E-content in the Learning Environment for Design and Verification of Communication Circuits

Galia I. Marinova, *Member, IEEE*

Abstract — **The paper presents the web-based e-content in the Learning environment for design and verification of communication circuits (LEDVCC). Several examples illustrate the e-content – the first example shows the importance of the priorities in the FSM specification and the second example illustrates a full cycle of design and verification in LEDVCC which includes VHDL description, simulation, realization and measurement of a demultiplexor.**

Keywords **— CPLD, Design and verification, E-content, Learning environment, VHDL.**

I. Introduction

THE Learning environment for design and verification of communication circuits (LEDVCC) is realized in the Laboratory for Computer-aided design in the Telecommunications faculty of Technical University – Sofia. The application of LEDVCC for FPGA of XILINX is described in [1]. The applications in [1] are mainly concentrated on modems and cryptoprocessor cores.

The application of LEDVCC in the present paper is oriented towards CPLDs from CYPRESS family [3].

The paper presents the e-content developed to help students using LEDVCC to follow the course Computer-aided design of digital communication circuits in Telecommunications faculty and in German faculty.

The web-based e-content for LEDVCC applying CYPRESS CPLDs is presented on Fig.1. The tools integrated in LEDVCC are:

- WARP6.2 – software tool for design creation, HDL description, compilation, synthesis, fitting and simulation. The elements of WAPR6.2 are:
  - GALAXY – VHDL editor;
  - ACTIVE-FSM – graphical FSM editor;
  - ACTIVE-HDL Sim – Simulator.
- ISR – In-system-reprogrammable software tool
- Development board with two CYPRESS CPLDs – DELTA39K and ULTRA 37000
- Generator of test signals MS-9160
- Logical analyzer MAX-8100

The web based e-content helps students to go step-by-step through different design and verification cycles in LEDVCC and to solve different examples beginning with design creation, description in VHDL text or FSM graph format, stimulus definition, synthesis, simulation, fitting, programming and measurement.

II. E-content in LEDVCC

The e-content for LEDVCC is available in three languages – Bulgarian, English and German, which can be found on the web addresses:
http://www.pueron.org/pueron/Sreda_Cypress.htm
http://www.pueron.org/pueron/en/LEDVCC.htm
http://www.pueron.org/pueron/inovazionniproekti/Sreda_CYPRESS/sreda_Cypress_DE.htm

German version is prepared for a special course in the German faculty in TU-Sofia.

The e-content includes the following information:

- Step-by step instructions for WARP6.2 applications:
  - Creating a new design in WARP6.2;
  - Simulation of a design in ACTIVE-HDL Sim;
  - Creation of FSM specification in ACTIVE-FSM;
  - Automatic conversion of the FSM specification in VHDL description for a given project.
- Step-by-step instructions for ISR applications:
  - Definition of a design to be programmed on CYPRESS board;
  - In-system reprogramming of a design;
  - Connection of CYPRESS board to PC by the programming cable;
  - Connection of CYPRESS board to power supply using an adapter for 3.3V or 5V;
  - Configuration of jumpers on the CYPRESS board to define which CPLD(s) to be programmed;
  - Additional operations: Erase device, Bypass device, Verify pattern, Read Silicon ID, Read user code, Program security bit, Read device.
- Step-by-step instructions of measurement and test of the programmed CYPRESS board:
  - Identification of CPLD pins and signals associated using the REPORT file *.rpt outcoming from design compilation in ACTIVE-HDL Sim.;
  - Connection of CYPRESS board to generator and logic analyzer;
  - Definition of test-vectors from the generator;
  - Collection of the output signals at the logic analyzer;

G. I. Marinova is with the Telecommunication faculty, Technical University of Sofia, Bulgaria (phone: 359-2-9653188; e-mail: gim@tu-sofia.bg).

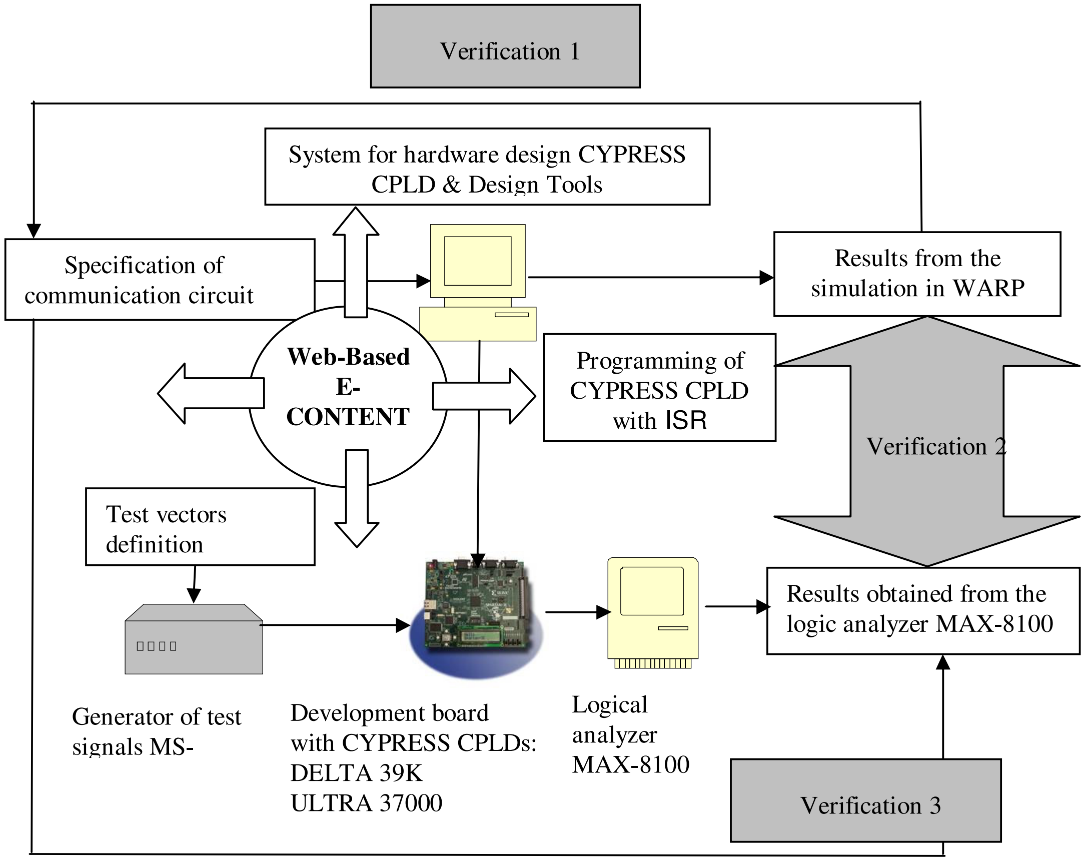

Verification 1
System for hardware design CYPRESS CPLD & Design Tools
Specification of communication circuit
Results from the simulation in WARP
Web-Based E-CONTENT
Programming of CYPRESS CPLD with ISR
Verification 2
Test vectors definition
Results obtained from the logic analyzer MAX-8100
Generator of test signals MS-
Development board with CYPRESS CPLDs: DELTA 39K ULTRA 37000
Logical analyzer MAX-8100
Verification 3

Fig. 1. Web-based E-content in the Learning environment for design and test of communication circuits

- Measurement data transfer from the logic analyzer to the PC;
- Analysis of the results obtained.

-

- Data formats and data conversion in LEDVCC:

  - projects in the editor GALAXY - *.pfg;
  - VHDL files in the text editor - *.vhd ;
  - FSM files in ACTIVE-FSM graphical editor- *.asf;
  - JEDEC and REPORT formats from the compilation in ACTIVE-HDL Sim - *.jed and *.rpt;
  - JAM and STAPL files for in-system-reprogramming;
  - *.isr and *.log files from the ISR tool;
  - *.max files from the logic analyzer MAX-8100.

- Design examples:

Multiplexor, Demultiplexor, D-latches (positive edge triggered, negative edge triggered, gate latches, with Set and Reset with different priorities), JK-flip-flops, Registers, Counters, Divider by 2, Half adder, Full adder, Adders (2,3,4,8,16-bits), Rotation of 8-bit vector etc.

Some of the examples in the e-content come from [2]. Each design example is structured as follows:

- Specification of the project;
- VHDL description in the test editor GALAXY;
- Definition of input signals using stimulus;
- Frequencies and timing parameters of test-vectors;
- Simulation results obtained from ACTIVE-HDL Sim;
- Some comments on the design behavior.

III. Examples from the E-content in LEDVCC

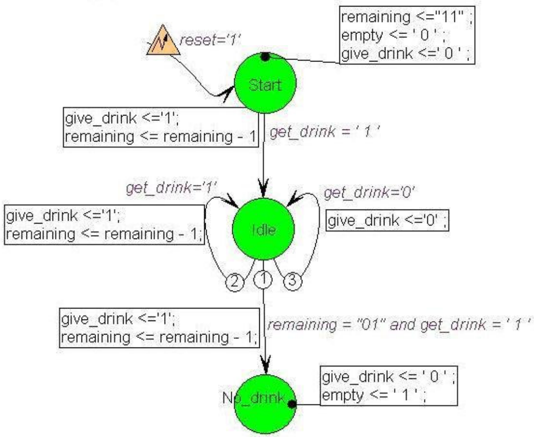
The step-by-step instructions illustrate through examples the three types of verifications in LEDVCC:

- verification 1 – simulation results versus specification;
- verification 2 – measurement versus simulation results;
- verification 3 – measurement versus specification.

Two illustration examples are given in the next section: Fig.2 and 3 present an example which illustrates the influence of changing priorities in the FSM specification. Fig.4 and 5 present an example which illustrates the full cycle of design, synthesis, fitting, simulation, programming and test of a demultiplexor in LEDVCC. Then the home page of LEDVCC is given.

A. Changing priorities in the FSM specification

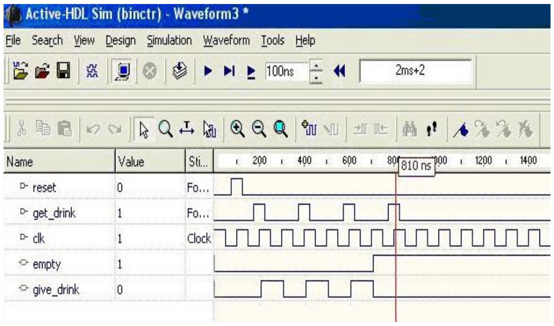
The task is to design a machine which gives a drink (give_drink='1') from a bin which contains 3 drinks. If the bin is empty a signal *empty* becomes *"1"*. The FSM graphs from Fig.2a and 3a differ from each other by the priorities 1,2,3. Fig.2b and 3b present the VHDL codes obtained from the corresponding FSM graphs. Fig. 2c and 3c present the simulation results in both cases. The FSM from Fig.2 generates a signal *empty='1'* when the bin is empty. The FSM from Fig.3 generates a signal *give_drink='1'* when the bin is empty, which is a wrong behavior.



(a) FSM graph with correct priorities

```
curstate_machine: process (CLK, reset)
begin
   if reset='1' then
   remaining <="11" ;
   empty <= ' 0 ' ;
   give_drink <=' 0 ' ;
curstate <= Start;
elsif CLK'event and CLK = '1' then
   case curstate is
     when Idle =>
      if remaining = "01" and get_drink = ' 1 '
then
    curstate <= No_drink;
    give_drink <='1';
    remaining <= remaining - 1;
        elsif get_drink='1' then
    curstate <= Idle;
    give_drink <='1';
```

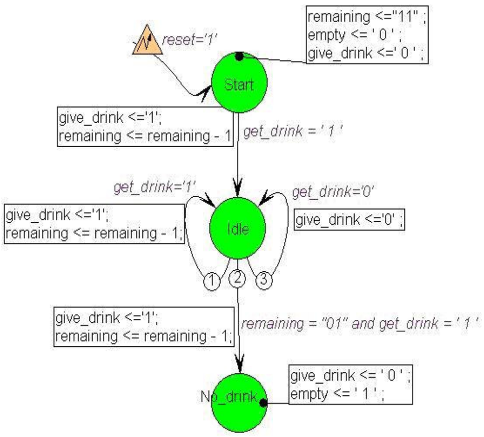
(b) VHDL code generated for the FSM with correct priorities



(c) Simulation results for the correct priorities

Fig.2. FSM graph of a drink machine with correct priorities

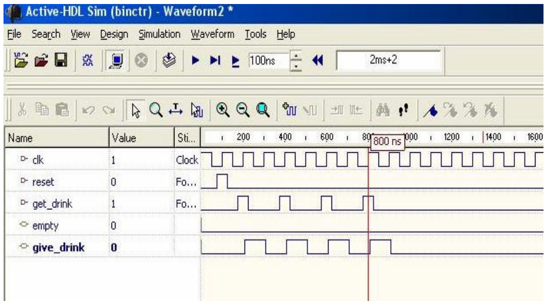
This example is a good illustration of the importance of priorities order in FSMs for the correct behavior of the designed system.



(a) FSM graph with changed priorities

```
curstate_machine: process (CLK, reset)
begin
if reset='1' then
   remaining <="11" ;
   empty <= '0' ;
   give_drink <='0' ;
   curstate <= Start;
elsif CLK'event and CLK = '1' then
   case curstate is
     when Idle =>
     if get_drink='1' then
    curstate <= Idle;
              give_drink <='1';
    remaining <= remaining - 1;
     elsif remaining = "01" and get_drink = '1'
then
    curstate <= No_drink;
    give_drink <='1';
```

(b) VHDL code generated for the FSM with changed priorities

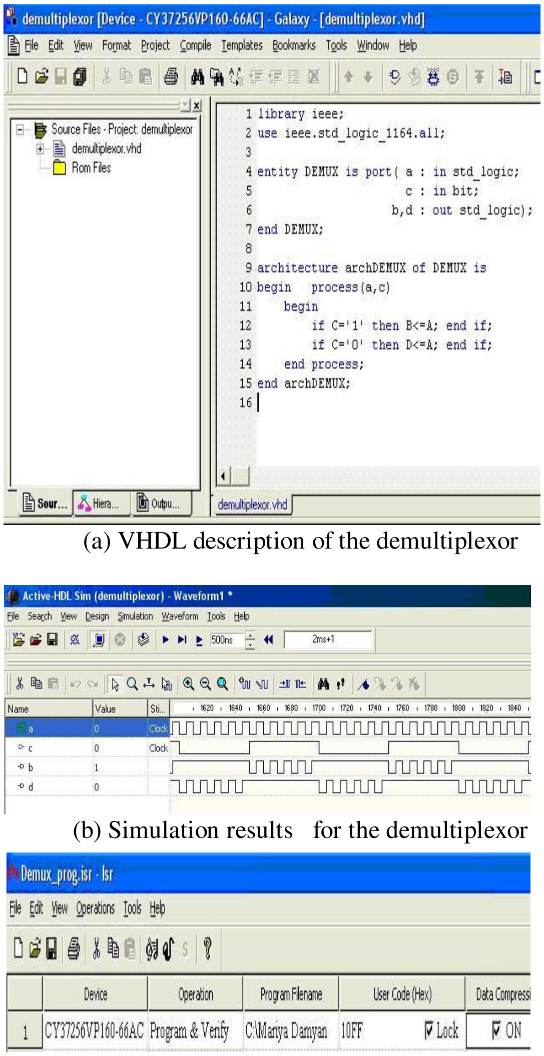


(c) Simulation results for the FSM with changed priorities

Fig. 3. FSM graph of a drink machine with changed priorities

B. Design, simulation, programming and test of a demultiplexor in LEDVCC

Fig.4a presents the VHDL description of the demultiplexor with 1 input, 2 outputs and 1 control signal. Fig.4b presents the simulation results for the demultiplexor obtained in ACTIVE-Sim, Fig.4c illustrates the definition of parameters in ISR tool for programming the demultiplexor on a CPLD. Fig 5a and 5b present the connection with signal generator and logical analyzer of the development board with the demultiplexor programmed on one of its CPLDs – CY37256VP160-66AC.


demultiplexor [Device - CY37256VP160-66AC] - Galaxy - [demultiplexor.vhd]
File Edit View Format Project Compile Templates Bookmarks Tools Window Help
Source Files - Project: demultiplexor
demultiplexor.vhd
Rom Files
Sour...
Hiera...
Outpu...
demultiplexor.vhd

```
library ieee;
use ieee.std_logic_1164.all;

entity DEMUX is port( a : in std_logic;
                      c : in bit;
                    b,d : out std_logic);
end DEMUX;

architecture archDEMUX of DEMUX is
begin   process(a,c)
    begin
        if C='1' then B<=A; end if;
        if C='0' then D<=A; end if;
    end process;
end archDEMUX;
```

(a) VHDL description of the demultiplexor

Active-HDL Sim (demultiplexor) - Waveform1 *
File Search View Design Simulation Waveform Tools Help
500ns
2ms+1
1620
1640
1660
1680
1700
1720
1740
1760
1780
1800
1820
1840

Name	Value	Sti...
a	0	Clock
c	0	Clock
b	1	
d	0	

(b) Simulation results   for the demultiplexor

Demux_prog.isr - Isr
File Edit View Operations Tools Help

	Device	Operation	Program Filename	User Code (Hex)	Data Compressi
1	CY37256VP160-66AC	Program & Verify	C:\Mariya Damyan	10FF ☑ Lock	☑ ON

(c) Programming the demultiplexor on the CPLD CY37256VP160-66AC in ISR

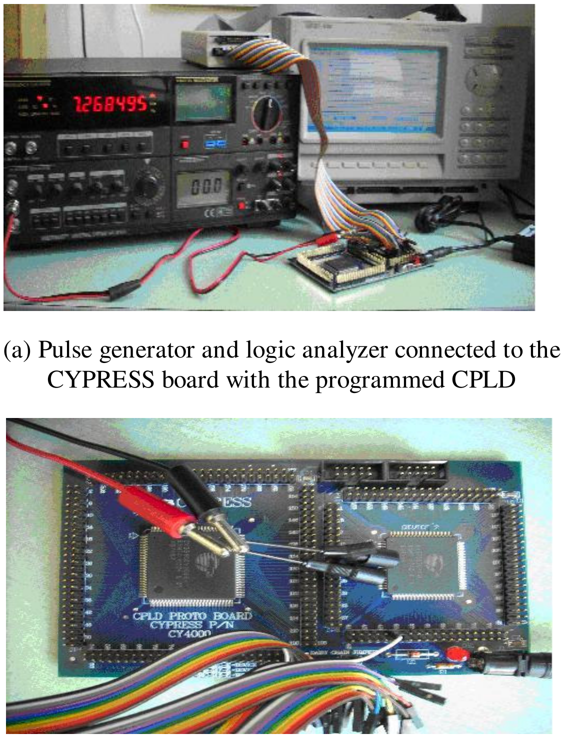
Fig.4. Design of demultiplexor in WARP6.2

C. *Home web-page*

Fig.6 presents the home web-page of the LEDVCC in Bulgarian language.

IV. Conclusion

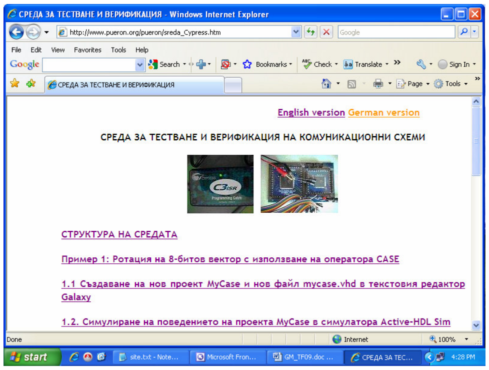
The e-content developed in LEDVCC is a good source for traditional and distance learning in computer-aided digital circuit design and verification. It is useful for students from telecommunications and computer science specialties. In future new teaching documents will be added to the e-content to enlarge its applications.



(a) Pulse generator and logic analyzer connected to the CYPRESS board with the programmed CPLD

(b) Connection of the cables to the Cypress board with demultiplexor on the CPLD

Fig.5. Measurement of a programmed circuit in LEDVCC


СРЕДА ЗА ТЕСТВАНЕ И ВЕРИФИКАЦИЯ - Windows Internet Explorer
http://www.pueron.org/pueron/sreda_Cypress.htm
Google
File Edit View Favorites Tools Help
Search
Bookmarks
Check
Translate
Sign In
СРЕДА ЗА ТЕСТВАНЕ И ВЕРИФИКАЦИЯ
Page
Tools
C3ISR
Done
Internet
100%
start
site.txt - Note...
Microsoft Fron...
GM_TF09.doc ...
СРЕДА ЗА ТЕС...
4:28 PM

English version German version

СРЕДА ЗА ТЕСТВАНЕ И ВЕРИФИКАЦИЯ НА КОМУНИКАЦИОННИ СХЕМИ

СТРУКТУРА НА СРЕДАТА

Пример 1: Ротация на 8-битов вектор с използване на оператора CASE

1.1 Създаване на нов проект MyCase и нов файл mycase.vhd в текстовия редактор Galaxy

1.2. Симулиране на поведението на проекта MyCase в симулатора Active-HDL Sim

Fig.6. Home page of LEDVCC

Acknowledgment

The author acknowledges the help of the student Marya Damyanova for her contribution to this project.

References

[1] G. Marinova, Learning Environment for Design and Verification of Communication Circuits Realized on FPGA, *ISTET 2009 (ITG-FB 217), XV International Symposium on Theoretical Engineering,* 22 - 24 June 2009, Lübeck, Germany , pp. 224-226

[2] G. Marinova, V. Guliashki, C. Fernandes, P. Daponte, M. Marinov, "Geographically distributed teaching laboratory for microelectronic design and testing", *Microelectronics education, Proceedings of the 3rd European Workshop on Microelectronic Education"*, Aix-en Provence, France, May 18-19, 2000, *Klewer Academic Publishers*, ISBN 0-7923-6456-2, pp.131-134

[3] www.cypress.com