Tabelul 3.12. Tabelul de funcționare al circuitului 74155.

Selecție	Strobare	Date	Ieşiri
B A	1G	1C	$\overline{1Y_0}\ \overline{1Y_1}\ \overline{1Y_2}\ \overline{1Y_3}$
x x	1	x	1 1 1 1
0 0	0	1	0 1 1 1
0 1	0	1	1 0 1 1
1 0	0	1	1 1 0 1
1 1	0	1	1 1 1 0
x x	x	0	1 1 1 1

Pentru realizarea unui circuit DMUX 1:8, terminalele 1*C* şi 2*C* se utilizează pentru selecție, iar 1*G* şi 2*G* pentru strobare sau pentru date (Figura 3.35).

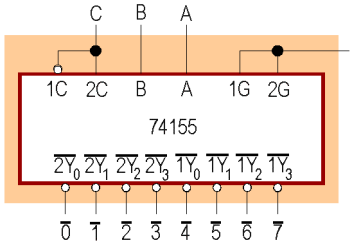

Figura 3.35. Utilizarea circuitului 74155 ca demultiplexor 1:8.

Pentru utilizarea ca decodificator, intrarea de date 1*C* se conectează la 1 logic. Numărul liniilor de ieşire se poate extinde prin conectarea în cascadă.

Deoarece la ieşirea unui demultiplexor se obțin toți termenii canonici minimali ai variabilelor de selecție multiplicați cu variabilele de intrare, cu ajutorul unui circuit DMUX 1:2^n se poate implementa orice funcție logică de *n* variabile sub forma canonică disjunctivă. Termenii canonici sunt introduşi într-o poartă SAU.

Demultiplexoarele se utilizează pentru transmiterea informației de la intrare la o anumită adresă, de exemplu la înscrierea informației în memorie.

3.5.6. Circuite logice programabile

Circuitele logice programabile, cunoscute şi sub forma acronimului PLD (*Programmable Logic Device*), sunt circuite integrate care conțin un număr mare de porți sau celule a căror interconexiune poate fi configurată sau "programată" pentru a implementa orice funcție combinațională sau secvențială dorită. Pentru programarea circuitelor PLD se utilizează două tehnici: programarea prin măşti, care se efectuează în timpul procesului de fabricație, şi programarea de către utilizator, pentru care se utilizează echipamente de programare cu costuri reduse. Multe circuite PLD pot fi

reprogramate de utilizator de multe ori, motiv pentru care ele sunt avantajoase pentru realizarea prototipurilor unui nou produs.

Conexiunile programabile între elementele logice ale unui circuit PLD conțin comutatoare realizare de obicei cu tranzistoare sau antifuzibile (uneori fuzibile). Porțile logice programabile ale unui circuit PLD pot fi reprezentate în mod simplificat ca în Figura 3.36 (b). În locul unor linii de intrare multiple la fiecare din aceste porți, se figurat o singură linie. Semnul × indică o conexiune programabilă a unei linii de intrare la o poartă logică. Absența semnului × indică faptul că respectiva conexiune a fost programată în starea deconectată.

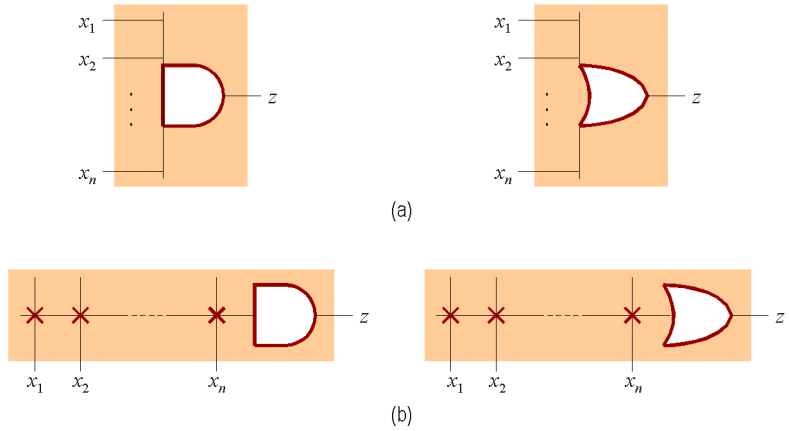

Figura 3.36. Porți ŞI, respectiv SAU: (*a*) reprezentare obişnuită; (*b*) reprezentare simplificată pentru circuitele PLD.

În continuare se prezintă unele categorii de circuite logice programabile: memorii ROM, rețele logice programabile şi circuite FPGA.

3.5.6.1. Memorii ROM

Memoriile ROM (*Read Only Memory*) reprezintă o categorie de circuite programabile care se pot utiliza pentru implementarea unui set de funcții logice. Aceste memorii conțin un set de locații utilizate pentru stocarea informației binare. Conținutul acestor locații este fixat în momentul fabricației, aceste memorii fiind programate prin măşti. Pe lângă acestea, există şi memorii ROM programabile care utilizează fuzibile, numite PROM (*Programmable* ROM). Acestea pot fi înscrise de către utilizator, cu ajutorul unui echipament de programare corespunzător. De asemenea, există memorii PROM care pot fi şterse cu ajutorul razelor ultraviolete, şi apoi pot fi reprogramate. Acestea se numesc EPROM (*Erasable Programmable* ROM). În fine, memoriile ROM care utilizează o tehnologie de ştergere prin impulsuri electrice se numesc EEPROM sau E^2PROM (*Electrically Erasable Programmable* ROM). Diferitele variante de memorii ROM sunt nevolatile, deci îşi păstrează conținutul şi după întreruperea tensiunii de alimentare.

O memorie ROM are *k* intrări şi *n* ieşiri. Intrările furnizează adresa pentru memorie, iar ieşirile furnizează conţinutul cuvântului selectat de adresa de la intrare. Numărul cuvintelor dintr-o memorie ROM este determinat de faptul că prin *k* linii de adresă se pot selecta 2^k cuvinte. Memoria ROM nu are date de intrare, deoarece nu este posibilă operaţia de scriere. Memoriile ROM integrate dispun de una sau mai multe intrări de validare şi ieşiri cu trei stări pentru a permite realizarea unor memorii de dimensiuni mai mari.

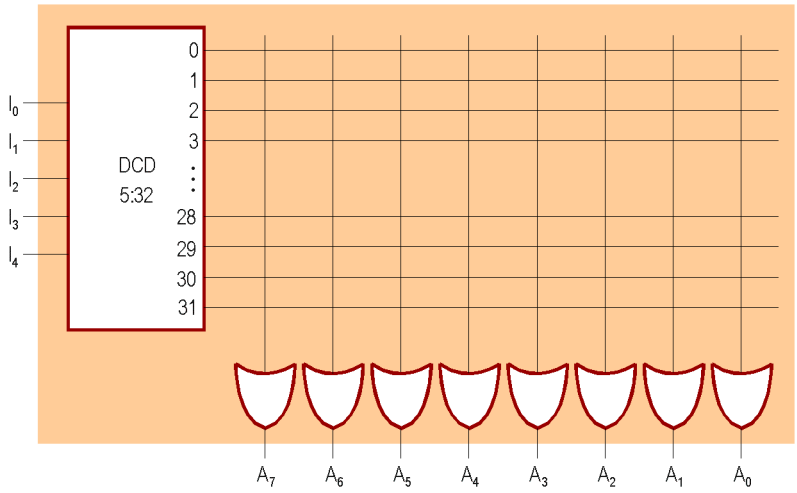

Figura 3.37. Logica internă a unei memorii ROM cu capacitatea de 32 × 8.

Considerăm, de exemplu, o memorie ROM cu capacitatea de 32×8, care conţine 32 de cuvinte de 8 biţi fiecare. Există 5 linii de intrare, care permit specificarea adreselor cuprinse între 0 şi 31. Structura internă a acestei memorii este indicată în Figura 3.37. Intrările sunt decodificate în 32 ieşiri distincte cu ajutorul unui decodificator 5:32. Fiecare ieşire a decodificatorului reprezintă o adresă de memorie. Cele 32 de ieşiri sunt conectate prin intermediul unor conexiuni programabile la fiecare din cele opt porţi SAU. Fiecare poartă SAU trebuie considerată ca având 32 de intrări. Pentru aceste porţi s-a utilizat reprezentarea simplificată. Fiecare ieşire a decodificatorului este conectată prin intermediul unui fuzibil la una din intrările fiecărei porţi SAU. Deoarece fiecare poartă SAU are 32 de conexiuni interne programabile, şi deoarece există opt porţi SAU, această memorie ROM conţine 32×8 = 256 de conexiuni programabile. În general, o memorie ROM cu capacitatea de $2^k \times n$ are un decodificator intern k:2^k şi n porţi SAU. Fiecare poartă SAU are 2^k intrări, care sunt conectate prin conexiuni programabile la fiecare din ieşirile decodificatorului.

Conţinutul locaţiilor unei memorii ROM poate fi specificat printr-o tabelă de adevăr. De exemplu, conţinutul unei memorii ROM cu dimensiunea de 32×8 poate fi specificat printr-o tabelă de adevăr similară cu cea din Tabelul 3.13. Fiecare combinaţie a intrărilor din tabel specifică adresa unui cuvânt de 8 biţi, a cărei valoare este indicată în coloanele ieşirilor. Tabelul 3.13 prezintă numai primele patru şi ultimele patru cu-

vinte ale memoriei ROM; tabelul complet trebuie să conțină lista tuturor celor 32 de cuvinte.

Tabelul 3.13. Tabelul de adevăr al unei memorii ROM (parțial).

Intrări	Ieşiri
I_4 I_3 I_2 I_1 I_0	A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0
0 0 0 0 0	1 0 1 1 0 1 1 0
0 0 0 0 1	0 0 0 1 1 1 0 1
0 0 0 1 0	1 1 0 0 0 1 0 1
0 0 0 1 1	1 0 1 1 0 0 1 0
⋮	⋮
1 1 1 0 0	0 0 0 0 1 0 0 1
1 1 1 0 1	1 1 1 0 0 0 1 0
1 1 1 1 0	0 1 0 0 1 0 1 0
1 1 1 1 1	0 0 1 1 0 0 1 1

Programarea memoriei ROM conform tabelului de adevăr de mai sus conduce la configurația din Figura 3.38. Fiecare valoare 0 din tabelul de adevăr specifică un circuit deschis, şi fiecare valoare 1 specifică un circuit închis. De exemplu, tabelul specifică un cuvânt cu valoarea 11000101 la adresa 00010. Cei patru biți de 0 din cadrul cuvântului sunt programați prin deschiderea conexiunilor dintre ieşirea 2 a decodificatorului şi intrările porților SAU asociate cu ieşirile A_5, A_4, A_3 şi A_1. Cei patru biți de 1 din cadrul cuvântului sunt marcați cu un $\times$ în schemă, pentru a indica un circuit închis. Atunci când intrarea memoriei ROM este 00010, ieşirea 2 a decodificatorului va fi la 1 logic, celelalte ieşiri fiind la 0 logic. Semnalul cu nivelul 1 logic de la ieşirea 2 a decodificatorului se propagă prin circuitele închise şi porțile SAU la ieşirile A_7, A_6, A_2 şi A_0, iar celelalte ieşiri rămân la 0 logic. Rezultatul este că la ieşirile de date se aplică cuvântul cu valoarea 11000101.

După cum s-a arătat anterior, un decodificator generează toți mintermii variabilelor de intrare. Prin inserarea unor porți SAU pentru însumarea mintermilor funcțiilor booleene, se poate implementa orice circuit combinațional. O memorie ROM conține atât un decodificator, cât şi porți SAU în cadrul aceluiaşi circuit. Prin închiderea conexiunilor pentru mintermii incluşi în cadrul funcției, ieşirile memoriei ROM pot fi programate pentru a reprezenta funcțiile booleene ale variabilelor de intrare dintr-un circuit combinațional.

Din punctul de vedere al unui circuit care implementează o funcție booleană, fiecare terminal de ieşire al unei memorii ROM este considerat separat ca ieşire a funcției booleene exprimate ca o sumă de mintermi. De exemplu, memoria ROM din Figura 3.37 poate fi considerată ca un circuit combinațional cu opt ieşiri, fiecare fiind o funcție a celor cinci variabile de intrare. Ieşirea A_7 poate fi exprimată ca o sumă a mintermilor, în felul următor (punctele reprezintă mintermii de la 4 la 27, care nu apar în figură):

$$A_7 (I_4, I_3, I_2, I_1, I_0) = \Sigma (0, 2, 3, \ldots, 29)$$

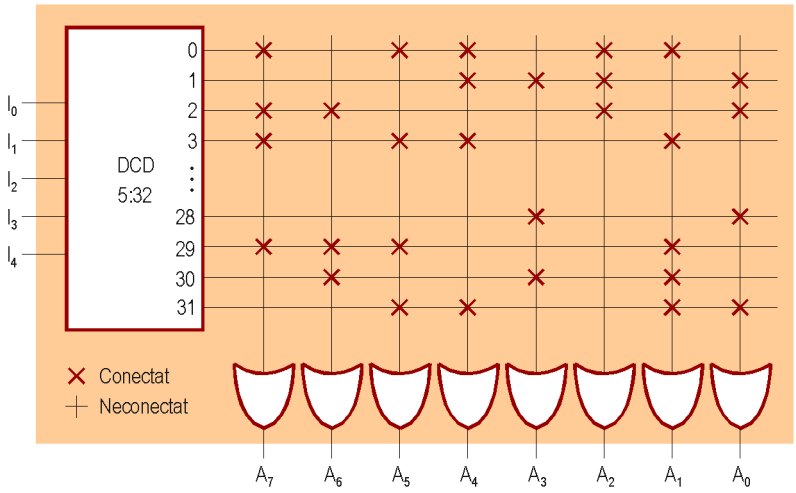

Figura 3.38. Programarea memoriei ROM conform Tabelului 3.13.

Deoarece o memorie ROM păstrează de fapt întregul tabel de adevăr al funcțiilor pe care le generează, o asemenea memorie poate fi utilizată pentru implementarea circuitelor combinaționale complexe direct din tabelul de adevăr. Procesul de citire a informațiilor memorate într-o memorie ROM este similar cu o căutare într-o tabelă (*table lookup*). Memoriile ROM sunt utile pentru implementarea unor circuite ale căror funcții sunt dificil de specificat prin ecuații logice, ca de exemplu convertoare de cod sau circuite pentru operații aritmetice complexe (înmulțire, împărțire), şi pentru aplicații care necesită un număr moderat de intrări şi un număr mare de ieşiri. Utilitatea memoriilor ROM este limitată de faptul că dimensiunea lor trebuie dublată cu fiecare nouă variabilă de intrare.

În practică, la implementarea unui circuit combinațional printr-o memorie ROM, nu este necesar să se deseneze schema logică sau să se indice conexiunile interne din cadrul memoriei. Trebuie să se specifice doar o anumită memorie ROM prin numărul circuitului şi tabelul de adevăr al memoriei. Tabelul de adevăr conține toate informațiile necesare programării memoriei.

De exemplu, se consideră proiectarea unui circuit combinațional care acceptă la intrare un număr de 3 biți şi generează la ieşire un număr binar egal cu pătratul numărului de la intrare. Prima etapă pentru proiectarea circuitului este întocmirea tabelului de adevăr. În cele mai multe cazuri, aceasta este singura operație necesară. În alte cazuri, se poate utiliza un tabel de adevăr parțial pentru memoria ROM ținând cont de anumite proprietăți ale variabilelor de ieşire. Tabelul 3.14 reprezintă tabelul de adevăr pentru circuitul proiectat.

Tabelul 3.14. Tabelul de adevăr pentru circuitul care generează pătratele valorilor de la intrare.

Intrări	Ieşiri	
A_2 A_1 A_0	B_5 B_4 B_3 B_2 B_1 B_0	Zecimal
0 0 0	0 0 0 0 0 0	0
0 0 1	0 0 0 0 0 1	1
0 1 0	0 0 0 1 0 0	4
0 1 1	0 0 1 0 0 1	9
1 0 0	0 1 0 0 0 0	16
1 0 1	0 1 1 0 0 1	25
1 1 0	1 0 0 1 0 0	36
1 1 1	1 1 0 0 0 1	49

Pentru circuitul proiectat, sunt necesare trei intrări şi şase ieşiri. Se observă că ieşirea B_0 este egală întotdeauna cu intrarea A_0, astfel încât nu este necesară generarea acestei ieşiri cu ajutorul memoriei ROM. De asemenea, ieşirea B_1 este întotdeauna 0. Astfel, trebuie să se genereze doar patru ieşiri cu memoria ROM. Memoria trebuie să aibă minimum trei intrări şi patru ieşiri. Cele trei intrări specifică opt cuvinte, deci dimensiunea minimă a memoriei necesare este de 8×4. Schema circuitului este prezentată în Figura 3.39.

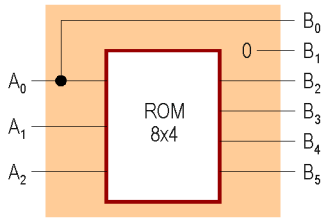

Figura 3.39. Implementarea cu o memorie ROM a circuitului care generează pătratele valorilor de la intrare.

3.5.6.2. Rețele logice programabile

O rețea logică programabilă PLA (*Programmable Logic Array*) este similară ca şi concept cu o memorie ROM, cu excepția faptului că nu realizează decodificarea completă a variabilelor şi nu generează toți mintermii. Decodificatorul este înlocuit cu o rețea de porți ŞI care poate fi programată pentru a genera termenii produs ai variabilelor de intrare. Termenii produs sunt apoi conectați în mod selectiv cu porți SAU pentru a genera suma termenilor produs pentru funcțiile booleene necesare. Structura de bază a unui circuit PLA este prezentată în Figura 3.40.

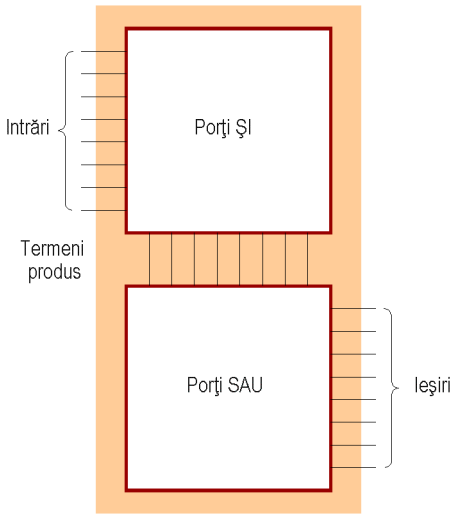

Figura 3.40. Structura generală a unui circuit PLA.

Un circuit PLA poate implementa în mod direct un set de funcţii logice exprimate printr-un tabel de adevăr. Fiecare intrare pentru care valoarea funcţiei este adevărată necesită un termen produs, şi acestuia îi corespunde o linie de porţi ŞI din primul etaj al circuitului PLA. Fiecare ieşire corespunde la o linie de porţi SAU din al doilea etaj al circuitului. Numărul de porţi SAU corespunde cu numărul de intrări din tabela de adevăr pentru care ieşirea este adevărată. Dimensiunea totală a circuitului PLA este egală cu suma dintre dimensiunea reţelei de porţi ŞI şi dimensiunea reţelei de porţi SAU. Din Figura 3.39 se observă că dimensiunea reţelei de porţi ŞI este egală cu numărul de intrări multiplicat cu numărul diferiţilor termeni produs, iar dimensiunea reţelei de porţi SAU este egală cu numărul de ieşiri multiplicat cu numărul termenilor produs.

Un circuit PLA are două caracteristici care determină o implementare eficientă a unui set de funcţii logice. Prima este că singurele intrări din tabelul de adevăr care necesită porţi logice sunt cele cărora le corespunde o valoare adevărată pentru cel puţin o ieşire. A doua este că un anumit termen produs va avea o singură intrare în circuitul PLA, chiar dacă termenul produs este utilizat în mai multe ieşiri.

Considerăm implementarea următoarelor funcţii booleene într-un circuit PLA:

$$\begin{aligned} F_1 &= A\overline{B} + AC + \overline{A}B\overline{C} \\ F_2 &= \overline{AC + BC} \end{aligned} \tag{3.25}$$

Structura internă a circuitului PLA care implementează aceste funcţii este prezentată în Figura 3.41. Fiecare intrare trece printr-un buffer şi un inversor. Există conexiuni programabile de la fiecare intrare şi complementul acesteia la intrările fiecărei porţi ŞI, indicate prin intersecţiile dintre liniile verticale şi orizontale. Ieşirile porţilor ŞI au conexiuni programabile la intrările fiecărei porţi SAU. Ieşirea fiecărei porţi SAU constituie intrare într-o poartă SAU EXCLUSIV, iar cealaltă intrare poate fi programată fie la

1 logic, fie la 0 logic. Ieşirea este inversată dacă această intrare este 1 logic (deoarece $X \oplus 1 = \overline{X}$), şi rămâne neschimbată dacă această intrare este 0 logic (deoarece $X \oplus 0 = X$).

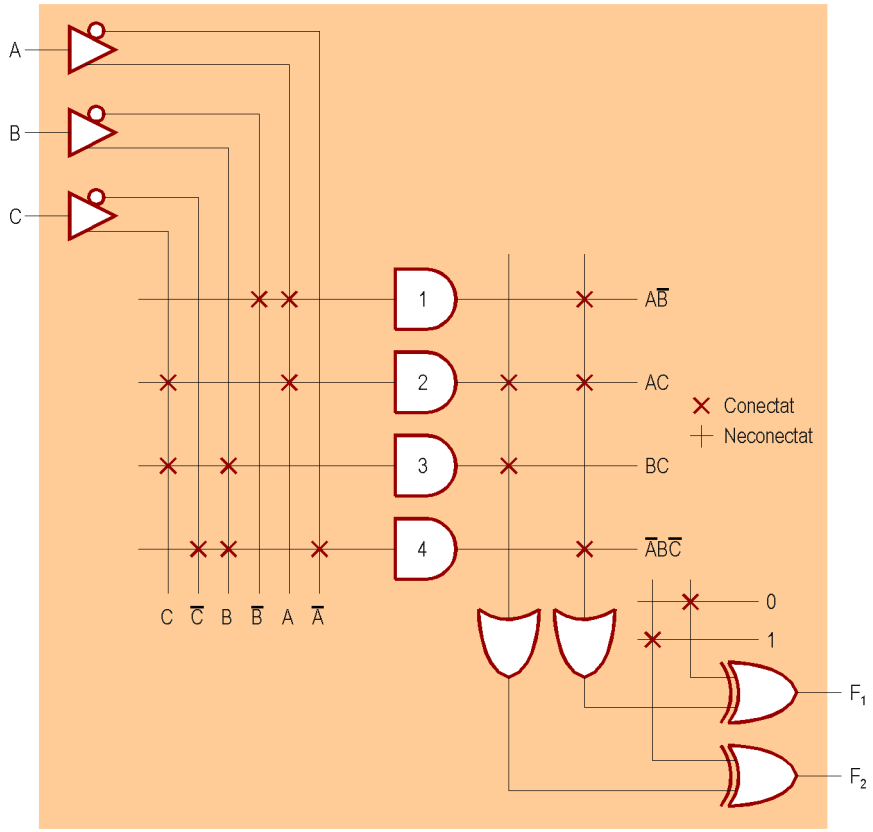

Figura 3.41. Structura internă a unui circuit PLA cu trei intrări, patru termeni produs şi două ieşiri.

Modul de programare a circuitului PLA se poate specifica sub formă tabelară. De exemplu, pentru circuitul din Figura 3.41 programarea este specificată în Tabelul 3.15. Tabelul constă din trei secţiuni. Prima secţiune listează numerele termenilor produs. A doua secţiune specifică conexiunile necesare dintre intrări şi porţile ŞI. A treia secţiune specifică conexiunile dintre porţile ŞI şi porţile SAU. Fiecare variabilă de ieşire poate fi adevărată (A) sau complementată (C), ceea ce se controlează prin poarta SAU EXCLUSIV. Pentru fiecare termen produs, intrările sunt marcate cu 1, 0 sau –. Dacă o variabilă dintr-un termen produs apare sub formă necomplementată, variabila este marcată cu 1. Dacă o variabilă dintr-un termen produs apare sub formă complementată, ea este marcată cu 0. Dacă variabila lipseşte din termenul produs, ea este marcată cu –.

Variabilele de ieşire sunt marcate cu 1 pentru termenii produs care sunt incluşi în cadrul funcției. O ieşire marcată cu A indică faptul că cealaltă intrare a porții SAU EXCLUSIV corespunzătoare trebuie conectată la 0, iar o ieşire marcată cu C indică o conectare la 1.

Tabelul 3.15. Tabelul de programare pentru circuitul PLA din Figura 3.41.

	Termen produs	Intrări	Ieşiri
		A B C	(A) (C) F_1 F_2
$A\bar{B}$	1	1 0 –	1 –
AC	2	1 – 1	1 1
BC	3	– 1 1	– 1
$\bar{A}\bar{B}C$	4	0 1 0	1 –

Dimensiunea unui circuit PLA este specificată prin numărul intrărilor, numărul termenilor produs şi numărul ieşirilor. Un circuit PLA tipic are 16 intrări, 48 de termeni produs şi 8 ieşiri. Pentru *n* intrări, *k* termeni produs şi *m* ieşiri, logica internă a circuitului PLA constă din *n* buffere-inversoare, *k* porți ŞI, *m* porți SAU, şi *m* porți SAU EXCLUSIV. Există $2n \times k$ conexiuni programabile între intrări şi porțile ŞI, $k \times m$ conexiuni programabile între porțile ŞI şi porțile SAU, şi *m* conexiuni programabile asociate cu porțile SAU EXCLUSIV.

Pentru proiectarea unui sistem digital cu un circuit PLA, nu este necesar să se indice conexiunile interne ale circuitului, ci trebuie să se specifice doar tabela de programare. Ca şi în cazul memoriilor ROM, circuitele PLA pot fi programate prin măşti sau programate de către utilizator. Circuitele PLA programate de către utilizator se numesc FPLA (*Field Programmable Logic Array*).

La implementarea funcțiilor logice cu ajutorul circuitelor PLA, trebuie să se reducă numărul termenilor produs în scopul reducerii complexității circuitului. Pentru aceasta trebuie simplificate funcțiile booleene astfel încât acestea să aibă un număr minim de termeni. Numărul de literale dintr-un termen este mai puțin important, deoarece toate variabilele de intrare sunt disponibile. Trebuie simplificată atât forma necomplementată, cât şi cea complementată a fiecărei funcții pentru a se determina care din acestea se poate exprima cu ajutorul unui număr mai mic de termeni produs şi care utilizează termeni produs care sunt utilizați şi de alte funcții.

Există şi circuite PLA care conțin bistabile ataşate prin conexiuni programabile la ieşirile rețelei de porți SAU, ceea ce permite implementarea unor circuite secvențiale de dimensiuni medii. O altă categorie de rețele logice programabile sunt circuitele PAL (*Programmable Array Logic*), care conțin o rețea de porți ŞI programabilă, dar rețeaua de porți SAU are conexiuni fixe. Fiecare linie de ieşire este conectată la un set fix de linii ale rețelei de porți ŞI, în mod tipic de opt linii. O asemenea ieşire a circuitului PAL poate implementa o expresie pe două nivele conținând cel mult opt termeni. Avantajele circuitelor PAL sunt simplitatea utilizării în anumite aplicații, ca şi viteza mai ridicată, deoarece factorul de încărcare la ieşire este limitat.

3.5.6.3. Circuite FPGA

Circuitele FPGA (*Field Programmable Gate Array*) au fost introduse în anul 1985 de compania Xilinx. De atunci au fost elaborate diferite tipuri de circuite FPGA de un număr de alte companii ca Actel, Altera, Atmel, Texas Instruments etc. Un circuit FPGA constă dintr-o rețea bidimensională de *celule* sau *blocuri logice*. De obicei, fiecare bloc logic poate fi programat pentru a implementa orice funcție logică a intrărilor sale. De aceea, aceste blocuri sunt numite de obicei blocuri logice configurabile (*Configurable Logic Block* - CLB). Cele mai multe blocuri logice conțin de asemenea unul sau două bistabile. Canalele şi blocurile de comutare dintre aceste blocuri conțin resurse de interconectare, după cum se ilustrează în Figura 3.42. Aceste resurse conțin de obicei segmente de interconectare de diferite lungimi. Interconexiunile conțin comutatoare programabile cu rolul de a conecta blocurile logice la segmentele de interconectare, sau un segment de interconectare la altul. În plus, există celule de I/E la periferia rețelei, care pot fi programate ca intrări sau ieşiri.

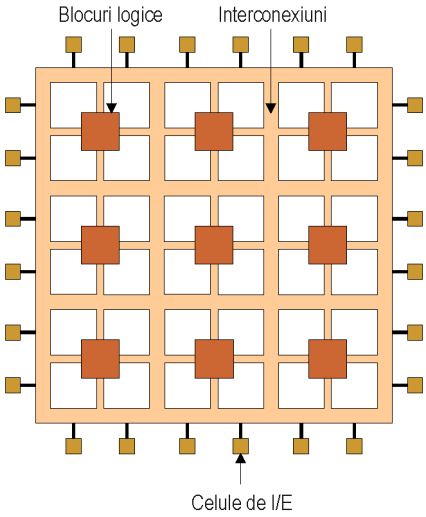

Figura 3.42. Structura unui circuit FPGA tipic.

Din punctul de vedere al tipului conexiunilor programabile, există două categorii principale de circuite FPGA: circuite cu memorii SRAM şi circuite cu antifuzibile.

Circuite cu memorii SRAM. Programarea acestor circuite se realizează prin celule de memorie statică. Logica este implementată cu ajutorul unor tabele (*lookup table*) realizate din celulele de memorie, intrările funcțiilor controlând liniile de adresă. Fiecare tabelă de 2^n celule de memorie implementează orice funcție cu n intrări. Una sau mai multe tabele, combinate cu bistabile, formează un bloc logic configurabil. Aceste blocuri sunt aranjate într-un tablou bidimensional, segmentele de interconectare formând canale, similar cu rețelele de porți. Segmentele se conectează la pinii blocurile

logice din canale şi la alte segmente din blocurile de comutare prin intermediul tranzistoarelor de trecere controlate de celule ale memoriei de configurare.

Un program de configurare pentru circuitele cu memorii SRAM constă dintr-un singur cuvânt lung de programare. Logica din circuit încarcă cuvântul de programare, pe care îl citeşte serial dintr-o memorie externă de fiecare dată când circuitul este alimentat. Biţii acestui cuvânt setează valorile tuturor celulelor memoriei de configurare din circuit, setând astfel valorile tabelelor şi selectând segmentele care se vor conecta între ele. Circuitele cu memorii SRAM sunt reprogramabile. Ele pot fi actualizate în sistem, punând la dispoziţia proiectanţilor noi opţiuni şi posibilităţi de proiectare.

Din această categorie de circuite FPGA fac parte cele ale firmelor Xilinx, Altera, AT&T.

Circuite cu antifuzibile. Un antifuzibil este un dispozitiv cu două terminale care în mod normal se află în starea de înaltă impedanţă, iar atunci când este expus la o tensiune ridicată, trece în starea cu rezistenţă redusă (300-500 Ω). Antifuzibilele au dimensiuni reduse, astfel încât o arhitectură bazată pe antifuzibile poate conţine sute de mii sau milioane de antifuzibile. Pentru simplificarea arhitecturii şi a programării, circuitele FPGA bazate pe antifuzibile constau de obicei din rânduri de elemente logice configurabile cu canale de interconectare între ele, ca şi reţelele de porţi tradiţionale. Un bloc logic poate fi programat prin conectarea pinilor săi de intrare la valori fixe sau la reţele de interconectare. Există antifuzibile la fiecare punct de intersecţie între interconexiuni şi pini din canal şi la toate punctele de intersecţie între interconexiuni în locurile în care canalele se intersectează.

Din categoria circuitelor FPGA cu antifuzibile fac parte circuitele firmelor Actel, Quicklogic, Cypress.

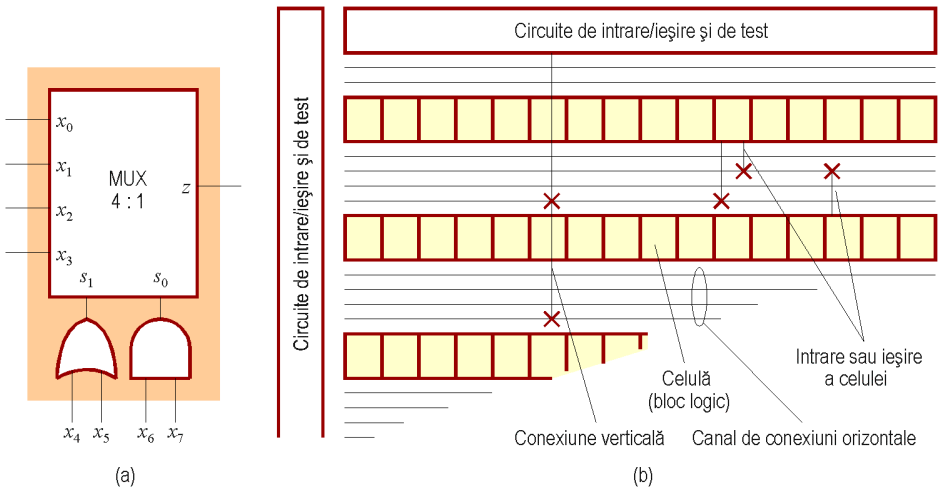

Figura 3.43. Circuit FPGA Actel din seria ACT: (*a*) celula de bază; (*b*) arhitectura circuitului.

Există două tipuri de celule logice ale circuitelor FPGA: celule bazate pe multiplexoare şi celule bazate pe memorii PROM. Figura 3.43(a) prezintă un tip de ce-

lulă (numită modul C) bazată pe multiplexoare utilizată de circuitele FPGA din seria ACT ale firmei Actel. Această celulă conține un multiplexor 4:1, la care se adaugă o poartă SAU şi o poartă ŞI. O variantă de celulă numită modul S conține un bistabil D conectat la ieşire; există de asemenea celule speciale ataşate la pinii de I/E ai circuitului FPGA. Un circuit FPGA din seria ACT conține o rețea de dimensiuni mari de asemenea celule organizate în rânduri separate prin canale de rutare orizontale, după cum se ilustrează în Figura 3.43(b). Există segmente de conexiuni verticale ataşate la terminalele de I/E ale fiecărei celule. Aceste segmente permit stabilirea conexiunilor între celule şi canalele de rutare prin intermediul antifuzibilelor poziționate la intersecția liniilor orizontale şi verticale. În plus, există linii lungi de conexiuni verticale pentru semnalele de I/E primare, de alimentare şi masă.

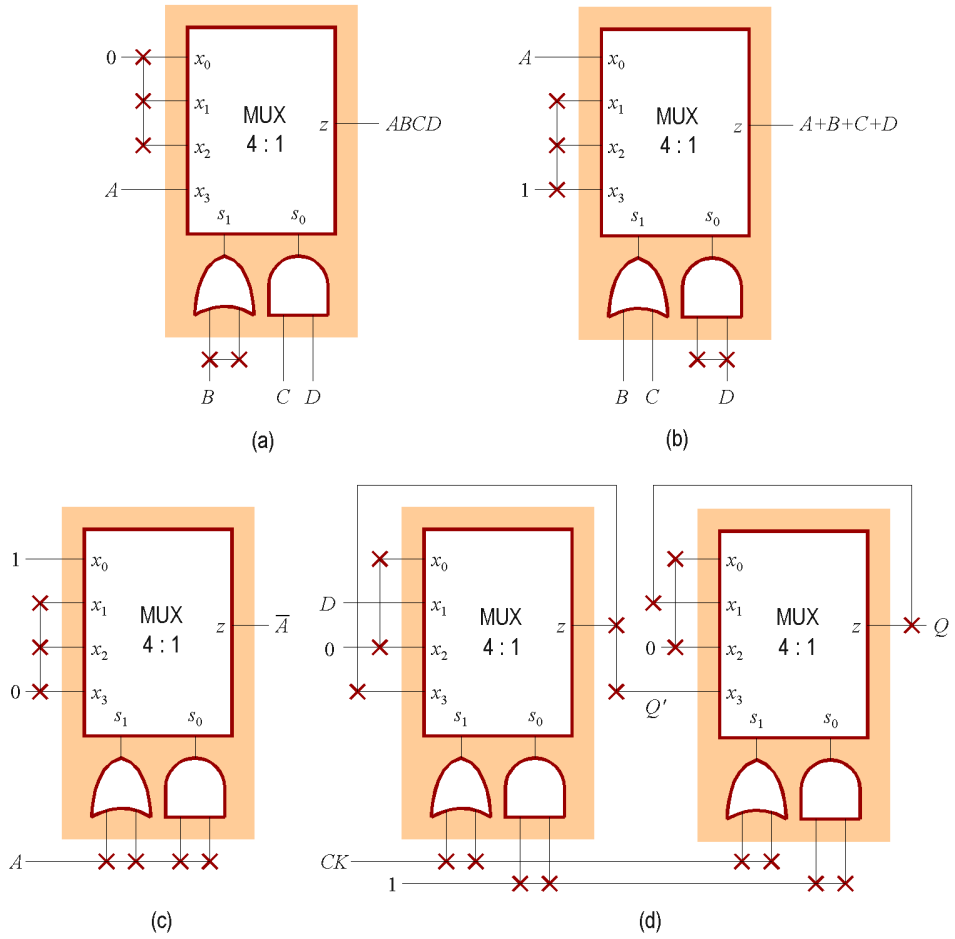

Figura 3.44. Celula FPGA din Figura 3.43(*a*) programată pentru a realiza: (*a*) o poartă ŞI cu patru intrări; (*b*).o poartă SAU cu patru intrări; (*c*) un inversor; (*d*) un bistabil D.

Celula FPGA din Figura 3.43(a) poate genera orice funcție booleană de până la trei variabile dacă intrările sunt furnizate atât sub formă complementată, cât şi necomplementată. Această celulă poate genera de asemenea diferite funcții utile de mai mult de trei variabile datorită prezenței celor două porți suplimentare. Figurile 3.44(a), 3.44(b) şi 3.44(c) ilustrează modul în care această celulă implementează un set complet de porți logice. Se observă modul în care porțile ŞI, respectiv SAU, permit realizarea funcțiilor ŞI, respectiv SAU cu patru intrări. Figura 3.44(d) ilustrează modul în care aceeaşi celulă combinațională implementează un bistabil D comutat pe front.

Avantajele circuitelor FPGA sunt costurile de prototipizare reduse şi durata scurtă de producție. Principalele dezavantaje sunt viteza de operare mai redusă şi densitatea mai redusă a porților. Comutatoarele programabile şi circuitele de programare asociate necesită un spațiu mai mare în cadrul circuitului comparativ cu conexiunile metalice din rețelele de porți. Aceste comutatoare au de asemenea o rezistență şi capacitate semnificativă care determină o viteză mai redusă de operare.

Circuitele FPGA sunt potrivite pentru proiectarea şi fabricația asistată de calculator, deoarece procesul de proiectare poate fi aproape în întregime automatizat. Acest proces necesită translatarea sau compilarea specificațiilor proiectului (sub forma schemelor logice sau a unei descrieri într-un limbaj de descriere hardware) într-un model la nivelul porților logice. Se utilizează apoi programe CAD de plasare şi rutare pentru asignarea elementelor logice la celule (plasare), configurarea celulelor pentru o anumită funcție şi pentru interconectarea celulelor (rutare). În final, circuitul FPGA este programat prin transmiterea şirului de configurare.

3.6. Circuite logice secvențiale

3.6.1. Prezentare generală a circuitelor secvențiale

Circuitele combinaționale implementează funcțiile esențiale ale unui calculator numeric. Aceste circuite se caracterizează prin faptul că starea ieşirilor depinde numai de starea intrărilor, şi nu depinde de timp. Deci, cu excepția memoriilor ROM, circuitele combinaționale nu furnizează informații de memorie sau de stare, care sunt de asemenea elemente esențiale pentru funcționarea unui calculator numeric. În acest scop se utilizează circuitele secvențiale. Un circuit secvențial are memorie, adică ieşirile curente ale circuitului nu depind numai de intrările curente, ci şi de intrările anterioare. Un alt mod de caracterizare a unui circuit secvențial este că ieşirile curente ale circuitului depind de intrările curente şi de starea curentă a circuitului.

Structura generală a unui circuit secvențial este prezentată în Figura 3.45.

Circuitul secvențial se compune dintr-un circuit combinațional, o parte a ieşirilor acestuia fiind conectate la intrările circuitului prin intermediul unor elemente de memorie (elemente de întârziere) Δ_1, …, Δ_p. Semnalele aplicate pe cele n intrări formează mulțimea $X = \{x_1, x_2, \ldots, x_n\}$ a variabilelor de intrare, numite şi *variabile de intrare principale*. Mulțimea formată din 2^n intrări distincte se numeşte *alfabet de intrare* I, $I = \{i_1, i_2, ..., i_{2^n}\}$. De exemplu, pentru două variabile de intrare alfabetul de intrare este $I = \{00, 01, 10, 11\}$. O combinație a intrărilor se numeşte simbol al alfabetului.