

2 Porți logice fundamentale

2.1 Obiective

Sunt prezentate noțiuni de bază și proprietăți de implementare tehnologică a porților logice fundamentale. Sunt descrise porțile logice fundamentale și asocierea acestora cu operațiile de bază din algebra booleană. Sunt introduse prioritățile operatorilor și se studiază elaborarea tabelului de adevăr și a schemei logice pornind de la expresia booleană. Sunt analizate diverse transformări și proprietăți ale algebrei booleene, care permit utilizarea porților logice fundamentale pentru a genera noi funcționalități.

2.2 Considerații de implementare tehnologică

Porțile logice fundamentale sunt cele mai elementare circuite utilizate în implementarea dispozitivelor numerice. Ele sunt caracterizate de o funcționalitate simplă și au în componența lor un număr redus de tranzistoare sau semiconductori. Raportat la modelarea în algebra booleană acestea implementează operațiile de bază: Inversor (NOT sau INV), SAU (OR), ȘI (AND). Datorită complexității reduse, tot în această categorie se încadrează și circuite care s-ar putea implementa din operațiile de bază precum: SAU-NU (NOR), ȘI-NU (NAND), SAU-EXCLUSIV (XOR), COINCIDENTĂ (XNOR).

Din punct de vedere al implementării la nivel electronic porțile logice fundamentale pot fi realizate în tehnologie TTL (Transistor-Transistor Logic) sau CMOS (Complementary Metal-Oxide Semiconductor). Există diferențe în ceea ce privește caracteristicile electrice ale celor două tehnologii, dintre care, printre cele mai cunoscute se pot enumera toleranța la variații de tensiune, viteza de comutare și puterea consumată.

- *toleranța la variații de tensiune* – conform valorilor din Figura 2. 1 se observă faptul că circuitele CMOS au un interval de tensiune mai extins asociat stării 0, ceea ce le conferă o toleranță mai bună la oscilații în jurul valorilor minime, probabilitatea ieșirii în mod necontrolat din această stare fiind mai scăzută, în comparație cu circuitele TTL.

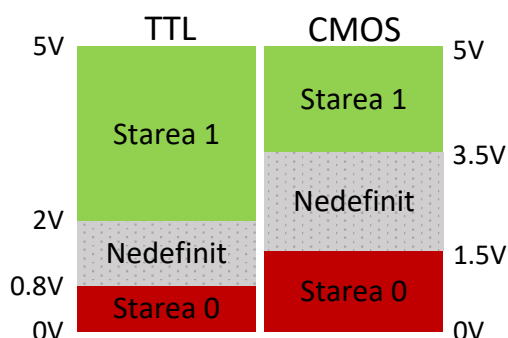


Figura 2. 1 Asocierea valorilor de tensiune pe intrări la nivelurile logice

- *viteza de comutare* – este strâns legată de timpul de reacție a circuitului, atunci când trebuie să-și schimbe starea pe ieșiri în urma unor modificări ale valorilor de intrare. Din acest punct de vedere circuitele CMOS sunt mai lente.
- *puterea consumată* – circuitele TTL consumă mai multă putere, motiv pentru care la echipamentele digitale mobile, cu alimentare pe bază de baterie sau acumulatori, este preferată implementarea cu circuite CMOS.



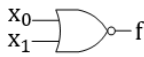




În continuare, vor fi studiate circuitele logice în tehnologie TTL. În catalogul de circuite ele sunt identificate prin coduri alcătuite din mai multe cifre, primele două fiind 74 sau 54. Conform proprietăților amintite anterior circuitele TTL se împart în mai multe subcategorii:

- standard;
- cu consum redus (low-power);
- rapide (high-speed);
- Schottky (cele mai rapide);
- combinații (ex. Low-Power Schottky).

2.3 Porțile logice fundamentale

La nivelul algebrei booleene porțile logice fundamentale implementează operații elementare de una sau două variabile de intrare, dar pot fi extinse la mai multe intrări. Acestea sunt enumerate în Tabelul 2. 1 alături de expresiile matematice, simbolurile grafice folosite în schema logică și tabelele de adevăr asociate.

Tabelul 2. 1 Porțile logice fundamentale

Inversor (NOT) $f = \overline{x_0}$ 			x_0 f 0 1 1 0
SAU (OR) $f = x_1 + x_0$ 	$x_1 \ x_0$ f 0 0 0 0 1 1 1 0 1 1 1 1	SAU-NU (NOR) $f = \overline{x_1 + x_0}$ 	$x_1 \ x_0$ f 0 0 1 0 1 0 1 0 0 1 1 0
SAU-EXCLUSIV (XOR) $f = x_1 \oplus x_0$ 	$x_1 \ x_0$ f 0 0 0 0 1 1 1 0 1 1 1 0	ȘI (AND) $f = x_1 \cdot x_0$ 	ȘI-NU (NAND) $f = \overline{x_1 \cdot x_0}$ 
COINCIDENTĂ (XNOR) $f = \overline{x_1 \oplus x_0} = x_1 \odot x_0$ 	$x_1 \ x_0$ f 0 0 1 0 1 0 1 0 0 1 1 1		

În ceea ce privește simbolurile folosite pentru reprezentarea grafică se observă faptul că operația de inversare este reprezentată de un cerculeț, iar în reprezentarea matematică, de o bară deasupra expresiei booleene. De asemenea, să nu se confunde operatorul + din algebra booleană, cu operația de adunare, respectiv operatorul \cdot , cu operația de înmulțire.

Conform tabelelor de adevăr asociate, efectul porților logice este următorul:

- operația NOT inversează valoarea de intrare;
- rezultatul unei operații SAU este 1 dacă cel puțin o intrare este 1 – echivalent cu valoarea maximă pe intrări;
- rezultatul unei operații ȘI este 1, dacă toate intrările sunt 1 – echivalent cu valoarea minimă pe intrări;
- operația SAU-NU este inversul operației SAU;
- operația ȘI-NU este inversul operației ȘI;
- rezultatul XOR este 1, dacă cele două intrări au valori diferite;
- rezultatul XNOR este 1, dacă cele două intrări au valori identice;
- operația XNOR este inversul operației XOR.

În funcție de operația implementată porțile logice pot fi identificate după codurile din tabelul următor:

Tabelul 2. 2 Codurile asociate porților logice fundamentale TTL

Poarta	Cod
NAND	7400
NOR	7402
NOT	7404
AND	7408
OR	7432
XOR	7486
XNOR	74266

Notă: În utilitarul Project Navigator porțile logice fundamentale sunt grupate în categoria *Logic* și se pot regăsi după denumirile lor (AND, OR, INV etc.). **INV înlocuiește NOT.**


Observație: Operația compusă SAU-NU se poate realiza și legând ieșirea porții SAU la intrarea porții NOT. O astfel de soluție presupune folosirea a două porți logice fundamentale, adică un număr dublu de resurse, o viteză de lucru mai scăzută din cauză că semnalele trebuie să se propage prin 2 porți și un consum de curent suplimentar. Din aceste motive se preferă implementarea operațiilor compuse de bază cu porți logice dedicate, care au comportamentul descris în tabelul de adevăr asociat. Același lucru este valabil și în cazul operațiilor ȘI-NU, XOR și XNOR.

2.4 Determinarea tabelului de adevăr al unei expresii booleene


La calculul rezultatului unei expresii se realizează operațiile, în ordinea priorităților (conform cu Tabelul 2. 3), pentru toate combinațiile de valori ale variabilelor. Dacă expresia are n variabile sunt 2^n combinații și tot atâtea valori de calculat pentru ieșire.

Considerând expresia $\overline{a \cdot b} + c$ se va realiza întâi operația ȘI-NU între variabilele a și b , urmată de operația SAU a rezultatului anterior cu variabila c . În calcule se vor utiliza tabelele de adevăr ale fiecărei operații în parte. Conform acestei metodologii valorile intermediare și finale sunt prezentate în Tabelul 2. 4.

Tabelul 2. 3 Prioritatea operațiilor în algebra booleană

Operator	Prioritate
()	mare
—	
•	
\oplus, \ominus	
+	
	mică

Tabelul 2. 4 Calculul tabelului de adevăr pentru expresia $\overline{a \cdot b} + c$

a b c	$\overline{a \cdot b} + c$		a b c	$\overline{a \cdot b} + c$
0 0 0 :	$\overline{0 \cdot 0} + 0 = 1 + 0 = 1$		0 0 0	1
0 0 1 :	$\overline{0 \cdot 0} + 1 = 1 + 1 = 1$		0 0 1	1
0 1 0 :	$\overline{0 \cdot 1} + 0 = 1 + 0 = 1$		0 1 0	1
0 1 1 :	$\overline{0 \cdot 1} + 1 = 1 + 1 = 1$		0 1 1	1
1 0 0 :	$\overline{1 \cdot 0} + 0 = 1 + 0 = 1$		1 0 0	1
1 0 1 :	$\overline{1 \cdot 0} + 1 = 1 + 1 = 1$		1 0 1	1
1 1 0 :	$\overline{1 \cdot 1} + 0 = 0 + 0 = 0$		1 1 0	0
1 1 1 :	$\overline{1 \cdot 1} + 1 = 0 + 1 = 1$		1 1 1	1

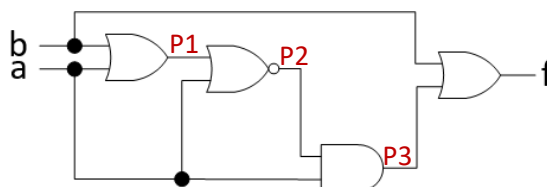
2.5 Realizarea schemei logice pentru o expresie

Conectarea porților în cadrul schemei logice este strict legată de ordinea operațiilor din expresia care trebuie implementată.

De exemplu, considerând funcția $f = b + a \cdot \overline{a + (b + a)}$ și ținând cont de prioritatea operațiilor (din Tabelul 2. 3) rezultă următorii pași de efectuare a operațiilor:

1. $P1 = b + a$ (operația SAU din paranteză)
2. $P2 = \overline{a + P1}$ (operația SAU-NU)
3. $P3 = a \cdot P2$ (operația ȘI)
4. $f = b + P3$ (operația SAU)

Dacă amplasarea porților se face de la stânga la dreapta, dinspre intrări spre ieșiri, atunci schema logică rezultată este prezentată în Figura 2. 2.

Figura 2. 2 Schema logică pentru funcția $f = b + a \cdot \overline{a + (b + a)}$

2.6 Proprietăți ale algebrei booleene

Două funcții booleene sunt considerate echivalente, dacă pentru toate combinațiile de intrare au rezultate identice.

Folosind tabelele de adevăr ale operațiilor de bază se pot demonstra următoarele proprietăți (egalități):

- Comutativitate: $a + b = b + a$; $a \cdot b = b \cdot a$
- Asociativitate: $(a + b) + c = a + (b + c)$; $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Distributivitate: $a + (b \cdot c) = (a + b) \cdot (a + c)$; $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
- Element neutru: $a \cdot 1 = a$; $a + 0 = a$
 - Consecințe ale elementului neutru: $a + 1 = 1$; $a \cdot 0 = 0$
- Idempotență: $a + a = a$; $a \cdot a = a$
- Dubla negație: $\bar{\bar{a}} = a$
- Operația XOR: $a \oplus b = \bar{a} \cdot b + a \cdot \bar{b}$
- Operația XNOR: $a \odot b = a \cdot b + \bar{a} \cdot \bar{b}$
- De Morgan: $\overline{a + b} = \bar{a} \cdot \bar{b}$; $\overline{a \cdot b} = \bar{a} + \bar{b}$

2.7 Transformări ale porților logice și generarea de noi funcționalități

Se pot realiza diverse tipuri de porți din alte tipuri folosind proprietățile algebrei booleene. Un exemplu elocvent îl reprezintă reducerea numărului de intrări la porțile ȘI, ȘI-NU, SAU și SAU-NU. O poartă ȘI cu 4 intrări care realizează expresia $a \cdot b \cdot c \cdot d$, poate fi redusă la 2 intrări $a \cdot b$ în mai multe feluri, conform echivalențelor: $a \cdot b = a \cdot 1 \cdot b \cdot 1 = a \cdot b \cdot a \cdot b = a \cdot b \cdot b \cdot b = a \cdot a \cdot a \cdot b = \dots$, etc. Valoarea constantă 1 reprezintă sursa de alimentare a circuitului (5V) și poartă denumirea de VCC (Voltage Common Collector). Ea este disponibilă în lista de simboluri a utilitarului Project Navigator, în categoria *General*. Câteva soluții pentru implementarea unei porți ȘI cu 2 intrări, dintr-o poartă ȘI cu 4 intrări sunt prezentate în Figura 2. 3. Soluțiile sunt similare în cazul porții ȘI-NU.

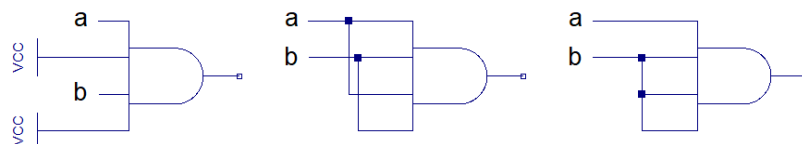


Figura 2. 3 Variante de transformare a unei porți ȘI cu 4 intrări, în poartă ȘI cu 2 intrări

Expresiile echivalente în cazul porții SAU implică folosirea constantei 0, denumită GND (Ground), adică masa circuitului (0V): $a + b = a + 0 + b + 0 = a + b + a + b = a + b + b + b = a + a + a + b = \dots$, etc. O parte din soluții sunt prezentate în Figura 2. 4 și sunt identice și pentru poarta SAU-NU. Simbolul GND se află în categoria *General*.

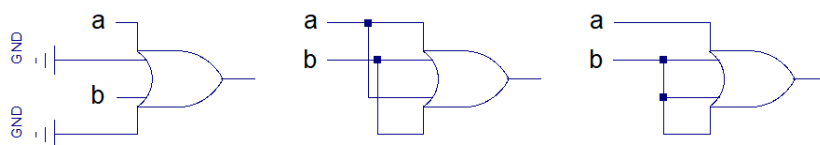


Figura 2. 4 Variante de transformare a unei porți SAU cu 4 intrări, în poartă SAU cu 2 intrări

O poartă inversor se poate realiza dintr-o poartă ȘI-NU sau o poartă SAU-NU cu 2 intrări conform expresiilor: $\bar{a} = \overline{a \cdot 1} = \overline{a \cdot a} = \overline{a + a} = \overline{a + 0}$ (Figura 2. 5).



Figura 2. 5 Variante de realizare a unei porți NOT din alte porți

Folosind relațiile De Morgan, o poartă SAU se poate realiza cu ȘI-NU și inversoare, iar o poartă ȘI se poate realiza cu SAU-NU și inversoare: $a \cdot b = \overline{\bar{a} + \bar{b}}$, $a + b = \overline{\bar{a} \cdot \bar{b}}$ (Figura 2. 6).

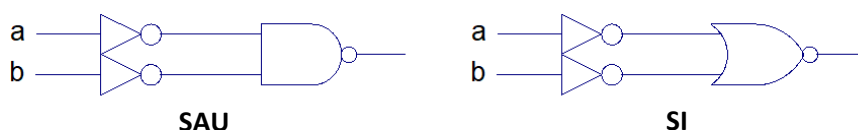
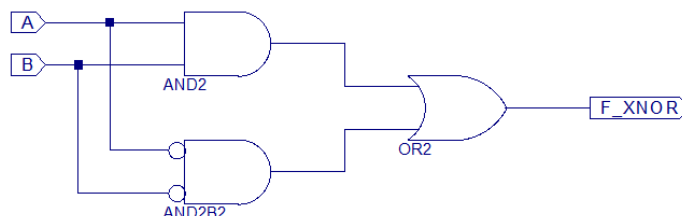


Figura 2. 6 Alternative de implementare a porților SAU (stânga) și ȘI (dreapta)

2.8 Activități practice

1. Implementați în cadrul proiectului *ttl_env*, într-o singură schemă, porțile fundamentale NOT, NOR și XOR cu 2 intrări și testați funcționalitatea acestora pe placă folosind tabelele de adevăr din Tabelul 2. 1.
2. Adăugați la proiect schema din figura următoare, care implementează expresia $a \cdot b + \bar{a} \cdot \bar{b}$ și testați pe placă faptul că această expresie are funcționalitatea porții XNOR folosind tabelul de adevăr al acesteia (din Tabelul 2. 1).



3. Pentru fiecare din expresiile următoare determinați tabelul de adevăr, adăugați câte o schemă corespunzătoare în cadrul proiectului și testați funcționalitatea pe placă cu tabelul de adevăr. Expresiile nu vor fi modificate folosind proprietățile algebrei booleene. Completați cerințele pentru fiecare punct, înainte de a trece la următorul.

- a. $\bar{a} \cdot b + \bar{c}$
- b. $a \cdot c + \overline{b \cdot c}$
- c. $a + \overline{b \cdot c}$
- d. $\overline{a + \bar{b} \cdot c}$
- e. $\overline{(a + b) \cdot \bar{a} + c}$
- f. $a + \overline{b \cdot \bar{a} + c}$