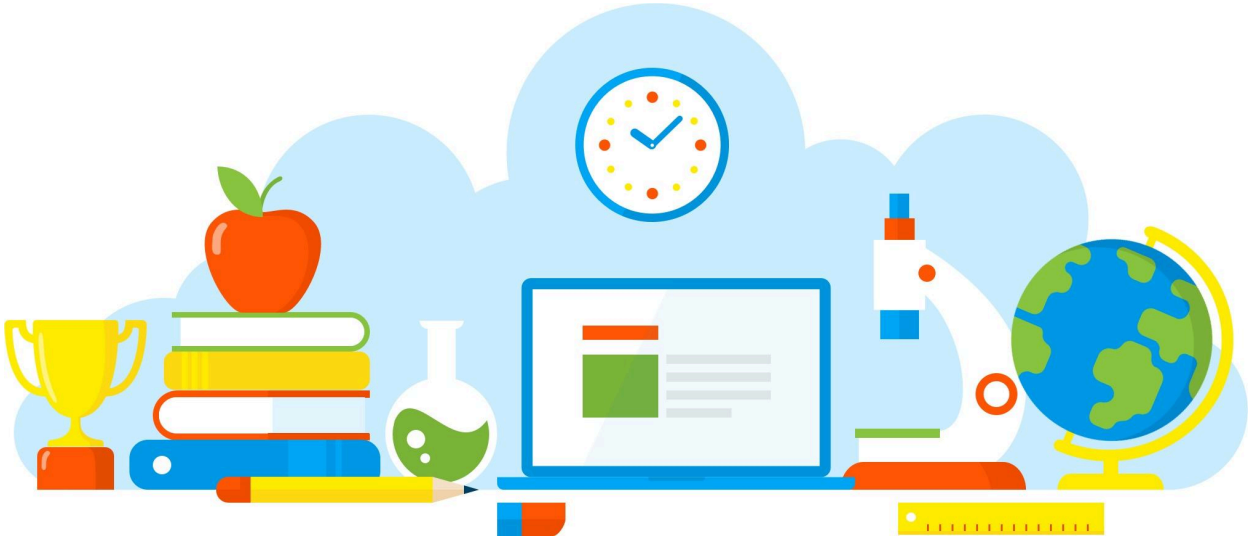


Arhitectura Calculatoarelor

Fetch, Decode, Execute

Laborator 2
2019

Circuitele combinaţionale din structura unui microprocesor

În această lucrare de laborator se va prezenta câteva aspecte la modul general cu privire la circuitele combinaţionale utilizate la construcţia unui microprocesor.

În figura 1 se prezintă structura internă a unui microprocesor şi s-a scos în evidenţă prin colorare părţile care conţin circuite combinaţionale. Blocul de logică pentru stările următoare, logica de ieşire, multiplexoarele, unitatea ALU şi circuitele de tampon sunt doar câteva exemple ale părţilor constituente.

Înainte de a proiecta aceste părţi constituente ale microprocesorului, va trebui să fim capabili în a optimiza circuitele proiectate în sensul spaţiului ocupat, al consumului de putere, al vitezei de procesare, şi într-un final să testăm funcţionalitatea microprocesorului creat.

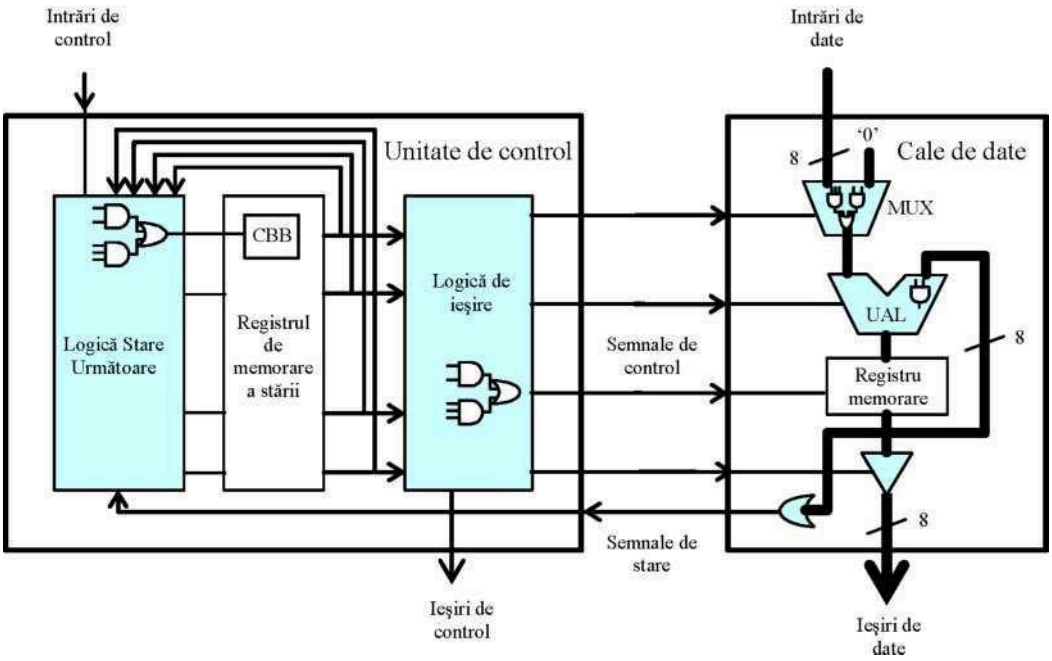


Figura 1. Structura internă a unui microprocesor

Maparea în funcţie de tehnologie

Pentru a reduce costurile de implementare in tehnologie a producţiei circuitelor numerice, proiectanţii pun la dispoziţie porţi logice dedicate (custom design) cu care se poate implementa pe un sigur chip structura circuitului numeric dorit. Multe arii logice prezintă o structură doar din porţi de tipuri SI-NU sau SAU-NU a căror intrări si ieşiri nu

sunt conectate, conexiunile urmând a fi realizate ulterior de către utilizator. Pentru a utiliza aceste porţi logice, proiectantului ii rămâne sarcina de a decide unde si cum sa conecteze aceste intrări respectiv ieşiri ale porţilor logice din structura ariei logice. Problema fundamentală care rămâne de rezolvat la proiectare este că în momentul în care se doreşte implementarea circuitului, trebuiesc convertite toate porţile SI, SAU si NU în porţi logice de tipul SI-NU sau SAU-NU. Mai trebuie de reţinut şi faptul că de obicei aceste tipuri de porţi logice (SI-NU sau SAU-NU) pot conţine un număr fix de intrări (de obicei trei intrări).

În continuare se va prezenta modul cum se pot obţine porţile logice de tip SI, SAU şi NU prin utilizarea de porţi SI-NU sau SAU-NU

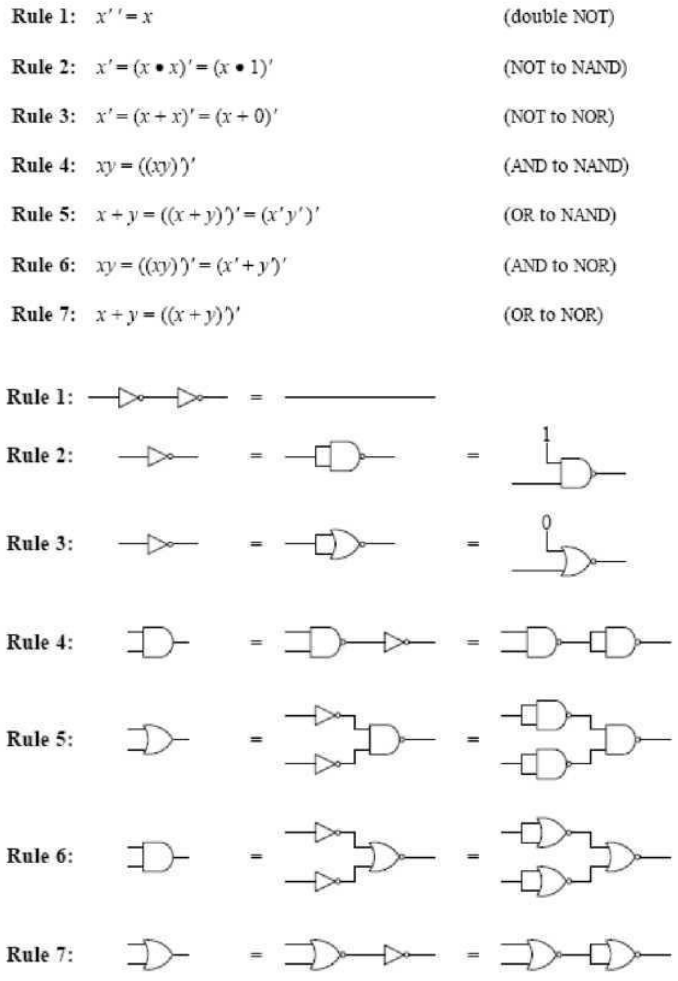


Figura 2. Circuite pentru conversia porţilor SI, SAU si NU în porţi de tipul SI-NU sau SAU-NU

Regula 1 spune ca o negaţie dublă poate fi eliminată din structura circuitului. Regulile 2 şi 3 convertesc o poartă NU la o poartă SI-NU sau SAU-NU. Regula 4 aplică regula 1 pe o poartă SI. Regula 5 schimbă o poartă SAU iar regula 6 converteşte o poartă SI.

Un alt lucru care se doreşte a fi implementat cu ajutorul a unei porţi logice de tipul SI-NU cu trei intrări (sau o poarta SAU-NU cu trei intrări) este obţinerea unei porţi logice SI-NU cu două intrări (sau a unei porţi logice de tip SAU-NU cu doua intrări). Regulile următoare fac posibilă această abordare:

Rule 8:	$(x \bullet y)' = (x \bullet y \bullet y)'$	(2-input to 3-input NAND)
Rule 9:	$(x + y)' = (x + y + y)'$	(2-input to 3-input NOR)
Rule 10:	$(abc)' = ((ab)\, c)' = ((ab)''\, c)'$	(3-input to 2-input NAND)
Rule 11:	$(a+b+c)' = ((a+b) + c)' = ((a+b)'' + c)'$	(3-input to 2-input NOR)

Unitatea de control este bazată pe o „maşină cu stări finite” cunoscută în limba engleză sub denumirea de Finite State Machine - FSM. Această maşină execută plecând dintr-o stare în altă stare, un număr finit de stări care trebuie să fie parcurse. În general o unitate de control este realizată din trei părţi principale, astfel: logica care conţine stările următoare ce urmează a fi executate, memoria de stare şi o logică de ieşire care de obicei produce un set de semnale necesare controlului unităţii de date. Rolul memoriei de stare este de a păstra starea curentă care se execută la un moment dat în maşina de stare.

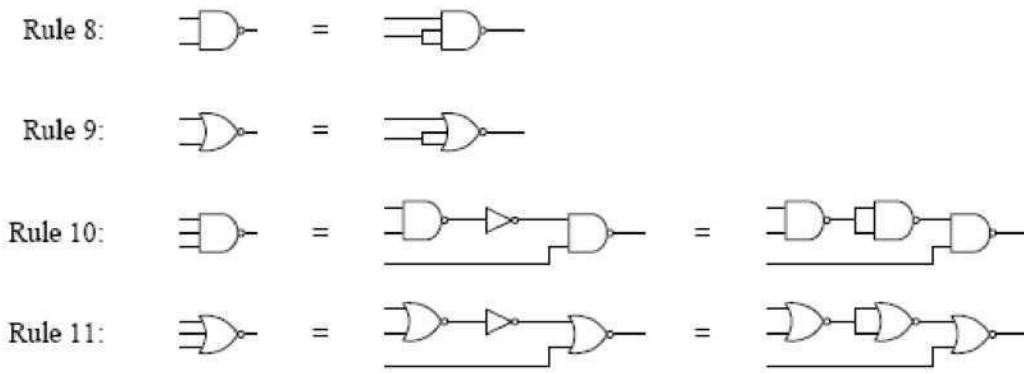
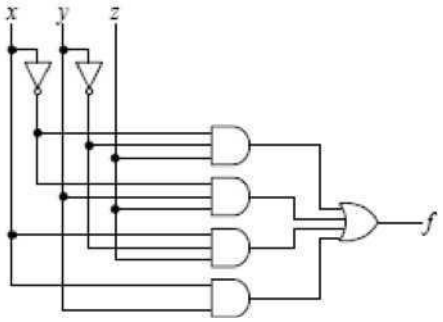


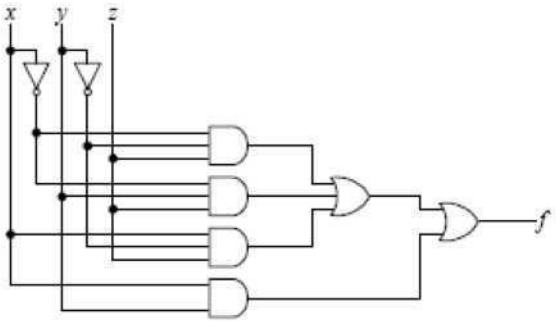
Figura 3. Circuite pentru conversia porţilor SI-NU, SAU-NU

Exemplu:

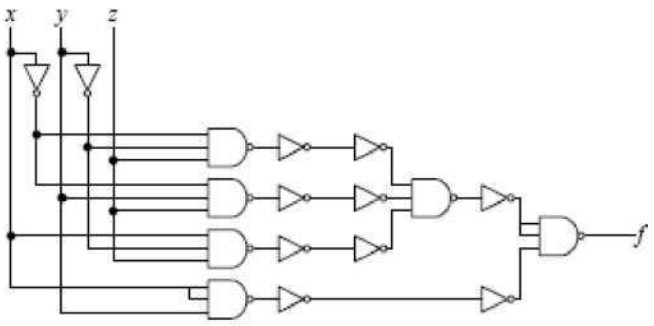
Fie circuitul numeric din figura de mai jos



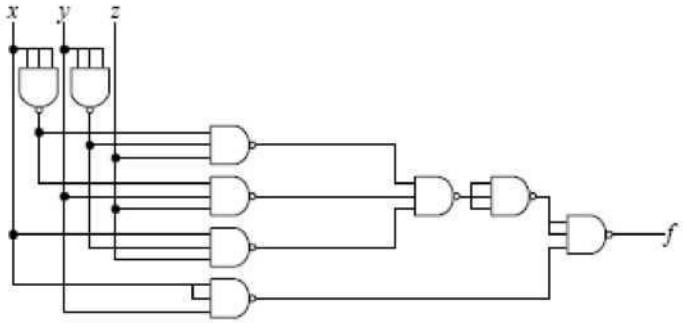
Prima dată vom face o schimbare asupra porţii logice de tip SAU cu patru intrări:



Apoi vom utilizeze regula 4 pentru a schimba toate porţile de tip SI în porţi logice de tip SI-NU cu trei intrări plus inversoare şi regula 5 pentru a schimba toate porţile logice de tipul SAU in porţi logice de tipul SI-NU cu trei intrări cu inversoare.



În final vom elimina toate inversoarele şi vom înlocui inversoarele rămase în circuit cu porţi SI-NU cu intrările conectate împreună.

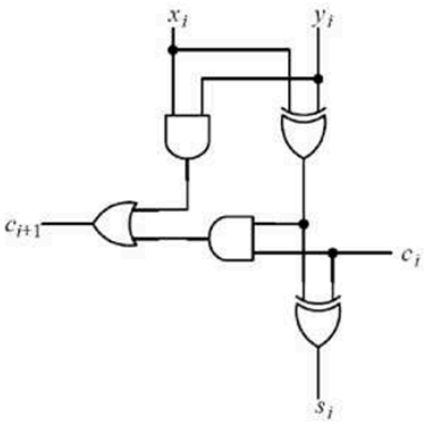


Componentele combinaţionale din structura unui microprocessor

După cum a fost precizat şi în laboratorul numărul unu, componentele combinaţionale sunt realizate cu ajutorul circuitelor combinaţionale. Componentele combinaţionale standard pot fi incluse în librării pentru a putea fi utilizate şi în alte proiecte. Câteva exemple de componente combinaţionale standard pot fi: circuit sumator/scăzător, unitate. ALU, multiplexor, circuit tampon de tip tri-state, decodificatoare, codificatoare, comparatoare, registri de deplasare/rotire şi multiplicatoarele.

Circuitul sumator elementar (Full Adder - FA)

În figura 4 se prezintă schema structurală a circuitului sumator.



xi	yi	ci	ci+1	si
0	0	0	0	0
0	0	1	0	1
0	i	0	0	1
0	i	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

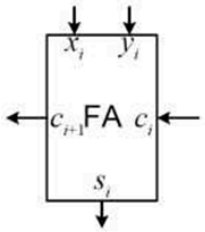


Figura 4. Circuit de sumare pe un bit

Circuit de sumare cu transport *(Ripple-Carry Adder)*

In figura 5 se prezintă circuitul de sumare a doi operanzi pe 4 biţi.

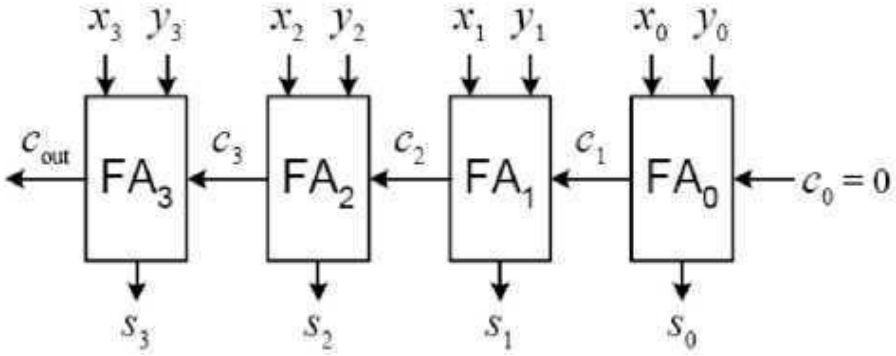
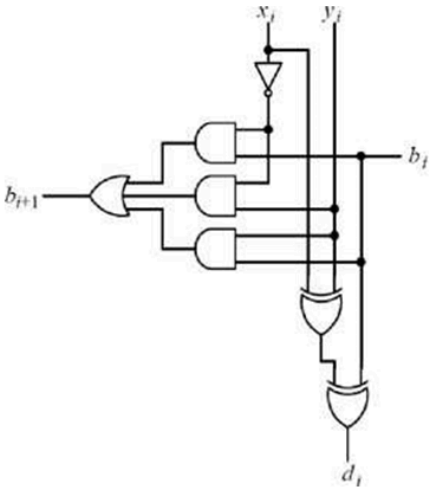


Figura 5. Circuit de sumare a doi operanzi pe 4 biţi cu transport

Circuit de scădere (Full Subtractor - FS)

În figura 6 se prezintă structura unui circuit de scădere.



xi	yi	bi	bi+1	di
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

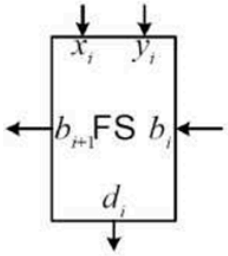


Figura 6. Circuit de scădere pe un bit

Circuit de adunare-scădere

În figura 7 se prezintă un circuit de adunare-scădere care lucrează cu operanzi pe 4 biţi.

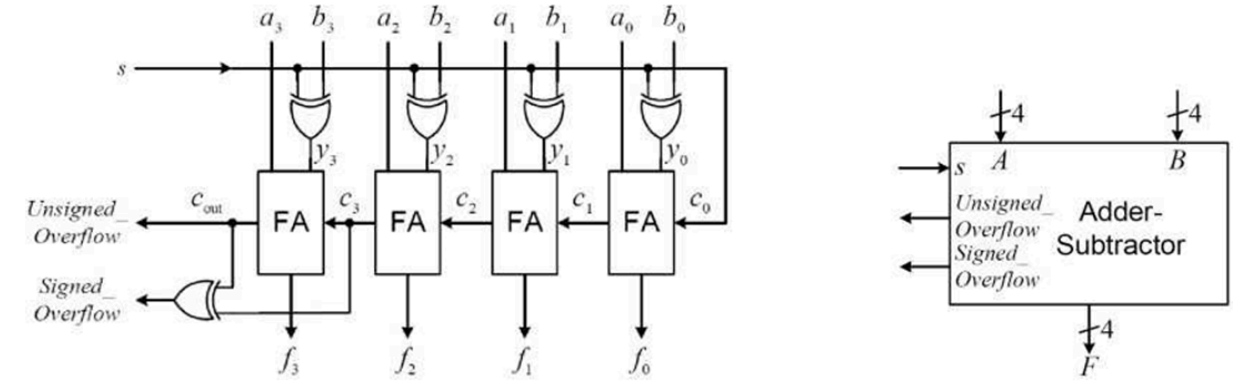


Figura 7. Circuit de adunare-scădere pe 4 biţi

Tabelul de adevăr asociat este următorul:

s	Function	Operation
0	Add	$F = A + B$
1	Subtract	$F = A + B' + 1$

s	b_i	y_i	c_0
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

Probleme

1) Să se realizeze schemele aferente circuitelor de mai jos în MaxPlus II:
   a) sumatorul elementar;
   b) sumatorul cu transport pentru operanzi pe 8 biţi;
   c) scăzătorul pe un bit;
   d) circuitul de adunare/scădere pentru operanzi pe 8 biţi.
2) Să se realizeze simulările funcționale ale circuitelor proiectate la punctul 1).