An Analytical Method to Determine the Data Inputs (R-S, D, J-K, T) for Synthesis of Finite Automata

[1] Alexandru VALACHI, [2] Călin-Mircea MONOR, [3] Mihai TIMIŞ
Technical University "Gh.Asachi"Iasi
Automatic Control and Computer Science Faculty
Computer Science & Engineering Department
[1] *avalachi@cs.tuiasi.ro*, [2] *cmonor@cs.tuiasi.ro*, [3] *mtimis@cs.tuiasi.ro*

***Abstract*—The first step of the automata design with registers is the synthesis of the combinational logic blocks which generates the data inputs (R-S, D, J-K, T). This paper presents an analytical method of synthesis which find a logical relation for the data inputs (R-S, D, J-K, T).**

***Index Term* —Finite State Machine, Finite Automaton, R-S, D, J-K , T - latch, Asynchronous Sequential Systems.**

I. INTRODUCTION

Let's consider a finite automata which contains (p) input variables, noted as $X = \{x_{p-1}, x_{p-2}, ... x_0\}$, (N) state variables noted as $Y = \{y_{N-1}, y_{N-2}, ... y_0\}$, (q) output variables noted as $Z = \{z_{q-1}, z_{q-2}, ... z_0\}$. It is described by the (1) equations:

$$\begin{Bmatrix} y_{i,n+1} = f_i(x_{p-1}, x_{p-2}, ... x_0; y_{N-1}, ... y_0)_n \\ z_{i,n+1} = g_j(x_{p-1}, x_{p-2}, ... x_0; y_{N-1}, ... y_0)_n \end{Bmatrix} \qquad (1)$$

The notation $u(t_n) = u_n$ means the value of the u(t) at the n moment of time.

If consider the vectorial notation as $F = \{f_{N-1}, ... f_0\}$ and $G = \{q_{q-1}, ... g_0\}$ the previous equations become (2):

$$\begin{aligned} y_{n+1} &= F(X_n, Y_n) \\ Z_n &= G(X_n, Y_n) \end{aligned} \qquad (2)$$

The finite automaton described by (2) is named Mealy machine.

For the Moore automaton, the equations are described by (3). The output variables depend only on internal variables.

$$\begin{aligned} y_{n+1} &= F(X_n, Y_n) \\ Z_n &= G(Y_n) \end{aligned} \qquad (3)$$

Neglecting the logic structure which generates the outputs variables, the finite automaton can be implemented with logic gates (1a), D latch (1b), RS latch (1c), JK latch (1d) or T latch (1e) . In figure 1a, Δ represents the total delay which is applied to the state variables while propagating through the combinatorial logic; (we considered the same propagation time for the state variables through defined loops).

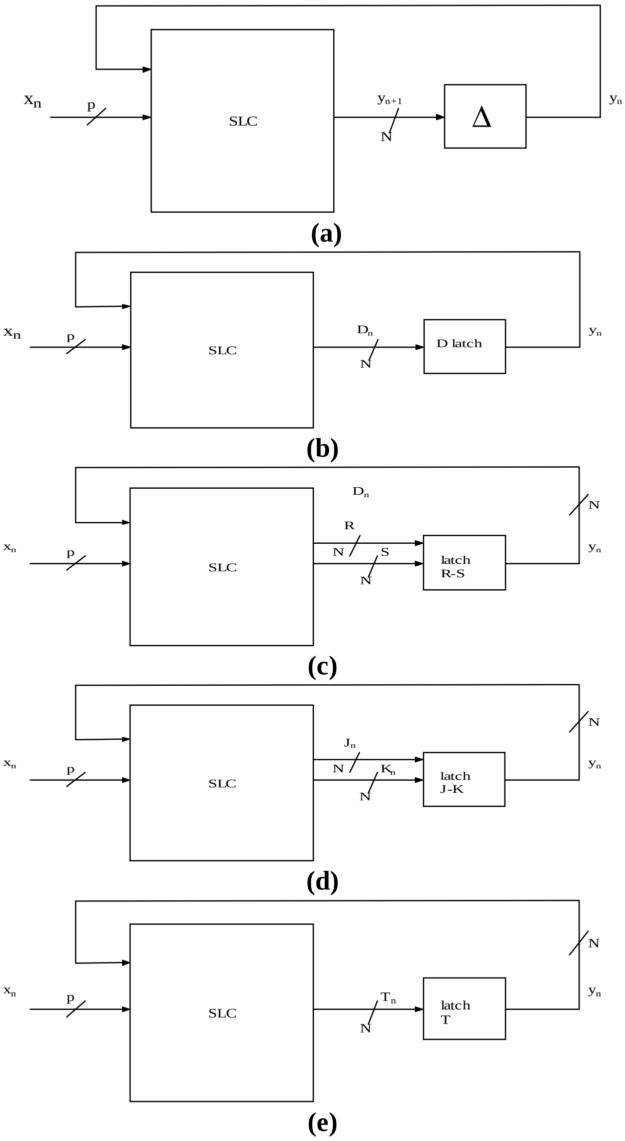

Figure 1. Implementation types for a finite automaton

The D latch implementation is easy to do using the next equation:

$$D_{i,n} = y_{i,n+1} = f_i(x_n, y_n)$$

II. IMPLEMENTATION OF THE ASS USING R-S LATCHES

$$\begin{cases} R_{i,n} = F_i(x_{p-1}, x_{p-2} ... x_0; y_{p-1}, ... y_0) \\ S_{i,n} = G_i(x_{p-1}, x_{p-2} ... x_0; y_{p-1}, ... y_0) \end{cases} \qquad (4)$$

The ASS(Asynchronous Sequential System) equations can be deducted by system synthesis or analysis.

Lets consider the analysis of the sequential system described in figure 2.

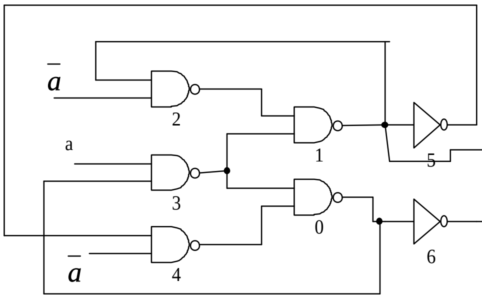

Figure 2. Ass logic gates design

If we consider two fundamental loops, formed by the (1-2) and (0-3) logic gates, the internal variables will be the outputs of the (1) and (0) logic gates, noted y_1 and y_0 [1].

The sequential system is described by equation (5):

$$\begin{aligned} y_{1,n+1} &= (\overline{a} \cdot y_1 + a \cdot y_0)_n \\ y_{0,n+1} &= (\overline{a} \cdot \overline{y_1} + a \cdot y_0)_n \end{aligned} \qquad (5)$$

From the equation of R-S latch (6), can be deducted the Veitch-Karnaugh tables for the R_i, S_i data inputs.

$$\begin{aligned} & y_{n+1} = S_n + \overline{R_n} \cdot y_n,\ R_n \cdot S_n = 0 \\ & y_n = 0 => S_n = y_{n+1}, R_n => R_n \cdot S_n = 0 \\ & y_n = 1 => R_n = \overline{y_{n+1}}, S_n => R_n \cdot S_n = 0 \end{aligned} \qquad (6)$$

Using the equations (5) will be deducted the transition table, figure 3.

	$(y_1y_0)_{n+1}$		S_1		R_1	
a $(y_1y_0)_n$	0	1	0	1	0	1
00	01	00	0	0	-	-
01	01	11	0	1	-	0
11	10	11	-	-	0	0
10	10	00	0	0	0	1

	S_0		R_0	
a $(y_1y_0)_n$	0	1	0	1
00	1	0	0	-
01	-	-	0	0
11	0	-	1	0
10	0	0	-	-

Figure 3. Transition table and veitch-karnaugh tables for S_I, R_I

Can be observed that when S are known, R will be chosen from equation $R \cdot S = 0$ (S=0=> R=- and S=1=> R=0).

The ASS equations are:

$$\begin{aligned} & S_1 = a \cdot y_0 \quad S_0 = \overline{a} \cdot \overline{y_1} \\ & R_1 = a \cdot \overline{y_0} \quad R_0 = \overline{a} \cdot y_1 \end{aligned} \qquad 7)$$

Next, we will present the proposed method; the $y_{i,n+1}$ variables can be written as:

$$y_{i,n+1} = (A + B \cdot \overline{y_i} + C \cdot y_i)_n \qquad (8)$$

and, with notation $A + B \cdot \overline{y_i} = u$, results equation:

$$y_{i,n+1} = u + c \cdot y_i = u + (c + u) \cdot y_i \qquad 9)$$

If we match the (9) equation with the (6) equation, results:

$$\begin{aligned} & S = u = A + B \cdot \overline{y_i} \\ & \overline{R} = (c + u) => R = \overline{c} \cdot \overline{u} \end{aligned} \qquad 10)$$

For example:

$$\begin{aligned} & y_{1,n+1} = \overline{a} \cdot y_1 + a \cdot y_0 \\ & A = a \cdot y_0 \\ & B = 0 \\ & C = \overline{a} \\ & S_1 = a \cdot y_0 \\ & R_1 = a \cdot (\overline{a} + \overline{y_0}) = a \cdot \overline{y_0} \end{aligned}$$

$$\begin{aligned} & y_{0,n+1} = \overline{a} \cdot \overline{y_1} + a \cdot y_0 \\ & A = \overline{a} \cdot \overline{y_1} \\ & B = 0 \\ & C = a \\ & S_0 = \overline{a} \cdot \overline{y_1} \\ & R_1 = \overline{a} \cdot y_1 \end{aligned}$$

Observation:

We cannot say that the (10) equations are unique.

The implementation looks like in figure 4:

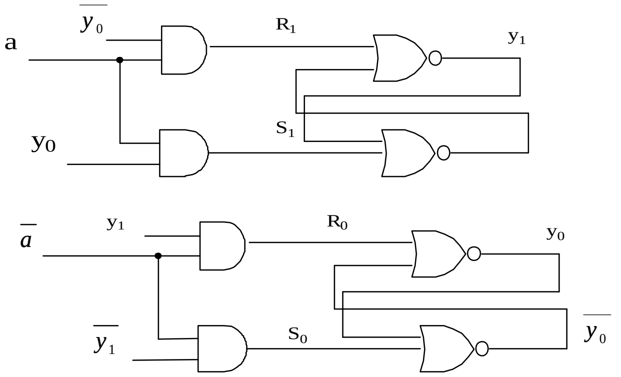

Figure 4. Latch implementation

The $a, \overline{a}$ can produce type of logic switching hazard. To avoid this situation the equations can be write as:

$$\begin{aligned} S_1 &= a \cdot y_0 \\ R_1 &= a \cdot (\overline{a} + \overline{y_0}) = a \cdot \overline{y_0} \\ S_0 &= \overline{a + y_1} \\ R_0 &= \overline{a + \overline{\overline{y_1}}} \end{aligned} \qquad 11)$$

The implementation is described in figure 5:

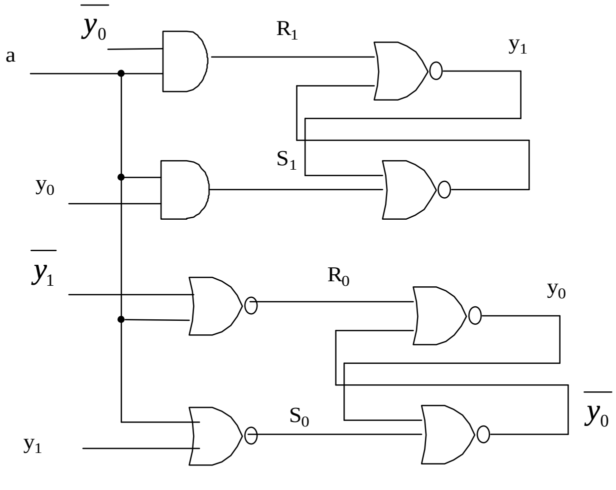

Figure 5. Synthesis without switching hazard

III. IMPLEMENTATION WITH J-K LATCHES

From the J-K latch equation, [5] results:

$$\begin{aligned} & y_{n+1} = (J \cdot \overline{y} + \overline{K} \cdot y)_n \\ & y_n = 0 => J_n = y_{n+1}; K_n = -(indifferent) \\ & y_n = 1 => K_n = \overline{y_{n+1}}; J_n = - \end{aligned} \qquad (12)$$

Using the same transition table with the (12) results the Veitch-Karnaugh table for the J$_n$, K$_n$.

To use the proposed analytical method, we need to use equation (8):

$$\begin{aligned} & y_{i,n+1} = A + B \cdot \overline{y_i} + C \cdot y_i = A \cdot (y_i + \overline{y_i}) + B \cdot \overline{y_i} + C \cdot y_i = \\ & = (A+B) \cdot \overline{y_i} + (A+C) \cdot y_i \end{aligned} \qquad (13)$$

$$\begin{aligned} J &= A + B \\ K &= \overline{A} \cdot \overline{C} \end{aligned} \qquad (14)$$

Using the proposed ASS example, applying the (14) equations, results equation:

$$\begin{aligned} & y_{1,n+1} = \overline{a} \cdot y_1 + a \cdot y_0 \\ & A = a \cdot y_0 \\ & B = 0 \\ & C = \overline{a} \\ & \quad J_1 = a \cdot y_0 \\ & \quad K_1 = \overline{A} \cdot \overline{C} = \overline{a \cdot y_0} \cdot \overline{\overline{a}} = a \cdot \overline{y_0} \end{aligned} \qquad (15)$$

$$\begin{aligned} & y_{0,n+1} = \overline{a} \cdot \overline{y_1} + a \cdot y_0 \\ & A = \overline{a} \cdot \overline{y_1} \\ & B = 0 \\ & C = a \\ & J_0 = \overline{a} \cdot \overline{y_1} \\ & K_1 = (a + y_1) \cdot \overline{a} = \overline{a} \cdot y_1 \end{aligned}$$

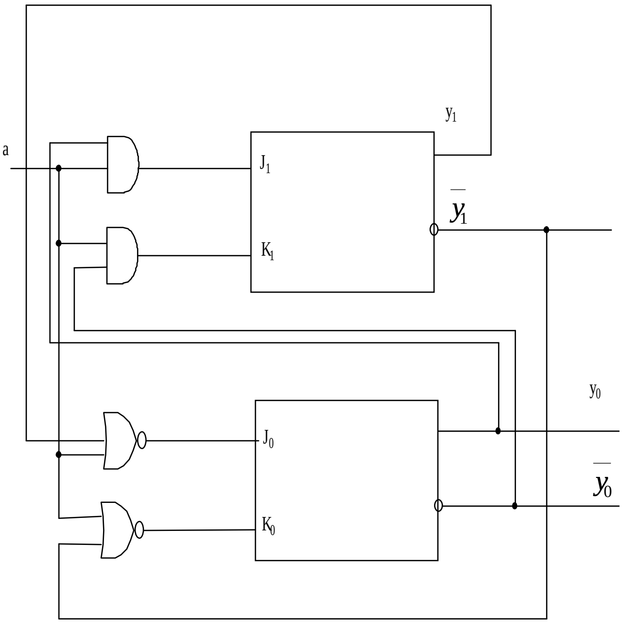

Figure 6. Implementation with jk latches

IV. IMPLEMENTATION WITH T LATCHES

From the (14) equations result the transition tables for the T inputs.

(14) $T_n = y_n \oplus y_{n+1}$

To apply the analytical method we are starting for the (8) equation:

$$T_{i,n} = y_{i,n} \oplus y_{i,n+1} = y_{i,n} \cdot \overline{y_{i,n+1}} + \overline{y_{i,n}} \cdot y_{i,n+1}$$

For the previous example we have, (16):

$$\begin{aligned} & T_1 = y_{1,n} \oplus y_{1i,n+1} = y_1 \oplus (\overline{a} \cdot y_1 + a \cdot y_0) = \\ & = y_1 \cdot \overline{\overline{a} \cdot y_1 + a \cdot y_0} + \overline{y_1} \cdot (\overline{a} \cdot y_1 + a \cdot y_0) = \\ & = a \cdot (y_1 \oplus y_0) \\ & T_0 = y_{0,n} \oplus y_{0i,n+1} = y_0 \oplus (\overline{a} \cdot \overline{y_1} + a \cdot y_0) = \\ & = \overline{a} \cdot \overline{(y_1 \oplus y_0)} = \overline{a + (y_1 \oplus y_0)} \end{aligned}$$

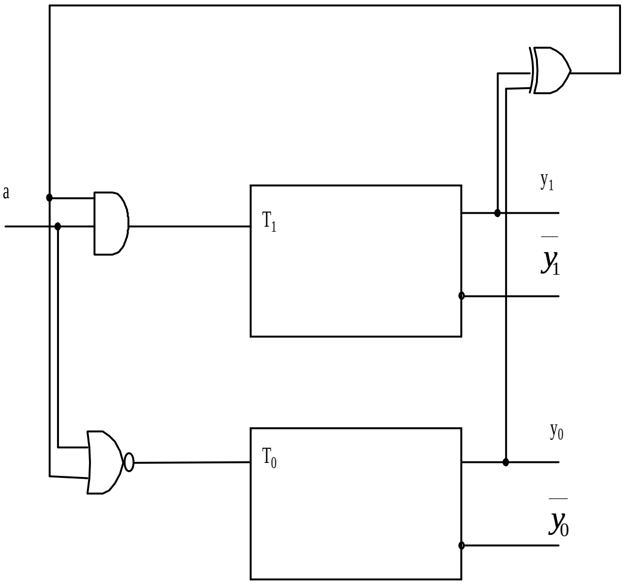

Figure 7. Implementation with t latches

V.CONCLUSION

This paper presents an original method for the calculation and synthesis of the data inputs for latches in asynchronous finite automata.

The proposed method can be succesfully applied in design of Mealy finite automata.

In cases of programable logic circuits which contains only logic gates it is useful to make the design with R-S latches because all the other types of latches (D,T,J-K) are built starting from the R-S latch.

REFERENCES

[1] M.Denouette, E.Daclin, J.P. Perrin - *Systemes Logiques- Tome I*, Dunod – Paris - 1967.

[2] Al. Valachi, Fl. Hoza, V. Onofrei, R.Silion – *Synthesis, Analysis and Testing of Digital Devices* - Ed. N.Est, 1993.

[3] M. Morris Mano – *Digital Design* – Pretince Hall PTR- 2002.

[4] Charles H. Roth - *Fundamentals of Logic Design,* West Publishing Co - 1992.

[5] Vlad Radulescu, Al. Valachi - *Contribution of the theory of Automata Design*, Bul. IPI, Tom VLVIII/LII,fasc.1-4 2002.