A Comparison between Coded-Decoded Mode Signals on Multifunctional Registers

Mihai TIMIS
Automatic Control and Computer Engineering Faculty
Technical University "Gh.Asachi"
Iasi, Romania
mtimis@tuiasi.ro

Alexandru VALACHI, Petru CASCAVAL,
Radu SILION
Automatic Control and Computer Engineering Faculty
Technical University "Gh.Asachi"
Iasi, Romania
avalachi@tuiasi.ro, pcascaval@tuiasi.ro, rsilion@tuiasi.ro

***Abstract*—This paper presents the research results regarding a comparison between two multifunctional registers types, with coded or decoded command validation signals. Also, the implementation costs, time parameters and the applicability in FSM synthesis are presented as conclusions.**

Keywords—Multifunctional Registers, Moore Sequential System, Veitch – Karnaugh, Logic Gates, Truth Table, Priority Inputs Signals.

I. INTRODUCTION

In this paper the authors continue the research of synthesis multifunctional registers, described in research papers [1],[2],[3]. The authors present in [1] the set of priority criteria orders for the multifunctional registers, denoted as RMF with the validation decode signals.

In this paper, the authors undertake a comparison between two multifunctional registers types, with the coded or decoded command validation signals. From this point of view, the authors analyze the implementation costs, time parameters and the applicability in FSM synthesis.

II. MULTIFUNCTIONAL REGISTERS WITH DECODED MODE SIGNALS

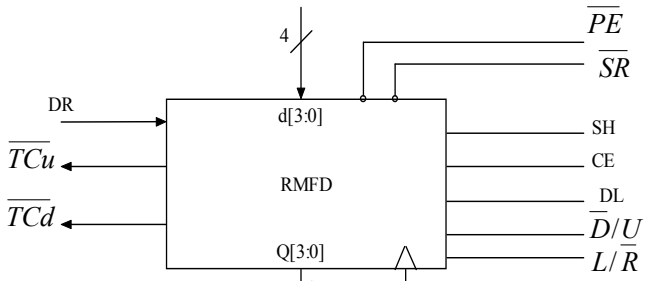

Fig. 1. RMF(decoded mode inputs)

Signals description:

Mode inputs: $\overline{SR}$ - Reset (High Priority), $\overline{PE}$ - Parallel Enable, SH – Logical Shift($L/\overline{R}=0-Right$, $L/\overline{R}=1$-Left), CE – Count Enable(Low Priority) ($U/\overline{D}=0\rightarrow Down, U/\overline{D}=1\rightarrow UP$)

Data: d[3:0] – input data in parallel load mode, Q[3:0] – output data, DR,DL – serial input data, right shift, left shift, $\overline{TC_u}$ - Terminal Count Up (Q[3:0]=1111, last state), $\overline{TC_D}$ - Terminal Count Down (Q[3:0]=0000, last state).

CK – clock input, active on positive edge signal.

The functional behaviour is described in Table I.

TABLE I. RMF FUNCTIONALITY

Mode	$\overline{SR}\,\overline{PE}$	SH	CE	$L/\overline{R}$	$U/\overline{D}$	CK	$Q[3:0]_{n+1}$
Hold	1 1	0	0	-	-	-	$Q[3:0]_n$
Reset	0	-	-	-	-	↑	0000
Parallel Load	1 0	-	-	-	-	↑	$d[3:0]_n$
Shift Right	1 1	1	-	0	-	↑	$DLQ[3:1]_n$
Shift Left	1 1	1	-	1	-	↑	$Q[2:0]_n\,DR$
Count Down	1 1	0	1	-	0	↑	$[Q_n-1]_n$ mod16
Count UP	1 1	0	1	-	1	↑	$[Q_n+1]_n$ mod16

Observation: From paper [1], we consider only 4 bits for input/output data. This limitation doesn't affect the presented research idea.

Based on functionality modes and priority orders, we deduced the following equations (Di- data inputs on flip-flop D, Qi- outputs from flip-flop D).

978-1-4799-5094-2/14/$31.00 ©2014 IEEE

$$
\begin{aligned}
D_i &= \overline{SR}\cdot(PE\cdot d_i + \overline{PE}\cdot(SH\cdot(\overline{L/\overline{R}}\cdot Q_{i+1} + L/\overline{R}\cdot Q_{i-1}) + \overline{SH}\cdot(CE\cdot(\overline{U/\overline{D}}\cdot(\overline{Q_i}\oplus\varphi_i) + \\
&+ U/\overline{D}\cdot(Q_i\oplus\psi_i)) + \overline{CE}\cdot Q_i))) \\
D_3 &= \overline{SR}\cdot(PE\cdot d_3 + \overline{PE}\cdot(SH\cdot(\overline{L/\overline{R}}\cdot DL + L/\overline{R}\cdot Q_2) + \overline{SH}\cdot(CE\cdot(\overline{U/\overline{D}}\cdot(\overline{Q_3}\oplus\varphi_3) + \\
&+ U/\overline{D}\cdot(Q_3\oplus\psi_3)) + \overline{CE}\cdot Q_3))) \\
D_0 &= \overline{SR}\cdot(PE\cdot d_0 + \overline{PE}\cdot(SH\cdot(\overline{L/\overline{R}}\cdot Q_1 + L/\overline{R}\cdot DR) + \overline{SH}\cdot(CE\cdot(\overline{U/\overline{D}}\cdot(\overline{Q_0}\oplus\varphi_0) + \\
&+ U/\overline{D}\cdot(Q_0\oplus\psi_0)) + \overline{CE}\cdot Q_0)))
\end{aligned}
\tag{1}
$$

where $i \in \{1,2\}$ and φ_i, ψ_i represent the "i^{th}" rank function for the down- or adder binary counter.

$$\varphi_i = \prod_{k=0}^{i-1}\overline{Q_k}\ ,\ \psi_i = \prod_{k=0}^{i-1} Q_k$$

or

$$\varphi_i = \varphi_{i-1}\cdot\overline{Q_{i-1}}\ ,\ \psi_i = \psi_{i-1}\cdot Q_{i-1} \text{ and } \varphi_0 = \psi_0 = 1$$

A possible implementation is shown in figure 2:

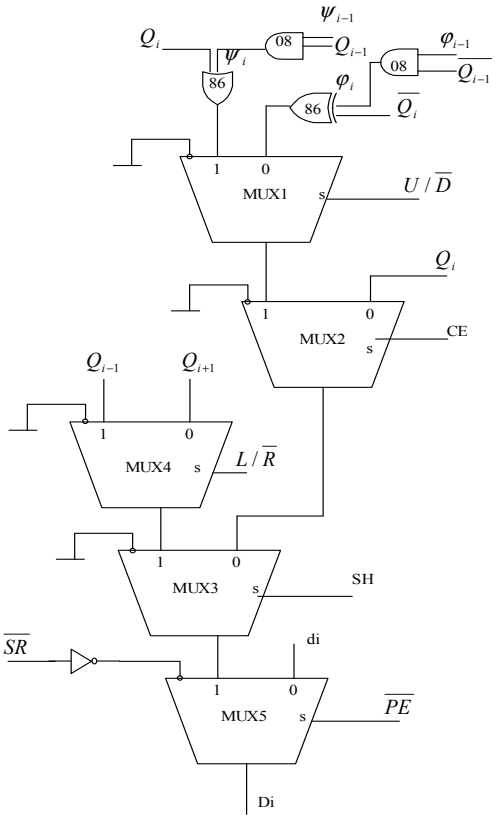

Fig. 2. Implementation of Table 1.

We used a MUX2x1, described in figure3, [1].

Taking into consideration the dynamic parameters for the digital circuits, like propagation delay times, set up times, hold times, one obtains the minimum expression for the clock period (equation (8) in [1]).

$$
\begin{aligned}
\underline{T_i} &= \underline{t_{s\_u}}(D) + \overline{t_b} + \overline{t_p}(\varphi_i,\psi_i) + \overline{t_p}(86) + 4\overline{t_p}(MUX) = \\
&= \underline{t_{s\_u}}(D) + \overline{t_b} + i\cdot\overline{t_p}(86) + \overline{t_p}(86) + 4\overline{t_p}(MUX) = 230n\sec
\end{aligned}
\tag{2}
$$

or $\overline{f_1} = 4{,}35MHz$

In the same paper [1], we calculated the same maximum clock parameters $(\overline{T}, \overline{f})$ for another priority order criteria when the best values were obtained.

So, the conclusion from [1] is that dynamic parameters depends on the priorities allocation modes.

III. MULTIFUNCTIONAL REGISTER WITH CODED MODE VALIDATION SIGNALS.

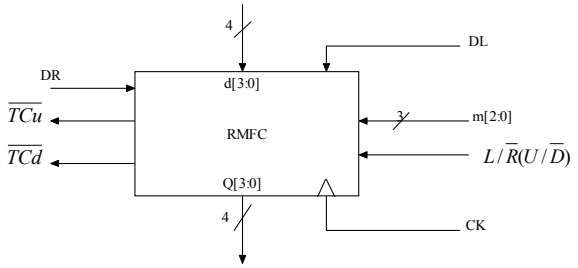

Fig. 3. Multifunctional Register with coded mode inputs

Signals: $m[2:0]$ - mode inputs, $L/\overline{R}(U/\overline{D})$ - operation type signal (shift Right/Left, Count Up/Down)

Data: d[3:0] – input data in parallel load mode, Q[3:0] – data output, DR,DL – serial input data, right shift,left shift, $\overline{TC_u}$ - Terminal Count Up (Q[3:0]=1111, last state), $\overline{TC_D}$ - Terminal Count Down (Q[3:0]=0000, last state).

Signals Description - mode input signals $m_2m_1m_0$ - functionality code mode, described in Table II.

TABLE II. FUNCTIONAL CODING MODE FOR RMF

$m_2m_1m_0$	$L/\overline{R}(U/\overline{D})$	Ck	Mode
000	Φ	Φ	NONE(Hold)
001	Φ	↑	Reset
010	Φ	↑	Parallel Load
011	0	↑	Count Down
011	1	↑	Count Up
100	0	↑	Shift Right
100	1	↑	Shift Left
101	Φ	Φ	NONE(Hold)
11Φ	Φ	Φ	NONE(Hold)

Taking into account the functionality Table, we obtained the following equations (D_i - data inputs on flip-flop D, Q_i - outputs from flip-flop D).

$$D_i = (\overline{m_2}\cdot\overline{m_1}\cdot\overline{m_0} + m_2\cdot\overline{m_1}\cdot m_0 + m_2\cdot m_1)Q_i + \overline{m_2}\cdot\overline{m_1}\cdot m_0\cdot 0 + \overline{m_2}\cdot m_1\cdot\overline{m_0}\cdot d_i +$$
$$+\overline{m_2}\cdot m_1\cdot m_0[\overline{U/\overline{D}}\cdot(\overline{Q_i}\oplus\varphi_i) + U/\overline{D}\cdot(Q_i\oplus\Psi_i)] + m_2\cdot\overline{m_1}\cdot\overline{m_0}[\overline{L/\overline{R}}\cdot Q_{i+1} + L/\overline{R}\cdot Q_{i-1}] =$$
$$= \overline{m_2}\{\overline{m_1}\cdot[\overline{m_0}\cdot Q_i + m_0\cdot 0] + m_1[\overline{m_0}d_i + m_0(\overline{U/\overline{D}}\cdot(\overline{Q_i}\oplus\varphi_i) + U/\overline{D}\cdot(Q_i\oplus\Psi_i))]\} +$$
$$+\overline{m_2}\{\overline{m_1}[\overline{m_0}\cdot(\overline{L/\overline{R}}\cdot Q_{i+1} + L/\overline{R}\cdot Q_{i-1}) + m_0 Q_i] + m_1 Q_i\} \tag{3}$$

$$D_3 = \overline{m_2}\{\overline{m_1}\cdot[\overline{m_0}Q_3 + m_0\cdot 0] + m_1\cdot[\overline{m_0}d_3 + m_0(\overline{U/\overline{D}}\cdot(\overline{Q_3}\oplus\varphi_3) +$$
$$+U/\overline{D}\cdot(Q_3\oplus\Psi_3))]\} + m_2\{\overline{m_1}\cdot[\overline{m_0}\cdot(\overline{L/\overline{R}}\cdot DR + L/\overline{R}\cdot Q_2) + m_0 Q_3] + m_1 Q_3\}$$

$$D_0 = \overline{m_2}\{\overline{m_1}\cdot[\overline{m_0}Q_0 + m_0\cdot 0] + m_1[\overline{m_0}d_0 + m_0(\overline{U/\overline{D}}\cdot(\overline{Q_0}\oplus\varphi_0) +$$
$$+U/\overline{D}\cdot(Q_0\oplus\Psi_0))]\} + m_2\{\overline{m_1}\cdot[\overline{m_0}\cdot(\overline{L/\overline{R}}\cdot Q_1 + L/\overline{R}\cdot DL) + m_0 Q_0] + m_1 Q_0\}$$

$$\overline{TC_u} = \overline{\overline{m_2}m_1m_0(U/\overline{D})\cdot\prod_{i=0}^{3}Q_i}$$

$$\overline{TC_d} = \overline{\overline{m_2}m_1m_0(\overline{U/\overline{D}})\cdot\prod_{i=0}^{3}\overline{Q_i}}$$

Figure 4 illustrates a possible implementation for the (i^{th}) rank:

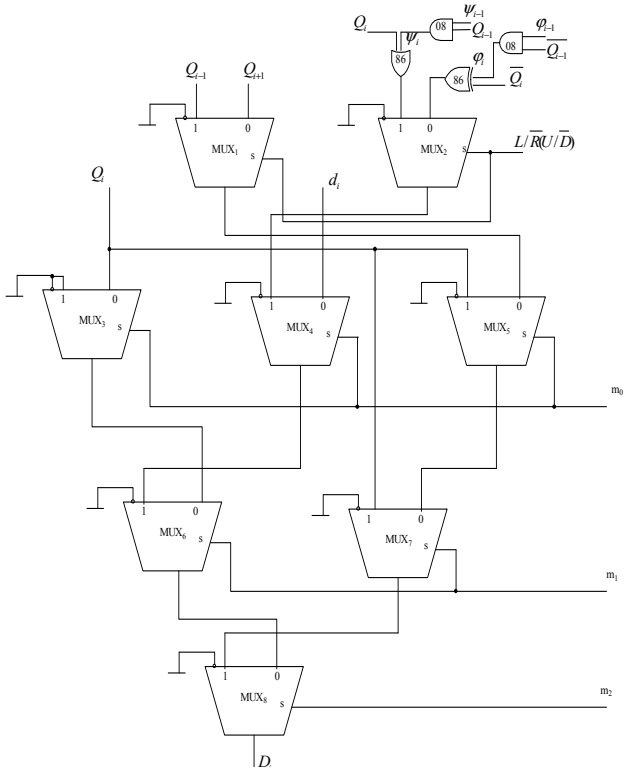

Fig. 4. Implementation of flip-flop D data inputs (Coded mode RMF inputs)

The same idea is used for the 3…0 ranges. For results comparison [1], we consider the implementation of the RMF with the same components like those used for decoded RMF inputs:

- the 3x1 MUXes are made with 3-State buffers SN74LS126A(4 buffers on chip) - $\overline{t_{pLH}} = \overline{t_{pHL}} = 15n\sec$

- flip-flop D with SN74LS74A ($\overline{t_p} = 40n\sec$, $\underline{t_{su}} = 20ns$)

- AND circuits, 74LS08 ($\overline{t_p}(08) = 20n\sec$)

- XOR circuits, 74LS86 ($\overline{t_p}(86) = 30n\sec$)

The signal wave diagram is presented in figure 5.

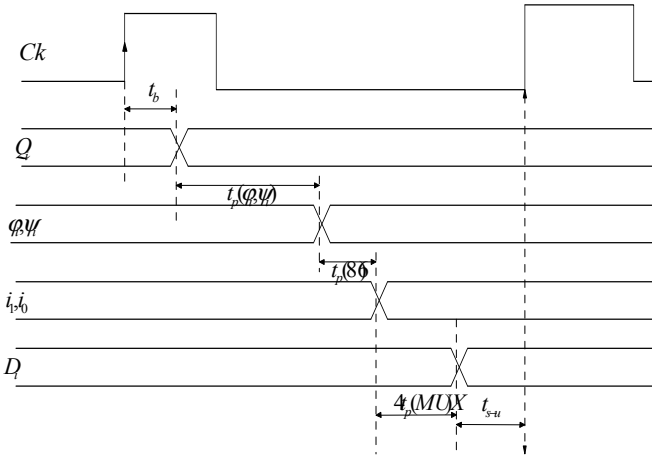

Fig. 5. Signal wave diagram

To satisfy the minimum flip-flop -D time set-up we imposed $t_{s\_u} \geq \underline{t_{s\_u}}$, and obtained:

$$\underline{T} = \underline{t_{s\_u}(D)} + \overline{t_p} + \overline{t_p(\varphi_i,\psi_i)} + \overline{t_p(86)} + 4\cdot\overline{t_p(MUX)} \tag{4}$$

Introducing the specific values for the circuits in [1], we obtained eq.(4.1):

$$\underline{T} = \underline{t_{s\_u}(D)} + \overline{t_b} + i\cdot\overline{t_p(08)} + \overline{t_p(86)} + 4\cdot\overline{t_p(MUX)} = 230n\sec$$
$$\overline{f} = \frac{1000}{\underline{T}}MHz \cong 4.35MHz \tag{4.1}$$

We obtain a maximum frequency for the clock CK, same as in [1], for the most disadvantageous case of priority order.

In Table III are presented the following parameters: $\overline{f}$ - maximum frequency and C – cost for all the RMF variants (the parameters for the RMF with mode decode signals were

presented in [1]).We refer only to the logical structure for the signals producers Di. The rest of the signals are the same.

TABLE III. DIFFERENT RMF TYPES COMPARISON

RMF type		$\overline{f}[MHz]$	**C**
Decode mode	Priority order[1]	4.25	13/3 CI
	Priority order[2]	5	13/3 CI
Code mode	-	4.35	17/3 CI

The priority order is described in Table IV:

TABLE IV. THE PRIORITY ALLOCATION VARIANTS FOR THE MODE DECODE RMF

(1)	**(2)**
Reset	Reset High Priority
Paralell Load	Count .
Shift	Shift .
Count	Paralell Load Low Priority

The C cost is the same with the number of the CI (elementary circuits) used for range (i) RMF implementation (without flip-flop).

IV. SYNTHESIS OF SEQUENTIAL SYSTEM USING RMF

Let us consider a Moore sequential system with the fluence graph table from Fig.6 (the graph node is $\frac{s_n}{z_n}$, where s_n represents the current state and z_n represents the output.

We say that the 8 state is a parasitic one because the system may be casual in it.

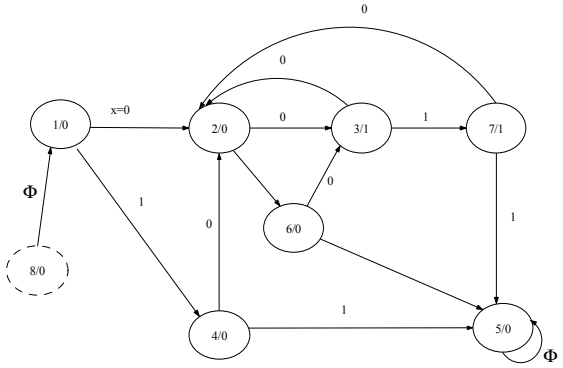

Fig. 6. Moore Automata

TABLE V. MOORE FLUENCE TABLE

s_n / z_n \ x	0	1
1/0	2	4
2/0	3	6
3/1	2	7
4/0	2	5
5/0	5	5
6/0	3	5
7/1	2	5
8/0	1	1

Table V represents the fluence table for the fluence graph in figure 6. We will make an inventory for possible transitions like:

$1 \to 2 \to 3 \to 7 \to 5$

$2 \to 3 \to 2$

$2 \to 6 \to 3,2 \to 6 \to 5$

$1 \to 4 \to 5,1 \to 4 \to 2$

$8 \to 1$

For the every state we attach a 3 bit binary code. For the first transitions we use the following encoding: $1 \leftrightarrow 000, 2 \leftrightarrow 001, 3 \leftrightarrow 010, 7 \leftrightarrow 011, 5 \leftrightarrow 100$

We chose this coding type because the count functions is the most simple.

For the next (4,6,8) states, we will use the following encoding: $4 \leftrightarrow 101, 6 \leftrightarrow 110, 8 \leftrightarrow 111$

Using Table 4 and the states codification, results the Table VI , who represents an evidence for the working mode.

TABLE VI. WORKING MODE STATES

s_n / z_n \ x	0	1
1/0	INC	PL
2/0	INC	PL
3/1	DEC	INC
4/0	PL	DEC
5/0	HOLD	HOLD
6/0	PL	PL
7/1	PL	INC
8/0	RES	RES

V. SYNTHESIS FOR THE SEQUENTIAL SYSTEM USING RMF WITH DECODED MODE SIGNALS

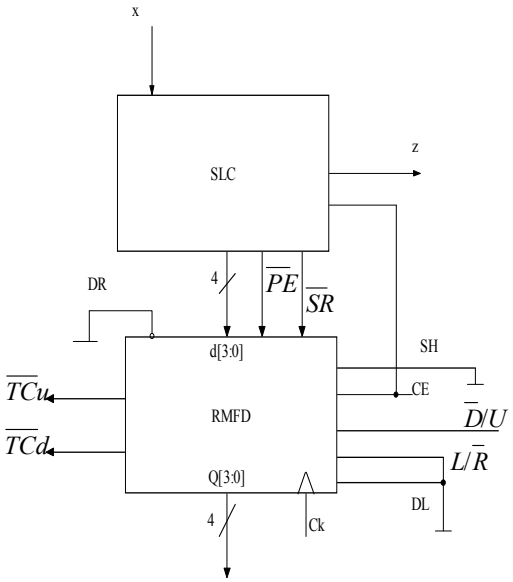

Fig. 7. Automata implementation architecture with RMFD

Considering the (1) priority order ($\overline{SR}$ - High Priority, SH – Low Priority), using Tables 3 and 4, results the next truth table, figure 8.

$Q[2:0]_{n+1}$ / X, $Q_2Q_1Q_0$	$\overline{SR}$ 0	$\overline{SR}$ 1	$\overline{PE}$ 0	$\overline{PE}$ 1	CE 0	CE 1	Z 0	Z 1	Z -
000	001	101	1	1	1	0	1	-	0
001	010	110	1	1	1	0	1	-	0
011	001	100	1	1	0	1	-	1	1
010	001	011	1	1	1	1	0	1	1
110	010	100	1	1	0	0	-	-	0
111	000	000	0	0	-	-	-	-	0
101	001	100	1	1	0	1	-	0	0
100	100	100	1	1	1	1	-	-	0

X, $Q_2Q_1Q_0$	d_2 0	d_2 1	d_1 0	d_1 1	d_0 0	d_0 1	$\overline{D}/U$ 0	$\overline{D}/U$ 1
000	-	1	-	0	-	1	1	-
001	-	1	-	1	-	0	1	-
011	0	-	0	-	1	-	-	1
010	-	-	-	-	-	-	0	1
110	0	1	1	0	0	0	-	-
111	-	-	-	-	-	-	-	-
101	0	-	0	-	1	-	-	0
100	-	-	-	-	-	-	-	-

Fig. 8. Truth table

The unused mode or data inputs are disabled by connection to ground (0 logic). The truth table was completed as:

if priority comand signal is activated, the next low priority signals are *x* (0 or 1 logic)

$\overline{D}/U$ has 0 logic value for the DEC mode and 1 logic value for INC mode

in the Hold state, the commands are disabled

Using the Veitch-Karnaugh method, the following equations result:

$$\begin{aligned}
&\overline{SR} = \overline{Q_2}\,\overline{Q_1}\,\overline{Q_0}\\
&\overline{PE} = \overline{x}\cdot[Q_1Q_0 + Q_2Q_1 + Q_0Q_2] + x\cdot(Q_2 \oplus \overline{Q_1})\\
&CE = \overline{Q_2}\,\overline{Q_0}\\
&\overline{D}/U = \overline{x}\cdot Q_1 + xQ_2\\
&Z = \overline{Q_2}Q_1\\
&d_3 = 0\\
&d_2 = x\\
&d_1 = \overline{Q_0} \oplus x\\
&d_0 = \overline{x}\cdot Q_0 + x\cdot\overline{Q_2}\,\overline{Q_0}
\end{aligned} \tag{5}$$

If the SLC system would be implemented with logic gates as: AND (2 inputs), OR(2 inputs), XOR(2 inputs), NAND(2 inputs), 23 elementary circuits will be obtained, $C_{dec} = 23$.

VI. SYNTHESIS OF SEQUENTIAL SISTEM USING RMF WITH CODED MODE SIGNALS (RMFC)

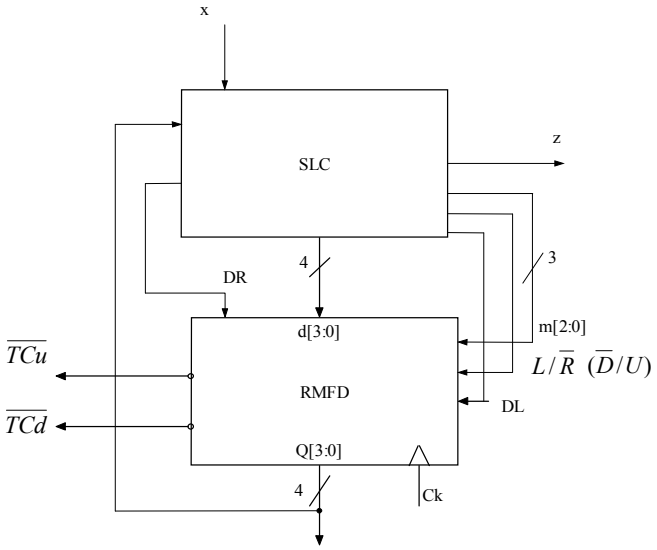

Fig. 9. Architecture of automata implemented with RMFC

If we consider the same Moore automata, we use only the Hold, Reset, INC/DEC, Parallel Load functions as:

- $m_2 \equiv 0$ (use only 2 codify bits)
- DL=DR=0 (the Shift function isn't used)
- $d_3 = 0$ (the automaton has only 8 states)

The codifying function is defined in Table VII.

TABLE VII. MODE CODING

$m_2m_1m_0$	Mode	$L/\overline{R}(U/\overline{D})$
000	HOLD	-
001	Reset	-
010	Parallel Load	-
011	Count Up	1
011	Count Down	0

The proposed system use only one RMFC and SLC which generates the output *z* signals, m[2:0], $L/\overline{R}(U/\overline{D})$.

The fluence table from figure 7 became the one illustrated in Table VIII.

TABLE VIII. STATE TRANSITION MATRIX

	$Q[2:0]_{n+1}$		m_1m_0		Z
X $Q_2Q_1Q_0$	0	1	0	1	-
000		101	011	010	0
001		010	011	010	0
011	001		010	011	1
010			011	011	1
110	010	100	010	010	0
111			001	001	0
101	001		010	011	0
100			000	000	0

	$L/\overline{R}(U/\overline{D})$		d_2		d_1		d_0	
X $Q_2Q_1Q_0$	0	1	0	1	0	1	0	1
000	1	-	-	1	-	0	-	1
001	1	-	-	1	-	1	-	0
011	-	1	0	-	0	-	1	-
010	0	1	-	-	-	-	-	-
110	-	-	0	1	1	0	0	0
111	-	-	-	-	-	-	-	-
101	-	0	0	-	0	-	1	-
100	-	-	-	-	-	-	-	-

Using one of the synthesis methods (Veitch-Karnaugh, Quine-McCluskey), we obtain the following equations:

$$
\begin{aligned}
&m_1 = \overline{Q_2} + Q_1 \oplus Q_0 \\
&m_0 = \overline{Q_2} \cdot (x \oplus \overline{Q_1}) + x \cdot Q_2Q_0 + Q_1 \cdot (Q_2 \oplus \overline{Q_0}) \\
&z = \overline{Q_2}Q_1 \\
&U/\overline{D} = x \oplus \overline{Q_1} \\
&d_2 = x \\
&d_1 = x \oplus \overline{Q_0} \\
&d_0 = \overline{x}Q_0 + x\overline{Q_1} \cdot \overline{Q_0}
\end{aligned}
\qquad (6)
$$

If the above equations are implemented with the same elementary circuits, results a cost of 18, C=18.

VII. CONCLUSIONS:

1. In this paper, the authors made a comparison between the 2 types of RMF: RMFD – Multifunctional Registers with validation signals of decoded commands and RMFC – Multifunctional Registers with coded mode signals.

2. The authors, already presented RMFD in [1] and the conclusion is that the maximum frequencies depends on the priority allocation order for the mode signals.

3. From the comparison of 2 multifunctional register types, RMFD and RMFC results: RMFC allows a maximum frequency(lower than RMFD) but a cost greather than RMFD (See Table 5).

4. In section 3, the authors syntesized a Moore automata and proposed an original constribution to the states coding, taking into account that the most used RMF functions must be used mosed often (INC/DEC most often and Parallel Load as little).

The logical structure (SL) will contain less elementary circuits for the RMFC implementation as compared with RMFD, so there is cost compensation (RMFC more expensive than RMFD).

5. To design an automaton with separate inputs, which can be activated simultaneously, it is necessary to use RMFD.

REFERENCES

[1] Al. Valachi, M. Timis, B. Aignatoaie, S. Tarcau, "Orders Priorities Settings Criteria for Multifunctional Registers," Electronics and Electrical Engineering ISSN 1392 – 1215, T 120, System Engineering, Computer Technology, The 14th International Conference ELECTRONICS'2010 Kaunas and Vilnius University – Lituania, http://www.ee.ktu.lt/page.php?227 - T120, page.87.

[2] Al. Valachi, M. Timis, M. Danubianu, "Some to Synthesis and Implementation of Multifunctional Registers," 11th WSEAS Int.Conf. on Automatic Control, Modelling & Simulation (ACMOS'09), Proceedings of the 2nd WSEAS International Conference on Multivariate Analysis and its Application in Science and Engineering Istanbul, Turkey, May 30 - June 1, 2009, ISSN: 1790-5117, ISBN: 978-960-474-083-3, WSEAS Press, www.wseas.org, pp..146-149.

[3] C. H. Roth, Fundamentals of Logic Design, West Publishing Company, 1999.

[4] Al. Valachi, R.Silion, M. Timis, "Improvement of FSM Synthesis using MSI and LSI Circuits," Advances in Electrical and Computer Engineering. Academy of Technical Sciences of Romania, "Stefan cel Mare" University of Suceava, vol 5(12), no.1/23, 2005.

[5] Al.Valachi, M.Chis-Ster, "States Coding Program for Wired Sequencers Designed with Shift Registers", 4th International Symposium on Automatic Control and Computer Science, SACCS'93, 29-30 Oct. 1993, Iasi.

[6] V. Onofrei, Al. Valachi, "Systematic Synthesis of Wired Sequencers Using Counters and Registers," MICROCAD'1999 – Miskolc, Hungary.

[7] ***, XXX – Fast and LS TTL Data. Motorola INC, 1992.