5 Circuite logice combinaționale – optimizare și sinteză

5.1 Obiective

Se prezintă funcțiile incomplet definite și modul de reprezentare a acestora. Se studiază tehnici de simplificare a funcțiilor booleene. Se realizează circuite combinaționale pentru implementarea unor funcții booleene în formă simplificată. Se studiază și se verifică funcționarea unor circuite combinaționale: generator de cod Excess-3, comparator de numere pe 2 biți, sumatoare pe 1 sau mai mulți biți.

5.2 Considerații teoretice

Circuitele logice se pot clasifica în *combinaționale* și *secvențiale*. Circuitele *combinaționale* sunt caracterizate de faptul că ieșirile implementează funcții a căror variabile sunt determinate numai de intrările circuitului. Circuitele combinaționale compuse din porți logice fundamentale sunt ușor de identificat, prin faptul că nu prezintă conexiuni cu reacție, adică de la pini de ieșire la pini de intrare. Funcțiile implementate pot fi de două feluri: *incomplet definite* și *complet definite*.

O funcție este *incomplet definită* când nu se cunoaște sau nu reprezintă interes rezultatul pentru un subset din combinațiile de intrare. Pentru aceste combinații rezultatul se notează cu X sau Ø. De exemplu, un circuit care primește cifre zecimale codificate binar pe 4 biți de intrare, va avea un comportament nedefinit pentru valori binare în intervalul 1010÷1111, deoarece nu au echivalent cifre zecimale. Un astfel de caz este circuitul care implementează codul Excess-3 (BCD incrementat cu 3) din Tabelul 5. 1. Circuitul va prezenta 4 biți pentru codul binar de intrare și 4 biți pentru codul Excess-3 de ieșire. Ieșirile vor implementa funcțiile W, X, Y, Z cu variabilele comune A, B, C, D.

Tabelul 5. 1 Tabelul de adevăr pentru codul Excess-3 (BCD incrementat cu 3)

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

5.2.1 Tehnici de optimizare a circuitului

Prin procesul de optimizare se determină un circuit cu aceeași funcționalitate, dar cu o structură mai simplă. Se urmărește reducerea numărului de operații și a numărului de variabile implicate, ceea ce va determina reducerea numărului de porți, a conexiunilor necesare și a lungimii acestora. O parte din tehnicile folosite sunt descrise în continuare:

a) **Folosirea variabilelor auxiliare**: De exemplu, expresiile $\begin{cases} a = (c \odot d) \cdot (e \oplus f) \\ b = \overline{c \odot d} + (e \oplus f) \end{cases}$ se pot rescrie cu variabilele auxiliare $m = c \odot d$, $n = e \oplus f$, astfel: $\begin{cases} a = m \cdot n \\ b = \overline{m} + n \end{cases}$.

b) **Aplicarea proprietăților algebrei booleene**: Implică aplicarea de relații din algebra booleană, precum: $\overline{\overline{x}} = x$, $x + 1 = 1$, $x \cdot 1 = x$, $x + \overline{x} = 1$ sau $x \cdot \overline{x} = 0$, etc. Astfel, în expresia $x \cdot y \cdot z + x \cdot y \cdot \overline{z}$ se poate da factor comun $x \cdot y$ și se obține echivalența: $x \cdot y \cdot z + x \cdot y \cdot \overline{z} = x \cdot y \cdot (z + \overline{z}) = x \cdot y \cdot 1 = x \cdot y$.

c) **Utilizarea de tehnici cu caracter general**, precum minimizarea cu diagrame Karnaugh sau metoda Quine-McCluskey.

5.2.2 Sinteza funcțiilor booleene

Procesul de sinteză a funcțiilor booleene presupune definirea funcțiilor sub o formă grafică (tabel de adevăr, diagrama Karnaugh), aducerea la o reprezentare analitică optimizată și implementarea circuitului corespunzător. În continuare, se vor studia câteva exemple de sinteză a unor circuite uzuale.

5.2.2.1 Generator de cod Excess-3

Folosind reprezentarea funcțiilor W, X, Y, Z din Tabelul 5. 1, care realizează conversia de la codul binar la codul Excess-3, se realizează diagramele Karnaugh pentru fiecare dintre acestea, ca în Figura 5. 1.

Minimizarea funcțiilor booleene la Forma Disjunctivă Minimă (FDM): În cadrul diagramelor se urmărește realizarea de grupuri dreptunghiulare cu celule vecine de 1 și X, care se încercuiesc (Figura 5. 1). Prima și ultima coloană, precum și primul și ultimul rând, sunt considerate vecine. Grupurile trebuie să conțină un număr de celule putere a lui 2 cât mai mare, iar numărul grupurilor să fie cât mai mic, încât toate celulele de 1 din diagramă să fie incluse în cel puțin un grup. Nu se vor realiza grupuri care să conțină numai celule de X. De asemenea, este posibil ca celule de X să nu fie incluse în niciun grup. O celulă de 1 sau X poate să facă parte din mai multe grupuri, dacă acest lucru ajută la creșterea dimensiunii acestora.

Pentru fiecare grup identificat se va obține un termen, prin aplicarea operației de conjuncție (ȘI) peste variabilele cu valori constante în dreptul celulelor grupului. În cadrul termenului variabilele apar negate, dacă în dreptul celulelor din grup au valoarea constantă 0 sau nenegate, dacă au valoarea 1. Forma Disjunctivă Minimă se obține ca disjuncție peste termenii rezultați asociați grupurilor.

Minimizarea funcțiilor booleene la Forma Conjunctivă Minimă (FCM): Tehnica este simetrică celei de minimizare la forma disjunctivă, deoarece se urmărește, după

aceleași criterii, formarea de grupuri cu celule de 0. La nevoie, grupurile pot include și celule de X. Termenii rezultați din grupuri aplică o disjuncție (SAU) numai peste variabilele constante în dreptul grupului: variabilele apar negate, dacă sunt constante la 1 și nenegate, altfel. Forma Conjunctivă Minimă se obține prin conjuncția termenilor rezultați din grupuri.

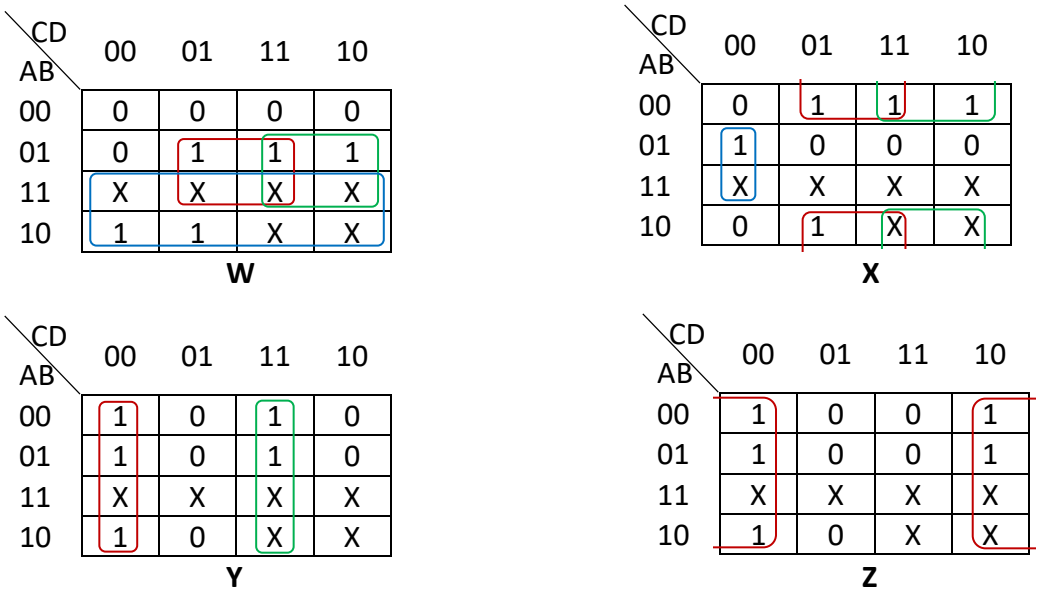


Figura 5. 1 Diagramele Karnaugh pentru ieșirile generatorului de cod Excess-3

Din Figura 5. 1 rezultă următoarele expresii la Forma Disjunctivă Minimă, pentru cele 4 funcții de ieșire ale circuitului:

$$\begin{aligned} W &= B \cdot D + B \cdot C + A \\ X &= \overline{B} \cdot D + \overline{B} \cdot C + B \cdot \overline{C} \cdot \overline{D} \\ Y &= \overline{C} \cdot \overline{D} + C \cdot D \\ Z &= \overline{D} \end{aligned} \qquad (5.\ 1)$$

5.2.2.2 Comparator de numere fără semn, pe 2 biți

Circuitul de comparare a 2 numere fără semn, pe 2 biți, $N_1=A_1A_0$, respectiv $N_2=B_1B_0$, generează 3 ieșiri F_1, F_2, F_3, care indică relația de mărime dintre acestea, după următoarele reguli:

- F_1=1, dacă $N_1>N_2$, altfel 0;
- F_2=1, dacă $N_1=N_2$, altfel 0;
- F_3=1, dacă $N_1<N_2$, altfel 0.

Schema bloc a circuitului este redată în figura următoare.

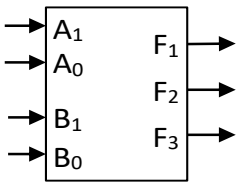


Figura 5. 2 Schema bloc a comparatorului fără semn, pe 2 biți

Așa cum s-a văzut în lucrarea anterioară, comparatorul fără semn, pe 2 biți, se poate realiza cu comparatoare fără semn pe 1 bit și porți logice. O soluție alternativă, bazată integral pe porți logice fundamentale, presupune generarea diagramelor Karnaugh ale ieșirilor, ca funcții de cele 4 intrări: A_1, A_0, respectiv B_1, B_0. De exemplu, în diagrama pentru F_1 se completează cu 1 în celulele pentru care $A_1A_0>B_1B_0$ și cu 0 în rest. Completarea pentru F_3 folosește o abordare simetrică, iar în cazul F_2 se pune 1, dacă $A_1A_0=B_1B_0$. Diagramele rezultate sunt redate în figura următoare:

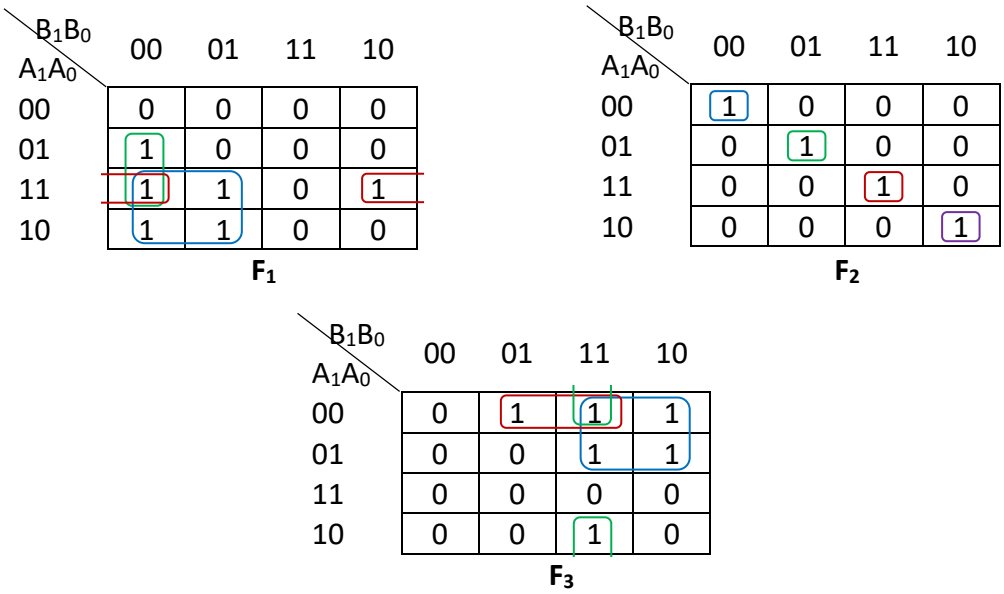


Figura 5. 3 Diagramele Karnaugh pentru ieșirile comparatorului fără semn, pe 2 biți

Expresiile la Forma Disjunctivă Minimă rezultate pe baza grupărilor efectuate în diagramele Karnaugh sunt următoarele:

$$\begin{aligned} F_1 &= A_1 \cdot \overline{B_1} + A_0 \cdot \overline{B_1} \cdot \overline{B_0} + A_1 \cdot A_0 \cdot \overline{B_0} \\ F_2 &= \overline{A_1} \cdot \overline{A_0} \cdot \overline{B_1} \cdot \overline{B_0} + \overline{A_1} \cdot A_0 \cdot \overline{B_1} \cdot B_0 + A_1 \cdot A_0 \cdot B_1 \cdot B_0 + A_1 \cdot \overline{A_0} \cdot B_1 \cdot \overline{B_0} \\ F_3 &= \overline{A_1} \cdot B_1 + \overline{A_0} \cdot B_1 \cdot B_0 + \overline{A_1} \cdot \overline{A_0} \cdot B_0 \end{aligned} \quad (5.\ 2)$$

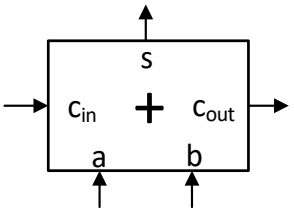
Aplicând axiomele și teoremele algebrei booleene funcția F_2 se poate rescrie într-o formă simplificată:

$$\begin{aligned} F_2 &= \overline{A_1} \cdot \overline{B_1} \cdot (\overline{A_0} \cdot \overline{B_0} + A_0 \cdot B_0) + A_1 \cdot B_1 \cdot (\overline{A_0} \cdot \overline{B_0} + A_0 \cdot B_0) = \\ &= (\overline{A_1} \cdot \overline{B_1} + A_1 \cdot B_1) \cdot (A_0 \odot B_0) = (\mathbf{A_1} \odot \mathbf{B_1}) \cdot (\mathbf{A_0} \odot \mathbf{B_0}) \end{aligned} \quad (5.\ 3)$$

5.2.2.3 Sumatorul complet pe 1 bit

Sumatorul complet pe 1 bit este unitatea de bază utilizată la implementarea sumatoarelor pe mai mulți biți, prin cascadarea acestuia de *n* ori, unde *n* este numărul de biți pe care se dorește să se realizeze operația de adunare. Concret, această unitate realizează suma a 3 intrări pe 1 bit și returnează rezultatul pe 2 biți. Cele 3 intrări sunt valorile a, b și transportul c_{in} (Carry In) rezultat din suma biților de rang inferior. Cei 2 biți de ieșire sunt transportul rezultat c_{out} (Carry Out) către rangul superior, respectiv

rezultatul s (Sum), de la rangul curent. Schema bloc a circuitului și tabelul de adevăr sunt prezentate în Figura 5. 4.



a	b	c_{in}	c_{out}	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

Figura 5. 4 Schema bloc și tabelul de adevăr pentru sumatorul complet pe 1 bit

Pe baza tabelului de adevăr se pot realiza diagramele Karnaugh pentru cele 2 ieșiri:

Figura 5. 5 Diagramele Karnaugh pentru ieșirile sumatorului complet pe 1 bit

Expresiile rezultate prin minimizare la Forma Disjunctivă Minimă sunt următoarele:

$$c_{out} = a \cdot c_{in} + b \cdot c_{in} + a \cdot b$$
$$s = a \cdot \bar{b} \cdot \overline{c_{in}} + \bar{a} \cdot \bar{b} \cdot c_{in} + a \cdot b \cdot c_{in} + \bar{a} \cdot b \cdot \overline{c_{in}} = \ldots = a \oplus b \oplus c_{in} \quad (5.4)$$

5.2.2.4 Sumatorul pe 4 biți (semioctet) obținut prin cascadare

Implementarea unui sumator pe 4 biți (semioctet) între 2 numere $A_{3:0}$ și $B_{3:0}$ se poate realiza cu 4 sumatoare pe 1 bit, conectate încât transportul să se propage de la unitatea de rang inferior spre cele superioare (cascadare). Intrarea de transport la rangul cel mai mic va fi conectată la 0 (GND) – se obține un semi-sumator pe 4 biți. Transportul pe ieșire la rangul cel mai mare va indica depășirea numărului de 4 biți rezervat rezultatului $S_{3:0}$. Schema logică este prezentată în figura următoare:

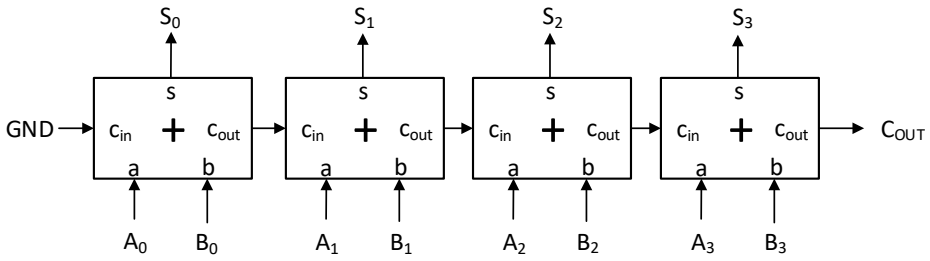


Figura 5. 6 Sinteza unui semi-sumator pe 4 biți prin cascadarea sumatoarelor pe 1 bit

Notă: Într-o manieră asemănătoare, sumatoarele pe 4 biți pot fi conectate pentru a implementa sumatoare pe orice număr de biți, multiplu de 4: intrarea de transport la rangul cel mai mic va fi conectată la 0, iar celelalte linii de transport vor fi interconectate pentru a realiza propagarea acestuia la sumatoarele de rang superior.

5.2.2.5 Implementarea sumatoarelor pe mai mulți biți folosind tehnici de minimizare

Alternativ, se pot implementa sumatoare pe mai mulți biți exprimând cifrele rezultatului ca funcții de biții de intrare. Astfel, se generează tabelul de adevăr al ieșirilor, se aplică tehnici de minimizare și se implementează expresiile obținute cu porți logice fundamentale. De exemplu, un semi-sumator pe 2 biți, care realizează suma între numerele A_1A_0 și B_1B_0, va prezenta la ieșire suma S_1S_0 și transportul C_{OUT}, cu funcționalitatea descrisă în Tabelul 5. 2. Cele 3 funcții, S_1, S_0, respectiv C_{OUT}, se pot minimiza folosind diagrame Karnaugh și proprietățile algebrei booleene, expresiile rezultate fiind următoarele:

Tabelul 5. 2 Semi-sumator pe 2 biți

A_1	A_0	B_1	B_0	C_{OUT}	S_1	S_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

$$\begin{aligned} C_{OUT} &= A_1 \cdot B_1 + A_0 \cdot B_1 \cdot B_0 + A_1 \cdot A_0 \cdot B_0 \\ S_1 &= A_1 \oplus B_1 \oplus (A_0 \cdot B_0) \\ S_0 &= A_0 \oplus B_0 \end{aligned} \quad (5.5)$$

Observație: Avantajul acestei metode constă în faptul că numărul de porți logice cumulate este mai redus față de metoda implementării cu cascadare de sumatoare pe număr redus de biți. Dezavantajul îl reprezintă procesul de minimizare costisitor, odată cu creșterea exponențială a numărului de combinații posibile pe intrări. Diagramele Karnaugh pot fi folosite pentru minimizări de funcții cu maxim 4 variabile. Așadar, pentru sumatoare cu intrări pe mai mult de 2 biți sunt necesare tehnici de minimizare alternative, precum metoda Quine-McCluskey.

5.3 Activități practice

1. Aduceți la Forma Disjunctivă Minimă funcția: $f = \sum(0, 4,7,10,12,13) + \sum_{\Phi}(1,8,15)$.
2. Implementați și testați pe placă ieșirea W a generatorului de cod Excess-3.
3. Implementați și testați pe placă funcția F_2 a comparatorului fără semn, pe 2 biți, cu porți XNOR.
4. Implementați și testați pe placă sumatorul complet pe 1 bit. Pentru implementarea ieșirii s (Sum) se va folosi varianta cu porți XOR.
5. Implementați și testați în Logisim un semi-sumator pe 4 biți folosind sumatoare pe 1 bit (cu atributul `Data Bits = 1`) din librăria *Arithmetic*.
6. Implementați și testați în Logisim un convertor de cod din 8421 (cod BCD) în 2421 (cod Aiken, https://en.wikipedia.org/wiki/Aiken_code). Pentru implementare, se va realiza tabelul de adevăr și se vor determina Formele Disjunctive Minime ale ieșirilor.