

# REALIZAREA UNEI MEMORII ADRESABILE DUPĂ CONȚINUT

Iacob Petrescu

*iacobp@quadrilogic.ro*

Costică Nitu

*cnitu\_upb2001@yahoo.com*

**Rezumat:** În lucrare, se propune o soluție pentru implementarea unei memorii adresabile după conținut atât cu ajutorul structurilor reconfigurabile de tip FPGA, cât și sub forma de ASIC. Sunt detaliate componentele structurale ale memoriei asociative și se prezintă descrierea Verilog, implementarea VLSI (ASIC), cât și simularea celulei de baza, inclusiv a logicii asociate.

**Cuvinte cheie:** memorii asociative, memorii adresabile după conținut, tablouri asociative, comparând, repondent, FPGA, ASIC, Verilog.

**Abstract:** The paper presents a new content addressable memory implementation based on reconfigurable logical structures-Field Programmable Gates Arrays and on Application Specific Integrated Circuits. Details concerning associative memory structural components are provided, including as well: Verilog description, elements of ASIC implementation, basic cell and associate logic simulation.

**Key Words:** associative memories, content addressable memories, associative arrays, comparand, repondent, FPGA, ASIC, Verilog.

## 1. Introducere

*Memoriile adresabile după conținut* (MAC) reprezintă o categorie specială de memorii, care facilitează accelerarea operațiilor de regăsire a informațiilor [1], [2], [3]. MAC mai poartă și numele de *memorii asociative* (MA) sau *tablouri asociative* (TA), ultima denumire fiind utilizată frecvent în domeniul structurilor de date [4].

Spre deosebire de memoriile obișnuite, cu acces aleator (RAM-Random Access Memory), la care regăsirea informației se realizează pe baza adresei locației în care este stocată aceasta, la MAC se explorează în diferite moduri (paralel/serial la nivel de cuvânt/bit etc.) toate locațiile de memorie pentru a stabili acele locații care conțin, eventual, informația căutată. Informația căutată trebuie să corespundă unui *comparand* (C). Rezultatul căutării este reprezentat sub forma unui vector *repondent* (R), ale cărui componente egale cu 1 sunt asociate cu locațiile în care s-a regăsit informația corespunzătoare comparandului, în timp ce componentele egale cu 0 sunt asociate cu locațiile care nu conțin informația căutată. Uneori, informațiile returnate pot reprezenta diferite date asociate cu câmpurile din locațiile al căror conținut a coincis cu comparandul. MAC reprezintă o implementare hardware a ceea ce, în software, este denumit TA.

*Implementări ale MAC.* Având în vedere faptul că majoritatea aplicațiilor impun examinarea în paralel a tuturor locațiilor, sistemele de regăsire a informațiilor bazate pe MAC sunt mult mai rapide decât sistemele care utilizează RAM. Pentru a realiza această performanță, spre deosebire de celulele individuale RAM, celulele MAC trebuie să dispună de circuite care detectează relația (=, <, > etc.) între bitul stocat și bitul corespunzător al comparandului. În plus, la nivelul fiecărui locație/cuvânt din MAC trebuie să existe circuite suplimentare, care generează bitul corespunzător al repondentului, în conformitate cu îndeplinirea sau neîndeplinirea relației dorite pentru toți biții din cuvânt. Circuitele adiționale contribuie la creșterea dimensiunilor, a puterii disipate și a costurilor de producție pentru MAC. Circuitele MAC, de regulă, sunt utilizate în aplicațiile speciale de regăsire a informației, care necesită o viteză mare de operare. În ultimii ani, s-au propus noi implementări, de perspectivă, bazate pe tehnologii optice/holografice. Având în vedere aceste aspecte critice, s-a încercat găsirea unui compromis între viteză, cost și dimensiunile memoriei, prin emularea funcțiilor realizate de către MAC, prin tehnici hardware de căutare în arbori, "hashing", cât și prin tehnici de replicare sau bandă de asamblare pentru creșterea performanței. În lucrarea de față, se propune o soluție care facilitează realizarea MAC, utilizând circuite de tip FPGA (Field Programmable Gate Array – Tablouri de porți programabile de către utilizator), având în vedere versatilitatea și costurile relativ scăzute ale acestora, cât și ușurința de implementare. Implementare nu își propune minimizarea ariei ocupate sau a puterii disipate.

Cea mai simplă celulă MAC este *celula binară*, care stochează valorile 0/1 și care permite căutarea/compararea numai după aceste două valori. În realitate, se întâlnesc aplicații în care anumiți biți din informația căutată nu sunt relevanți, ceea ce sugerează stocarea celei de-a treia valori "X", care este indiferentă și care presupune prezența unei *celule ternare*. Astfel, MAC poate avea stocată informația 1X0X1, care corespunde, în condițiile date, mai multor comparanți: 10001, 10011, 11001, 11011. De regulă, comparandul C este însoțit de un cuvânt mască M, folosit de logica de comparație, pentru a inhiba biții indiferenți din cuvântul stocat în MAC. Celulele ternare vor folosi două celule binare, ocupând astfel un spațiu mai mare.

Memoriile asociative holografice realizează termenii indiferenți pe baza unui model matematic, care utilizează reprezentarea de valori complexe.

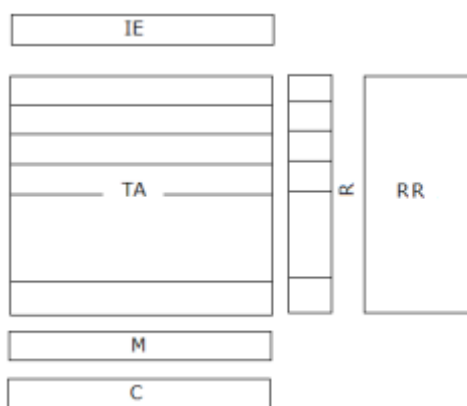
În domeniul MAC, au fost elaborate o serie de standarde cărora li se aliniază producătorii de asemenea dispozitive: Cypress Semiconductor, Integrated Device Technology, IBM etc. [5].

Memoriile adresabile după conținut își găsesc aplicații în implementarea tabelor de rutare, ale rutelor, în controloarele memoriilor cache și ale tampoanelor de translatare asociativă (TLB), în sistemele orientate pe baze de date, în dispozitivele de compresie a datelor, în rețelele neuronale artificiale, în sistemele de securitate, în aplicațiile speciale de tip C4 ș.a.

Lucrarea de față furnizează o soluție pentru memoria și unitatea de execuție ale unor coprocesoare/procesoare asociative atașate, care se pot folosi în conjuncție cu sistemele de calcul convenționale, în vederea accelerării operațiilor de regăsire a informației în aplicațiile care necesită o viteză mare de prelucrare.

## 2. Structura unei memorii adresabile după conținut

O memorie adresabilă după conținut, complet paralelă, este alcătuită, de regulă, din următoarele componente (figura 1):



**Figura 1. Una dintre structurile generale ale unei memorii adresabile după conținut**

- **TA:** Tabloul Asociativ (m linii și n coloane) - memorie și resursele hardware pentru căutare;
- **C:** Registrul Comparand (n ranguri) - conține data care se compară cu conținutul celulelor memoriei;
- **M:** Registrul Masca (n ranguri) - maschează zone din datele care nu participa (sunt indiferente) la operația de căutare;
- **IE:** Registrul de I/E (n ranguri) - joacă rol de tampon între memoria asociativă și mediul extern;
- **R:** Registrul Respondenților (m ranguri) - stochează rezultatul căutării la nivelul fiecărui cuvânt din memoria asociativă;
- **RR:** Resolver Respondenți (m ranguri) – rețea care soluționează situațiile în care mai multe cuvinte din **TA** satisfac condiția de potrivire.

Memoria adresabilă după conținut poate efectua, ca și o memorie obișnuită RAM, operații de citire/scriere sau operații specifice de căutare/modificare după diverse criterii. În cele ce urmează, vor fi examinate numai operațiile cu caracter asociativ, care pot fi descrise la nivelul transferurilor între registre (RTL) astfel:

*Căutare asociativă:*

$$R \leftarrow \text{ASOC}(\mathbf{TA}, C) \quad (1)$$

*Modificare/Scriere asociativă:*

$$\mathbf{TA} * R \leftarrow \text{RIE}; \quad (2)$$

unde  $\text{ASOC}(\mathbf{TA}, C)$  este o funcție logică combinațională, care are drept argumente tabloul asociativ **TA** și Comparandul **C**.

ASOC generează un vector **R** cu  $2^m$  componente binare, dintre care nici una, una sau mai multe pot fi egale cu unu; în organizarea din figura 1, se presupune că vectorul generat de ASOC poate avea 0, 1 sau mai multe componente egale cu 1.

### 3. Implementarea componentelor hardware ale CAM

Spre deosebire de abordarea convențională a realizării MAC [6], [7], în cele ce urmează se propun soluții bazate pe porți logice și bistabile, care conduc la implementări atât sub formă de FPGA, cât și de ASIC (Application Specific Integrated Circuits). În continuare, vor fi examinate propunerile de implementare pentru registrul/registrele Respondenților, a logicii asociate, cât și pentru registrele Comparand și Masca.

#### 3.1. Celula de bază

În această lucrare, celula de bază se centrează pe un bistabil M-S de tip D, cu intrări de: date, ceas, reset și ieșiri directă/negată (figura 2).

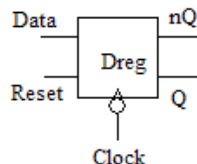


Figura 2. Bistabil MS, de tip D, cu intrare de ceas activă pe front negativ

Implementarea la nivelul porților/tranzistoarelor este prezentată în figura 3.

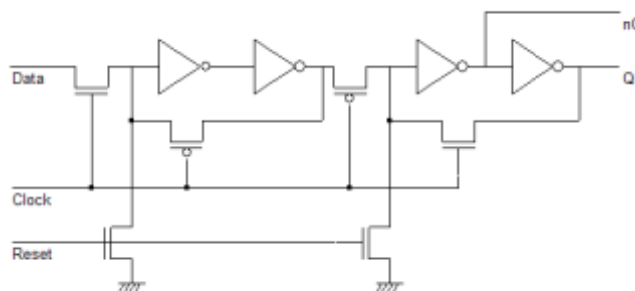


Figura 3. Reprezentarea bistabilului din figura 2 la nivelul implementării cu ajutorul inversoarelor și al tranzistoarelor

Celula de bază trebuie să asigure operațiile de: comparare a conținutului cu comparandul, citire și scriere. În ceea ce privește compararea, s-au implementat numai operațiile pentru egalitate și mai mic (figura 4.), întrucât rezultatul comparării pentru mai mare se obține pe cale logică, din rezultatele primelor două.

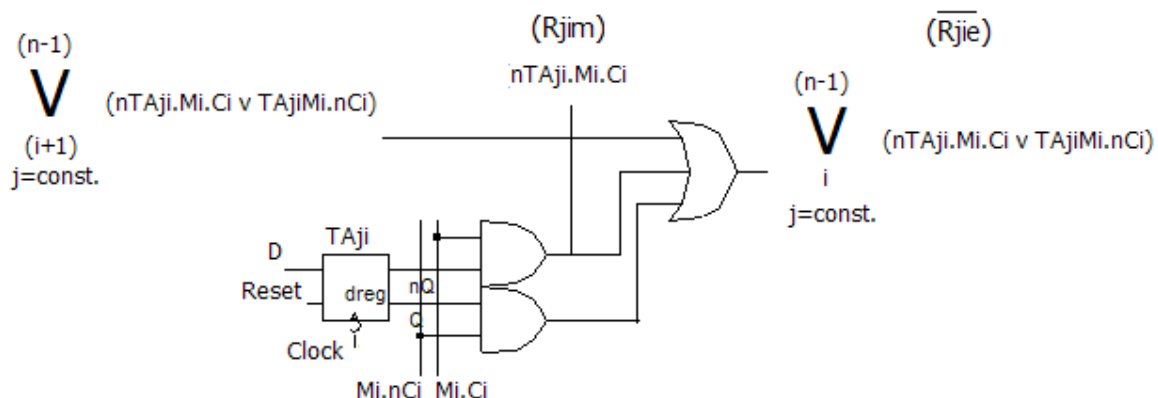


Figura 4. Celula de bază TA<sub>ji</sub> și logica asociată pentru obținerea rezultatelor mai mic R<sub>ji</sub> și egal R<sub>je</sub>

Se remarcă faptul că, pentru a masca unele ranguri ale cuvintelor din tabloul asociativ, se generează semnalele Mi.nCi și Mi.Ci, care traversează vertical tabloul, la nivelul coloanelor, într-o manieră întrepesută. În desen, semnalele negate sunt precedate de prefixul “n”.

Considerând un rang i, din cuvântul TA<sub>j</sub> al Tabloului Asociativ, rang notat cu TA<sub>ji</sub>, se poate constata

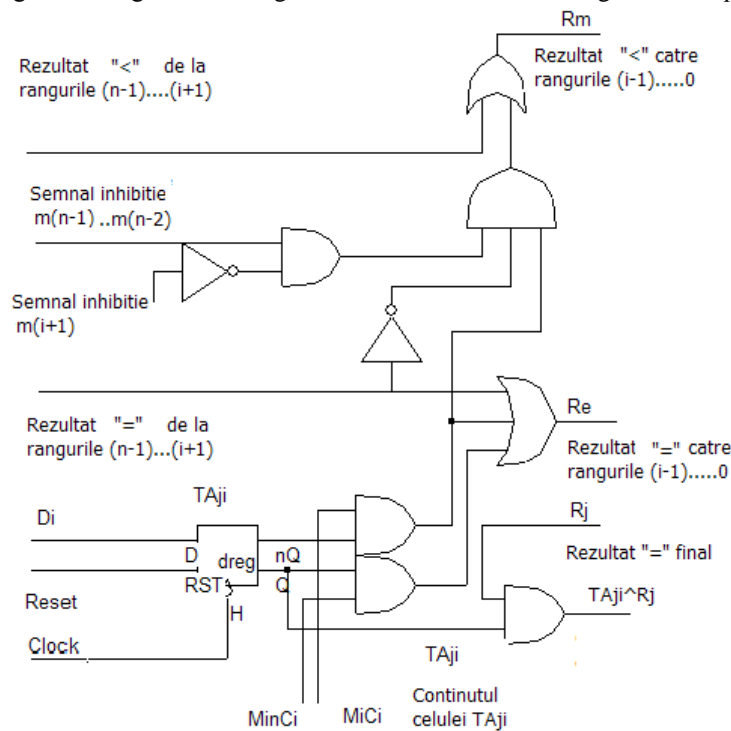
cu ușurință că rezultatele/respondenții operațiilor de comparare  $R_{jim}$  și  $R_{jie}$ , pentru relațiile mai mic ( $TA_{ji} < C_i$ ) și pentru egal ( $TA_{ji} = C_i$ ), în condițiile în care unii termeni ai Comparandului pot fi mascați cu biții  $M_i$ , ai Măștii, se materializează în ecuațiile (3) și (4):

$$R_{jim} = \overline{TA_{ji}} \cap \overline{M_i} \cap C_i \quad (3)$$

$$\overline{R_{jie}} = \overline{TA_{ji}} \cap \overline{M_i} \cap C_i \cup TA_{ji} \cap M_i \cap \overline{C_i} \quad (4)$$

Informația stocată în  $TA$ ,  $C$  și  $M$  reprezintă numere pozitive sau numere pozitive și negative deplasate, ca și în cazul exponenților în virgula mobilă. Astfel, operațiile de comparare a cuvintelor din  $TA$  vor conduce la o logică relativ simplă, la nivelul fiecărei locații.

Structura rangului  $i$ , din linia  $j$ , a tabloului asociativ  $TA$  este dată în figura 5. Se observă, în jumătatea inferioară a desenului, bistabilul  $TA_{ji}$ , traseele verticale pentru semnalele  $M_i \cdot nC_i$  și  $M_i \cdot C_i$ , porțile SI, care realizează termenii neidentității logice, cât și poarta SAU, cu trei intrări, prin care se propagă, sub formă de sumă logică, inegalitatea logică de la rangul cel mai semnificativ la rangul cel mai puțin semnificativ.

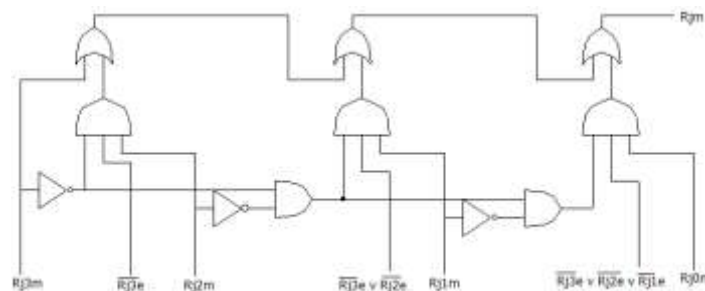


**Figura 5. Structura rangului  $i$ , din linia  $j$ , a tabloului asociativ  $TA$**

Pentru operația de comparare mai mic, se va explora, la nivelul locației  $j$ , din  $TA$ , vectorul  $R_{j(n-1)m}$ , de la stânga la dreapta, respondenții  $R_{jm}$  fiind generat de o schemă bazată pe priorități (figura 5). Schema implementează expresia (5):

$$R_{jm} = \dots (R_{j(n-1)m} \cup (R_{j(n-1)m} \cap R_{j(n-1)e} \cap R_{j(n-2)m}) \dots \cup \dots (R_{j(n-1)} \cap \dots \cap R_{j1m} \cap R_{j1e} \cap R_{j0m}) \quad (5)$$

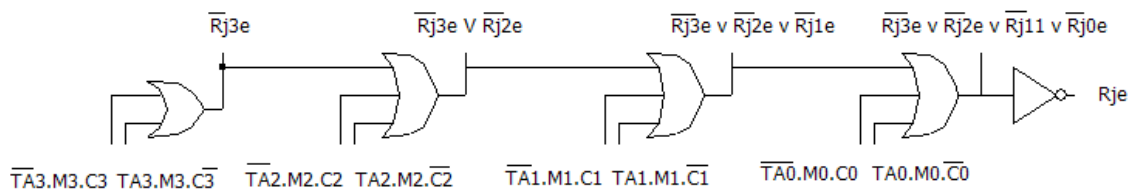
În figura 6, se prezintă schema pentru generarea rezultatului „mai mic” în cazul unei structuri pe 4 biți.



**Figura 6. Logica pentru generarea  $R_{jm}$  la nivelul locației  $j$  din  $TA$ , în cazul unei structuri pe 4 biți**

În mod asemănător, se obțin expresia logică pentru operația de comparare de egalitate (6), cât și schema de implementare (pentru o structură pe 4 ranguri) (figura 6):

$$R_{je} = (\dots((\overline{R_j(n-1)e} \cup \overline{R_j(n-2)e} \cup \overline{R_j(n-3)e} \dots) \cup \overline{R_j0e}) \quad (6)$$



**Figura 6. Logica pentru generarea Rje la nivelul locației j din TA**

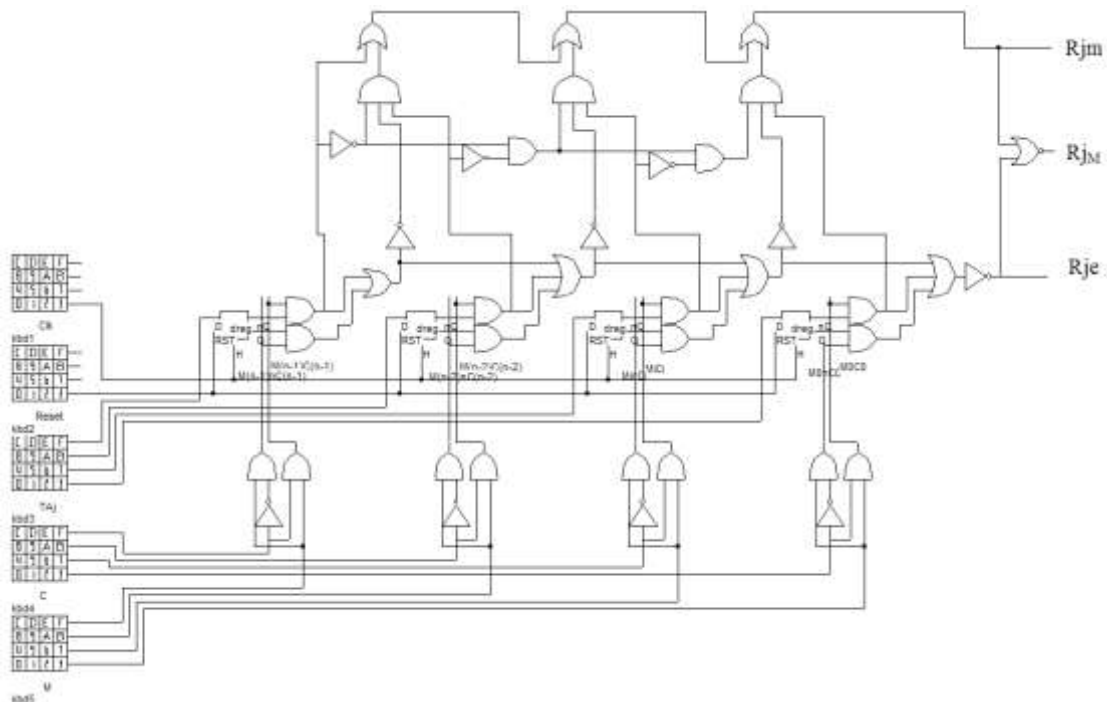
Pentru realizarea relațiilor “>”, “≥” și “≤”, materializate prin respondenții:  $R_{jM}$ ,  $R_{jMe}$  și  $R_{jme}$ , se pot utiliza la nivelul fiecărei linii j a TA combinații de porți, care prelucrează logic semnalele  $R_{je}$  și  $R_{jm}$ , după cum urmează:

$$R_{jM} = (R_{jm} \cup R_{je}) \quad (7)$$

$$R_{jMe} = (R_{jM} \cup R_{je}) \quad (8)$$

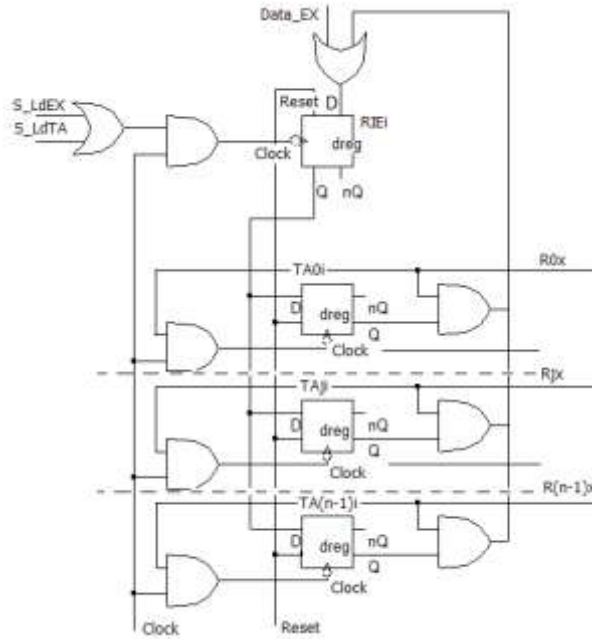
$$R_{jme} = (R_{jm} \cup R_{je}) \quad (9)$$

În figura 7, se prezintă schema de simulare, cu ajutorul aplicației Dsch2 [8], a structurii liniei j, pe 4 ranguri, a tabloului TA, cât și a logicii de obținere a respondenților:  $R_{jm}$ ,  $R_{jM}$  și  $R_{je}$ :



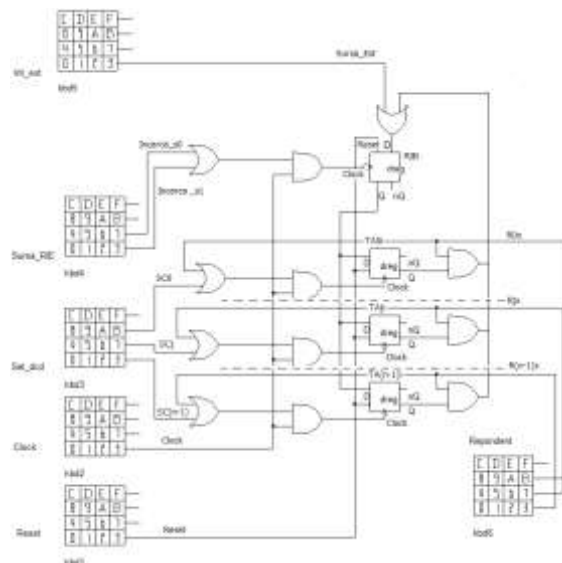
**Figura 7. Schema de simulare a structurii liniei j, pe 4 ranguri, a tabloului TA, cât și a logicii de obținere a respondenților:  $R_{jm}$ ,  $R_{jM}$  și  $R_{je}$**

Circuitele de scriere/citire în/din celula de baza, pentru coloana i, a tabloului TA, sunt prezentate în figura 8.



**Figura 8. Circuitele de scriere/citire în/din celula de baza, pentru coloana i, a tabloului TA**

În partea superioară a figurii 8, se află celula i a registrului de I/E (RIE), care poate avea două surse pentru scriere: o sursă externă, asociată cu semnalul de comandă S\_LdEX, și o sursa din coloana i a tabloului asociativ TA, controlată de semnalul S\_LdTA, în cazul în care este activată o linie a respondenților R0x,...,R(n-1)x. Litera "x" se substituie uneia dintre relațiile: "<" (m), "=" (e), ">" (M), "≤" (me), "≥" (Me). Partea inferioară a figurii conține coloana i, a tabloului TA, împreună cu circuitele de eșantionare a semnalului de ceas, de către semnalele corespunzătoare respondenților, în cazul scrierii asociative.



**Figura 9. Schema de simulare a structurii coloanei i, pe n ranguri, a tabloului TA, cât și a logicii de scriere/citire, folosind registrul de I/E (RIE)**

### 3.2. Registrul/registrele Respondenților și logica asociată

În implementarea de față, structura corespunzătoare respondenților, la nivelul bitului j, este dată în figura 10.

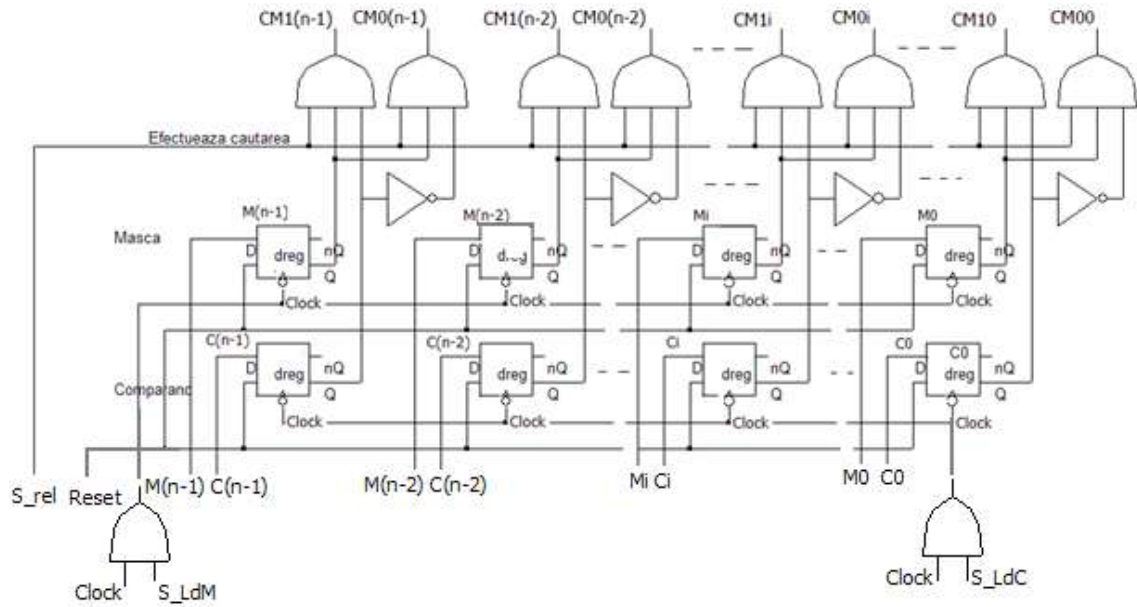


CMi0, care corespund formulelor (6) si (7), relațiilor următoare:

$$CMi1 = Mi \cap Ci \quad (10)$$

$$CMi0 = Mi \cap \overline{Ci} \quad (11)$$

Încărcarea registrelor C și M se realizează cu ajutorul semnalelor de comandă S\_LdC și S\_LdM, în timp ce ieșirile CMi1 și CMi0 devin active sub controlul semnalului S\_rel

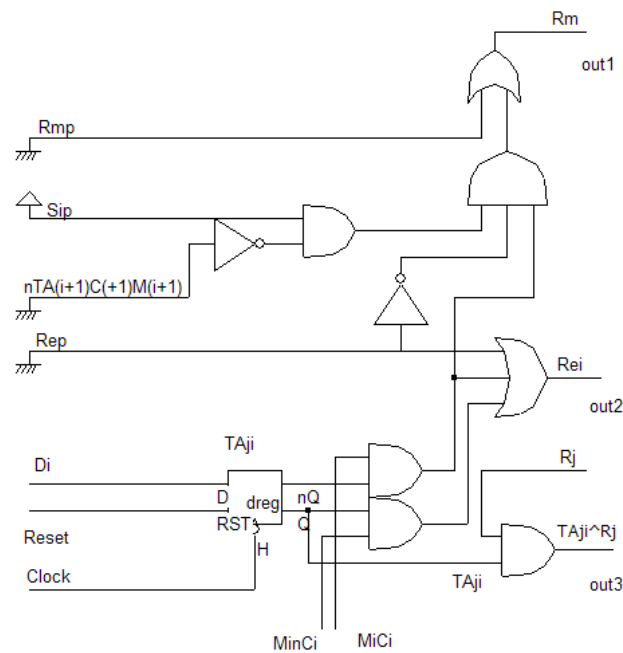


**Figura 12. Registrele Comparând, Masca și logică asociată**

În Tabloul Asociativ, traseele semnalelor CMi1 și CMi0 traversează vertical structura acestuia.

### 3.4. Implementarea/simularea celulei de baza TAji în VLSI

Implementarea în VLSI a celulei de bază, prezentată în figura 13, se bazează pe descrierea Verilog, la nivel structural, care este dată mai jos:



**Figura 13. Celula de baza TAji**

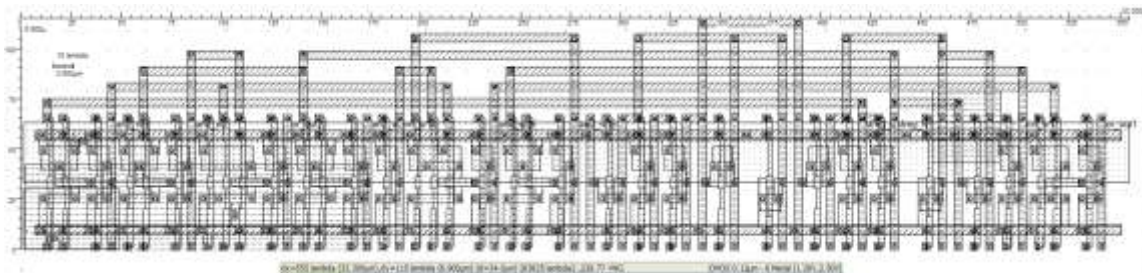


```

// DSCH 2.7f
// 9/9/2007 12:05:32 PM
// D:\Jacob_Unelte\Export dsch2\TA_Cell_5_bun.sch
module TA_Cell_5_bun( Di ,Clock, MinCi,MiCi,Rj,Reset,out2,out3,out1);
input Di ,Clock,MinCi,MiCi, Rj,Reset;
output out2,out3,out1;
and #(23) and(w5,w3,MiCi);
and #(16) and(w8,MinCi,w7);
or #(19) or(out2,vss,w5,w8);
and #(16) and(out3,w7,Rj);
not #(10) inv(w14,vss);
and #(16) and(w16,w15,w14,w5);
and #(16) and(w15,w17,vdd);
dreg #(19) dreg(w7,w3,Di,Reset,Clock);
not #(10) inv(w17,vss);
or #(16) or(out1,vss,w16);
endmodule
// Simulation parameters în Verilog Format
// Simulation parameters
// Clock CLK 10 10
// Di CLK 20 20
// MinCi CLK 40 40
// MiCi CLK 80 80
// Rj CLK 160 160
// Reset CLK 640 640

```

Compilarea în Si a programului de mai sus a generat ansamblul de măști din figura 14.



**Figura 14. Măștile pentru zona activă a celulei de baza TA<sub>ji</sub>**

Pentru implementarea VLSI, s-a considerat tehnologia CMOS 0,12  $\mu\text{m}$  ( $\lambda=0.06 \mu\text{m}$ ), Tensiune de alimentare 2,5 V. în cadrul acestei tehnologii tranzistoarele NMOS și PMOS au următoarele dimensiuni ale canalelor:  $L_n=0,120 \mu\text{m}$ ,  $W_n = 0,240 \mu\text{m}$ ;  $L_p=0,120 \mu\text{m}$ ,  $W_p = 0,720 \mu\text{m}$ . Dimensiunile rezultate ale celulei TA<sub>ji</sub> sunt:  $dx = 33,30 \mu\text{m}$  și  $dy = 6,90 \mu\text{m}$ , ceea ce conduce la o arie a structurii active egală cu  $229,77 \mu\text{m}^2$ .

Formele de undă rezultate în urma simulării celulei de baza TA<sub>ji</sub>, din figura 14, sunt prezentate în figura 15. Zona de interes este asociată cu intervalele de timp în care  $\text{Reset} = 0$ , semnalele MinCi și MiCi au valori diferite sau sunt zero, întrucât  $M=0$ . Pentru verificarea corectitudinii operării celulei de baza s-a marcat o asemenea zonă în figura 15. Examinarea relațiilor între semnalele MiCi, MinCi și TA<sub>ji</sub>  $\cap$  Rj, ca intrări, pe de-o parte, și semnalele de ieșire Re<sub>i</sub> (out1) și Re<sub>i</sub> (out2), pe de alta parte, pune în evidență corectitudinea operării celulei de baza, inclusiv a logicii asociate.

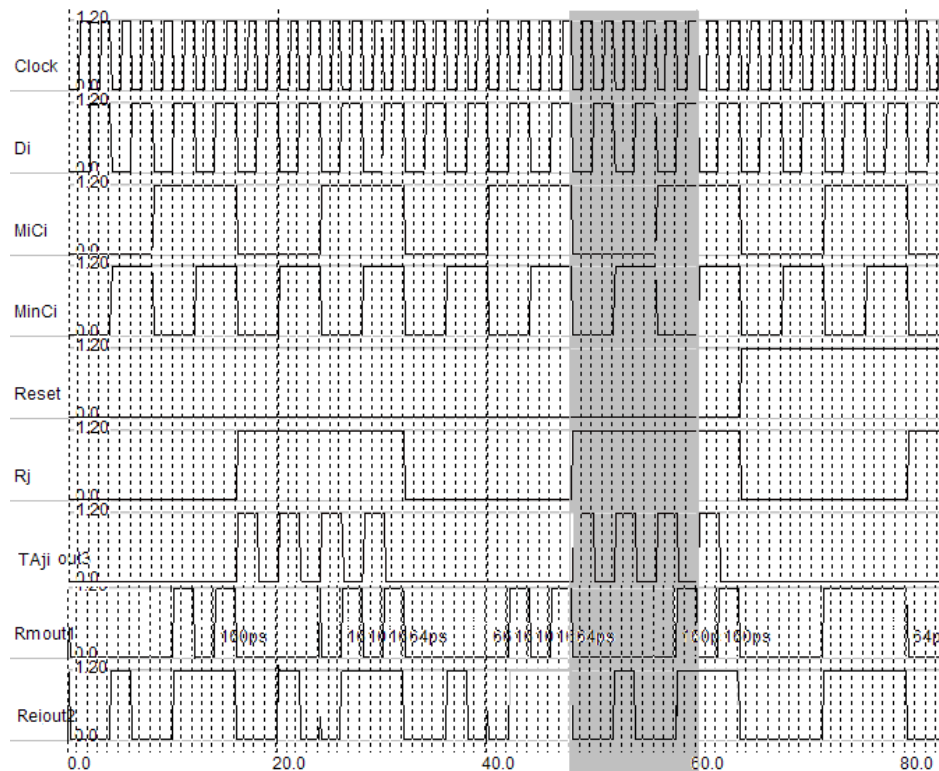


Figura 15. Rezultatele simulării operării celulei de baza TAji, inclusiv a logicii asociate

## 4. Concluzii

În lucrare s-a propus o implementare a celulei de bază și a logicii asociate acestuia, inclusiv a circuitelor de citire și scriere pentru o memorie de tip asociativ. Soluția propusă are avantajul obținerii, sub forma de respondenți, a rezultatelor comparațiilor de tip: “<”, “=”, “>”, “≤” și “≥”. Între conținutul celulei TAji și bitul mascat al comparandului, CiMi. Rezultatele comparațiilor se obțin simultan, într-o singură perioadă de ceas. Atât simularea logică, la nivelul porților logice, cât și simularea comportamentală a celulei de bază, implementată sub forma VLSI demonstrează valabilitatea soluției propuse. Astfel, se creează premisele realizării unității de execuție pentru un coprocesor/ procesor atașat de tip asociativ, care poate fi implementat, fie cu ajutorul circuitelor FPGA, fie sub formă de circuit ASIC.

## Bibliografie

1. **KRIKELIS, A., C. WEEMS** (eds.): Associative Processing and Processors, IEEE Computer Science Press. ISBN 0-8186-7661-2. 1997.
2. **PAGIAMTZIS, K., A. SHEIKHOESLAMIA.**: Content-addressable Memory (CAM) Circuits and Architectures: A Tutorial and Survey. IEEE Journal of Solid-State Circuits, Vol. 41, No. 3, March 2006, pp. 712–727.
3. **CHO, S, J. R. MARTIN, R. XU, M. H. HAMMOUD, R. MELHEM:** CA-RAM: A High-performance Memory Substrate for Search-intensive Applications. International Symposium on Performance Analysis of Systems and Software, April 2007, pp. 333–338.
4. **AANNUM et all.:** System and Method for Resetting and Initaliasing a Full Associative Array to a Known State at Power on or through Machine Specific State. U.S. Patent 6823434.
5. \*\*\* [www.oiforum.com](http://www.oiforum.com).
6. **SCHULTZ, K.J.:** Content-Addressable Memory Core Cells: A Survey, Integration, the VLSI Journal 23, No. 2, November 1997, pp. 171-188.
7. **XILINX Corp.** [www.xilinx.com/ipcenter/catalog/logicore/docs/cam.pdf](http://www.xilinx.com/ipcenter/catalog/logicore/docs/cam.pdf)
8. **SICARD, E., S. BENDHIA:** [Microwind & Dsch Version 3.0](#).