TERNARY REVERSIBLE REGISTERS AND THEIR UTILISATION

Oleg Cernian

University of Craiova
Computer and Communications Engineering Department
13, Alexandru Ioan Cuza Street
200585 CRAIOVA, ROMANIA

Abstract: Much attention is nowadays afforded for non-binary logic as an alternative for future development of computer technology. In this paper there are presented some experiments run on a ternary simulator based on HDL-Aldec platform. There is described in detail a reversible ternary register realized with "MS Rev Tristables" and a direct application with it for implementation of a ternary frequency divider. All structures are captured on the ternary simulator and appropriate functional experiments are run proving the correctness of their operation. Any timing analysis can be carried out revealing eventual functional abnormalities.

Keywords: Ternary devices, Ternary registers, Ternary frequency dividers, Ternary simulation

1. INTRODUCTION

At present a particular attention is devoted to the theoretical and practical aspects of development of technologies for non-binary logic as an alternative for to-day binary logic. This paper deals with a slice referring to functional analysis of two ternary devices; ternary reversible register and ternary frequency divider. The ternary reversible registers are based on a special type of tristable as designed and presented in (Higuchi and Kameyama, 1977). It was redefined the structure of a T-gate (Lee and Chen, 1969; Porat, 1969; Higuchi and Kameyama, 1975; Cernian, 1998b) as implemented in the Post algebra technical system (Hurst, 1979; Porat, 1969) and proved that it works hazard free for certain selection signal transitions. Some widely used criteria of tristables classification are discussed in (Cernian, 1998a; Hurst, 1979; Porat, 1969).

By using the hazard free T-gates there is designed a MS Rev Tristable (Higuchi and Kameyama, 1977) which is versatile for construction of registers, counters and frequency dividers. A functional simulation of the tristables on a ternary simulator (Cernian, 2004), was carried out (Cernian, 2004; Cernian, 2007). Here are presented applications with these tristables consisting in construction of a four stage reversible register RTR4, its extension on a hierarchical base,as well as it is designed an implementation in T-gate logic of a frequency divider. All proposed devices were captured and functionally tested on the designed ternary simulator. In all treated cases the symmetric ternary logic (-1, 0, 1) was used.

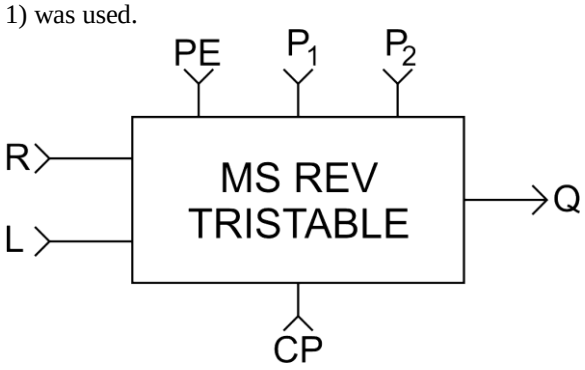

Fig. 1. Block diagram of the MS Rev Tristable

2. IMPLEMENTATION OF A FOUR STAGE REVERSIBLE TERNARY REGISTER

The basic component for realization of the reversible ternary register (RTR) is a dedicated tristable called MSRevTristable(Cernian,2007;Higuchi&Kameyama, 1977). The block diagram of this tristable is given in Fig. 1.

PE represents the Mode of Operation control signal; if PE = 0 then the synchronous operation is selected; if PE = 1 then load of P_1 takes place; if PE = -1 then load of P_2 takes place.

The type of synchronous operation is defined by the CP (clock pulse) waveform: for $0 \rightarrow 1 \rightarrow 0$ evolution a right shift takes place consisting in writing the input R; for $0 \rightarrow -1 \rightarrow 0$ evolution a left shift takes place consisting in writing the input L.

The block diagram of the four stage reversible register (RTR4) is presented in Fig. 2.

The internal structure of this ternary reversible register is presented in Fig. 3.

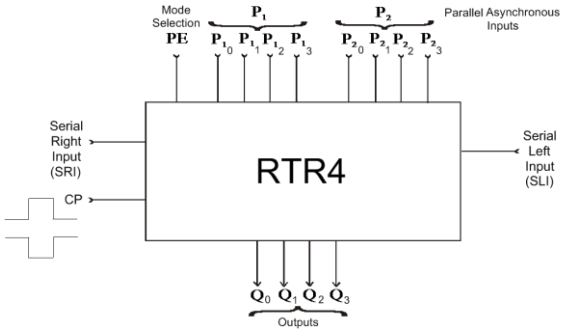

Fig. 2. Block diagram of RTR4

3. OPERATION OF THE RTR4

PE Mode of Operation control signal is connected to all MS Rev Tristables selecting the same operation mode (Synchronous or Asynchronous).

The right shift is realized by connecting the output Q_i to the input R_{i+1}, while left shift is realized by connecting the output Q_i to the input L_{i-1}, where R_1 represents the serial Right Shift signal (R) for RTR4 and L_4 represents the serial Left Shift signal (L) for the RTR4.

Clock pulse (CP) is applied to all CP inputs acting synchronously on all stages of RTR4.

If PE = 0, then the synchronous mode is enabled.

For ⎍‾⎍ evolution of CP a right shift is derived, thereby Q = (r q_0 q_1 q_2), while for ‾⎍‾ evolution of CP a left shift is derived, thereby Q = (q_1 q_2 q_3 l), where q, r, l are the logical values one set up time before the active edge of CP.

If PE = 1, then the asynchronous input of the word P_1 occurs, thereby Q = (P_{10} P_{11} P_{12} P_{13}).

If PE = -1, then the asynchronous input of the word P_2 occurs, thereby Q = (P_{20} P_{21} P_{22} P_{23}).

Obviously, cases PE = 1 and PE = -1 correspond to the preset of the register mode of operation.

The extension of these functional units is straightforward, by tying Q_3 to the input SRI of the next stage for Right Shift, and the output Q_0 to the input SLI of the preceding stage for Left Shift. In Fig. 4 it is given the structure of an RTR8 realized from two RTR4. It is pointed up the hierarchical structure of the realizations (T-gates, MS Rev Tristables, RTR4, RTR8 etc.).

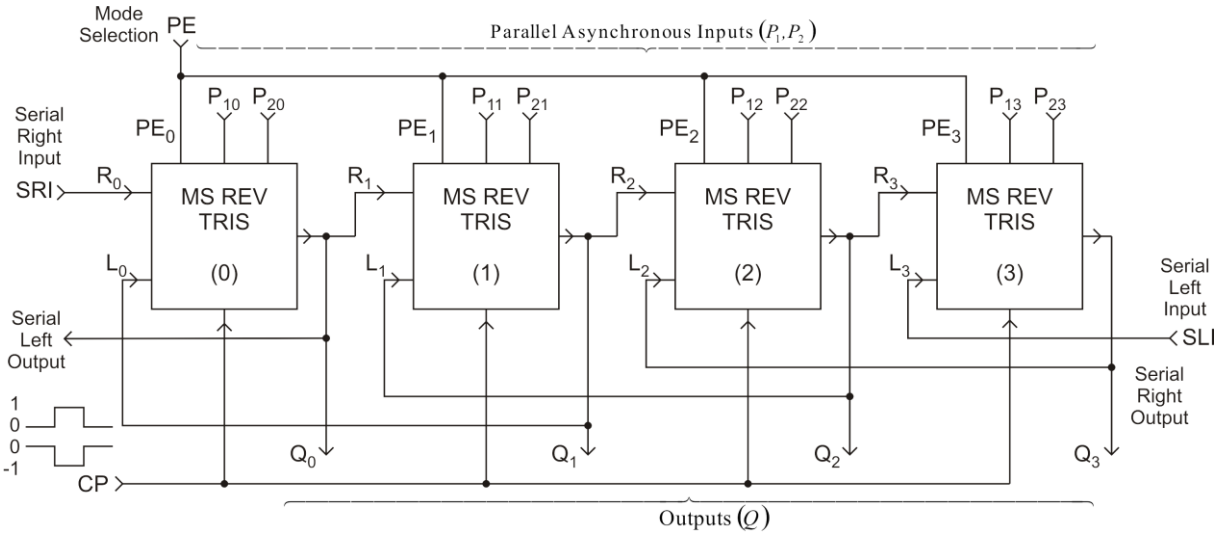

Fig. 3. Internal Structure of RTR4

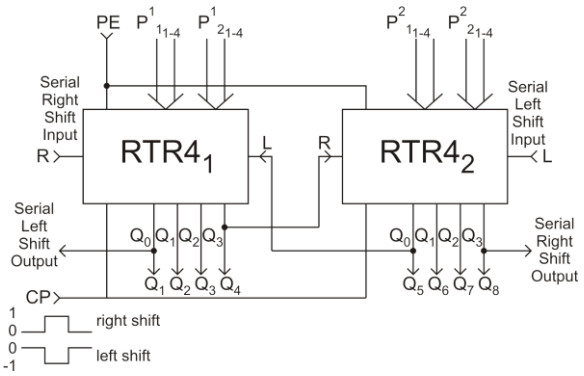

Fig. 4. Extension of two RTR4

4. SIMULATION OF RTR4

The simulation of RTR4 operation is realized by means of a ternary simulator briefly described in (Cernian, 2004).

Firstly the structure of the RTR4 is captured as depicted by the appropriate screenshot (Fig. 5).

The parallel asynchronous words were defined as four dimensional vectors $P_1(3:0)$ and $P_2(3:0)$. The output of the RTR4 was also defined as a four dimensional vector Q(3:0).Q(3) represents the most significant stage of RTR4. SRI and SLI represent the Serial Right Input and, respectively, Serial Left Input. The values for P_1 and P_2 were fixed to (HHHH), i.e. (1111) and, respectively to (MMMM), i.e. (0000). PE was defined by a formula, whereas CP was defined in such a way to highlight the correctness of RTR4 operation. To simplify the simulation the values of SLI and SRI were fixed to H, i.e. 1, and, respectively, to L, i.e. -1.

When PE = H, i.e. 1, it is realized the asynchronous load of P_1, that is HHHH, and therefore the outputs $Q_3Q_2Q_1Q_0$ become HHHH.

After that, PE is changed to M, i.e. to 0, and the synchronous mode of operation is selected. It is pointed up the left shift and right shift actions in accordance with the evolution of CP. For the MHM (010) evolution of CP it is derived a right shift, where Q_0 become SRI, i.e. -1, while the content of cells Q_0Q_1 and Q_2 are right shifted with one position; the content of Q_3 is lost.

For MLM (0-10) evolution of CP it is derived a left shift, where Q_3 becomes SLI, i.e.1, the rest being shifted one position to the left; the content of Q_0 is lost.

The evolution of the realized simulation is depicted in the screenshot (Fig. 6).

If necessary, the detailed structure of the output vector Q is available. By means of two cursors, on the selected timescale, the experimenter can pinpoint the delays between different events. The experiments performed on the designed RTR4 proved the correctness of the operation.

5. USE OF THE TERNARY SHIFTING REGISTER FOR IMPLEMENTATION OF A FREQUENCY DIVIDER

It is used a ternary reversible shifting register on three stages (RTR3) realized with MS Rev Tristables implementing a frequency divider with division factor D = 6.

The block diagram of the frequency divider, consisting from a state generator and two CLNs, is given in Fig. 7.

Initially, PE is tied to 1 selecting the asynchronous parallel load for RTR3; hence, the initial value 1000 is stored in the RTR3. RTR3 is used as a state generator. Since the factor D = 6 it results that RTR3 must produce six successive states followed by a return in the initial state.

It is selected shift right operation mode by applying $\begin{smallmatrix}1\\0\end{smallmatrix}\!\!\sqcap\!\!\_$ on CP. The main design process is centered on the correct generation of the serial shift right input R for RTR3, which is synthesized by Input Control Block, whose output is designated f_{CR}.

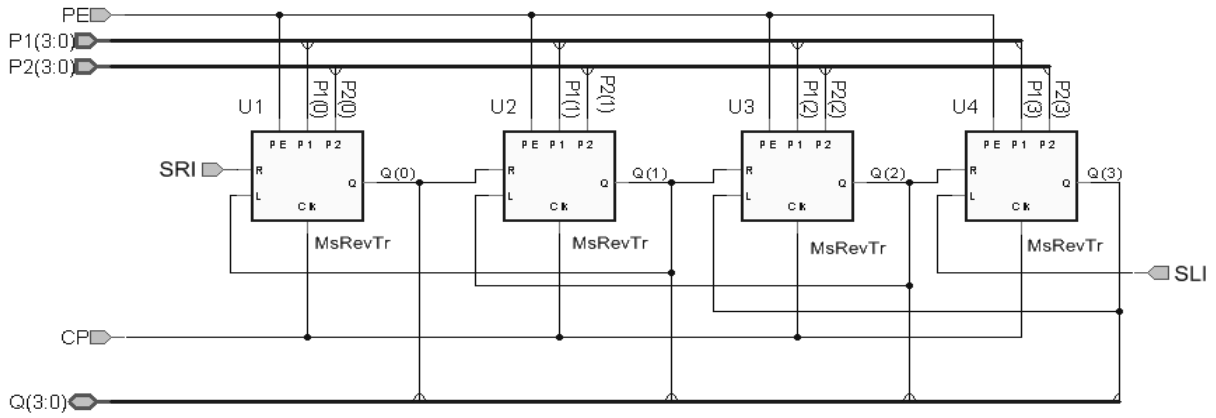

Fig. 5. Screenshot of the RTR4

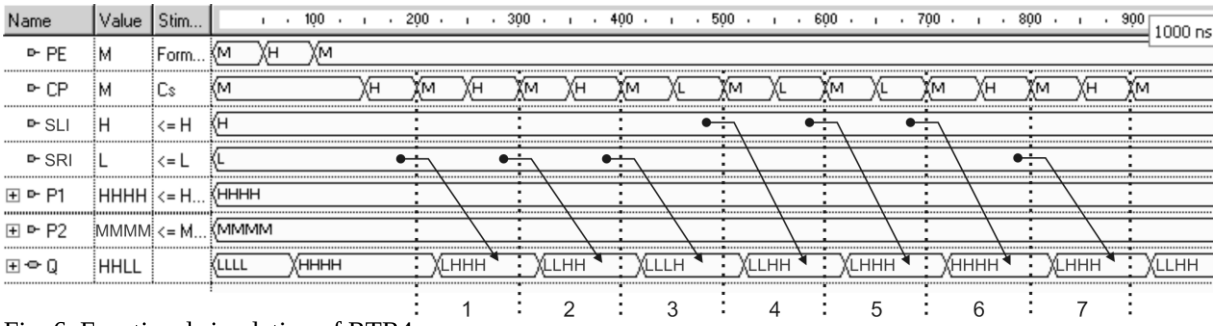

Fig. 6. Functional simulation of RTR4

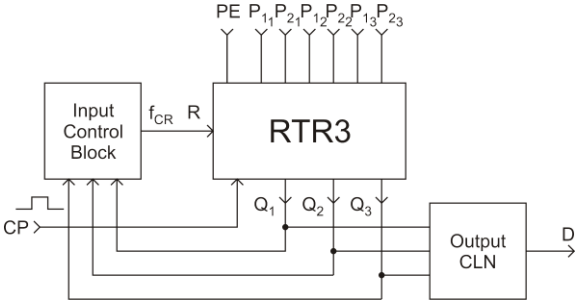

Fig. 7. Block diagram of the frequency divider

The entire design process is based on the general graph for shift right on an RTR3 (Cernian, 2004). At each pulse the new value R is inputted in cell Q_1 while the rest is shifted one position to the right; the rightmost position is pushed out. The following evolution graph is derived (Fig. 8), representing a cycle with 6 states from the general graph associated to RTR3.

The truth table for Input Control Block is given in Fig. 9.

The coarse implementation of the ternary switching function f_{CR} with T-gates is presented in Fig. 10.

After applying specific methods for T-gates logic simplification it is derived a less complex structure (Cernian,1998b), which is given in Fig. 11.

The output D of the frequency divider can be generated, for instance, from the state -7, i.e. $Q_3 = \bar{1}$, $Q_2 = 1$, $Q_1 = \bar{1}$, as presented in Fig.12. For each cycle of six consecutive states, i.e. for each six clock pulses ⎍⎍, the output D = 1.

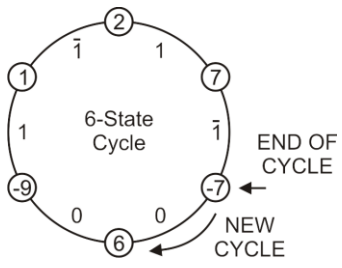

Fig. 8. Evolution graph for RTR3 implementing a frequency divider with D = 6

6. FUNCTIONAL SIMULATION OF THE FREQUENCY DIVIDER WITH D = 6

It is used the ternary HDL simulator that is briefly presented in (Cernian, 2004). The screenshots for the described in paragraph 5 devices are depicted in Fig. 13, Fig. 14 and Fig. 15.

The detailed capture for the RTR3 is given in Fig. 16 The reason to use an RTR3 instead of the RTR4, presented and tested previously, is linked to the complexity of the states graph associated to a four stage register (81 states),as compared to a three stage register (only 27 states).

Q_1	Q_2	Q_3	f_{CR}
1	0	0	$\bar{1}$
$\bar{1}$	1	0	1
1	$\bar{1}$	1	$\bar{1}$
$\bar{1}$	1	$\bar{1}$	0
0	$\bar{1}$	1	0
0	0	$\bar{1}$	1
Rest of combinations			X

Fig. 9. Truth table for Input Control Block

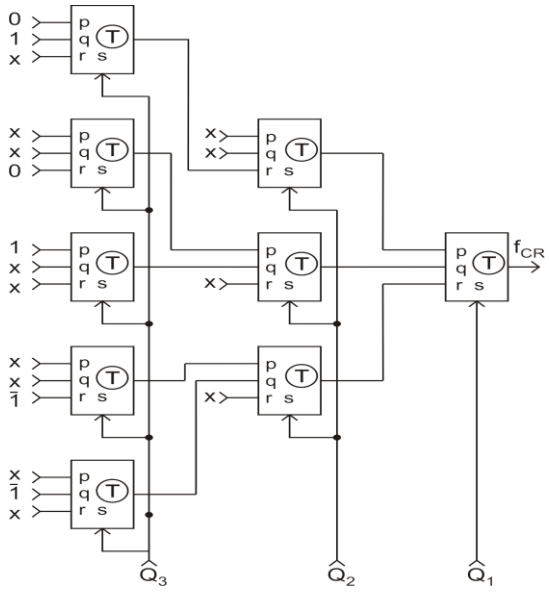

Fig. 10. Internal structure of Input Control Block

After capturing all blocks of the divider the functional analysis was run. The screenshot for the evolution of the divider is given in Fig. 17. Initially, PE = 1 selecting asynchronous load of P_1 established by the vector at 100 (decimal 1). Afterwards, PE = 0 selecting synchronous mode, where the right shift is enabled by $\begin{matrix} H \\ M \end{matrix}$ ⎍ at CP.

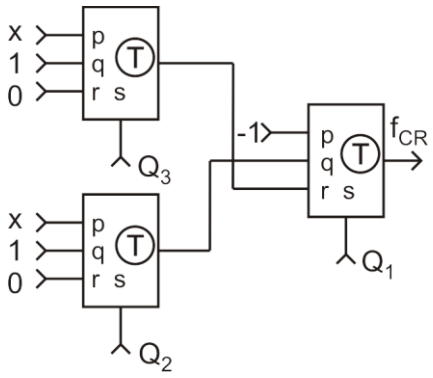

Fig. 11. Simplified structure of Input Control Block

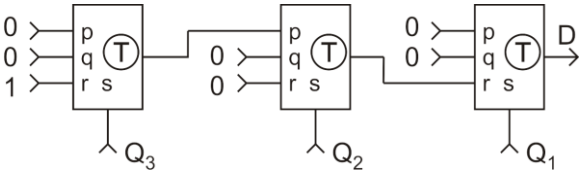

Fig. 12. Generation of D from state $\bar{1}\bar{1}\bar{1}$

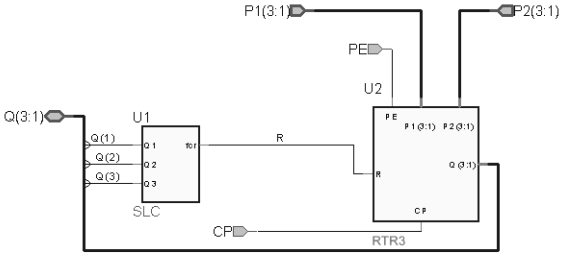

Fig. 13. Screenshot for the frequency divider

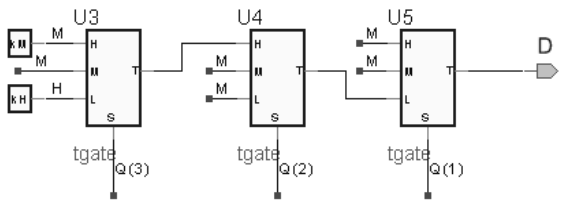

Fig. 14. Screenshot for CLN generating the output D

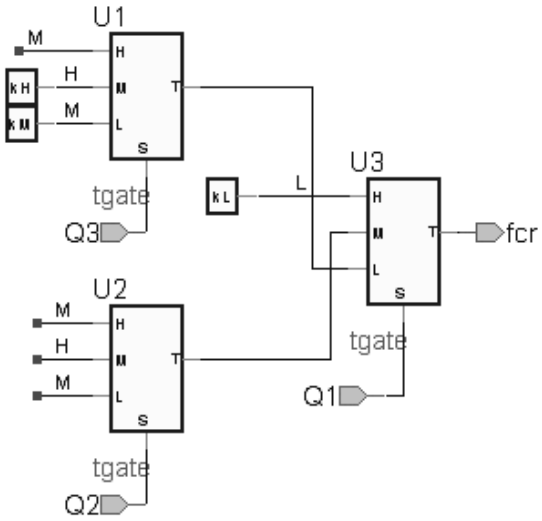

Fig. 15. Screenshot for the Output Control Block

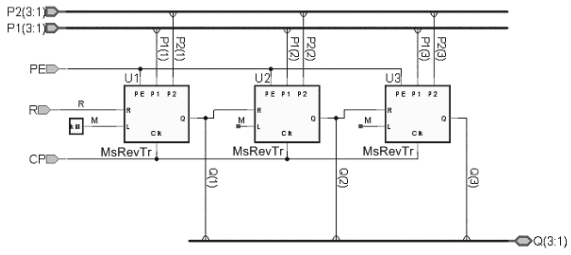

Fig. 16. Screenshot for RTR3

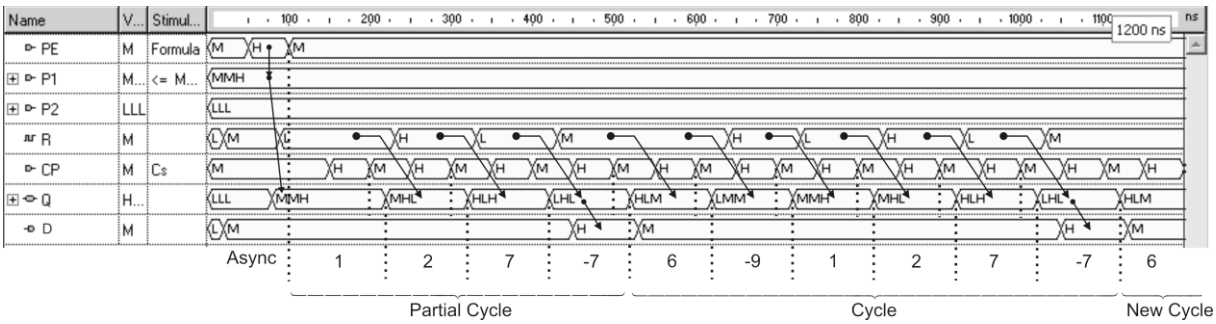

Fig. 17. Functional analysis of the frequency divider

The Input Control Block is generating through its output f_{CR} the serial right shift input R. For every H→ M transition of CP the value of R is inputted in Q_1 and the rest is shifted one position to the right. The final state in the cycle is -7, that is -1 1 -1 (LHL), which generates through the Output CLN the effective output D = 1. The simulation highlights a complete cycle with 6 states (since D = 6) and a partial cycle with only 4 states, as an initial cycle, since the initial loaded state was 1.The next cycles contain six states.

By means of two cursors there can be pinpointed all delays between different transitions of signals. The experiments proved the correctness of the synthesized frequency divider.

REFERENCES

Cernian, O., (2007). Simulations of different homogeneous T-gate tristables. Proceedings of the 13th SINTES Symposium, Craiova.

Cernian, O. (1998a). Clasificări de circuite basculante tristabile. Universitatea din Craiova, Raport intern de cercetare, Catedra de Calculatoare.

Cernian, O. (1998b). Implementarea structurată a funcţiilor ternare cu ajutorul portilor T. Universitatea din Craiova, Raport intern de cercetare, Catedra de Calculatoare.

Cernian, O. (2004). *Contribuţie la Analiza si Sinteza Sistemelor Logice Ternare*. Teză de doctorat, Universitatea Tehnică Gh. Asachi Iaşi.

Higuchi, T. and Kameyama, M. (1975). Ternary System based on T-Gate. Proceedings of IEEE Int. Symp. on Multiple-Valued Logic.

Higuchi, T. and Kameyama, M. (1977). Static Hazard Free T-Gate for Ternary Memory Element and its Application to Ternary Counters. *IEEE Trans on Computers*, **vol. C26, No.12**.

Hurst, S.L. (1979). *The Logical Processing of Digital Signals*. Crane-Russek

Lee, C. Y., Chen, N. H. (1956). Several Valued Combinational Switching Circuits, *Trans AIEE*, **vol. 75**.

Porat, D. I. (1969). Three Valued Digital Systems. *Proc of IEE*, **vol. 110**.

XXX (2005), *Active-HDL 6.3 Student Edition*. Prentice Hall.