

CURS - 9

CONVERTOARE NUMERIC-ANALOGICE – DAC

Autor: Conf.dr.ing. Constantin Harja

Măsurarea și prelucrarea numerică a mărimilor cu variație continuă în timp necesită două operații esențiale pe lanțul de prelucrare a informației de măsurare. Mai întâi mărimile analogice trebuie exprimate sub formă numerică și această operație se realizează cu ajutorul convertoarelor analog-numerice, *ADC (Analog-to-Digital Converter)*. În esență, un *ADC* realizează o corespondență biunivocă între nivelul mărimii analogice de intrare și un cod numeric rezultat la ieșire. Apoi, sunt frecvente situațiile în care o mărime exprimată sub formă numerică trebuie convertită în semnal analogic. Această operație se realizează cu ajutorul convertoarelor numeric-analogice, *DAC (Digital-to-Analog Converter)* care realizează tot o corespondență biunivocă, dar de data aceasta între codul numeric de intrare și o mărime analogică rezultată la ieșire.

Atât *DAC* cât și *ADC*, operează cu numere zecimale exprimate sub formă binară (*baza de numerație 2*). Prin urmare, prezintă interes cunoașterea codurilor numerice.

IV.1. CODURI UZUALE ÎN CONVERSIILE A/D ȘI D/A

Pentru reprezentarea binară a mărimilor analogice se utilizează diferite coduri numerice care prezintă anumite particularități. O primă departajare se referă la polaritatea semnalului pe care îl reprezintă. Din acest punct de vedere, există coduri numerice unipolare și coduri numerice bipolare.

IV.1.1. CODURI NUMERICE UNIPOLARE

Pentru reprezentarea numerică a semnalelor analogice care prezintă o singură polaritate, pozitivă sau negativă, codul numeric exprimă numai valoarea normalizată a mărimii analogice corespunzătoare. În acest scop, cel mai folosit cod numeric este așa-zisul *cod binar natural*, în *logică pozitivă*. Acest cod poate fi utilizat și sub formă complementată, sub denumirea de *cod binar complementar*. Trecerea de la unul la celălalt din aceste coduri se realizează prin complementarea fiecărui bit component.

În afară de cele două coduri menționate mai sus, se mai utilizează și alte coduri binare, în funcție de aplicația practică concretă. Astfel, în cazul aparatelor de măsurat numerice, a căror afișaj funcționează în cod zecimal, sunt deosebit de utile codurile *binar-zecimale*, *BCD*. Codurile *BCD* utilizează o reprezentare bazată pe cifre zecimale, fiecare dintre ele fiind exprimată printr-un număr de 4 cifre binare (4 biți).

Deoarece un număr binar de 4 biți poate lua $2^4 = 16$ valori, iar un număr zecimal de o cifră poate lua numai 10 valori, rezultă că codurile *BCD* renunță la o parte din valorile unui număr binar de 4 biți. Din acest motiv, pentru reprezentarea unui anumit număr în cod *BCD* este nevoie de mai mulți biți decât în codul binar natural.

Codurile *BCD* sunt de obicei coduri ponderate, adică fiecărui bit i se atribuie o anumită pondere, iar valoarea cifrei zecimale rezultă prin însumarea produselor dintre ponderea și valoarea bitului. Cele mai folosite coduri *BCD* ponderate sunt codurile *BCD-8421* și *BCD-2421*, conform *Tabel IV.1*. Primul dintre acestea, numit cod *BCD natural*, atribuie cifrelor zecimale 0-9 codurile

binare naturale corespunzătoare.

Tabel IV.1. Reprezentarea codurilor BCD.

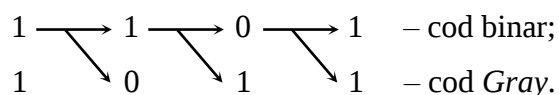
Valoare zecimală	Cod BCD 8 4 2 1	Cod BCD 2 4 2 1
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 1 1
6	0 1 1 0	1 1 0 0
7	0 1 1 1	1 1 0 1
8	1 0 0 0	1 1 1 0
9	1 0 0 1	1 1 1 1

Codul BCD-2421 este identic cu codul BCD-8421 până la cifra zecimală 4. Apoi, cifrele 5-9 reprezintă complementul binar al cifrelor 0-4, în ordine inversată.

Un alt cod unipolar, ceva mai deosebit, este codul *Gray*. Acesta este un cod neponderat care prezintă particularitatea că între două valori consecutive se modifică de fiecare dată doar valoarea unui singur bit, după cum se observă din Tabel IV.2.

Conversia din cod binar natural în cod *Gray*, se efectuează după următoarea regulă:

- **MSB (Most Significant Bit)** din codul binar este egal cu **MSB** din codul *Gray*;
- apoi, spre bitul **LSB (Least Significant Bit)** tranzițiile 1→0 sau 0→1, din codul binar, dau valoarea 1 pentru biții din codul *Gray*, în caz contrar biții din codul *Gray* au valoarea 0, așa cum se exemplifică în exemplu de mai jos:



Tabel IV.2. Reprezentarea codului Gray comparativ cu codul binar natural.

Valoare zecimală	Cod binar	Cod Gray	Valoare zecimală	Cod binar	Cod Gray
0	0 0 0 0	0 0 0 0	8	1 0 0 0	1 1 0 0
1	0 0 0 1	0 0 0 1	9	1 0 0 1	1 1 0 1
2	0 0 1 0	0 0 1 1	10	1 0 1 0	1 1 1 1
3	0 0 1 1	0 0 1 0	11	1 0 1 1	1 1 1 0
4	0 1 0 0	0 1 1 0	12	1 1 0 0	1 0 1 0
5	0 1 0 1	0 1 1 1	13	1 1 0 1	1 0 1 1
6	0 1 1 0	0 1 0 1	14	1 1 1 0	1 0 0 1
7	0 1 1 1	0 1 0 0	15	1 1 1 1	1 0 0 0

Regula de conversie inversă, din codul *Gray* în codul binar natural, se poate deduce cu ușurință din cea precedentă. Prin urmare:

- *MSB* din codul *Gray* are aceeași valoare cu *MSB* din codul binar;
- apoi, spre bitul *LSB*, dacă un bit din codul *Gray* este 1, bitul corespunzător din codul binar va fi complementul bitului precedent, iar dacă un bit din codul *Gray* este 0, bitul corespunzător din codul binar va fi identic cu bitului precedent, astfel:

$$\begin{array}{ccccccc} 1 & & 0 & & 1 & & 1 \\ & \swarrow & & \swarrow & & \swarrow & \\ 1 & \longrightarrow & 1 & \longrightarrow & 0 & \longrightarrow & 1 \end{array} \quad \begin{array}{l} \text{– cod Gray;} \\ \text{– cod binar.} \end{array}$$

Plecând de la regulile de conversie din cod binar în cod *Gray* și invers, se pot sintetiza circuitele logice apte să realizeze aceste conversii, conform Fig.IV.1.

Datorită faptului că la codul *Gray*, între două valori consecutive ale codului se schimbă valoarea unui singur bit, se evită fenomenele tranzitorii, foarte supărătoare, care pot apărea în schemele fizice, datorită schimbării nesimultane a valorii biților.

Astfel, considerând un *DAC* de 4 biți care lucrează în cod binar și că numărul de intrare se schimbă de la valoarea 7 la 8, adică 0111 → 1000, dacă comută mai întâi bitul *MSB* se poate ajunge la starea finală prin tranzițiile 0111 → 1111 → 1000, sau dacă *MSB* comută ultimul, tranzițiile până în starea finală sunt 0111 → 0000 → 1000.

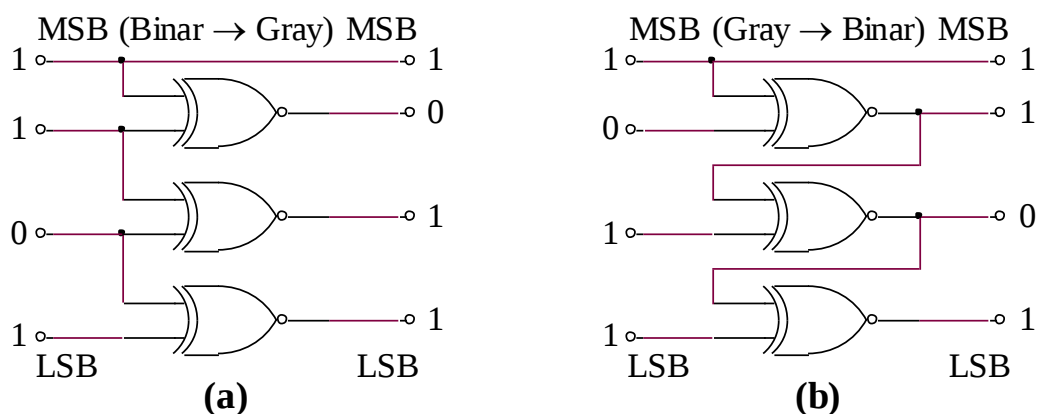


Fig.IV.1. Circuite logice de conversie cod binar-cod *Gray* și invers.

Ca urmare, pe durata tranziției, datorită stării intermediare care poate apărea prin comutarea nesimultană a biților, tensiunea de ieșire a *DAC* poate înregistra salturi imprevizibile generatoare de erori. Dacă în exemplul considerat, *DAC* lucrează în cod *Gray*, comutând un singur bit, fenomenele parazite descrise mai sus nu mai apar.

IV.1.2. CODURI NUMERICE BIPOLARE

Pentru a fi posibilă funcționarea cu semnale bipolare, facilitate oferită de majoritatea circuitelor *DAC* sau *ADC*, sunt necesare coduri binare care să permită exprimarea atât a valorii, cât și a semnelui mărimii analogice. În acest caz, codurile se numesc *coduri binare bipolare*. În tehnica numerică de calcul, există posibilitatea utilizării mai multor variante de coduri binare, cum ar fi: codul *semn-modul*, codul *complementar față de 2*, codul *complementar față de 1*, codul *binar*

deplasat, codul inversat analogic, precum și codurile inversate (complementare) ale acestora.

- **Codul semn-modul.** Acest cod, spre deosebire de codul binar natural, utilizează un bit suplimentar pentru indicarea semnului, care se plasează înaintea bitului *MSB*. De obicei, bitul de semn are valoarea 0 pentru numere pozitive și 1 pentru cele negative. Ceilalți biți, care indică modulul, rămân neschimbați indiferent de semn.

Considerând un număr binar de n biți, $a_{n-1} a_{n-2} \dots a_1 a_0$, valoarea acestuia în zecimal poate fi exprimată și sub forma compactă următoare:

$$N = \sum_{i=0}^{i=n-1} a_i 2^i . \quad (IV.1)$$

Dacă *MSB* indică semnul, (IV.1) poate fi transcrisă astfel:

$$N = a_{n-1} 2^{n-1} + \sum_{i=0}^{i=n-2} a_i 2^i , \quad (IV.2)$$

în care, $a_{n-1} = 0$ pentru numere pozitive și $a_{n-1} = 1$ pentru numere negative. De exemplu, codul 0111 = +111 = +7, iar codul 1111 = -111 = -7. Din cele arătate, se poate ușor anticipa faptul că, în codul semn-modul există două reprezentări pentru valoarea 0 și anume 0000 = 0^+ și 1000 = 0^- . Acest aspect, constituie un dezavantaj care complică atât *software*-ul, cât și *hardware*-ul circuitelor utilizate.

- **Codul binar deplasat.** Altă posibilitate de a exprima numeric mărimi analogice bipolare este separarea în două a intervalului de variație unipolar și atribuirea unei jumătăți valorilor pozitive și a celeilalte valorilor negative. Practic, acest lucru se poate realiza printr-o deplasare de nivel pe partea analogică a circuitelor, egală cu jumătate din intervalul de variație, adică cu valoarea *MSB*. Astfel, un interval de variație unipolar ($0 \dots v_{\max}$) va deveni ($-v_{\max}/2 \dots +v_{\max}/2$). Ca urmare, o valoare nulă la intrare va genera codul 1000...0. În acest caz, valoarea zecimală are expresia:

$$N = \sum_{i=0}^{i=n-1} a_i 2^i - 2^{n-1} . \quad (IV.3)$$

Codul binar deplasat prezintă avantajul că implică modificări simple la partea de circuite, constând într-o deplasare de nivel și eventual o dublare a factorului de scară dacă se dorește ca intervalul de variație bipolar să fie ($-v_{\max} \dots +v_{\max}$). În plus, acest cod este compatibil cu sistemele de intrare/ieșire ale calculatoarelor, fiindcă poate fi ușor convertit în codul complementar față de 2, prin complementarea bitului *MSB*. Și de asemenea, prezintă o valoare unică pentru zero.

Însă codul binar deplasat are și dezavantaje, cum ar fi erorile dinamice și statice care apar la trecerea prin zero. Erorile dinamice apar din cauza schimbării masive de biți ($01 \dots 1 \rightarrow 10 \dots 0$), dacă biții nu comută simultan. Erorile statice apar din cauză că zeroul analogic rezultă prin diferență între două tensiuni, una fiind referința *ADC/DAC*, iar cealaltă tensiunea cu care se realizează deplasarea de nivel.

- **Codul complementar față de 2.** În acest cod, un număr pozitiv are reprezentarea:

$$N^+ = 0 \cdot 2^{n-1} + \sum_{i=0}^{i=n-2} a_i 2^i , \quad (IV.4)$$

iar un număr negativ are reprezentarea:

$$N^- = 1 \cdot 2^{n-1} + \sum_{i=0}^{i=n-2} \bar{a}_i 2^i + 1, \quad (IV.5)$$

unde $\bar{a}_i = 1 - a_i$ este complementul față de 1 al bitului a_i .

Codul complementar față de 2 este foarte util în calculatoarele numerice, fiindcă simplifică circuitele aritmetice. El poate fi considerat ca un set de numere negative și astfel, operația de scădere se rezumă la în adunarea unui număr cu complementul față de 2 al celui alt.

- **Codul complementar față de 1.** În acest cod, numerele pozitive se exprimă la fel ca și în codul complementar față de 2, iar numerele negative au reprezentarea:

$$N^- = 1 \cdot 2^{n-1} + \sum_{i=0}^{i=n-2} \bar{a}_i 2^i. \quad (\text{IV.6})$$

Codul complementar față de 1 prezintă două valori pentru zero (00...0 și 11...1) și în general nu este utilizat pentru funcționarea circuitelor numerice.

- **Codul inversat analogic.** Numit și *cod cu referință negativă*, nu este un cod diferit de cele menționate mai sus, ci prezintă doar o inversare a sensului de variație al acestora. Astfel, dacă pentru codurile normale numerele cresc pentru variația semnalului analogic în intervalul $(0 \dots v_{\max})$ sau $(-v_{\max} \dots +v_{\max})$, în cazul codului inversat analogic, creșterea numerelor corespunde variației în sens invers a mărimii analogice și anume în intervalul $(v_{\max} \dots 0)$ sau $(+v_{\max} \dots -v_{\max})$.

În Tabel IV.3 sunt date valorile codurile bipolare prezentate mai sus.

Tabel IV.3. Exemplificarea codurilor binare bipolare

Valoare zecimală	Cod semn-modul	Cod binar deplasat	Cod complement față de 2	Cod complement față de 1
+7	0111	1111	0111	0111
+6	0110	1110	0110	0110
+5	0101	1101	0101	0101
+4	0100	1100	0100	0100
+3	0011	1011	0011	0011
+2	0010	1010	0010	0010
+1	0001	1001	0001	0001
0 ⁺	0000	1000	0000	0000
0 ⁻	1000	(1000)	(0000)	1111
-1	1001	0111	1111	1110
-2	1010	0110	1110	1101
-3	1011	0101	1101	1100
-4	1100	0100	1100	1011
-5	1101	0011	1011	1010
-6	1110	0010	1010	1001
-7	1111	0001	1001	1000
-8	—	0000	1000	—

IV.2. CARACTERISTICA DE TRANSFER A UNUI DAC

Correspondența biunivocă între mărimea analogică de ieșire și numărul prezentat la intrarea unui DAC, poate fi exprimată în principiu prin relația:

$$X = kN, \quad (IV.7)$$

în care X reprezintă mărimea analogică rezultată la ieșire, N – numărul prezentat la intrare, iar k – o constantă de proporționalitate. Expresia binară a numărului N fiind:

$$N: \quad a_{n-1} a_{n-2} \dots a_i \dots a_1 a_0, \quad (IV.8)$$

unde s-a considerat că N are n ranguri binare, adică n biți și $\forall a_i \in \{0,1\}$, valoarea zecimală a numărului N se poate calcula cu relația:

$$N = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_i2^i + \dots + a_12^1 + a_02^0 = \sum_{i=0}^{i=n-1} a_i2^i, \quad (IV.9)$$

iar valoarea maximă a numărului N este:

$$N_{\max} = 2^n - 1. \quad (IV.10)$$

De obicei, mărimea analogică de ieșire, X , se exprimă sub formă normată, adică prin valorile ei raportate la o valoare X_0 , corespunzătoare numărului $N_0 = 2^n$:

$$N_0 = 2^n = N_{\max} + 1, \quad \text{deci} \quad X_0 = kN_0. \quad (IV.11)$$

Raportând membru cu membru (IV.7) și (IV.11), se obține:

$$X = X_0 \frac{N}{2^n} = \frac{X_0}{2^n} \sum_{i=0}^{i=n-1} a_i2^i = X_0 \sum_{i=0}^{i=n-1} a_i2^{i-n}, \quad (IV.12)$$

sau:

$$X = X_0 (a_{n-1}2^{-1} + a_{n-2}2^{-2} + \dots + a_i2^{i-n} + \dots + a_12^{-(n-1)} + a_02^{-n}), \quad (IV.13)$$

Dacă în (IV.13) se inversează ordinea indicilor coeficienților a_i , rezultă:

$$X = X_0 (b_12^{-1} + b_22^{-2} + \dots + b_k2^{-k} + \dots + b_{n-1}2^{-(n-1)} + b_n2^{-n}) = X_0 \sum_{k=1}^{k=n} b_k2^{-k}. \quad (IV.14)$$

Din (IV.10) și (IV.12), se poate calcula valoarea maximă a mărimii analogice:

$$X_{\max} = X_0 \frac{2^n - 1}{2^n} = X_0 (1 - 2^{-n}). \quad (IV.15)$$

Valoarea $X_{\max}$ este denumită, conform terminologiei de limbă engleză, *domeniul maxim al ieșirii* (**Full Output Range – FR**) și reprezintă valoarea mărimii analogice de ieșire corespunzătoare valorii maxime a codului numeric de intrare, iar valoarea X_0 este denumită *capăt de scală* (**Full Scale – FS**). Din (IV.15), se poate ușor constata că valoarea $FS = X_0$ nu poate fi atinsă niciodată de către mărimea analogică de ieșire.

Spre exemplificarea celor menționate mai sus, valabile indiferent de numărul de biți, în Fig.IV.2 este reprezentată caracteristica de transfer a unui DAC doar de 3 biți.

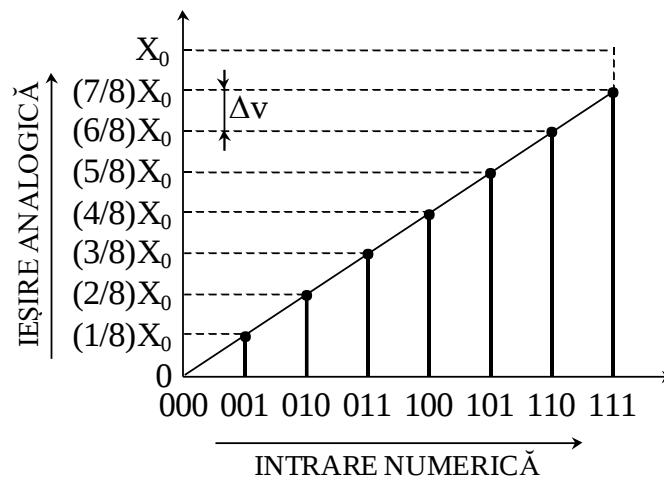


Fig.IV.2. Caracteristica de transfer a unui DAC de 3 biți.

Semnalul analogic de ieșire înregistrează valorile discrete $0, (1/8)X_0, \dots, (7/8)X_0$, denumite și *nivele de cuantificare*, care corespund fiecărei secvențe consecutive a codului numeric de intrare. Diferența Δv dintre două nivele de cuantificare, deci cea mai mică variație a mărimii analogice de ieșire care corespunde la o variație a codului numeric între două valori consecutive, constituie *rezoluția DAC*. Rezoluția poate fi exprimată direct prin Δv , fie prin numărul de biți, n ($\Delta v = X_0/2^n$), prin procente din domeniul maxim al ieșirii ($\Delta v = a[\%] FR$) sau din capăt de scală ($\Delta v = a[\%] FS$).

Prin urmare, deși mărimea de ieșire a unui DAC este numită *analogică*, aceasta nu este o mărime analogică în sensul propriu al cuvântului, adică nu este o mărime cu variație continuă în timp. Acest aspect este evident din (IV.14) și Fig.IV.2, conform cărora semnalul analogic de ieșire a unui DAC rezultă ca o sumă de componente analogice ponderate binar, în funcție de valorile biților corespunzători, numărul de componente fiind egal cu numărul de biți, iar valorile conform (IV.14).

IV.3. REȚELE REZISTIVE UTILIZATE LA REALIZAREA DAC

Soluția cea mai simplă, performanță utilizată de obținere a unui set de valori, ponderate binar, ale unei mărimi analogice, tensiune sau curent, este cea bazată pe utilizarea unor rețele de rezistențe cu valori adecvate. Rețelele de rezistențe cu această destinație pot fi *ponderate binar* sau în *scară*.

IV.3.1. REȚELE REZISTIVE PONDERATE BINAR

După cum sugerează și denumirea, aceste rețele sunt constituite dintr-un set de rezistențe cu valori în secvența $2^1R, 2^2R, \dots, 2^iR, \dots, 2^nR$, n fiind numărul de biți. Dacă o astfel de rețea este conectată la o sursă de tensiune de referință curenții care parcurg rezistențele din rețea vor rezulta cu valori ponderate binar, conform Fig.IV.3.

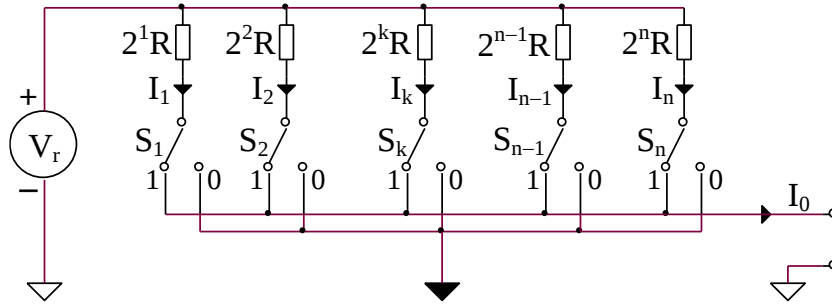


Fig.IV.3. Rețea rezistivă ponderată binar.

Neglijând rezistența comutatoarelor în starea închis, curenții prin rezistențele rețelei ponderate binar rezultă cu următoarele valori:

$$I_1 = \frac{V_r}{2^1 R}; \quad I_2 = \frac{V_r}{2^2 R}; \quad I_k = \frac{V_r}{2^k R}; \quad (IV.16)$$

iar curentul total, reprezentând curentul de ieșire, are valoarea:

$$I_0 = b_1 I_1 + b_2 I_2 + \dots + b_n I_n = \frac{V_r}{R} \sum_{k=1}^{k=n} b_k 2^{-k}, \quad (IV.17)$$

unde coeficienții b_k au valorile 1 sau 0, după cum S_k sunt pe poziția 1 sau 0.

Comutatoarele S_k mai pot fi conectate și în alte moduri, pentru a rezulta curenți ponderați binar, însă modul de conectare din Fig.IV.3 are unele avantaje, cum ar fi:

- comutatoarele fiind conectate spre masa de semnal, efectul cuplajului dintre sursa de comandă și calea de semnal devine nesemnificativ;
- prin rezistențe circulând același curent, indiferent de poziția comutatoarelor, curentul total absorbit de la sursa de referință rezultă constant, deci solicitarea sursei de referință în regim dinamic este practic nulă.

Rețeaua ponderată binar prezintă un dezavantajul major prin aceea că rezistențele componente ajung la valori mult diferite între ele, dacă numărul de biți este mare. Acest fapt afectează negativ împerecherea valorilor și a coeficienților de variație cu temperatura ale rezistențelor, deci precizia și stabilitatea rețelei. Apoi, rezistența minimă din rețea trebuie să fie mult mai mare decât rezistența r_{on} a comutatoarelor. De exemplu, dacă $n = 10$ și $R = 1 \text{ k}\Omega$, rezultă $R_{max} = 2^{10} R = 1024 \text{ k}\Omega$.

Din motivele menționate mai sus rețeaua rezistivă ponderată binar nu se utilizează la realizarea DAC cu un număr mare de biți. Ea este folosită de regulă doar la realizarea unor surse multiple de curent de 4 biți sau ca celulă de bază pentru o cifră zecimală din structura unor DAC care funcționează în cod BCD.

IV.3.2. REȚELE REZISTIVE ÎN SCARĂ

În ideea de a se reduce numărul de valori de rezistențe cât și raportul dintre ele, se utilizează rețele rezistive în scară, care nu necesită decât trei valori de rezistență: R_0 , numită *rezistență caracteristică* a rețelei și R_1, R_2 , după rezultă din Fig.IV.4.

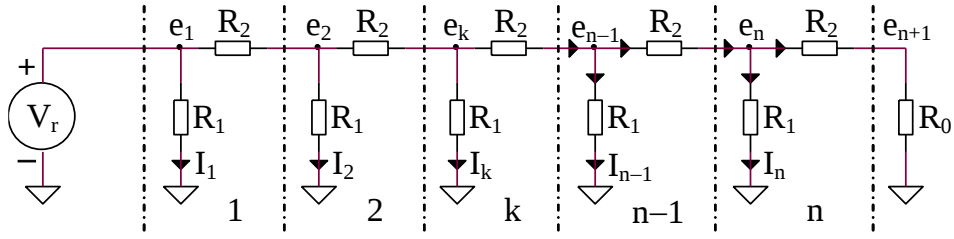


Fig.IV.4. Rețea rezistivă în scară.

Pentru calculul valorilor rezistențelor R_1 , R_2 și R_0 , se impun condiții similare celor din cazul atenuatoarelor rezistive, cum ar fi:

- rezistența de intrare în rețea trebuie să fie egală cu rezistența caracteristică, dacă rețeaua este închisă pe o rezistență egală cu cea caracteristică;
- raportul între valorile a doi curenți consecutivi trebuie să fie constant.

Relațiile care reflectă aceste condiții sunt următoarele:

$$R_1 \parallel (R_2 + R_0) = R_0 \quad \text{și} \quad I_{n-1}/I_n = K, \quad (\text{IV.18})$$

care prin explicitare pot fi puse sub forma:

$$\begin{aligned} I_{n-1} R_1 &= I_n \frac{R_1 + R_2 + R_0}{R_2 + R_0} [R_2 + R_1 \parallel (R_2 + R_0)] \\ K &= \frac{R_1 + R_2 + R_0}{R_1} = \frac{R_2 + R_0}{R_0}, \end{aligned} \quad (\text{IV.19})$$

din care se obțin valorile rezistențelor din structura rețelei în scară:

$$R_2 = (K - 1)R_0 \quad \text{și respectiv} \quad R_1 = \frac{K}{K - 1} R_0. \quad (\text{IV.20})$$

În practică, se impun rezistența caracteristică și raportul de divizare și apoi în funcție de valorile acestora se calculează valorile rezistențelor R_1 , R_2 . De exemplu:

- pentru un cod binar: $K = 2$, rezultă $R_1 = 2R_0$ și $R_2 = R_0$;
- pentru un cod zecimal: $K = 10$, rezultă $R_1 = (10/9)R_0$ și $R_2 = 9R_0$.

Se observă că în cazul codurilor binare, rețeaua rezistivă în scară conține numai două valori de rezistențe – R și $2R$, de unde și denumirea de *rețea rezistivă $R/2R$* .

Realizarea rețelei $R/2R$ asigură din punct de vedere tehnologic obținerea unor performanțe maxime, în raport cu celelalte tipuri de rețele (precizie a raportului, împerecherea coeficienților de temperatură etc.). De regulă, aceste rețele se realizează în structuri integrate, prin tehnologia straturilor subțiri, tehnologia straturilor groase din componente discrete, cea mai performantă fiind tehnologia straturilor subțiri.

În practică, la realizarea DAC, rețeaua $R/2R$ este utilizată într-o multitudine de moduri de conectare. Cu toate acestea, proprietățile ei se conservă. Analizând tensiunile care apar în nodurile rețelei, se observă că acestea satisfac relațiile:

$$e_{n+1} = e_n/2; \quad e_n = e_{n-1}/2; \quad \dots \quad e_2 = e_1/2; \quad (\text{IV.21})$$

adică, tensiunile în nodurile rețelei $R/2R$ sunt la rândul lor ponderate binar, reprezentând o consecință a faptului că valorile curenților sunt ponderate binar.