

2 SISTEME COMBINAȚIONALE

Aplicațiile din acest capitol își propun să prezinte metodele de analiză și sinteză folosite la implementarea funcțiilor binare cu porți logice, dar și cu unele circuite integrate pe scară medie (MSI), cum ar fi multiplexoarele și demultiplexoarele.

2.1 Considerații teoretice

2.1.1 Implementarea cu porți logice

Structurile numerice combinaționale sau circuitele logice combinaționale (CLC) implementează funcții binare. O funcție binară de n variabile binare independente este o aplicație $f: \{0,1\}^n \rightarrow \{0,1\}$, unde domeniul de definiție este mulțimea $\{0,1\}^n = \{(x_1 x_2 \dots x_n) \mid x_1 \in \{0,1\}, x_2 \in \{0,1\}, \dots, x_n \in \{0,1\}\}$.

Numărul maxim de elemente ale mulțimii $\{0,1\}^n$ este 2^n , dar nu este absolut necesar ca funcția să fie definită pentru toate aceste combinații de variabile. Numărul maxim de funcții binare de n variabile este:

$$N = \sum_{i=0}^{2^n} C_{2^n}^i = 2^{2^n}$$

Analiza unui circuit combinațional are ca scop determinarea funcției logice intrare-ieșire. Aceasta se poate exprima fie în limbaj natural, fie algebric prin expresii logice, fie prin tabele de adevăr, fie prin diagrame Veitch-Karnaugh.

Sinteza unui circuit combinațional urmărește determinarea schemei logice, pornind de la unul dintre modurile de reprezentare enumerate mai sus.

Orice funcție binară poate fi implementată numai cu porți ȘI-NU. Se spune că operatorul logic ȘI-NU formează un set complet de operatori. Panoul logic conține 3 tipuri de circuite integrate realizate în tehnologie TTL standard: porți inversoare, porți ȘI-NU cu 2 intrări și porți ȘI-NU cu 3 intrări.

Pentru a găsi circuitul optim care implementează funcția căutată este necesară operația de minimizare a funcției binare, în urma căreia rezultă o expresie algebrică cât mai simplă, care permite construcția circuitului folosind un număr minim de porți.

Există numeroase tehnici de minimizare: pentru calculul manual se preferă utilizarea **diagramelor Veitch-Karnaugh**, iar pentru calculul automat metode tabelare cum ar fi metoda Quine-McCluskey sau metoda Espresso.

Vom prezenta în continuare un exemplu de minimizare a funcției binare f , reprezentată prin tabelul de adevăr din figura 2.1, folosind diagramele Veitch-Karnaugh:

- funcția dată are 4 variabile, deci diagrama Veitch-Karnaugh are 2^4 compartimente, fiecare corespunzând uneia dintre cele 16 valori ale funcției. Fiecare compartiment este "adresat" de o anumită combinație a variabilelor A, B, C și D. După completarea compartimentelor cu valorile funcției din tabelul de adevăr, se grupează compartimentele vecine (diferă valoarea unei singure variabile) care conțin 1 logic, astfel încât fiecare grupare să conțină un număr de 2^i compartimente, unde $i = 0,1,2,3$ sau 4. Fiecare grupare este definită de un produs de variabile, sub formă directă sau negată, iar expresia funcției se obține prin aplicarea funcției SAU acestor "produse logice".

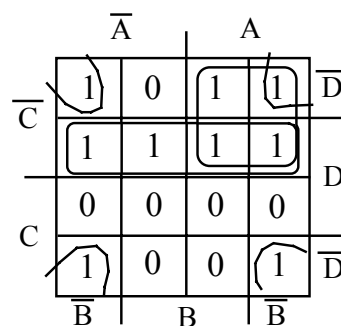
Pentru exemplul considerat:

$$f(A, B, C, D) = \bar{B} \cdot \bar{D} + \bar{C} \cdot D + A \cdot \bar{C} = \overline{\bar{B} \cdot \bar{D} \cdot \bar{C} \cdot D \cdot A \cdot \bar{C}}$$

S-a folosit o teoremă fundamentală a algebrei boolene cunoscută sub numele de "Legile lui DeMorgan" care permite transformarea operațiilor binare SAU în ȘI și invers. Ea permite implementarea funcțiilor binare numai cu porți ȘI-NU:

$$\overline{X + Y} = \bar{X} \cdot \bar{Y}, \text{ sau relația duală: } \overline{X \cdot Y} = \bar{X} + \bar{Y}$$

A	B	C	D	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0



Veitch

CD \ AB	00	01	11	10
00	1	0	1	1
01	1	1	1	1
11	0	0	0	0
10	1	0	0	1

Karnaugh

Fig. 2.1 Exemplu de folosire a diagramelor Veitch-Karnaugh

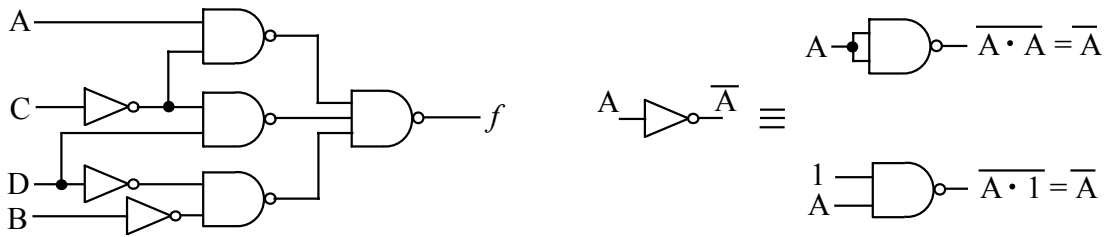


Fig. 2.2 Schema logică cu număr minim de porți ȘI-NU

Inversoarele se pot obține din porți ȘI-NU prin conectarea împreună a intrărilor (legile de idempotență) sau prin conectarea celorlate intrări la 1 logic (elementul neutru pentru operația binară ȘI).

2.1.2 Implementarea cu multiplexoare/demultiplexoare

Funcțiile binare pot fi implementate cu ajutorul unor structuri combinaționale MSI, fără a mai fi necesară minimizarea lor.

Una dintre aceste structuri este **decodificatorul**(DCD). Decodificatorul este un circuit integrat pe scară medie, care identifică un cod de intrare prin activarea unei singure linii de ieșire. Dacă circuitul are n variabile binare de intrare, atunci numărul liniilor de ieșire este 2^n . Figura 2.3 arată structura circuitului TTL pentru $n = 2$.

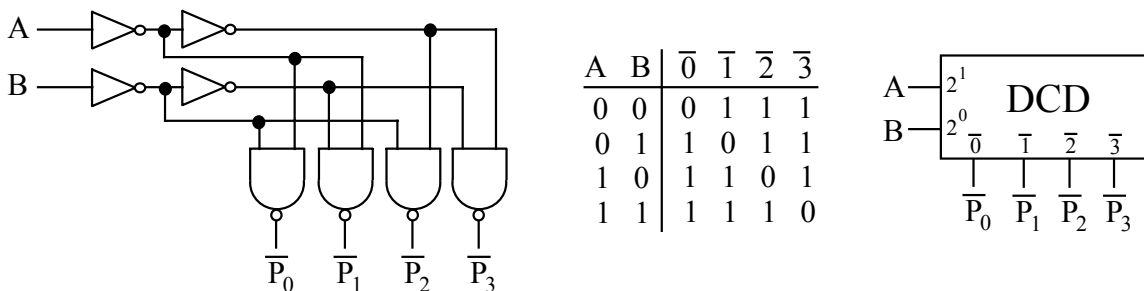


Fig. 2.3 Structura, tabelul de adevăr și reprezentarea decodificatorului TTL pentru $n=2$

Demultiplexorul(DMUX) este un circuit construit pe structura decodificatorului, care permite transmiterea datelor de pe o singură cale de intrare pe una din cele 2^n căi de ieșire. Selecția liniilor de ieșire se face prin aplicarea unui cod binar pe n linii de intrare, care devin acum intrări de selecție. Structura demultiplexorului TTL pentru $n = 2$ este prezentată în figura 2.4.

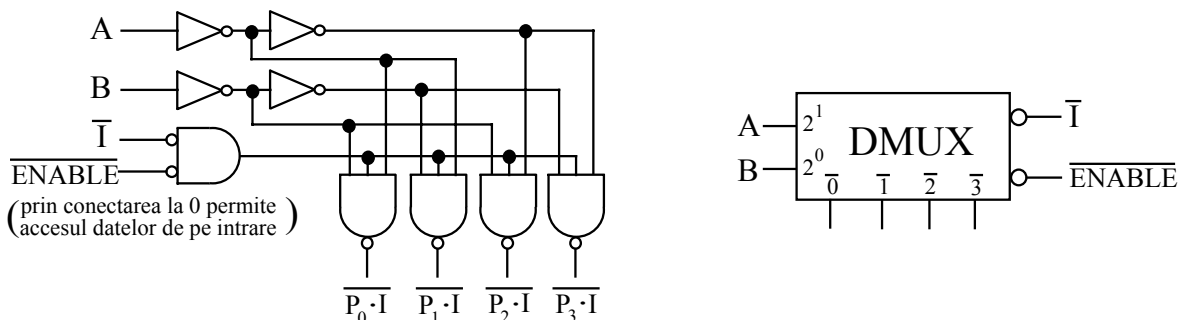


Fig. 2.4 Structura și reprezentarea demultiplexorului TTL pentru $n=2$

Multiplexorul (MUX) realizează funcția inversă demultiplexorului, adică permite transmiterea datelor de la una din cele 2^n căi de intrare la o singură cale de ieșire. Selecția unei anumite intrări de date se face prin aplicarea unui cod binar pe n linii de intrare, linii care sunt intrările de selecție. Structura multiplexorului TTL pentru $n = 2$ este prezentată în figura 2.5.

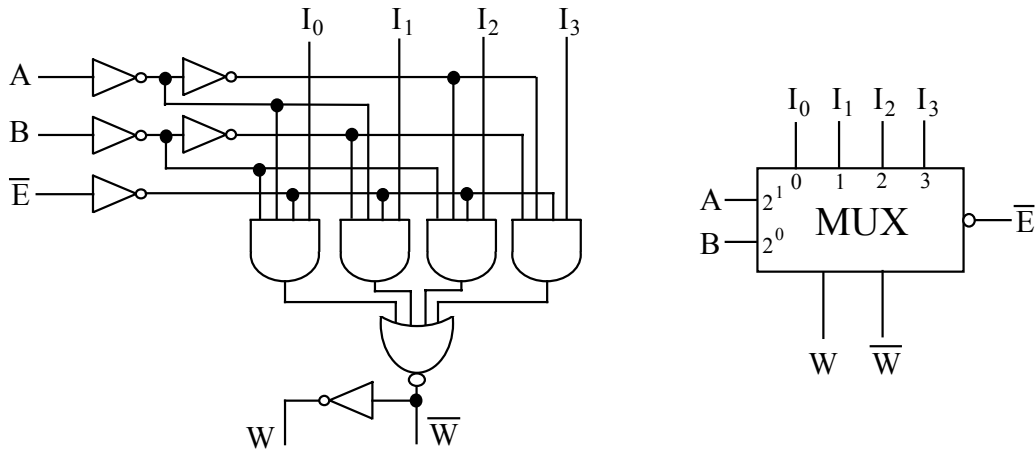


Fig. 2.5 Structura și reprezentarea multiplexorului TTL pentru $n=2$

În descrierile de mai sus, s-a precizat că aceste structuri corespund tehnologiei TTL. În tehnologia CMOS s-ar putea repeta aceste structuri, dar s-au descoperit alte structuri mai eficiente, bazate pe poarta de transmisie CMOS. Circuitele pot fi asimilate cu niște comutatoare programabile. Aceste comutatoare sunt bidirecționale, deci același circuit poate fi utilizat fie ca multiplexor, fie ca demultiplexor, iar semnalele transmise pot fi și analogice. Schema simplificată a unui MUX/DMUX cu 3 intrări de selecție în tehnologie CMOS, care selectează intrarea/ieșirea cu numărul 5, este dată în figura 2.6.

Aceste structuri combinaționale MSI implementează toți termenii produs fundamentali ai unei funcții cu număr de variabile mai mic sau egal cu numărul de intrări de selecție n . Din acest motiv, implementarea funcțiilor binare nu necesită operații de minimizare, ci numai alegerea corectă a conexiunilor. Dacă numărul de variabile ale funcției este mai mic sau egal cu numărul de intrări de selecție n , atunci implementarea se face direct, iar în caz contrar mai sunt necesare o serie de transformări algebrice pentru a găsi o altă structură adițională, formată de obicei din porți, care să completeze lipsa intrărilor de selecție de la MUX sau DMUX.

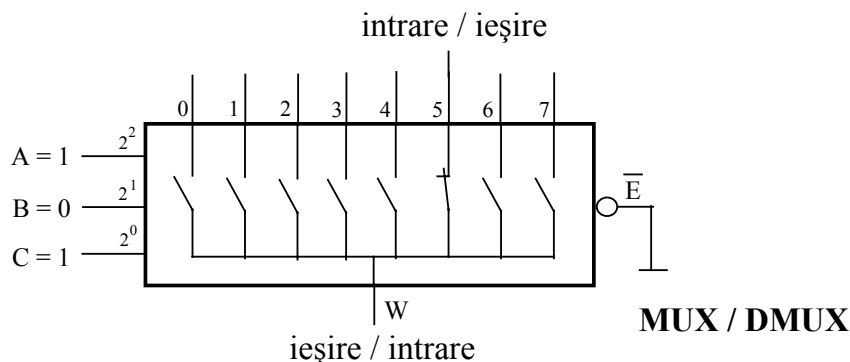


Fig. 2.6 Implementarea funcției cu demultiplexor și cu multiplexor CMOS

Dacă ne propunem să implementăm cu MUX și apoi cu DMUX funcția binară $f = P_0 + P_1 + P_2 + P_6 + P_7$, cele două soluții TTL sunt prezentate în figura 2.7.

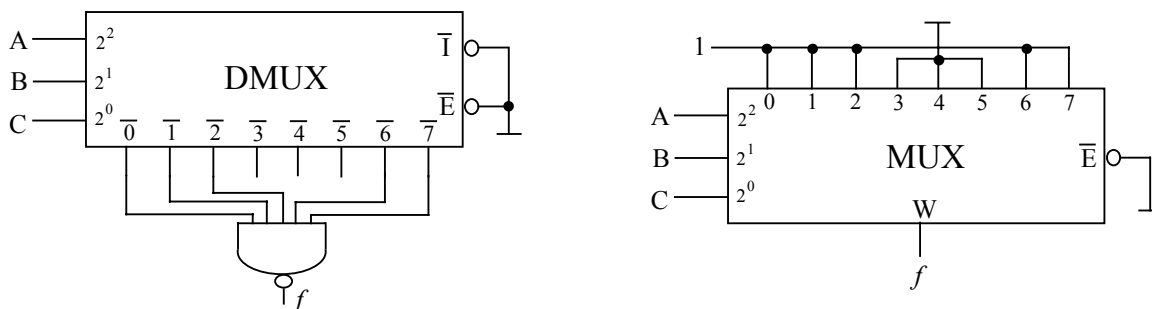


Fig. 2.7 Implementarea funcției cu demultiplexor și cu multiplexor

Implementarea cu MUX este imediată. Intrările corespunzătoare indicilor termenilor P_i din suma logică se conectează la 1 logic, iar celelalte intrări la 0 logic. Pentru implementarea cu DMUX ne interesează termenii \overline{P}_i , deci scriem funcția sub forma $f = \overline{P}_0 \cdot \overline{P}_1 \cdot \overline{P}_2 \cdot \overline{P}_6 \cdot \overline{P}_7$. Ieșirile corespunzătoare indicilor termenilor \overline{P}_i se conectează la intrările unei porți ȘI-NU.

Dacă demultiplexorul este o structură CMOS, atunci la fiecare dintre ieșirile $\overline{0}, \dots, \overline{7}$ se conectează câte o rezistență la V_+ . În caz contrar, ieșirile care nu sunt selectate rămân în aer (stare de înaltă impedanță).

2.2 Demonstrații practice

Considerațiile asupra alimentării panoului logic, formulate în capitolul anterior, rămân valabile pentru toate aplicațiile practice din această lucrare. Circuitele integrate MSI au deja o complexitate mai mare decât porțile logice și utilizarea lor la implementarea unor scheme electrice nu se mai poate face fără consultarea foii de catalog.

Panoul logic conține porți logice, două circuite integrate MSI realizate în tehnologie TTL standard și un circuit integrat în tehnologie CMOS – seria 4000. Circuitul **SN 74150** este un multiplexor TTL cu 4 intrări de selecție, deci cu 16 căi de intrare, iar circuitul **SN 74155** este un circuit TTL cu funcție dublă: fie un demultiplexor cu 3 intrări de selecție, fie două demultiplexoare cu câte 2 intrări de selecție. Circuitul **MMC 4097** este un circuit CMOS care conține două multiplexoare/demultiplexoare cu câte 3 intrări de selecție.

Foile de catalog furnizează informația completă pentru utilizarea corectă a acestor circuite integrate. Chiar dacă circuitele sunt fabricate de alt producător, ele trebuie să respecte standardul elaborat de firma **TEXAS INSTRUMENTS** pentru circuitele TTL, respectiv **RCA**, pentru circuitele CMOS. Fragmente din foile de catalog, care conțin numai informațiile strict necesare pentru utilizarea acestor circuite integrate sunt prezentate în figurile 2.8, 2.9 și respectiv 2.10. În cazul structurilor combinaționale, aceste informații strict necesare sunt cele care permit realizarea conexiunilor conform schemei logice, adică configurația pinilor și tabelul de adevăr.

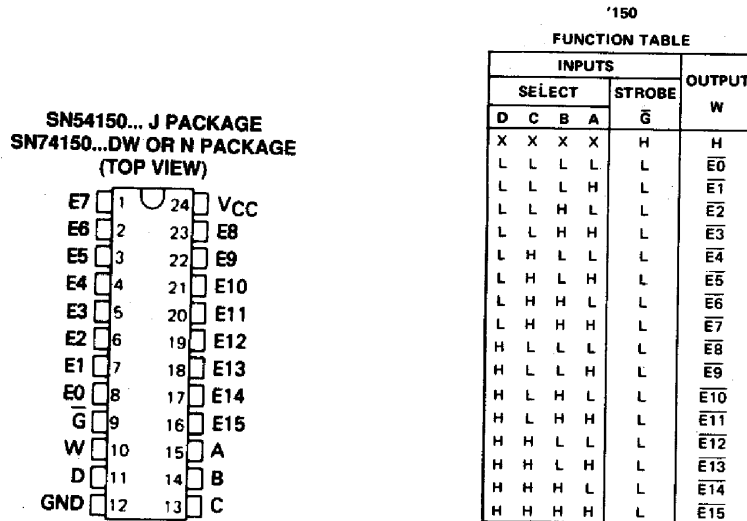


Fig. 2.8 Configurația pinilor și tabelul de adevăr pentru circuitul SN 74150

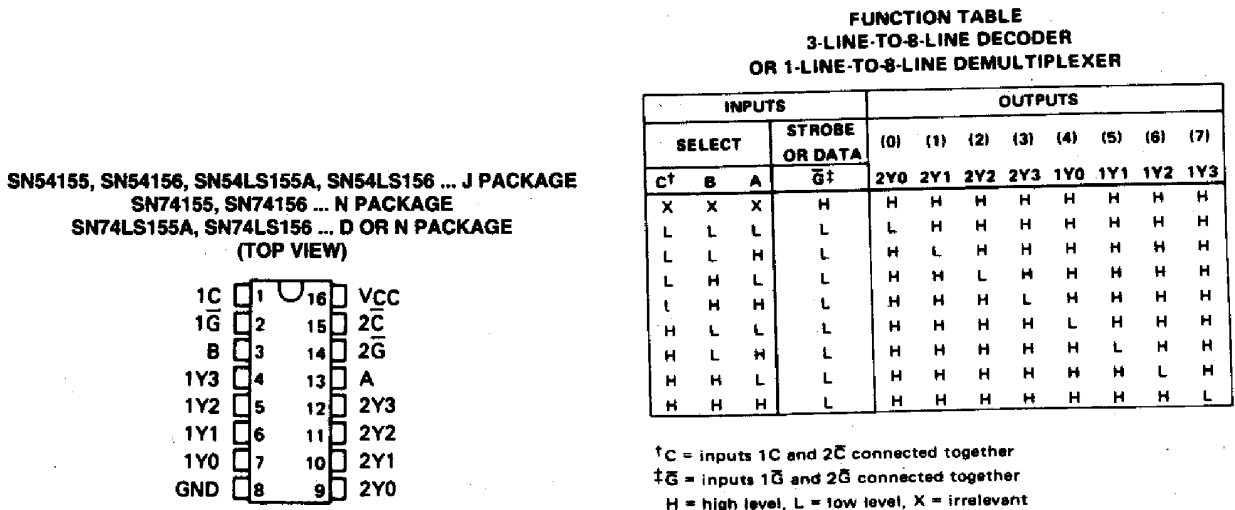


Fig. 2.9 Configurația pinilor și tabelul de adevăr pentru circuitul SN 74155

Observăm că pinul ENABLE din descrierea teoretică de mai sus se numește aici STROBE și are rolul de a permite accesul datelor la ieșire. El trebuie conectat la masă (0 logic), după cum rezultă din tabelele de adevăr. O altă deosebire importantă constă în notarea variabilelor aplicate pe intrările de selecție. La sinteza structurilor combinaționale cu porți logice, sau la prezentarea teoretică de mai sus, s-a notat cu A variabila mai semnificativă, iar cea mai puțin semnificativă cu B, C sau D, funcție de numărul total de variabile. Convenția adoptată de firma TEXAS INSTRUMENTS, după cum observăm din tabelele de adevăr, este exact inversă: C sau D este intrarea de selecție mai semnificativă, iar A este intrarea de selecție cea mai puțin semnificativă. În fond este vorba de o simplă convenție de care trebuie însă să ținem seamă pentru folosirea corectă a circuitelor. Iată de ce, printre altele, este obligatorie consultarea foii de catalog atunci când proiectăm diverse structuri folosind circuite integrate numerice!

Circuitul integrat MMC 4097(dacă este de fabricație românească, sau pur și simplu 4097, precedat de alte caractere, dacă este fabricat de altă firmă) are un pin numit INHIBIT, cu acțiune contrară pinului ENABLE. Este activ pe 1 logic, având rolul de a bloca accesul datelor

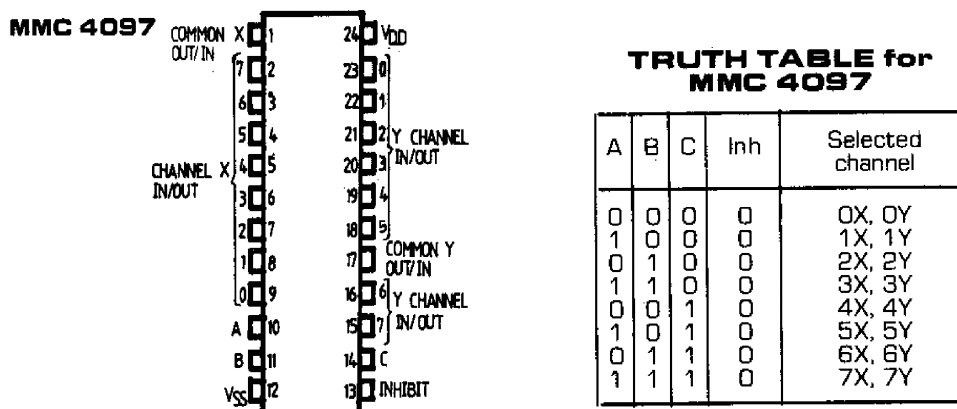


Fig. 2.10 Configurația pinilor și tabelul de adevăr pentru circuitul MMC 4097

la ieșire. El trebuie conectat la masă (0 logic), după cum rezultă din tabelele de adevăr. Convenția de alocare a ponderilor pentru variabilele de selecție se păstrează la fel ca la TTL: C este intrarea mai semnificativă, iar A este intrarea cea mai puțin semnificativă.

2.2.1 Se face analiza schemei logice din figura 2.11, reprezentând cele 4 funcții binare prin expresii algebrice, tabele de adevăr și diagrame temporale. Se verifică experimental rezultatele obținute în urma analizei teoretice a circuitului.

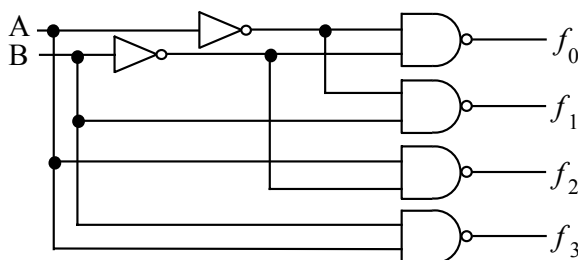


Fig. 2.11 Schema logică a unui decodificator realizat cu porți □

2.2.2 Se implementează cu număr minim de porți ȘI-NU funcția binară reprezentată prin tabelul de adevăr din figura 2.12 (x – este 0 sau 1, după cum ne convine). Se realizează montajul din figură și se face verificarea sintezei prin analiza circuitului folosind tabelul de adevăr. Schema logică obținută este unică?

A	B	C	D	f
0	0	0	0	0
0	0	0	1	1
0	0	1	0	x
0	0	1	1	x
0	1	0	0	x
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	x

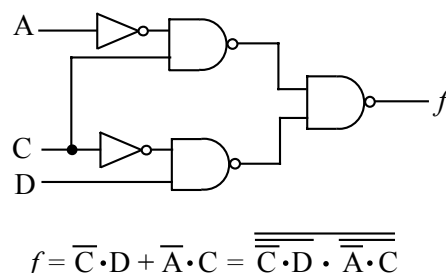
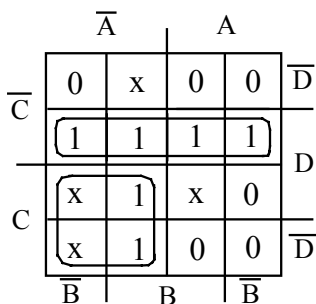


Fig. 2.12 Sinteza funcției binare reprezentată prin tabelul de adevăr □

2.2.3 Se consideră funcția binară definită de tabelul de adevăr din figura 2.13. Se implementează folosind circuitul integrat **SN 74155** ca demultiplexor cu 8 căi de ieșire și se verifică funcționarea circuitului obținut folosind tabelul de adevăr. Se repetă implementarea funcției cu circuitul **MMC4097** și se verifică din nou funcționarea corectă a montajului.

C	B	A	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

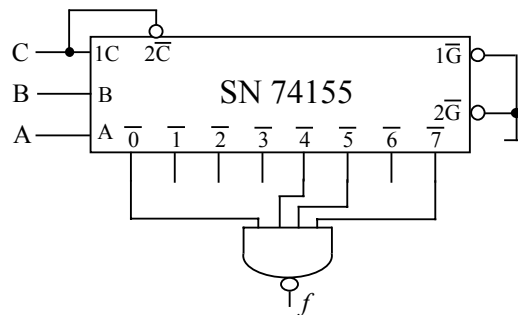


Fig. 2.13 Implementarea unei funcții binare cu circuitul SN 74155 □

2.2.4 Un juriu format din 4 persoane decide asupra reușitei unui concurent prin majoritate de voturi, semnalizate prin aprinderea unui element de afișaj. Se implementează funcția care comandă aprinderea LED-ului, reprezentată prin tabelul de adevăr din figura 2.14, folosind circuitul integrat **SN 74150**. Se verifică funcționarea corectă a montajului (în schemele logice cu multiplexoare am folosit pentru intrări notația I_i ; în foaia de catalog a circuitului **74150** de la TEXAS INSTRUMENTS, ele sunt notate cu E_i , iar complementarea lor la ieșire indică că este disponibilă ieșirea \overline{W} ; informația din catalog trebuie adaptată la cunoștințele noastre și este posibil ca în cataloagele altor firme să întâlnim alte notații).

D	C	B	A	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

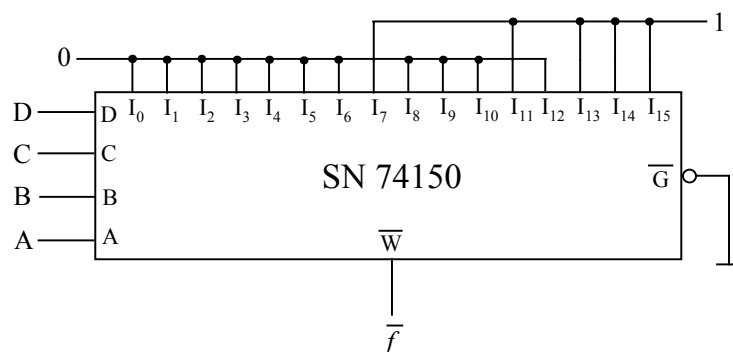


Fig. 2.14 Implementarea funcției de vot majoritar cu circuitul SN 74150 □

2.2.5 Se repetă implementarea funcției de la punctul 2.2.4, folosind de această dată circuitul integrat **SN 74155** ca demultiplexor cu 8 căi de ieșire și se verifică funcționarea circuitului obținut folosind tabelul de adevăr dat în figura 2.14. Să se arate cum s-a făcut sinteza schemei logice prezentate în figura 2.15.

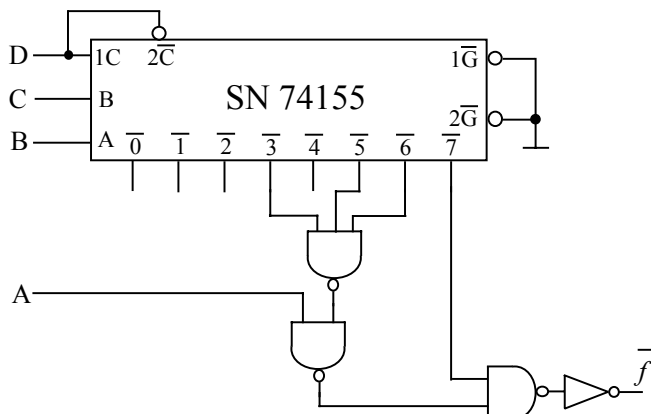


Fig. 2.15 Implementarea funcției de vot majoritar cu circuitul SN 74155 și porți □

2.2.6 Una dintre aplicațiile funcțiilor de multiplexare/demultiplexare este și conversia din format serie în format paralel, sau invers, al datelor. Această proprietate poate fi urmărită folosind circuitul din figura 2.16. Mai mult decât atât, circuitul demonstrează că porțile de transmisie din interiorul acestor structuri permit transmiterea semnalelor analogice, iar transmiterea acestor semnale este bidirecțională (nu contează care este intrarea sau ieșirea din circuit, ele pot fi oricând interschimbate între ele). Se introduce pe intrarea IN/OUT un semnal sinusoidal fără componentă continuă, de amplitudine mai mică de 5V. Acest semnal este regăsit, practic fără atenuare, la ieșirea OUT/IN a circuitului, pentru orice combinație a variabilelor de intrare A, B și C. Pentru o anumită combinație binară a acestor intrări, semnalul introdus se va regăsi și la una dintre ieșirile/intrările corespunzătoare ale DMUX/MUX-ului, numerotate de la 0 la 7. Se aplică semnalul și pe intrarea OUT/IN a circuitului, fiind regăsit la ieșirea IN/OUT. Semnalul sinusoidal (sau de orice altă formă) este generat cu ajutorul unui generator de semnal, iar vizualizarea lui la intrarea și la ieșirea din circuit se face cu ajutorul unui osciloscop cu 2 canale. Circuitul integrat **MMC 4097** este alimentat cu tensiunea de $\pm 5V$, deci excursia semnalului analogic de intrare trebuie să fie cuprinsă între $-5,5V$ și $+5,5V$, depășirea acestor limite având ca efect deschiderea diodelor din rețeaua de protecție a intrării și posibilitatea distrugerii lor.

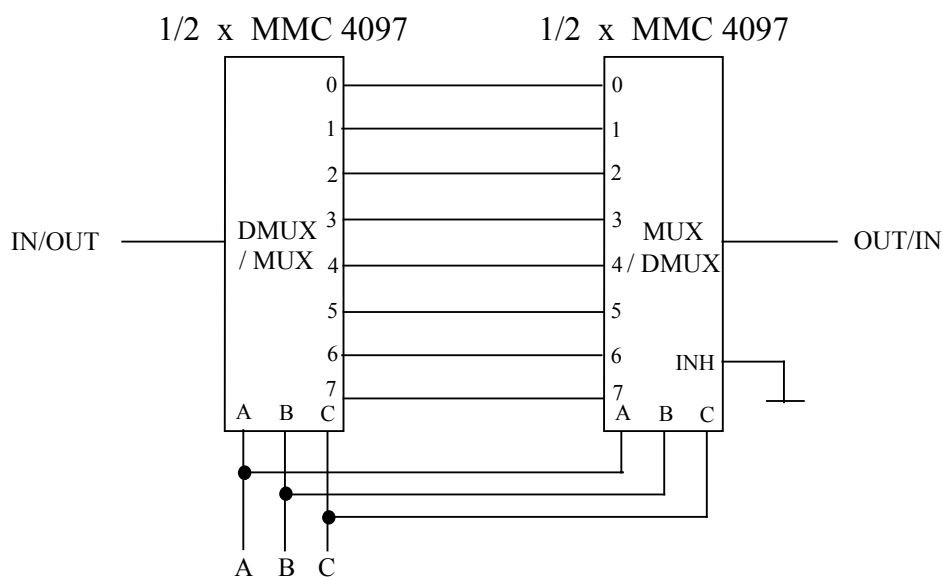


Fig. 2.16 Trecerea unui semnal analogic printr-un DMUX și un MUX analogic □

2.3 Probleme rezolvate

2.3.1 Să se proiecteze un circuit logic combinațional de tip "cheie electronică" cu 3 intrări și o singură ieșire. Fiecare intrare reprezintă starea unui contact: dacă contactul este deschis, acesta furnizează 1 logic pe intrarea respectivă. În starea inițială toate contactele sunt deschise și ieșirea circuitului este în 1 logic, adică alarma este dezactivată.

Dacă închiderea contactelor (care pot acționa, de exemplu, niște zăvoare electromagnetice) se face într-o anumită ordine prestabilită, atunci ieșirea rămâne în 1 logic și alarma nu este declanșată. Dacă ordinea de închidere a contactelor nu este respectată, ieșirea trece în 0 logic și se afișează starea de alarmă prin aprinderea unui LED.

Câte combinații posibile există? Dar pentru 5 contacte? Explicați cum trebuie să fie contactele și de ce.

Rezolvare:

Să presupunem că ordinea prestabilită de închidere a contactelor este B, C, A. Tabelul de adevăr pentru implementarea funcției de alarmare, minimizarea funcției cu ajutorul diagramei Veitch și schema logică obținută sunt date în figura 2.17.

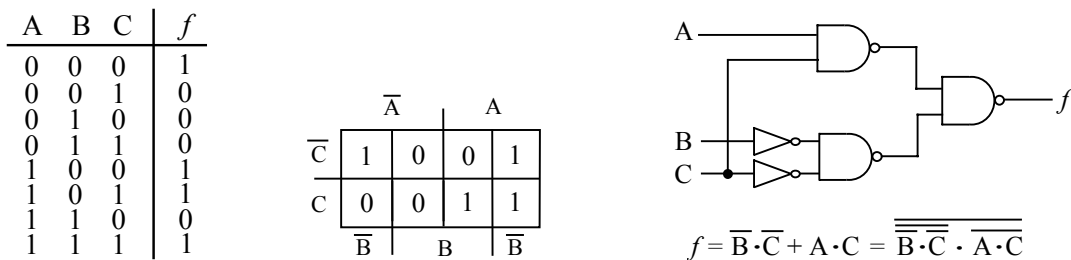


Fig. 2.17 Sinteza funcției binare de alarmare

Pentru 3 contacte există $3! = 6$ combinații posibile, iar pentru 5 contacte avem 120 de combinații distincte. Se folosesc întrerupătoare cu menținere, deoarece este necesară memorarea stărilor contactelor la fiecare pas. În lipsa acestor memorii mecanice, sunt necesare memoriile electronice, adică circuite secvențiale, deci problema nu este rezolvabilă prin metodele prezentate până acum în lucrare. □

2.3.2 Să se proiecteze un circuit de decodificare pentru afișajul cu 7 segmente din figura 2.18, care să permită afișarea a 4 stări distincte, prin literele O, L, H și E. Cele 4 stări distincte sunt date de toate combinațiile posibile realizate cu 2 variabile de intrare, A și B. Segmentele elementului de afișaj sunt aprinse pentru 0 logic și stinse pentru 1 logic.

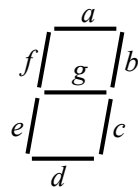


Fig. 2.18 Notății folosite pentru cele 7 segmente ale elementului de afișaj

Rezolvare:

Pentru fiecare segment de afișaj se generează câte o funcție binară conform cerințelor problemei. Rezultă tabelul de adevăr din figura 2.19. Prin minimizare se obțin ecuațiile circuitului.

A	B	a	b	c	d	e	f	g	Simbol
0	0	0	0	0	0	0	0	1	
0	1	1	1	1	0	0	0	1	
1	0	1	0	0	1	0	0	0	
1	1	0	1	1	0	0	0	0	

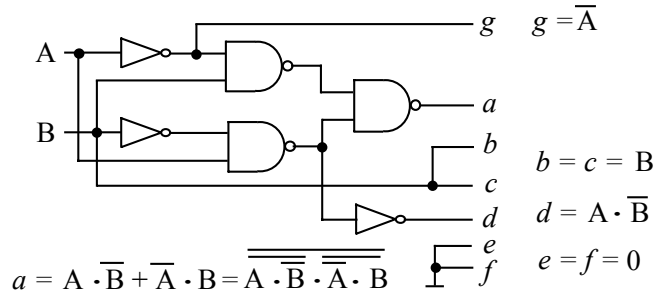


Fig. 2.19 Sinteza circuitului de decodificare pentru afișajul cu 7 segmente

2.3.3 O companie face angajări de personal pentru diferite compartimente. Candidații trebuie să îndeplinească cel puțin una dintre următoarele 5 condiții:

- a: bărbat, necăsătorit, cu studii superioare
- b: necăsătorit(ă), cu studii superioare, vârsta sub 30 de ani
- c: femeie, necăsătorită, fără studii superioare
- d: bărbat, vârsta sub 30 de ani
- e: necăsătorit(ă), vârsta peste 30 de ani

Să se proiecteze un circuit pentru selecția candidaților.

Rezolvare:

Se pune problema să găsim o funcție binară f , de mai multe variabile, pentru selecția candidaților. Dacă $f = 1$ se consideră candidat acceptat, iar dacă $f = 0$, atunci avem de-a face cu un candidat respins.

Stabilim pentru început care sunt variabilele funcției:

- A - vârsta (A = 1 dacă vârsta este sub 30 de ani, și A = 0 în caz contrar)
- B - sexul (B = 1 dacă este bărbătesc și B = 0 dacă este femeiesc)
- C - starea civilă (C = 1 dacă este necăsătorit(ă) și C = 0 dacă este căsătorit(ă))
- D - studii (D = 1 dacă are studii superioare și D = 0 dacă nu are studii)

Aceste variabile binare rezultă din datele problemei, iar proiectantul este liber să aleagă modul în care se atribuie valorile logice acestor variabile. Candidatul este acceptat dacă cel puțin una dintre condițiile cerute este îndeplinită, adică:

$$f = a + b + c + d + e, \text{ unde}$$

$$a = BCD, \quad b = ACD, \quad c = \overline{BCD}, \quad d = AB, \quad e = \overline{AC}$$

Expresia algebrică a funcției căutate este $f = BCD + ACD + \overline{BCD} + AB + \overline{AC}$, dar această formă a funcției nu este minimă. Pentru o minimizare rapidă și eficientă se trec termenii produs într-o diagramă Veitch. Soluția problemei este dată în figura 2.20.

	\overline{A}	A	
\overline{C}	0	0	1
C	0	0	1
	1	1	1
	1	1	1

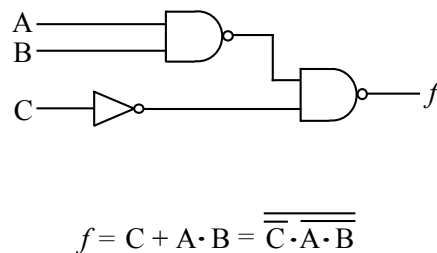


Fig. 2.20 Sinteza funcției binare $f(A, B, C, D)$

2.3.4 Să se facă sinteza funcției de vot majoritar, reprezentată prin tabelul de adevăr din figura 2.14, cu ajutorul unui demultiplexor cu 8 căi de ieșire.

Rezolvare:

Funcția poate fi scrisă în forma canonică disjunctivă sub forma: $f = P_7 + P_{11} + P_{13} + P_{14} + P_{15}$. Observăm însă că numărul de variabile ale funcției este mai mare decât numărul intrărilor de selecție ale demultiplexorului. Din acest motiv vom face o serie de transformări algebrice care să pună în evidență mintermenii unei funcții de 3 variabile, mintermeni care sunt disponibili la ieșirile demultiplexorului. Cea de-a patra variabilă va fi introdusă într-o logică combinațională suplimentară, realizată de obicei cu porți logice.

$$\begin{aligned} f &= P_7 + P_{11} + P_{13} + P_{14} + P_{15} = \overline{D}CBA + D\overline{C}BA + DC\overline{B}A + DCB\overline{A} + DCBA = \\ &= A(\overline{D}CB + D\overline{C}B + DC\overline{B} + DCB) + \overline{A}DCB = A(P'_3 + P'_5 + P'_6 + P'_7) + \overline{A} \cdot P'_7 = \\ &= A(P'_3 + P'_5 + P'_6) + P'_7 = A \cdot \overline{P'_3} \cdot \overline{P'_5} \cdot \overline{P'_6} + P'_7 = A \cdot \overline{P'_3} \cdot \overline{P'_5} \cdot \overline{P'_6} \cdot \overline{P'_7} \end{aligned}$$

Expresia algebrică obținută permite implementarea funcției f , conform schemei din figura 2.15. Este evident că oricare alte 3 variabile puteau fi aplicate pe intrările de selecție ale demultiplexorului, cu condiția refacerii calculelor de mai sus. □

2.3.5 Să se proiecteze un convertor de cod din cod binar în cod Gray pentru numere reprezentate pe 3 biți, folosind:

- un decodificator cu 3 intrări de selecție.
- multiplexoare cu câte 2 intrări de selecție.
- un număr minim de circuite.

Rezolvare:

Două reprezentări succesive în cod Gray diferă printr-un singur bit. Tabelul din figura 2.21 indică corespondența cod binar - cod Gray, iar schema logică alăturată prezintă soluția de la punctul a. Pentru sinteza schemelor logice în probleme, propunem ca variabila A să fie aplicată pe intrarea de selecție cea mai semnificativă. La problema 2.3.4 am păstrat convenția din catalog, pentru că schema logică obținută este implementată cu circuitul SN 74155, așa cum se vede în figura 2.15.

A	B	C	A_G	B_G	C_G
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

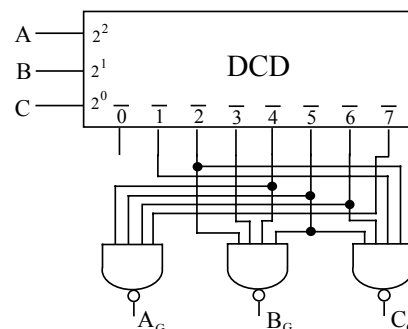


Fig. 2.21 Convertor de cod binar - Gray cu decodificator și porți

b) Se pot folosi multiplexoare cu câte 2 intrări de selecție prin transformările:

$$\begin{aligned} A_G &= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC = (\overline{A}\overline{B} + \overline{A}B) \cdot \overline{C} + (\overline{A}\overline{B} + \overline{A}B) \cdot C = P'_2 + P'_3 \\ B_G &= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} + A\overline{B}C = (\overline{A}\overline{B} + A\overline{B}) \cdot \overline{C} + (\overline{A}\overline{B} + A\overline{B}) \cdot C = P'_1 + P'_2 \\ C_G &= \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + A\overline{B}\overline{C} + A\overline{B}C = (P'_0 + P'_2) \cdot C + (P'_1 + P'_3) \cdot \overline{C} \end{aligned}$$

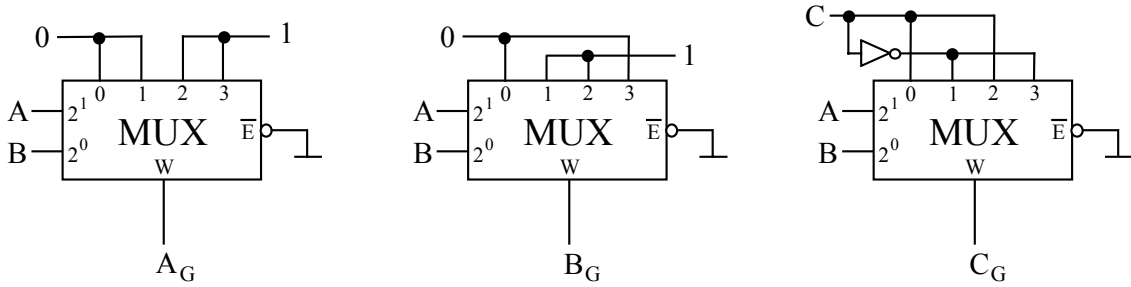


Fig. 2.22 Convertor de cod binar - Gray cu multiplexoare

Schema logică din figura 2.22 prezintă o soluție a problemei. Sunt posibile și alte soluții, funcție de alegerea variabilelor care se aplică pe intrările de selecție.

c) Implementarea cea mai simplă este cu porți logice. Prin minimizare cu ajutorul diagramei Veitch-Karnaugh rezultă următoarele relații:

$$A_G = A, \quad B_G = \bar{A}B + A\bar{B} = A \oplus B, \quad C_G = \bar{B}C + B\bar{C} = B \oplus C.$$

Deci sunt necesare numai două porți logice SAU-EXCLUSIV. □

2.3.6 Să se implementeze un sumator complet de 1 bit, folosind:

- a) multiplexoare cu 4 căi de intrare;
- b) demultiplexor cu 4 căi de ieșire și porți;

Comparați cele două soluții din punct de vedere al numărului de circuite integrate și al timpului de propagare.

Rezolvare:

Sumatorul de 1 bit are două intrări pentru operanzi, notate aici cu A și B, și încă o intrare pentru transportul de la sumatorul de rang inferior, notată aici cu C₋₁. Circuitul are două ieșiri, suma, notată cu S, și transportul, notată cu C. Tabelul de adevăr este dat în figura 2.23.

$$C = P_3 + P_5 + P_6 + P_7 = \bar{C}_{-1}AB + C_{-1}(\bar{A}B + A\bar{B} + AB) = P_3' + C_{-1}(P_1' + P_2')$$

$$S = P_1 + P_2 + P_4 + P_7 = \bar{C}_{-1}(\bar{A}B + A\bar{B}) + C_{-1}(\bar{A}\bar{B} + AB) = \bar{C}_{-1}(P_1' + P_2') + C_{-1}(P_0' + P_3')$$

Schemele logice care se obțin cu aceste ecuații sunt date tot în figura 2.23.

C ₋₁	A	B	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

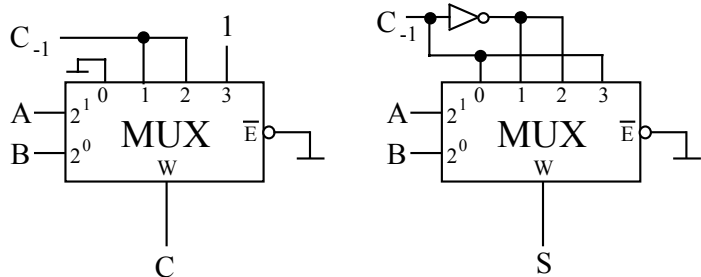


Fig. 2.23 Sumator complet de 1 bit implementat cu multiplexoare

b) Pentru implementarea cu demultiplexor, se fac următoarele transformări (punem în evidență termenii produs \bar{P}_i'):

$$C = P_3' + C_{-1}(P_1' + P_2') = \overline{\bar{P}_3' \cdot C_{-1} \cdot \bar{P}_1' \cdot \bar{P}_2'}$$

$$S = \bar{C}_{-1}(P_1' + P_2') + C_{-1}(P_0' + P_3') = \overline{\bar{C}_{-1} \cdot \bar{P}_1' \cdot \bar{P}_2' \cdot C_{-1} \cdot \bar{P}_0' \cdot \bar{P}_3'}$$

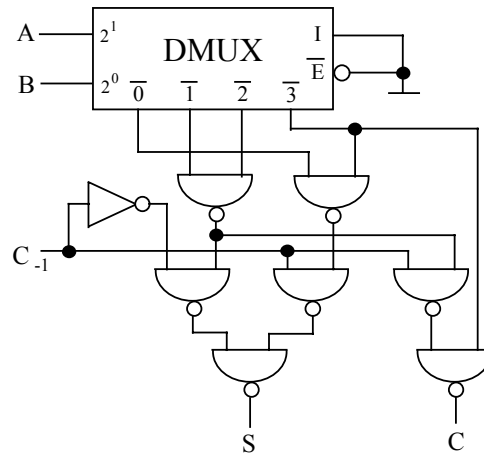


Fig. 2.24 Sumator complet de 1 bit implementat cu decodificator și porți ȘI-NU

Dacă se compară schemele logice astfel obținute, observăm că implementarea cu demultiplexor este mult prea costisitoare, atât din punct de vedere al numărului de circuite integrate, cât și al timpului de propagare. Implementarea cu multiplexoare este mai rezonabilă, dar cea mai simplă soluție se obține prin minimizarea funcțiilor și implementare cu porți. Această soluție a fost prezentată în curs. □

2.3.7 Un circuit logic combinațional are 4 intrări, 2 ieșiri și funcționează astfel încât:

- dacă $F = 00$, atunci $Q = I$;
- dacă $F = 01$, atunci $Q = I + 1(\text{mod } 4)$;
- dacă $F = 10$, atunci $Q = I - 1(\text{mod } 4)$;
- dacă $F = 11$, atunci $Q = \bar{I}$,

unde F , Q și I sunt cuvinte de câte doi biți.

- a) Să se scrie funcțiile logice Q_1 și Q_0 asociate ieșirilor circuitului.
- b) Să se implementeze circuitul anterior descris.

Rezolvare:

a) Se poate folosi tabelul de adevăr pentru fiecare dintre funcțiile cerute, dar și o metodă mai rapidă care presupune scrierea directă a funcției pornind de la o formulare convenabilă în limbaj natural. Este evident că pentru $F_1 = 0$ și $F_0 = 0$, $Q_1 = I_1$ și $Q_0 = I_0$. Pentru $F_1 = 0$ și $F_0 = 1$, se impune ca intrarea să fie $I_1 = 0$ și $I_0 = 1$, sau exact invers, $I_1 = 1$ și $I_0 = 0$, deoarece numerele 01 și 10 generează prin incrementare bitul cel mai semnificativ, $Q_1 = 1$. Se pot folosi și diagramele Veitch – Karnaugh de două variabile, I_1 și I_0 , pentru fiecare combinație F_1 , F_0 . Expresiile cerute în problemă devin:

$$Q_1 = \bar{F}_1 \bar{F}_0 I_1 + \bar{F}_1 F_0 (I_1 \oplus I_0) + F_1 \bar{F}_0 (\overline{I_1 \oplus I_0}) + F_1 F_0 \bar{I}_1$$

$$Q_0 = \bar{F}_1 \bar{F}_0 I_0 + \bar{F}_1 F_0 \bar{I}_0 + F_1 \bar{F}_0 \bar{I}_0 + F_1 F_0 I_0$$

b) Probabil că implementarea cea mai simplă se poate face folosind două multiplexoare cu câte 2 intrări de selecție (F_1 și F_0) și porți (două inversoare, o poartă SAU-EXCLUSIV și o poartă SAU-EXCLUSIV NEGAT). Lăsăm în seama cititorului găsirea altor soluții posibile. □