

10. CIRCUITE DE MEMORIE, CIRCUITE LOGICE PROGRAMABILE

10.1.Circuite de memorie

Circuitele de memorie sunt circuite care asigură posibilitatea de regăsire a unor informații reprezentate sub forma binară și care au fost stocate anterior. Din punct de vedere al modului de utilizare, memoriile se clasifică astfel:

- ROM, PROM, EPROM - memoriile ce pot fi doar citite în sistemul care le utilizează; ștergerea informației este posibilă doar în cazul EPROM și nu este selectivă
- RAM, EEPROM - memoriile ce permit atât citirea cât și înscrisura informației.

10.1.1 Memorii ROM

Memoriile ROM (*Read Only Memories*) sunt circuite de memorie care permit citirea informației înscrisă în prealabil de către fabricant. Aceste memoriile sunt nevolatile, adică informația nu se distruge la dispariția tensiunii de alimentare. Structura lor internă este concepută matricial, în fiecare nod al matricii fiind memorat câte un bit de informație.

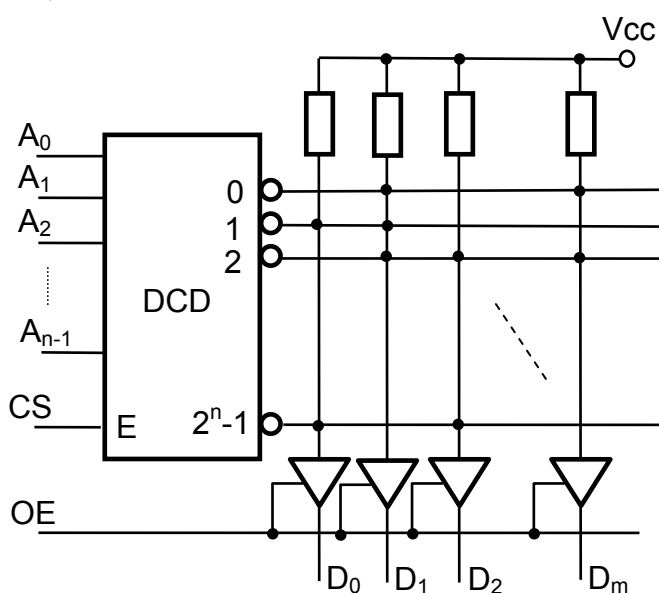


Figura 10.1 Structura de principiu a unei memorii ROM

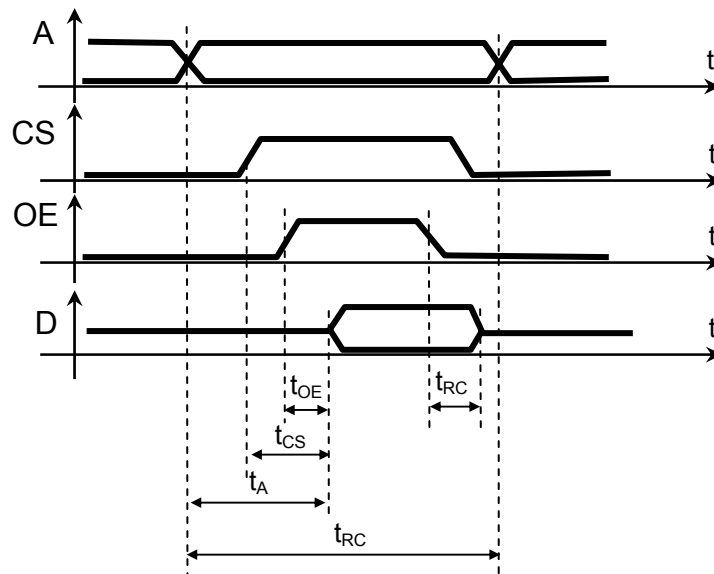


Figura 10.2 Diagrame de timp pentru o memorie ROM

Linile matricii ($0 \div 2^n - 1$) sunt activate de ieșirile unui decodificator ale cărui intrări sunt conectate la liniile de adresare. Coloanele matricii sunt conectate la liniile de date prin intermediul unor amplificatoare având ieșiri three-state.

Semnalul CS realizează selecția circuitului (el poate controla și starea ieșirilor). Semnalul OE controlează numai starea circuitelor de ieșire. Dacă $CS = 1$, cuvântul aplicat pe intrările de adresă determină activarea unei linii a matricii de memorare. Dacă în nodurile de pe aceasta linie există conexiuni pe coloanele respective potențialul va fi determinat de ieșirile decodificatorului, respectiv "0" logic. Dacă nu sunt realizate conexiuni în noduri, potențialul coloanelor respective este corespunzător lui „1” logic, determinat prin rezistențele legate la V_{cc} . Informația de pe cele m coloane de date este prezentată la ieșire dacă $OE = 1$.

Pentru funcționarea corectă este necesară satisfacerea anumitor condiții temporale (figura 10.2), principalii parametri de timp fiind cei de mai jos.

- t_{RC} - durata ciclului de citire (intervalul de timp între două schimbări de adresă)
- t_A - timpul de acces la memorie (timpul scurs de la stabilirea adreselor până la obținerea datelor de ieșire)
- t_{CS} - timpul scurs de la $CS = 1$ până la obținerea datelor la ieșire
- t_{OE} - timpul scurs de la $OE = 1$ până la apariția datelor la ieșire
- t_{OH} - timpul de menținere a datelor la ieșire după ce OE este dezactivat ($OE = 0$).

Memoriile PROM (*Programmable Read Only Memories*) sunt circuite de memorie a căror conținut poate fi programat o singură dată, de către utilizator. După înscrierea informației, aceasta nu mai poate fi ștearsă.

Memoriile EPROM (*Electrically Programmable Read Only Memories*) au posibilitatea de înscriere repetată a conținutului, ștergerea acestora făcându-se prin expunere la radiație ultravioletă. Ștergerea este neselectivă.

Memoriile EEPROM sau E²PROM (*Electrically Erasable Read Only Memories*) elimină dezavantajele circuitelor EPROM: ștergerea întregului conținut, timp de ștergere ridicat, necesitatea scoaterii circuitului din circuit pentru ștergere. Ștergerea

informației din celule de memorare se realizează în cazul EEPROM prin aplicarea unor tensiuni de comandă a ștergerii, selectiv, pentru o celulă sau un grup de celule.

10.1.2 Memorii RAM

Memoriile RAM sunt memorii volatile la care memorarea informației logice se face folosind structuri de tip circuit basculant bistabil. Denumirea (Random Access Memory) se datorează faptului că accesul la orice cuvânt al memoriei se realizează în același interval de timp, spre deosebire de memoriile cu acces serial, la care trebuie parcurse toate cuvintele, de la adresa prezentă la adresa dorită.

Structura unei memorii RAM este tot matricială, în fiecare nod al matricii existând o celulă de memorare. Dacă celula de memorare are la bază o structură de bistabil, memoria este de tip static (SRAM) iar dacă celula de memorare are la bază înmagazinarea, pentru o durată finită, a unei sarcini electrice pe un condensator, memoria este de tip dinamic (DRAM) și necesită reîmprospătarea periodică a sarcinii electrice pe capacitățile de stocare (*refresh*). Memoriile DRAM prezintă capacitate de integrare mult superioară memoriilor SRAM dar necesită o logică suplimentară pentru *refresh*.

Structura unei memorii SRAM este prezentată în figura 10.3 iar diagramele de timp corespunzătoare citirii, respectiv scrierii sunt descrise în figura 10.4.

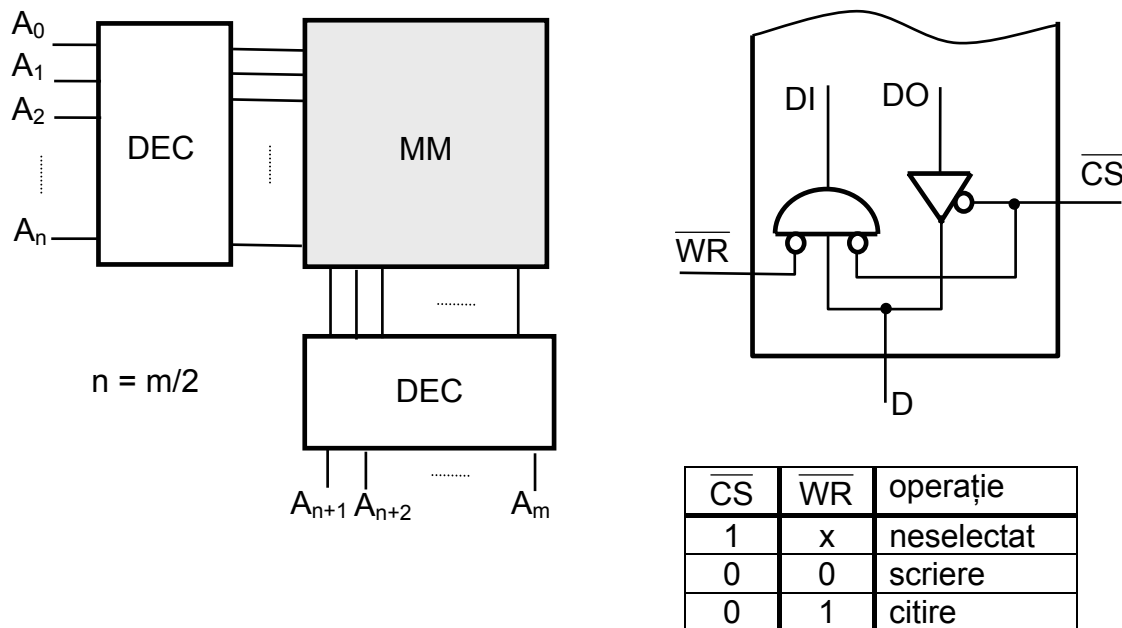


Figura 10.3 Structura de principiu a unei memorii SRAM

Căile de înscriere (DI – data in) și de citire (DO – data out) pot fi separate sau pot fi comune, caz în care intrarea se face printr-o poartă cu validare ($\overline{CS} + \overline{WR} = 0$), iar ieșirea printr-un amplificator *three-state* comandat corespunzător ($\overline{CS} = 0, \overline{WR} = 1$).

Structura internă poate fi detaliată ca în figura 10.5.

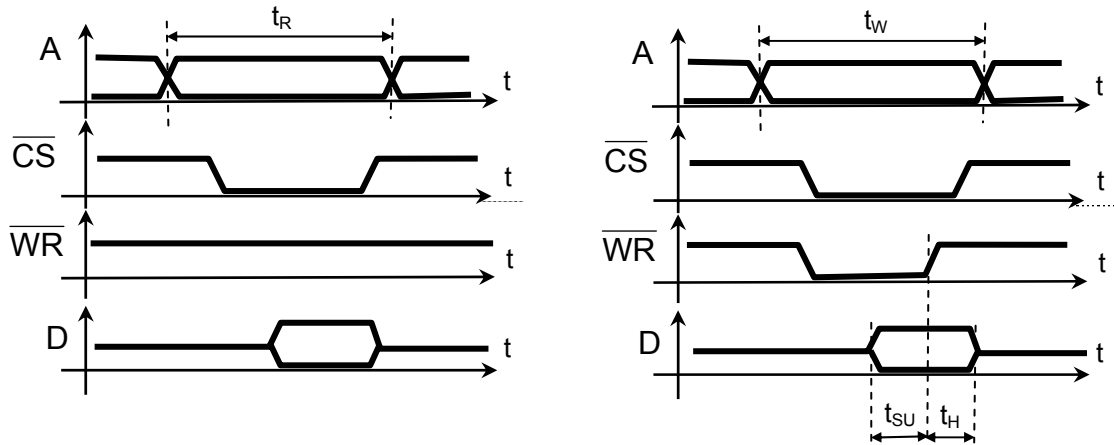


Figura 10.4 Diagrame de timp citire/scriere SRAM

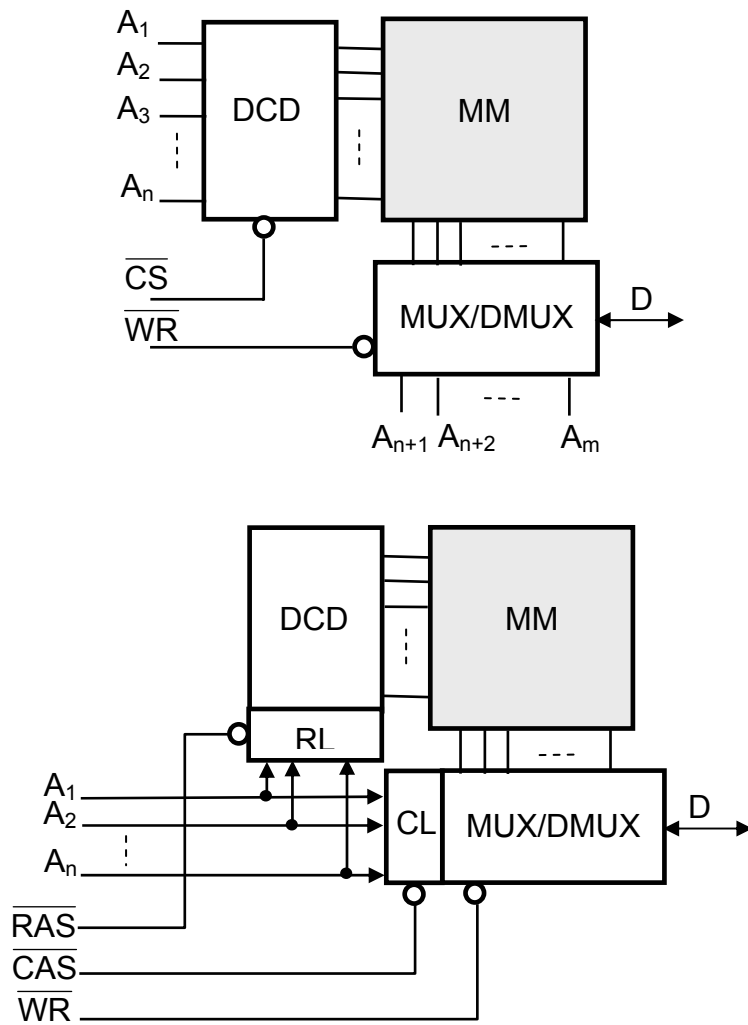


Figura 10.5 Structura internă SRAM, DRAM

Cu ajutorul liniilor de adresă $A_1 \div A_n$ se poate selecta, prin intermediul decodificatorului DCD, una din liniile matricii de memorare. Biții ce constituie linia adresată de decodificator sunt aplicați unui multiplexor/demultiplexor (MUX/DMUX) comandat de semnalul \overline{WR} . Liniile de adresă $A_{n+1} \div A_m$ selectează una dintre coloanele de date, conectând-o la linia externă de date D. Decodificatorul este validat de semnalul \overline{CS} . Un ciclu de memorie constă în selectarea unei linii (\overline{CS} , $A_1 \div A_n$) și apoi, după o întârziere dictată de propagarea prin decodificator și matricea de memorie, selectarea unei coloane ($A_{n+1} \div A_m$, \overline{WR}). Adresele liniilor și coloanelor nu sunt deci necesare simultan. Utilizând *latch*-uri, se poate imagina o schemă în care liniile pentru $A_1 \div A_n$ să fie comune cu cele pentru $A_{n+1} \div A_m$, dacă $n=m/2$. Acesta este principiul folosit la realizarea memoriilor DRAM. Operarea se face prin aplicarea pe liniile de adresă a adresei de linie, care este memorată în RL pe frontul negativ al \overline{RAS} , apoi pe aceleași linii de adresă se aplică adresa de coloană care este strobată în CL pe frontul negativ al \overline{CAS} . Sensul datelor și funcția structurii MUX/DMUX sunt determinate de polaritatea semnalului \overline{WR} .

În cazul memoriilor dinamice (DRAM), având în vedere că celulele memoriei DRAM funcționează pe baza stocării unei sarcini electrice pe o capacitate și că această capacitate se descarcă în timp, este necesară o operație de reîmprospătare a memoriei care să reîncarce capacitatea cel puțin o dată la 2ms. Reîmprospătarea se poate face simultan pentru toate celulele de pe o linie iar procesul de reîmprospătare a informației logice se numește *refresh*. Operațiunea de refresh presupune citirea conținutului memoriei înainte ca acesta să se deterioreze și reînscriserea celulelor, deci necesită o logică suplimentară.

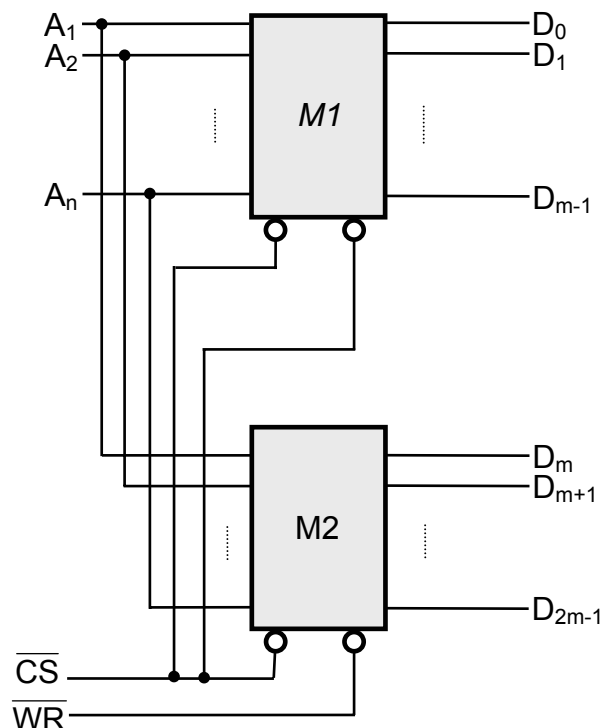


Figura 10.6 Modificarea geometriei memoriei prin expandarea cuvântului memorat

10.1.3 Modificarea geometriei memoriilor

Prin modificarea geometriei se realizează circuite de memorie cu număr de cuvinte sau număr de biți pe cuvânt diferit de cel al circuitelor disponibile. Pentru modificarea numărului de biți pe cuvânt se conectează în paralel circuitele de memorie disponibile. Semnalele de control și adresele sunt comune, funcționarea circuitelor fiind simultană (figura 10.6).

Pentru modificarea numărului de cuvinte, se aplică memoriilor folosite un număr de linii de adresă corespunzător capacității lor. Adresa de rang imediat superior se folosește pentru aplicarea selectivă a semnalelor de control (\overline{CS}) folosind o logică suplimentară cu porți OR (figura 10.7). Liniile de date sunt comune.

10.1.4 Aplicații ale memoriilor în sinteza dispozitivelor numerice

Circuitele de memorie pot fi folosite pentru a sintetiza circuite logice combinaționale sau circuite logice secvențiale, caz în care din schemă trebuie să facă parte și circuite pentru memorarea stării. Pentru a realiza un circuit combinațional, variabilele de intrare se aplică pe liniile de adresă. În fiecare locație a memoriei se înscriu valorile funcțiilor corespunzătoare vectorului de intrare ce reprezintă adresa locației respective (figura 10.8).

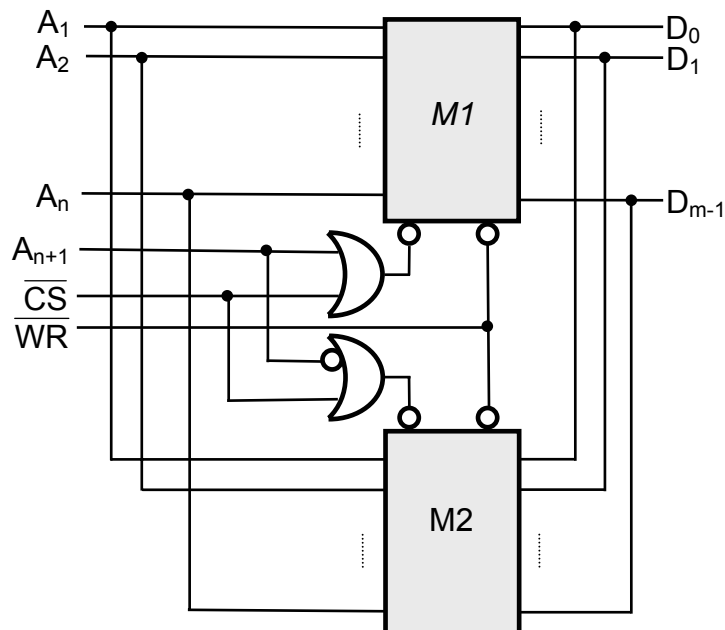


Figura 10.7 Modificarea geometriei memoriei prin expandarea numărului de cuvinte (expandarea adreselor)

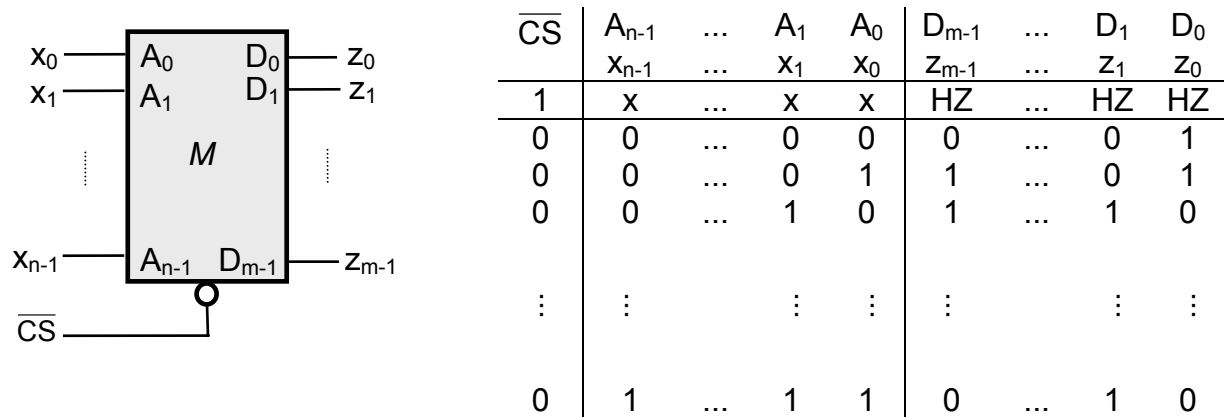


Figura 10.8 Sinteza unui circuit combinațional folosind memorii

Pentru sinteza unui sistem secvențial este necesară obținerea vectorului de stare, memorarea acestuia și realizarea vectorului de ieșire, în funcție de vectorul de stare și de vectorul intrare. Sistemul secvențial va fi constituit dintr-o memorie care conține în fiecare locație a sa vectorii y_{n+1} și z_n , adresa locației fiind fixată de y_n și x_n (ecuațiile 10.1, figura 10.9).

$$\begin{cases} y_{n+1} = f(x_n, y_n) \\ z_n = g(x_n, y_n) \end{cases} \quad (10.1)$$

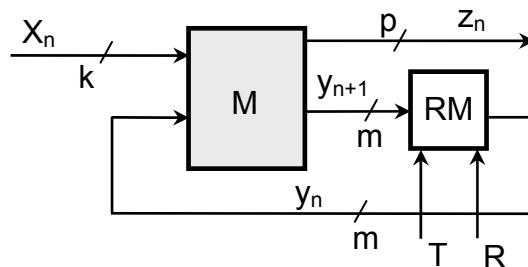


Figura 10.9 Sinteza unui circuit secvențial folosind memorii

Semnalul R este folosit pentru resetarea registrului de memorie RM, deci pentru aducerea sistemului în starea inițială. Evoluția sistemului este determinată de ceasul T.

10.2 Matrici logice programabile (PLA)

PLA-urile (*Programmable Logic Area*) sunt circuite care conțin o arie de porți logice interconectabile ce permit implementarea unei game foarte variate de CLC. În punctele de conexiune internă sunt plasate siguranțe fuzibile care pot fi arse, sau nu, de către utilizator, obținându-se astfel schema dorită.

Circuitul conține o matrice AND, o matrice OR și un circuit de ieșire. Forma canonică disjunctivă a unei funcții booleene este de forma 10.2.

$$z = \sum_{f=1}^{n-1} \left(\prod_{j=0}^{n-1} x_j^{\sigma_j} \right) = \sum_{f=1}^{n-1} Q_k^{(n)} \quad (10.2)$$

Matricea AND este constituită din circuite AND ce realizează termenii $Q_k^{(n)}$, care au pe intrări variabilele de intrare x_i, \bar{x}_i . Matricea OR conține circuite conectabile la toate ieșirile circuitelor AND și permite realizarea formei canonice disjunctive, FCD, a funcției booleene (z). Circuitul de ieșire conține circuite XOR care permit obținerea funcției logice z sau a inversei acesteia (\bar{z}) și buffere three-state controlate de un semnal de validare a ieșirii, \overline{OE} , (figura 10.10).

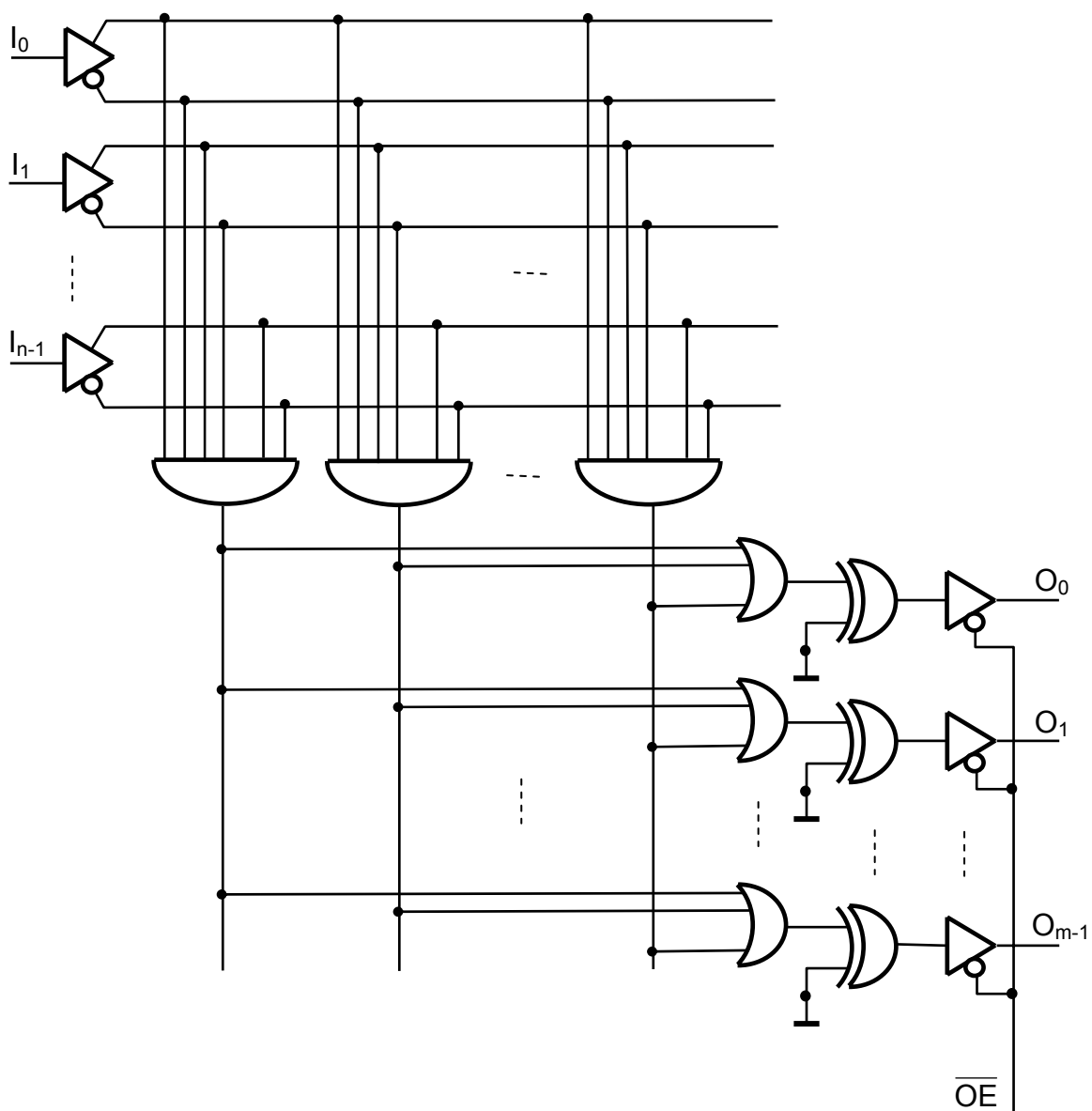


Figura 10.10 Structura de principiu a unei matrici logice programabile, PLA

PLA-urile pot fi folosite și pentru sinteza sistemelor secvențiale, permițând obținerea vectorilor de stare y_{n+1} și de ieșire z_n . Pentru întârzierea vectorului de stare se folosește un circuit de memorare (bistabil sau registru de memorie) ca și în cazul folosirii memoriilor pentru sinteza CLS.

10.3 Secvențatoare logice programabile (PLS)

PLS-urile (*Programmable Logic Sequencetor*) permit realizarea circuitelor logice secvențiale prin materializarea relațiilor cunoscute (vezi relațiile 10.1).

Schema bloc a unui astfel de circuit este prezentată în figura 10.11.

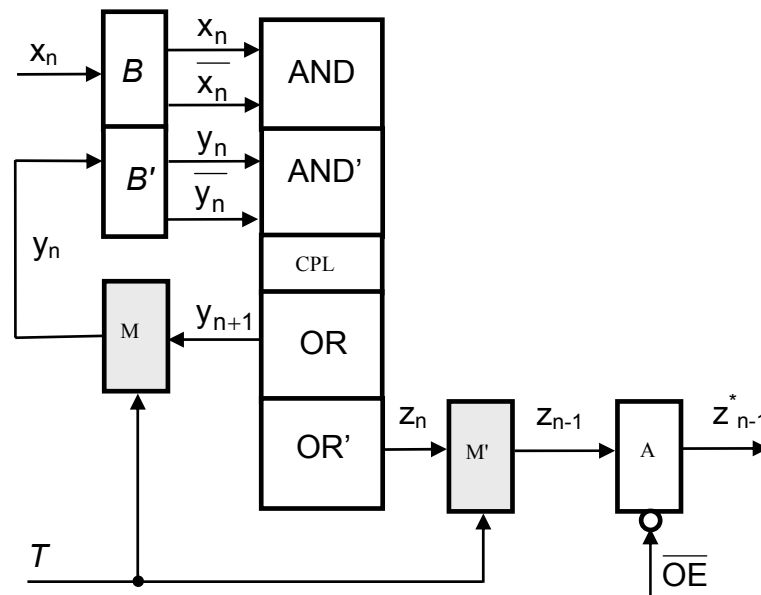


Figura 10.11 Schema bloc a unui secvențiator logic programabil (PLS)

Semnificația blocurilor funcționale și a semnalelor din figura 10.11 este următoarea:

- B, B' - buffere de intrare
- AND, AND' - matrici AND
- CPL - matrice complementară (permite realizarea unei reacții asincrone între OR și matricea AND în scopul minimizării numărului de termeni AND)
- OR, OR' - matrici OR
- M, M' - elemente de memorare (bistabile R-S)
- A - amplificatoare three-state de ieșire
- \overline{OE} - semnal de validare a ieșirilor
- T - semnal de ceas

Structura matricii complementare CPL este prezentată în figura 10.12.

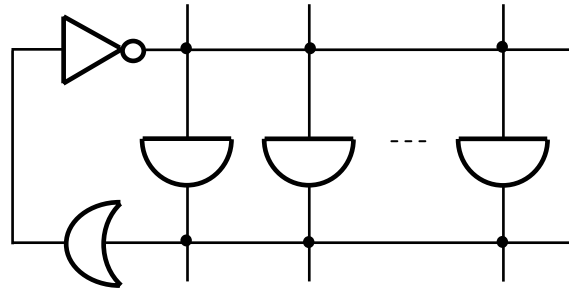


Figura 10.12 Structura matricii complementare CPL

Dacă nu se utilizează matricea CPL, siguranțele fuzibile vor fi arse în totalitatea lor. Dacă, de exemplu, două matrici AND furnizează mintermii Q_1 și Q_2 și este necesară și realizarea unui termen $Q_i = \overline{Q_1 + Q_2}$, se folosesc posibilitățile oferite de matricea CPL (figura 10.13).

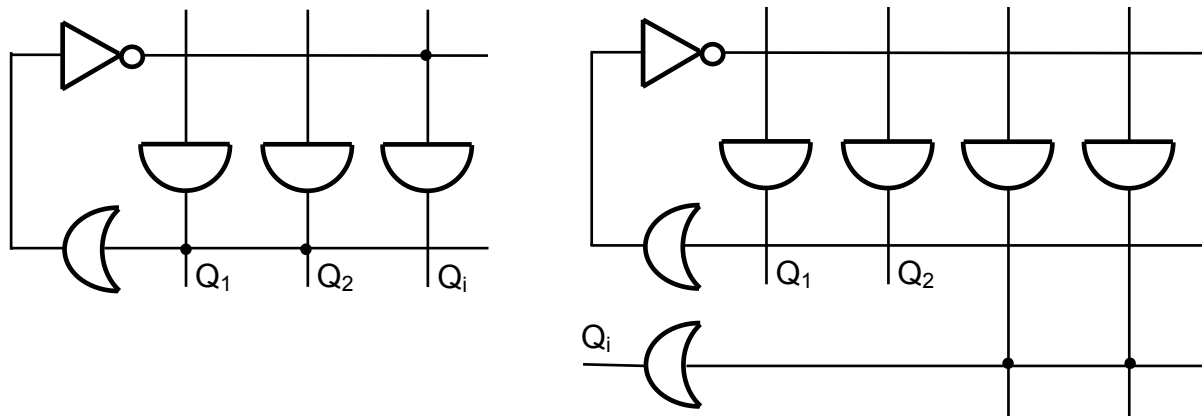


Figura 10.13 Realizarea unor termeni folosind matricea complementară CPL