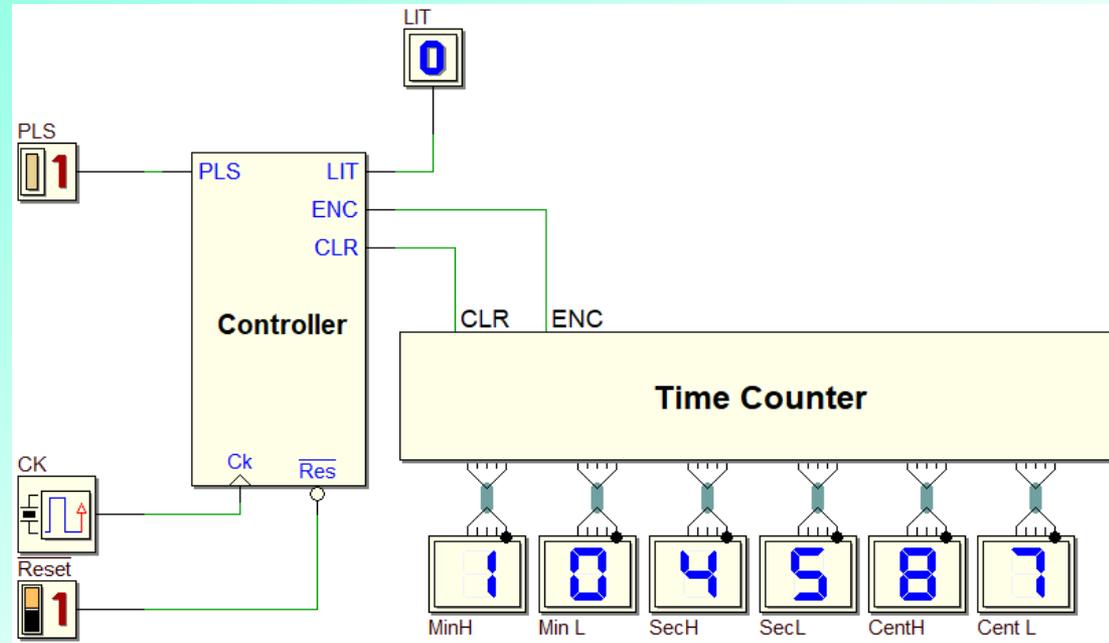
The FPGA Board

- The DE0-CV board makes available the required six seven-segments displays, push-buttons, switches and LEDs.
- The native 50 MHz clock generator will be scaled down automatically by Deeds, according to the project setting.

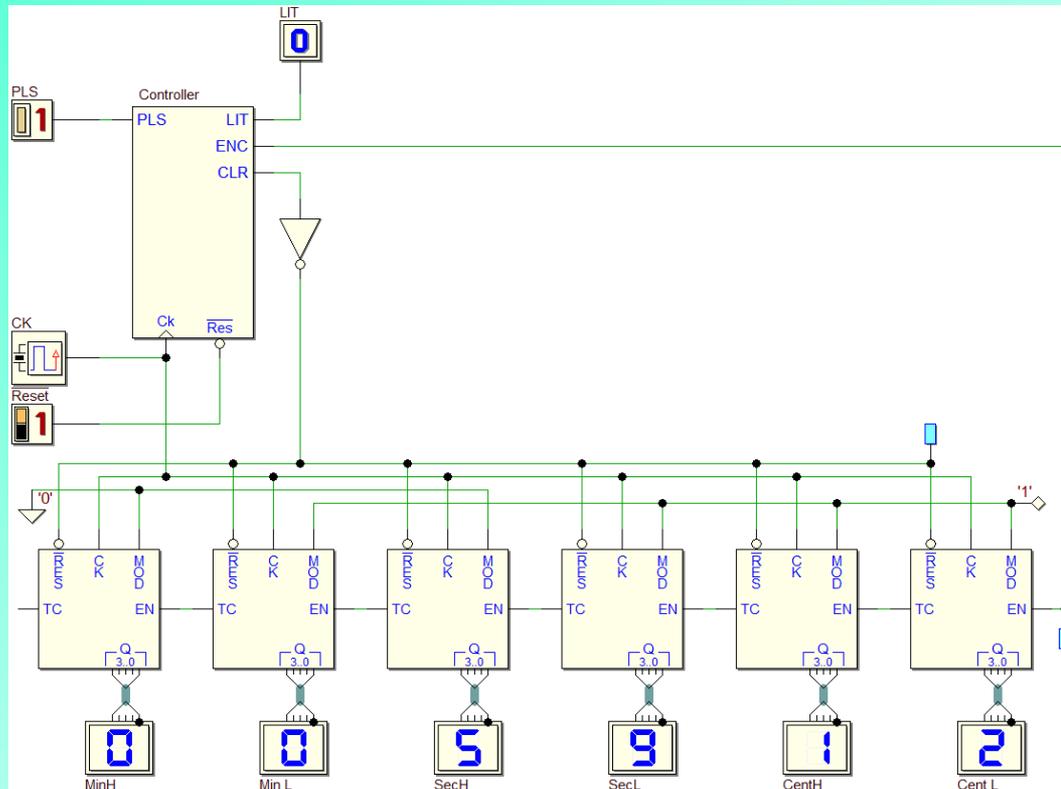


Design guidelines

- A controller-datapath structure is suggested to the student.
- The datapath includes the time counter and the associated displays.
- The controller reads the button PLS and handles the lamp LIT and the time counter controls.
- The time counter is cleared by CLR and will count when ENC='1'.
- We suggest to implement the controller as a Finite State Machine (FSM) that will track the PLS input.



Time counter architecture

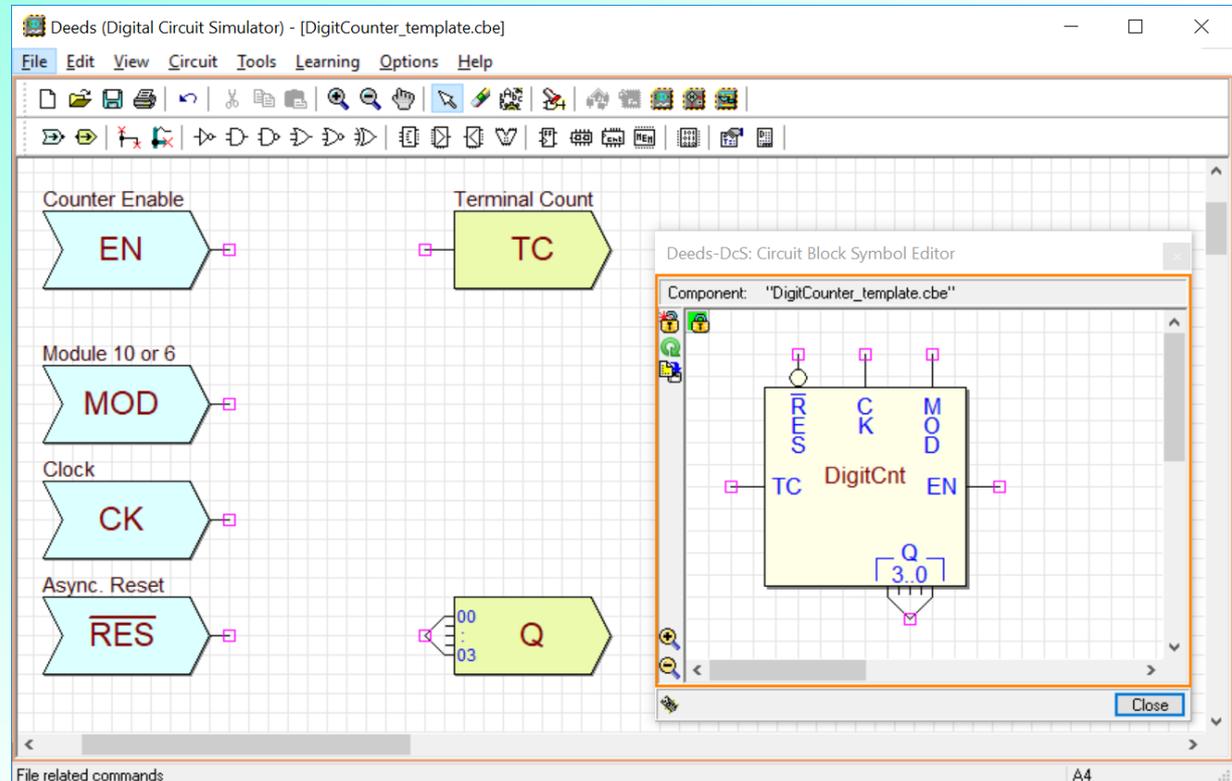


- The suggestion given to the students is to divide the counter in six BCD (Binary Coded Decimal) sub-modules.
 - The interior of each module is left to the creativity of each one.
 - In Deeds, each module element can be defined as a CBE (Circuit Block Element).
- Each module receives a count enable EN from the one on the right side, and generates a TC (Terminal Count) to enable the one placed on its left side.



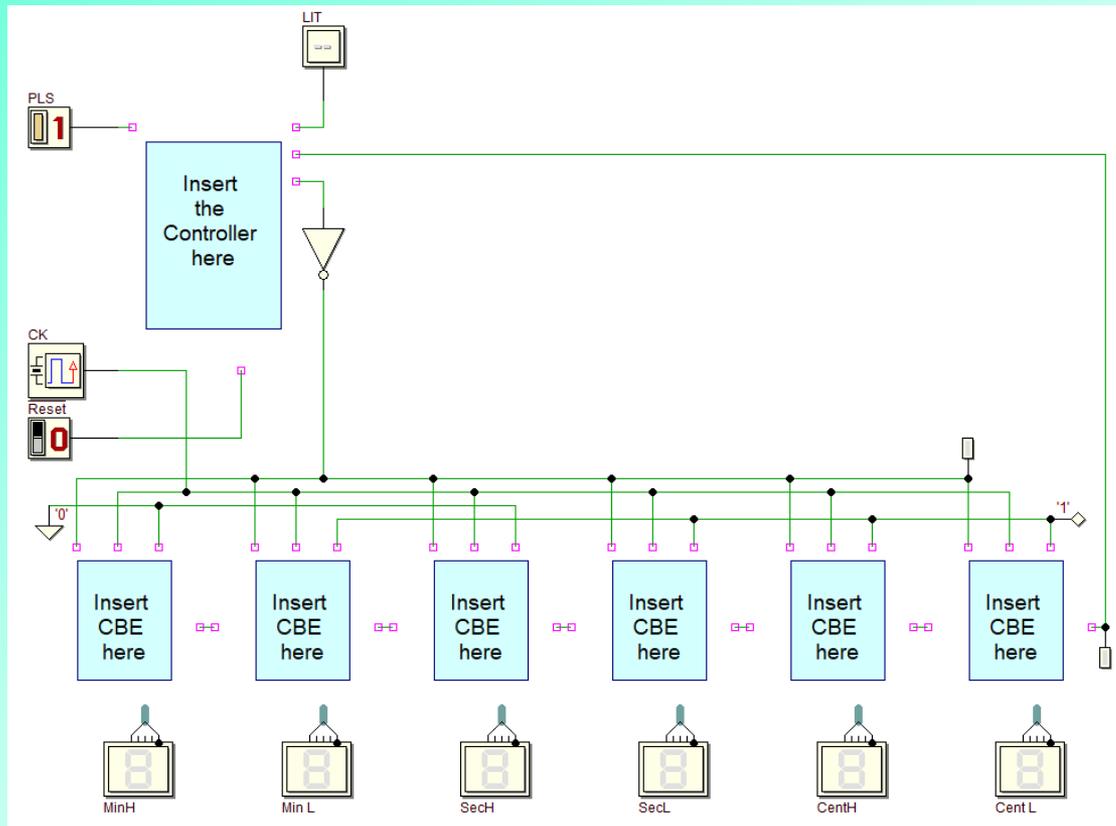
The 'templates' to speed up the design (1)

- A circuit 'template' can speed up the student work.
- The general uniformity given by templates is useful for the teacher too, to simplify checking of the quality of designs.
- CBE blocks are perfect to define circuit templates.
- In the figure, the CBE component template.
- I/O pins, and the component symbol are already defined.



The 'templates' to speed up the design (2)

- A template of the top level circuit is also very useful. *Why?*
- In Deeds, *Input / Output* components have a special role.

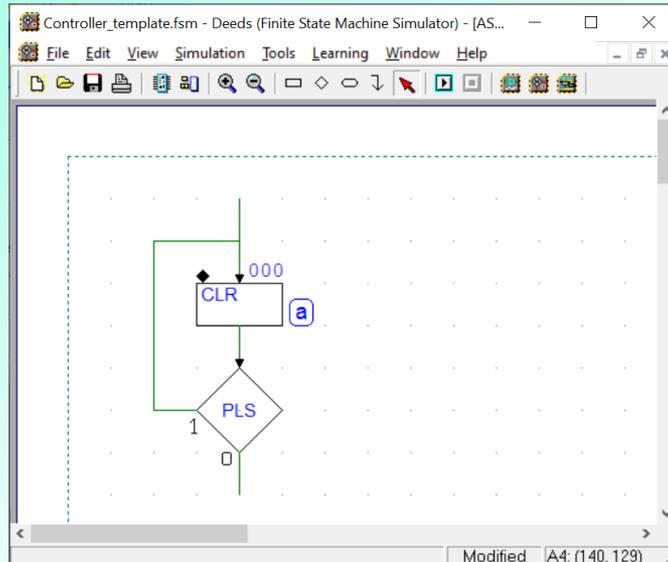
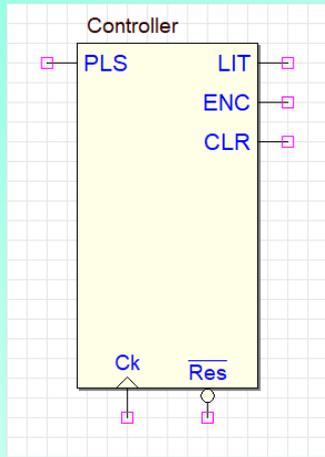


- *Timing Diagram:* Input components memorize the timing sequences used for the simulation.
- *Test on FPGA* window: Input and Output components store the association between the schematic and the physical devices on the FPGA board.



The 'templates' to speed up the design (3)

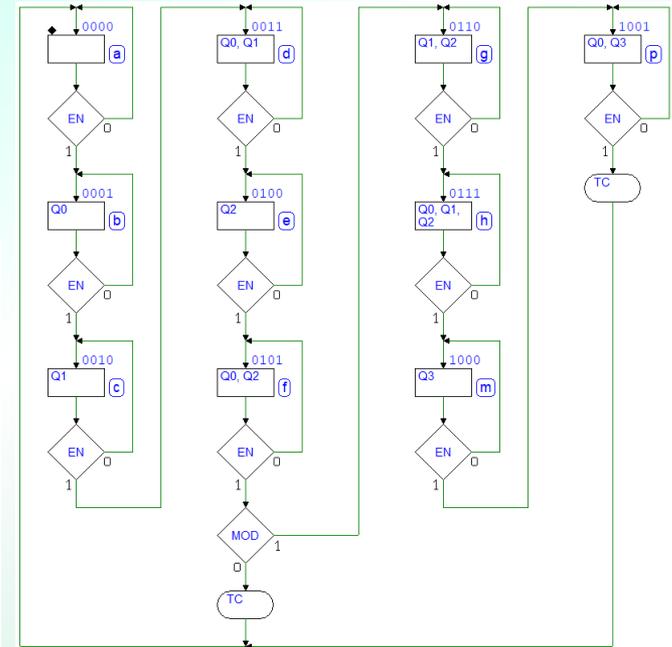
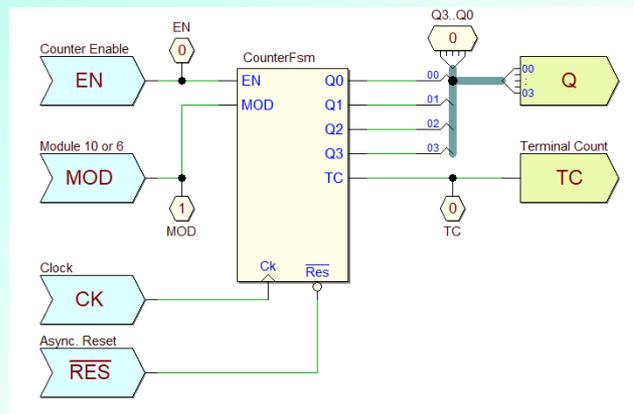
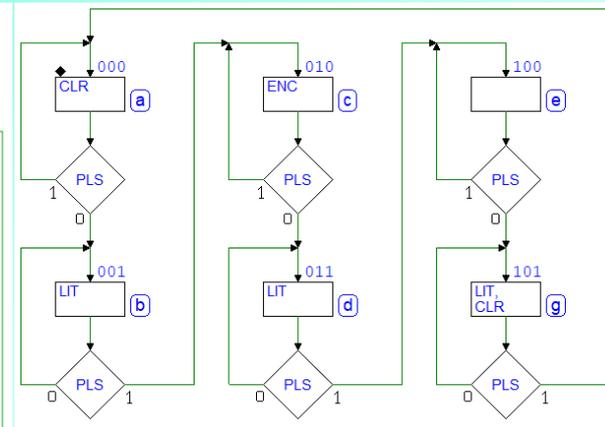
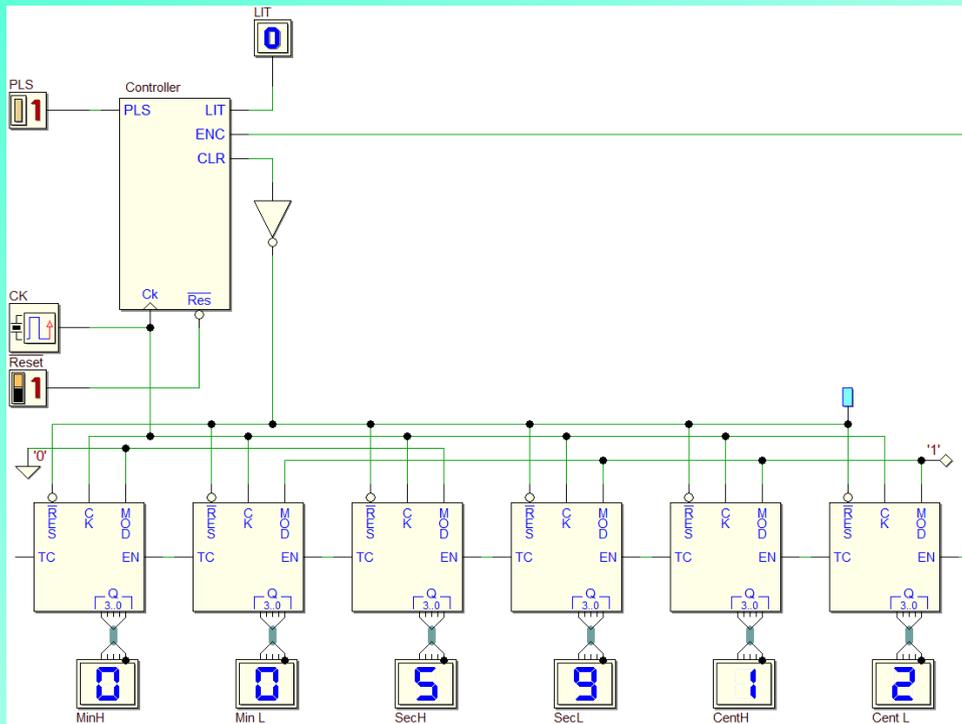
- You can give to the student a *top project file empty*, but already prepared for the FPGA prototyping.
 - You can test the solution in advance by simulating and prototyping it.
 - When all is working, it is only necessary delete from the circuit all, or part of, the circuit elements.
 - Leave in place the Input and output component, just to grant timing test sequences and the FPGA board configuration, to grant clear and comparable solutions from all the students.



- Templates can be defined also for the FSM components (see the figures aside).
- The component symbol has been defined, but the ASM-chart presents only a partial suggestion.



The final circuit of the chronometer



Specialized inputs: the programmable clock

- 'Clock'
- 'Slow Clock'
- $F_{ck} = 100\text{Hz}$
- **Sw[09]**: enables the slow mode.
- **Key[00]**: to pulse the clock in slow mode.
- **LEDR[09]**: to shows clock pulses.

The screenshot displays the Quartus II IDE interface for a project named "ChronometerF.pbs". The top panel shows the board configuration for "DE0 - CV (Altera Corporation / TerASIC)". The "On Board Clock Resources" section is highlighted with a red box, and a yellow arrow points to the "Clock Generator Frequency" dropdown menu, which is set to "100 Hz". Below this, the "Slow Clock Mode" is enabled by "Sw[09]". The "Clock Pulses" are set to "Key[00]" and the "Led" is "LEDR[09]". The bottom panel shows a schematic diagram of the clock circuit, with a red box around the "Key[00]" input and a green arrow pointing to the physical KEY[00] button on the board. Another yellow arrow points from the "Slow Clock Mode" dropdown to the physical SW[09] switch on the board. A red box highlights the LEDR[09] LED on the board, with a yellow arrow pointing from the "Led" dropdown in the software to it. The board image shows the FPGA, switches, buttons, and LEDs.

(The associations have been already all defined in the template)



Exporting the VHDL project

- Now students are ready to launch the specific FPGA tool to compile the FPGA project files (Quartus® II).

The screenshot shows the Quartus II software interface with the 'Test On FPGA' dialog box open. The dialog box displays the following information:

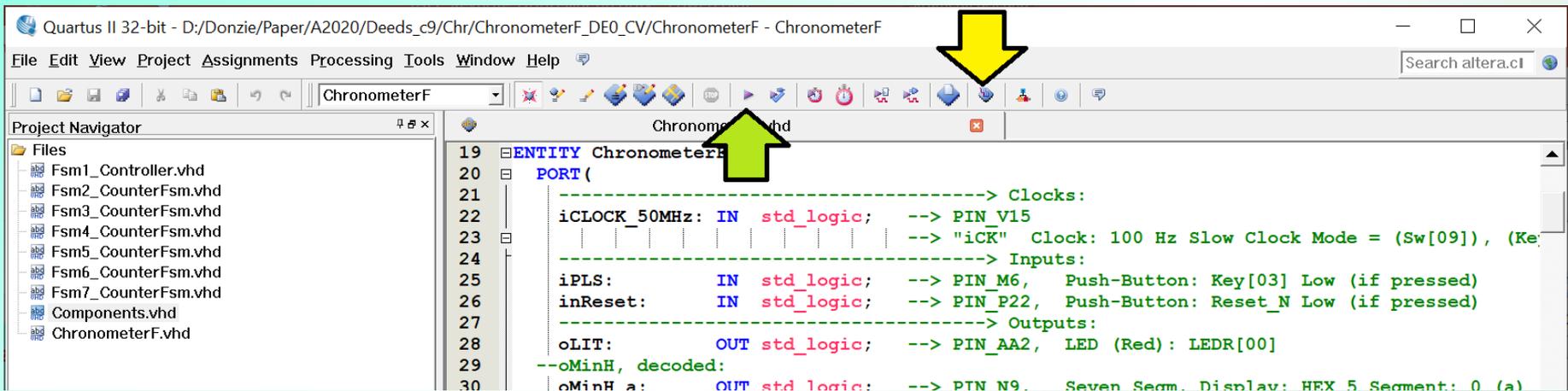
- FPGA Board / Brand:** Intel/Altera "DE0 - CV" (Quartus® II)
- Deeds Project Folder:** D:\Donzie\Paper\A2020\Deeds_c9\Chr\
- FPGA Project Subfolder:** D:\Donzie\Paper\A2020\Deeds_c9\Chr\Chrono
- FPGA Board:** Intel/Altera "DE0 - CV" (Quartus® II)
- Deeds Project:** ChronometerF.pbs in: D:\Donzie\Paper\A2020\Deeds_c9\Chr\
- FPGA Project:** ChronometerF.qpf in: D:\Donzie\Paper\A2020\Deeds_c9\Chr\ChronometerF_DE0_CV

A yellow arrow points to the 'Launch Quartus@ II' button, which is highlighted in a box. Other buttons include 'Open FPGA Project Folder', 'Configure', 'Cancel', and 'Help'.



Compiling the VHDL project

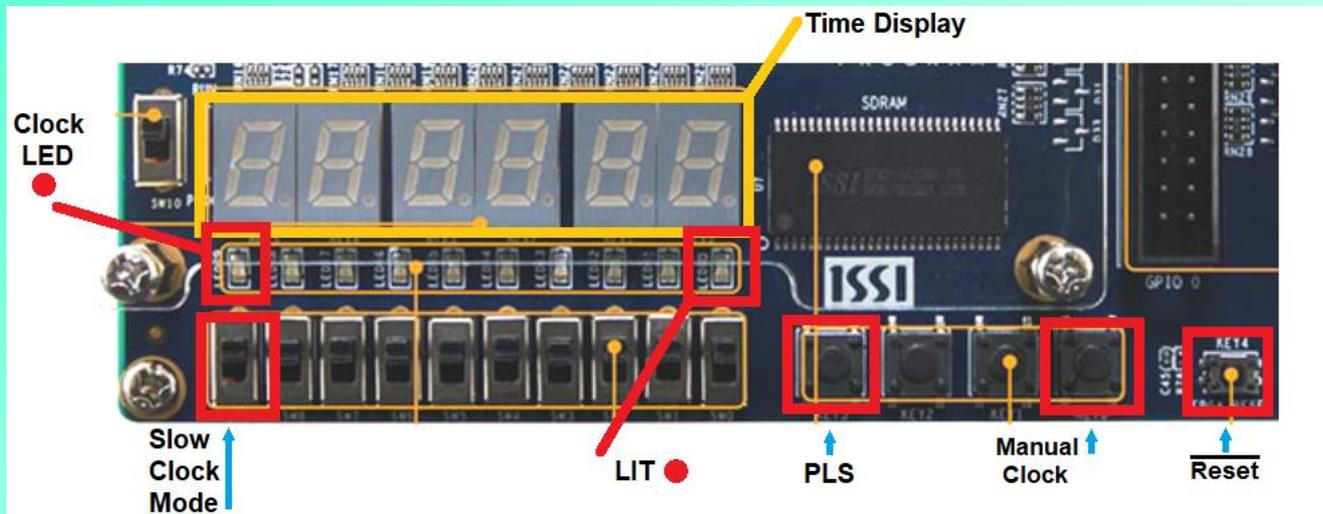
- The project in Quartus® II is ready to be compiled (green arrow).
- The top VHDL file contains the structural description of the digital network schematic.
- The other VHDL files collect the behavioral descriptions of all the components used by the network, included user-defined FSM, micro-computers, ROM contents...
- After the compilation, we are able to transfer the circuit to the FPGA using the programmer module and the USB connection (yellow arrow).



```
19 ENTITY ChronometerF
20 PORT (
21     -----> Clocks:
22     iCLOCK_50MHz: IN std_logic; --> PIN_V15
23     -----> "iCK" Clock: 100 Hz Slow Clock Mode = (Sw[09]), (Ke
24     -----> Inputs:
25     iPLS: IN std_logic; --> PIN_M6, Push-Button: Key[03] Low (if pressed)
26     inReset: IN std_logic; --> PIN_P22, Push-Button: Reset_N Low (if pressed)
27     -----> Outputs:
28     oLIT: OUT std_logic; --> PIN_AA2, LED (Red): LEDR[00]
29     --oMinH, decoded:
30     oMinH_a: OUT std_logic; --> PIN_N9, Seven Segm. Display: HEX 5 Segment: 0 (a)
```

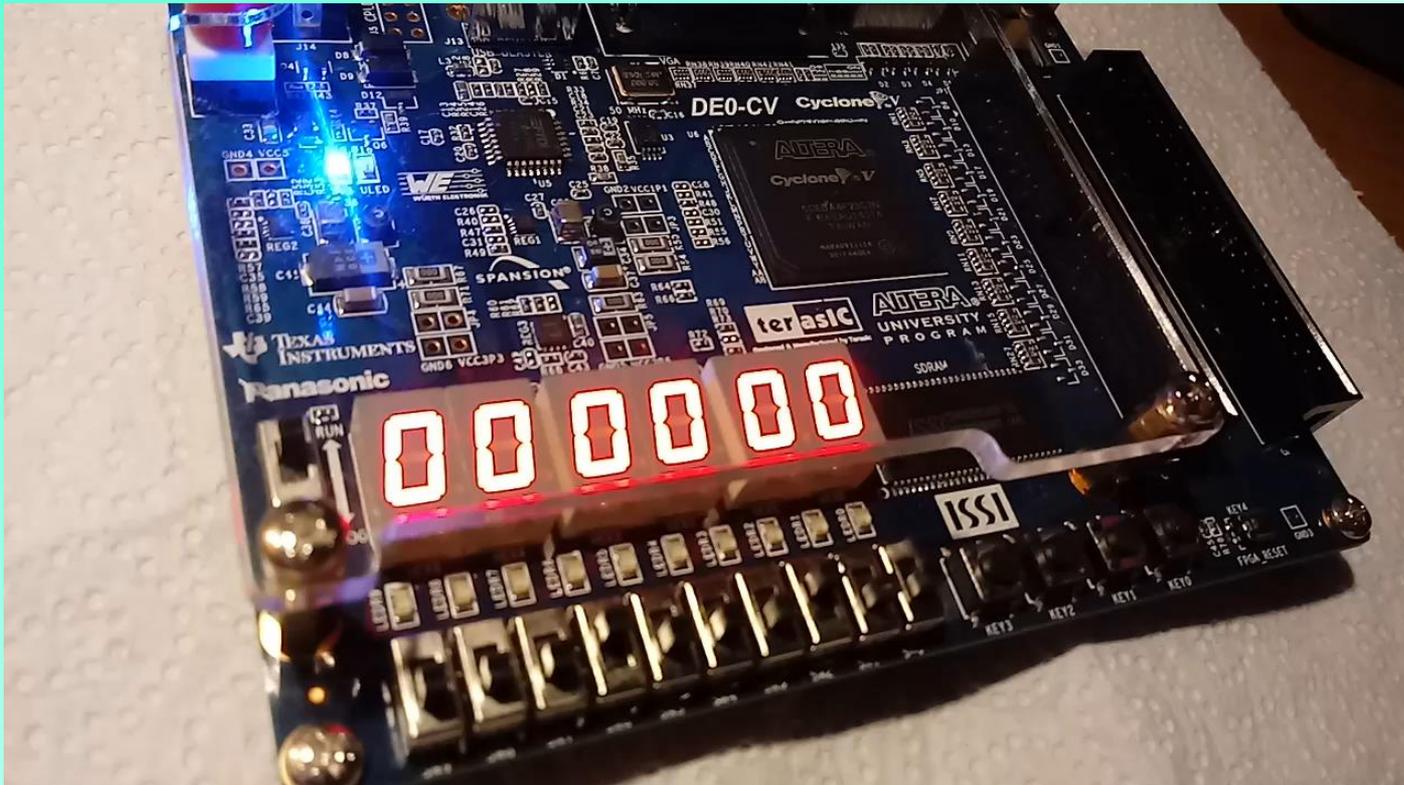


Testing the 'physical' chronometer



- Laboratory assignments always include a board synoptic view of the I/O of the circuit under test.
- Switches, push-buttons and LEDs of the FPGA board are put in evidence to facilitate the testing of the circuit.

The chronometer at work



Thank you for the interest and participation!

<https://www.digitalelectronicsdeeds.com>

