DESIGNING OF A STATE MACHINE WITH PROGRAMMABLE LOGIC

Ivan Simeonov Simeonov

Computer Systems and Technologies, Technical University of Gabrovo, 4 Hadji Dimitar Str., 5300 Gabrovo, Bulgaria, phone: +359 66 223 479, e-mail: isim@tugab.bg

In this paper is presented the development of an abstract state machine Mealy automat defined with table. For this purpose is used a technology for designing with Programmable Logic- Complex Programmable Logic Devices (CoolRunner-II XC2C128), based on limited version of the development system Xilinx ISE WebPACK 7.1i. The project is based over the family CoolRunner-II XC2C128 from the new generation of the CPLD devices, developed by Xilinx. For the definition of the automat is used graphical approach in the face of StateCAD(r). The work of the developed final state automat is simulated with the help of ModelSim XE III 6.0a. The CPLD device is configured in laboratory environment "in-system programmable". The programming of the system is done according to the IEEE 1149.1 JTAG standards, by means of specialized programmable device CPLD CoolRunner-II XC2C128 Design Board, which is connected with JTAG (Joint Test Action Group) cable for communication with the parallel port (LPT) of the personal computer.

Keywords: Programmable logic, WebPack ISE, VHDL, CPLD, Mealy

1. INTRODUCTION

The digital schematic technology constantly goes through fast evolution and improvements and there is constant need from uninterrupted, exhausting overtakes of the novelties. Very often the digital circuit's professional is stand against the need to synthesize or make betterments in some scheme and device. This in practice requires knowledge and be skilful at base theory for synthesis and analysis of digital circuits and over this base to apply new, modern and contemporary technologies. Such is the technology for designing with programmable logic - Complex Programmable Logic Devices (CPLDs) and Field Programmable Gate Arrays (FPGAs). Its applying is based on the use of integrated environment, which is offered by many known world firms. One of these firms is Xilinx. With the help of its software environment WebPack ISE, it is possible to go through all stages of designing and ends with loading of created project into the chosen chip. Together with this there is achievement of high level of automation of the process of designing.

Always in the education or in the practice there is need to make synthesis of Sequential schemes (SS). In relation with this is the purpose of the current paper and namely, to show one approach to make synthesis of Mealy final state machine by means of build in resources and potentialities in WebPack ISE by Xilinx. Reason for this is still low speak up for practical applying in our country of the contemporary electronically reprogrammable devices (in the face of CPLD and FPGA) [1 - 3].

2. DESCRIPTION OF THE STATE MACHINE DESIGNING PROCESS

The Mealy state machine which structure scheme will be synthesized is set with *Table 1-1* (table of transitions and outputs).

The Coding of the input signals (X), internal states (Q1 and Q2) and output reactions (Z_1^* and Z_2^*) of the Mealy state machine is shown accordingly in *Table 1-2*, *Table 1-3* and *Table 1-4*.

Table 1-1
Table of the transitions and outputs of the Mealy state machine

$X_i \backslash A_j$	a_0	a_1	a_2	a_3
X_1	a_1/Z_1	a_2/Z_3	a_0/Z_2	a_2/Z_1
X_2	a_2/Z_4	a_3/Z_2	a_0/Z_3	a_1/Z_3

Table 1-2
Coding of input signals X of the state machine

	X
X_1	0
X_2	1

Table 1-3
Coding of internal states Q1 and Q2 of the state machine

	Q_1	Q_2
a_0	0	0
a_1	1	1
a_2	0	1
a_3	1	0

Table 1-4
Coding of output reactions Z1* and Z2* of the state machine

	Z_1^*	Z_2^*
Z_1	0	0
Z_2	0	1
Z_3	1	0
Z_4	1	1

Note: With the purpose output reactions Z_1 and Z_2 to have difference by the physical outputs of the scheme, the last are marked with subscripts 1 and 2 and superscript star (*).

Possible approaches for designing of the state machine are:

- applying of no automated, classical, canonical method for synthesis of the state machine structure scheme, after what follows its practical realization and studying of the scheme;

- designing with programmable logical devices by firm Xilinx, in the development environment ISE WebPack with use of the synthesizer Xilinx Synthesis Technology (XST) – Schematic (Principal electronic circuit); State Diagram (Diagram of the final state machine), VHDL Module (Module which is wrote by means of VHDL with compound VHDL code, describing the functioning of the state machine) and by using of one standard Electronic Design Interchange Format (edif) file, generated by other software application.

For the designing ot the state machine set by Table 1-1 is used graphical approach in the face of StateCAD(r) in the development environment ISE WebPack 7.1i.

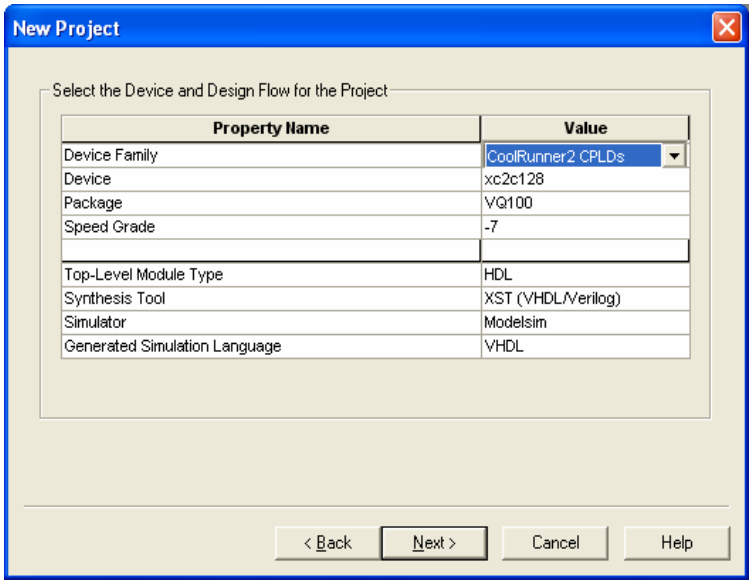

Fig. 1. The parameters of the new project

The next procedure for work is applied: **Project Navigator** is started from the menu **File → New Project.** In the field **Project Name** the name of the project has to be given (MY_SC). ISE creates a directory with this name in the place, pointed by the field **Project Location.**

In the field **Top-Level Module Type** is pointed the desired variation for represent of the project (HDL-represent by means of language for describing of hardware (VHDL)).

In the next window (fig. 1) the next parameters have to be entered: Device Family (family integrated circuits for the project - *CoolRunner2*); Device (current device from the family - *XC2C128*) [5]; Package (package -*VQ100*); Speed grade (Minimal delay from pin to pin -*7*); Top-Level Module (Variation for represent of the project); Synthesis Tool (*XST*); Simulator (Program-simulator – *ModelSim*); Generated Simulation Language (VHDL) [6].

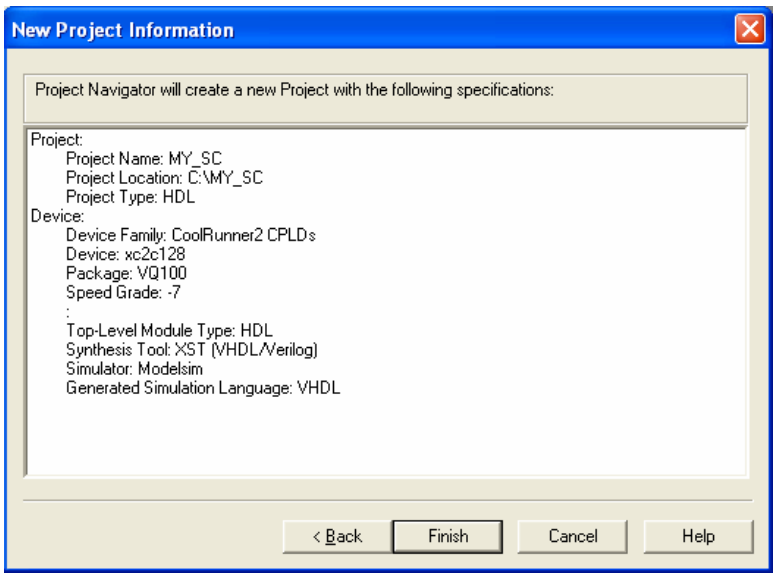

Fig. 2. Statistics for the new project

The next two windows are jumped over with pressing of the button **Next>.** In the display is visualized the window *New Project Information* (fig. 2). There are listed chosen parameters for the new project (name, path, type of represent, type of the device and other). The button **Finish** has to be pressed to open the main window of the environment for creation of new file to current project MY_SC. There from the menu **Project → New Source** opens new window **New Source** (фиг. 3), where are listed the next abilities for creation of the file (type of the file and description): IP (CoreGen - file containing IP core; Schematic - principal electronic circuit; State Diagram – Diagram of a state machine; Test Bench Waveform – Graphical set testing sequences; User Document – Text document; Verilog Module – Module written on Verilog; Verilog Test Fixture – Description of test module written on Verilog; VHDL Library – Library with instructions for VHDL; VHDL Module – Module written on VHDL; VHDL Package - VHDL module containing packets, and VHDL Test Bench – Description of test module on VHDL.

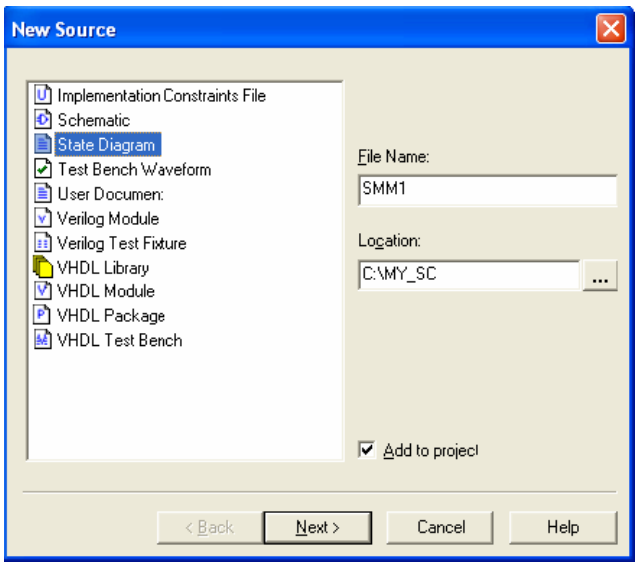

Fig. 3. Choice of the file type

There have to be entered the name of the file and the path toward it and with what tool will be created. The check box **Add to project** have to be enabled.

After pressing of the button **Next>**, on the display is visualized the window **New Source Information**. There is listed the type, the name of the file and the work directori of the project. After pressing of the button **Finish** the main window for the environment StateCAD will open (fig. 4). From the menu **File →Save As** the name

of the chosen file is set (smm1) with the extention *.DIA. The name of the file have to be in DOS format, eight symbols for name and three for extention (name.ext).

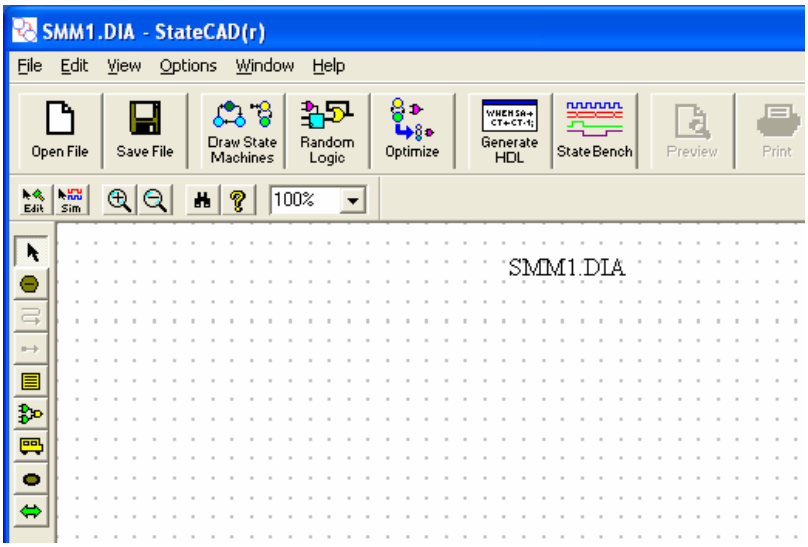

Fig. 4. Main window of the environment StateCAD

With the help of primitives from **StateCAD** the graph of the state machine have to be drawn. From the button **Draw State Machine** (fig. 4) the **State Machine Wizard** starts, which automates the choosing of the state machine's graph structure (**Multi-Column**) from **Shape of state machine** and from **Number of states**, the number of internal states of the state machine (4). After pressing of the button **Next>**, the window **Design Wizard: Reset The State Mashine** appears on the display. From there the form of the signal have to be chosen for the Reset (initialization at beginning) - Synchronous (fig. 5). Again with button **Next>**, on the display appears the window **Design Wizard: Setup Transitions** (fig. 6), for description of the transitions in the state machine - *Loop back* (loop back, keep the state of the state machine unchanged), it is marked as @ELSE over the graph; *Next* (transition toward next state) and *Previous* (transition toward previous state). Into the field next to Next (Previous) check box the condition for cause according transition is set. In current case this is input (X_in), which may have two values – 0 (!X_in) and 1 (X_in). After sequential execution of the steps in the **State Machine Wizard** the next state machine graph structure will be created (fig. 7). The building of the state machine graph is done by editing the result from the *State Machine Wizard*. Editing of the internal states is possible after double clicking over each one and making of the needed adjustments in the window from the fig.8.

Fig. 5. Choice of the type of the signal for Reset

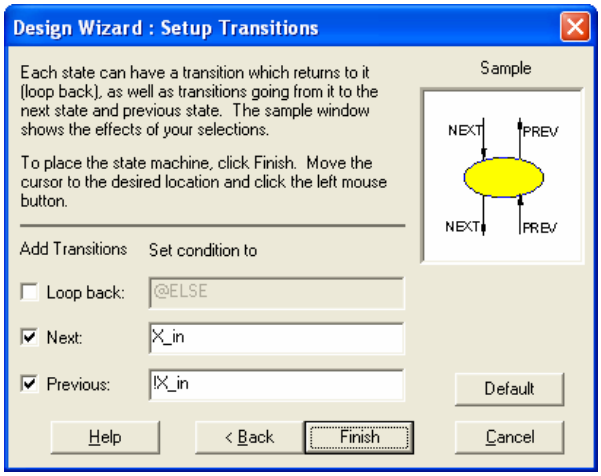

Fig. 6. Description of the state machine transitions

It is needed to be made changes in the transitions in the state machine graph. Adding of transitions in the graph can be made from the tool **Add Transition.** Editing of the transitions in the state machine for the change of it's the states, whit the window from fig. 9 **Edit Condition.** The last appears after double clicking over the arrow, pointing transition from old to new internal state.

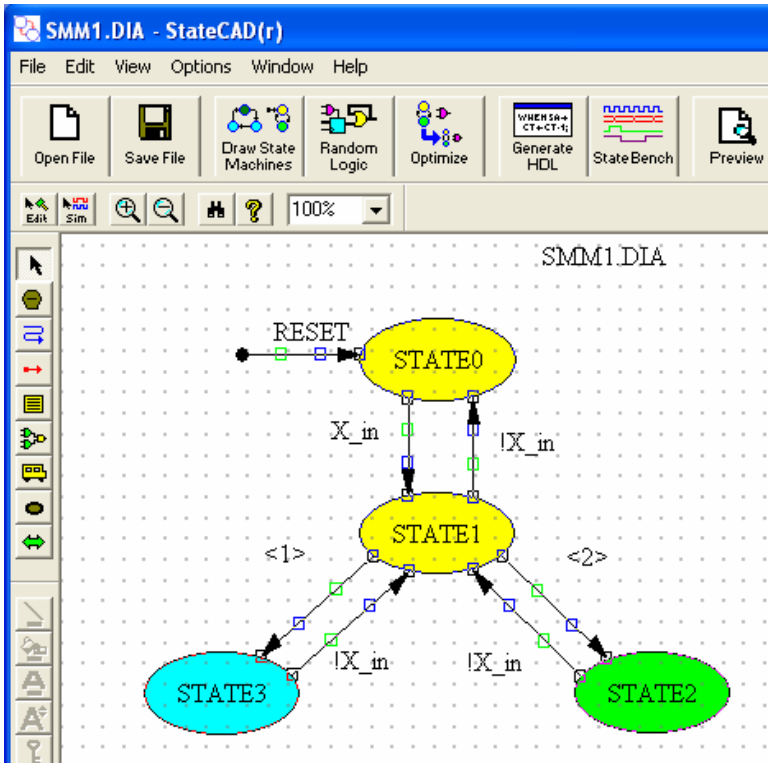

Fig. 7. Result after finishing of the State Machine Wizard

In the field *Condition* the condition has to be written causing the corresponding transition (internal signals in this case is one and is marked as !X_in (0) or X_in (1)).

In the field **Outputs** the output reaction have to be set, which is produced by transition from old toward new internal state.

By means of the button Output Wizard (fig. 9) the Logic Wizard can be started, which will automate designing of the data flow logic

The band with tools for graphical editing of the state machine is located in left part of the display after choosing of the menu **Window/Draw Mode Toolbar**.

On fig. 10 is shown the result after describing of the states and the transitions into the state machine's graph.

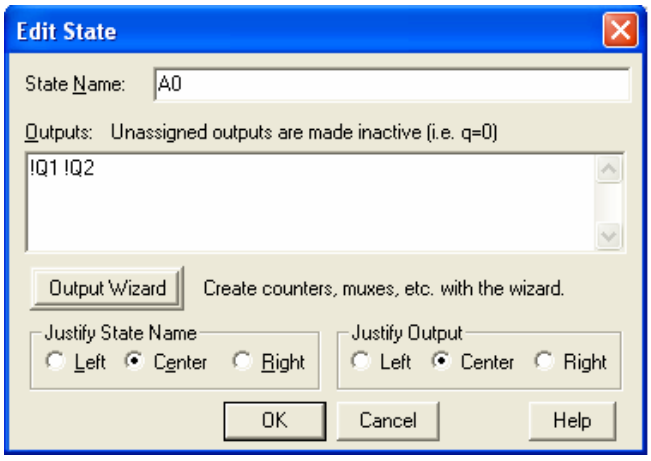

Fig. 8. Editing of the internal states

In fact on fig. 10 is presented the graph of the setted by table Mealy state machine.

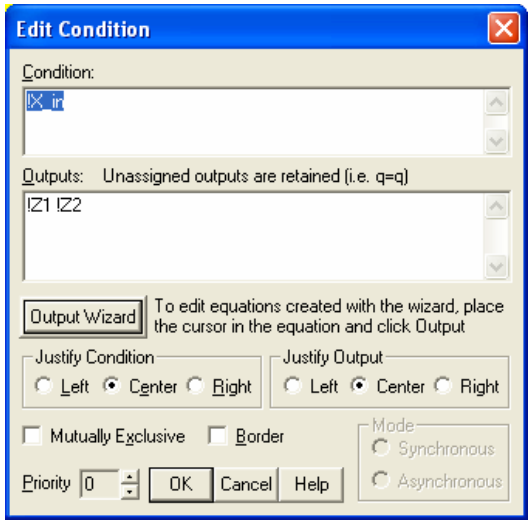

Fig. 9. Window for describing of the state machine transitions, affected by corresponding input signals and produced output reactions

Editing of the variables, inputs and outputs and states of the created state machine, can be made by the window called by menu **Options/Variables**. There are listed all variables after creation of the state machine's graph. All adjustments are done by the environment with default values. There is possible for their extra editing.

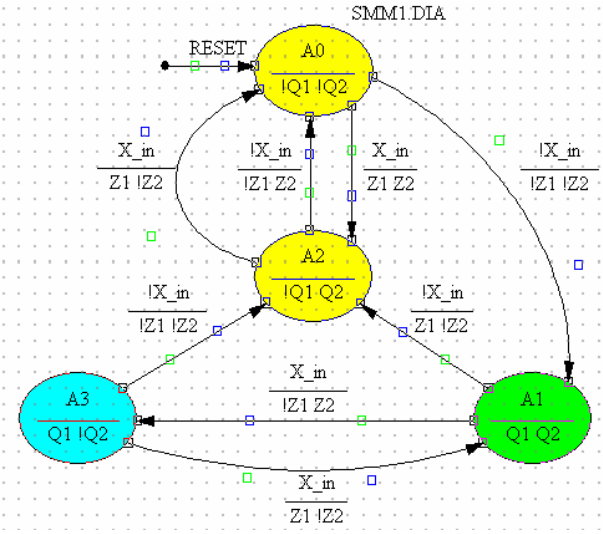

Fig. 10. Result after describing of the states and transitions in the state machine graph

3. CONCLUSION

With the help of the produced results can be created file for programming of the chosen CPLD chip XC2C128 from the family integrated circuits CoolRunner2. Its physical programming is done with specialized programmer, connected with JTAG cable to PC. Before this it is need to be generated VHDL code of the designed state machine, to be made overall statistics for applied analyses and to be made simulation of the project SMM1.DIA.

4. REFERENCES

[1] Mihov G., Digital schematic technology, TU – Sofia, 2005.

[2] Gizdarski E., *Designing with a programmable logic*, Rousse, 1998.

[3] Nikolay, I. I., I. S. Simeonov, *Teaching Aspects of Contemporary Reprogrammable Devices*, Proceedings of Papers, Volume 2, 2004, XXXIX International Scientific Conference, ICEST'2004, Bitola, Macedonia, 16-19 JUNI, 2004, pp. 843-844.

[4] Nancheva-Filipova K., Using (v)HDL for electronic hardware synthesis, Sofia, 2004.

[5] Spartan-II Data Sheet (www.xilinx.com)

[6] ISE Software Manual (support.xilinx.com)