

1 INTRODUCERE

1.1 Analogic versus numeric

O posibilă clasificare a sistemelor electronice se poate face după natura semnalelor prelucrate. **Sistemele analogice** se ocupă de prelucrarea semnalelor care au o **variație continuă** în timp, iar modelul cel mai simplu de semnal este semnalul pur sinusoidal:

$$s(t) = A \sin \omega t, t \in (-\infty, +\infty)$$

Sistemele numerice (sau **digitale**) se ocupă de prelucrarea semnalelor care au o **variație discontinuă** în timp, modelul de semnal fiind cel al impulsului dreptunghiular:

$$u(t) = \begin{cases} A & \text{pentru } t \in [t_1, t_2] \\ 0 & \text{pentru } t \notin [t_1, t_2] \end{cases}$$

Ambele semnale sunt modele idealizate. Primul presupune absența absolută a unei discontinuități, iar cel de-al doilea definește tranziții în timp nul. Faptul că aceste semnale nu sunt fizic realizabile arată formalismul conceptelor de continuu și discontinuu, concepte "utile pentru delimitări teoretice care, practic, au sens prin tehnicile distincte pe care le presupun, dar nu prin ceea ce în esență ar vrea să semnifice". ([Ștefan, 1993]).

Pornind de la cel de-al doilea tip de semnal se poate defini o **electronică a discontinuului** care să opereze cu semnale oricât de complexe obținute prin repetarea impulsului dreptunghiular. În figura 1.1a este reprezentată funcția $u(t)$, care aproximează semnalul real reprezentat în figura 1.1b. Realitatea impune tranziții în intervale de timp diferite de zero (t_{LH} , t_{HL}), iar atingerea **nivelelor** de **zero** sau A nu se poate face direct, ci prin aproximări succesive ce se stabilizează după un anumit interval de timp. Se consideră că tranziția pozitivă începe când mărimea semnalului este 10% din A și se termină la atingerea valorii de 90% din A . Intervalul de timp scurs între cele două repere se numește timp de

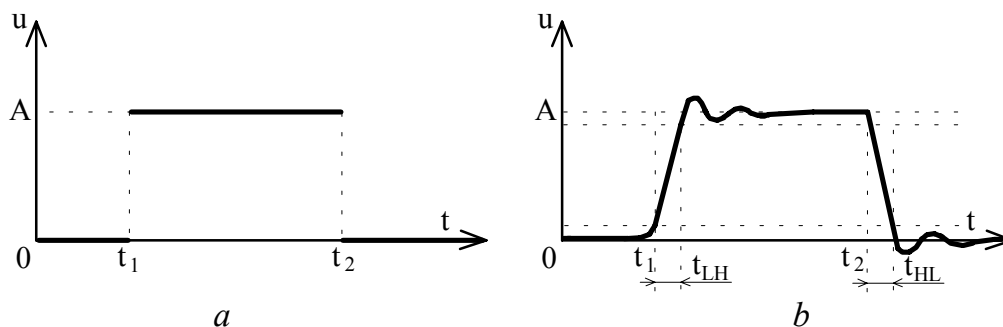


Fig. 1.1 Un semnal numeric real și modelul său

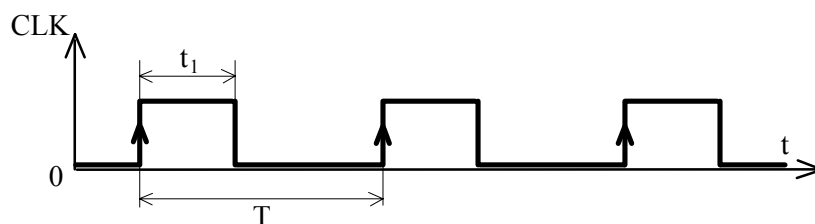


Fig. 1.2 Semnal de ceas

front pozitiv, sau **tim de creștere** (rise time) și se notează cu t_{LH} (time from LOW to HIGH). Aceleași repere sunt folosite și pentru definirea frontului negativ, sau a **timului de descreștere** (fall time), notat cu t_{HL} (time from HIGH to LOW). Mărimea supracreșterilor se exprimă procentual față de A .

Discontinuitatea semnalului se manifestă atât în **tim** cât și în **mărime**. Timpul este marcat discontinuu printr-un semnal periodic dreptunghiular denumit de regulă **ceas**, sau **tact** și notat prescurtat prin CLK (de la CLOCK). Acest semnal, care este reprezentat în figura 1.2, este caracterizat prin **perioada** de repetiție T , prin **factorul de umplere** t_1/T și prin **frontul activ** (crescător în reprezentarea noastră). Orice eveniment într-un **sistem discret sincron** este o consecință a frontului activ al semnalului de ceas. Timpul devine astfel o **mărime discretă**. Într-un sistem numeric care nu are un semnal de ceas aplicat pe intrare comutările între cele două nivele sunt **asincrone** și ansamblul este considerat **asincron**, variabila timp fiind o **mărime continuă**.

Mărimile semnificative ale semnalului sunt numai două: **zero** și A . Dacă semnalul este generat de un circuit electronic real, atunci nivelului de zero îi corespunde o tensiune de valoare foarte mică, apropiată de potențialul masei, iar nivelului A îi corespunde o tensiune mai mare, apropiată mai mult sau mai puțin de valoarea tensiunii de alimentare a circuitului. Deoarece aceste tensiuni pot varia între anumite limite, se iau în considerație un nivel **maxim** pentru zero și unul **minim** pentru A . Sistemele discrete acceptă **corespondența semnal-simbol**, adică se pot asocia celor două nivele ale semnalului $u(t)$ **valori logice**. Se obișnuiește să se asocieze nivelul **zero** cu **numărul binar 0** (fals) și nivelul A cu **numărul binar 1** (adevărat). Se spune că $u(t)$ comută la momentul t_1 din 0 în 1, sau din **zero logic** în **unu logic**, această din urmă exprimare introducând o redundanță, de multe ori necesară.

Pornind de la aceste corespondențe semnal - simbol, circuitele care prelucrează astfel de semnale sunt numite **circuite logice** sau **circuite numerice** (digitale). Fiecare ieșire a unui circuit logic furnizează un semnal de forma semnalului $u(t)$, adică una dintre cele două valori logice 0 sau 1, deci o informație de **un bit** (Binary Digit).

1.2 Apariția și dezvoltarea sistemelor numerice

Implementarea circuitelor numerice s-a făcut de fiecare dată cu mijloacele oferite de tehnologia momentului respectiv. Formalismul matematic care fundamentează logica binară este **algebra booleană**, elaborată de matematicianul englez George Boole în prima jumătate a secolului trecut. Creată ca o simplă abstracțiune matematică, ea nu este aplicată în practică până când Claude Shannon, pe atunci student la Massachusetts Institute of Technology, a arătat că este posibilă construirea unor circuite electrice echivalente cu expresii din algebra booleană, folosind relee de comutare. Ulterior au fost construite sisteme numerice realizate cu tuburi electronice (calculatorul ENIAC, construit în 1946, conținea 18000 tuburi electronice, 7500 relee și 7000000 rezistențe), cu tranzistoare și în cele din urmă cu circuite integrate ([Toacșe, 1996], [Mihalcea, 1992]).

Primul **circuit integrat** a fost inventat în mod independent de Robert Noyce (firma Fairchild) și Jack Kilby (Texas Instruments). Ambii au avut ideea combinării mai multor tranzistoare, diode și rezistențe pe aceeași pastilă de semiconductor. Noyce a creat un circuit monolitic pe siliciu, iar Kilby a construit mai degrabă un circuit hibrid pe germaniu.

Cu timpul, complexitatea circuitelor integrate numerice a crescut considerabil. O măsură a complexității sistemelor numerice este numărul de tranzistoare folosite în circuitele integrate din sistem. Gordon Moore, unul din colaboratorii lui Noyce și cofondator al firmei INTEL, a observat după primii 5 - 6 ani de dezvoltare a tehnologiei circuitelor integrate că numărul de tranzistoare integrate într-un singur cip se dublează la fiecare 18 luni. Această regulă a fost strict respectată în ultimii 30 de ani, fiind cunoscută ca **legea lui Moore**. Experții consideră că legea lui Moore va rămâne funcțională încă cel puțin 15 - 20 de ani de acum încolo. Figura 1.3 reprezintă evoluția numărului de tranzistoare pe cip, începând cu anul apariției primului circuit integrat și până la apariția procesorului Pentium 4 (circa 42 milioane de tranzistoare). Celelalte puncte de pe grafic marchează alte momente importante când au apărut diverse microprocesoare din familia Intel - x86: 286, 386, Pentium II sau Pentium III. Se poate estima că în mai puțin de 10 ani se va depăși probabil numărul de 1 miliard de tranzistoare pe cip ([Gwennap, 1996], [Hummel, 1996]).

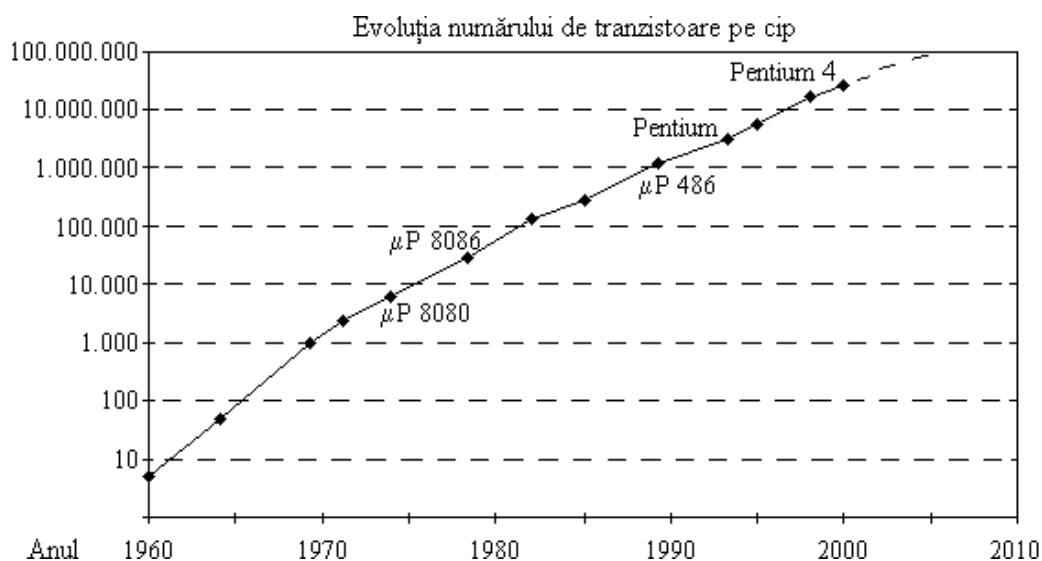


Fig. 1.3 Numărul de tranzistoare pe cip se dublează la aproximativ 18 luni

Creșterea complexității structurilor integrate a permis implementarea unor funcții tot mai complexe, care erau însă strict atașate structurilor inițiale, structuri care primeau la intrări numai șirul de biți ce trebuia prelucrat. Aceste structuri numerice se numesc **structuri funcționale**. O structură alternativă admite la intrare și un șir de biți prin care-i este definită funcția. În acest mod structura poate implementa **orice** funcție. Actualizarea funcțională se face printr-un șir de simboluri numit, de regulă, **program**, iar structura se numește **mașină universală**.

Structurile funcționale sunt adecvate scopului urmărit, atât în privința complexității structurale cât și a performanțelor de viteză. Mașinile universale au o maximă flexibilitate funcțională care se poate adapta unor probleme concrete în condiții în care nu întotdeauna se pot atinge performanțe maxime. Universalitatea se plătește prin creșterea complexității și limitarea vitezei de execuție ([Ștefan, 1993]).

Este interesant că evoluția tehnologică repune periodic în discuție oportunitatea soluțiilor oferite de cele două puncte de vedere. Revoluția indiscutabilă care a marcat începutul erei microprocesorului, odată cu celebra invenție din 1971 a lui M. E. Hoff de la firma INTEL - microprocesorul 4004 de 4 biți, reprezintă un moment istoric în care conceptul mașinii universale a avut câștig de cauză. Etapa prin care trecem acum tinde să impună structurile funcționale datorită performanțelor și flexibilităților tehnologice oferite.

Cert este că, cel puțin până la ora actuală, tehnologiile disponibile depășesc gândirea structurală a proiectanților de sisteme numerice. Este încă justificată afirmația din 1982 a profesorului Edward A. Feigenbaum de la Universitatea Stanford: "Hardware engineers are concept limited, not component limited". Gândirea structurală în noul context funcțional nu este încă la nivelul tehnologiilor disponibile ([Ștefan, 1991]).

1.3 Clasificări

Clasificarea circuitelor integrate numerice devine o sarcină tot mai dificilă datorită complexității structurale și funcționale tot mai mari. O clasificare după **numărul de tranzistoare pe capsulă** este propusă în tabelul 1.1. Circuitele integrate au fost împărțite în 5 categorii, dar limitele care separă o categorie de alta sunt discutabile, ele depinzând de momentul de timp la care s-a făcut clasificarea. Dacă acum 15-20 ani structurile VLSI se limitau la circa 50000 tranzistoare, astăzi probabil că și valoarea de 1000000 tranzistoare este prea mică. De exemplu, microprocesorul 486 care conține 1,2 milioane de tranzistoare este mai degrabă un circuit VLSI decât unul ULSI.

O clasificare după **tehnologia de realizare** a circuitelor integrate digitale este propusă în tabelul 1.2. Informația prezentată este utilă pentru a face o comparație între performanțele diverselor familii logice importante. Valorile tipice date pentru timpii medii de propagare și pentru consumul mediu de putere pe poartă pot diferi în funcție de frecvența de lucru sau de valoarea tensiunii de alimentare. Consumul mediu de putere pe poartă este dat de produsul dintre tensiunea de alimentare și curentul mediu consumat de o poartă logică din familia respectivă. Pentru calculul timpului mediu de propagare se folosește relația:

$$t_p = \frac{t_{pLH} + t_{pHL}}{2},$$

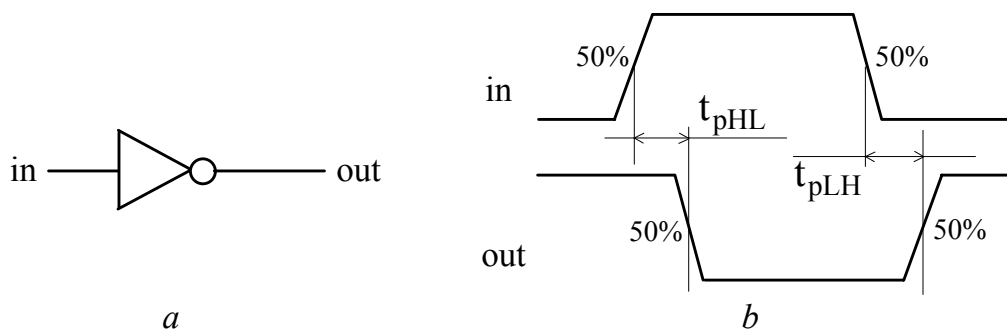
Tabelul 1.1

Denumire	Notăție	Anul apariției	Tehnologie	Dimensiune tranzistor	Număr de tranzistoare	Exemple
Small Scale Integration	SSI	1960	bipolară	10 μm	≤ 50	porți logice
Medium Scale Integration	MSI	1965	bipolară MOS	8 μm	≤ 500	MUX, DMUX, registre
Large Scale Integration	LSI	1970	bipolară MOS	5 μm	≤ 10000	memorii, μP Z80
Very Large Scale Integration	VLSI	1980	bipolară MOS	1,5 - 3 μm	≤ 1000000	memorii, $\mu\text{control}$, μP 486
Ultra Large Scale Integration	ULSI	1990	bipolară MOS	0,13 - 1 μm	> 1000000	calculator, transputer Pentium II, III, 4

unde, t_{pLH} reprezintă timpul scurs din momentul modificării semnalului la intrarea porții logice și până la modificarea răspunsului la ieșire în sensul tranziției din 0 logic (LOW) în 1 logic (HIGH), iar t_{pHL} reprezintă timpul scurs din momentul modificării semnalului la intrare și până la tranziția din 1 logic (HIGH) în 0 logic (LOW) a ieșirii.

În exemplul din figura 1.4a s-a considerat o poartă inversoare. Semnalul aplicat pe intrarea *in* generează un răspuns furnizat la ieșirea *out*, ca în figura 1.4b. Reperele care marchează limitele timpilor de propagare au fost fixate prin convenție la 50% din valoarea tensiunii corespunzătoare palierului de 1 logic. Atragem atenția că este posibil ca palierul de 1 logic pentru cele două semnale să fie dat de valori diferite ale tensiunii.

Primul circuit integrat logic obținut de Noyce a fost realizat în tehnologie RTL. În 1962 a apărut poarta DTL, care implementa o logică cu diode cuplate cu un tranzistor inversor de ieșire. Înlocuirea diodelor cu un tranzistor multiemitor și introducerea unui etaj de ieșire cu tranzistoare în contratimp a generat o structură reușită, care a fost standardizată în 1964 de firma TEXAS INSTRUMENTS sub numele de familia logică TTL. Această structură a dominat timp de peste două decenii familiile logice bipolare, cunoscând perfecționări continue în tot acest timp (vezi tabelul 1.2).

Fig. 1.4 Definierea timpilor de propagare t_{pLH} și t_{pHL}

Tabelul 1.2

Tehnologia	Familia logică	Notăție	Timpi de propagare	Consum pe poartă logică	Faza de dezvoltare
bipolară	Diode Transistor Logic	DTL	30 ns	9 mW	desuet
	Resistor Transistor Logic	RTL	12 ns	20 mW	desuet
	TransistorTransistorLogic	TTL	9 ns	10 mW	declin
	High Speed TTL	HTTL	6 ns	16 mW	declin
	Low Power TTL	LTTL	33 ns	1 mW	declin
	Schottky TTL	STTL	3 ns	20 mW	declin
	LowPower Schottky TTL	LSTTL	10 ns	2 mW	declin
	Advanced Schottky TTL	ASTTL	2 ns	20 mW	declin
	Advanced LSTTL	ALS	4 ns	1 mW	maturitate
	Fast TTL	FTTL	3 ns	5 mW	maturitate
	Emitter Coupled Logic	ECL	2 ns	25 mW	declin
	Integrated Injection Logic	I ² L	10 ns	1 mW	declin
MOS	N channel MOS	NMOS	30 ns	1 mW	desuet
	P channel MOS	PMOS	250 ns	1 mW	desuet
	CMOS	CD4000	100 ns	1 μW	declin
	High Speed CMOS	HC/T	8 ns	10 μW	maturitate
	Advanced CMOS	AC/T	3 ns	50 μW	creștere
	Low Voltage CMOS	LVC	4 ns	10 μW	acceptare
	Advanced LVC	ALVC	2 ns	20 μW	acceptare
hibridă	Low Voltage Technology	LVT	2 ns	0,08/2 mW	creștere
	Advanced BICMOS Tech	ABT	1,5 ns	0,1/20 mW	creștere
	Adv.LowPower BICMOS	ALB	1 ns	0,3/2 mW	acceptare

Primul circuit integrat MOS apare în 1962. Deși densitatea de integrare era superioară tehnologiilor bipolare, sensibilitatea la sarcini electrostatice, timpii prea mari de propagare și unele probleme de fiabilitate în realizarea oxidului de poartă au întârziat oarecum dezvoltarea structurilor integrate MOS. Abia după anul 1970 familiile logice MOS au început să se impună, mai ales după apariția familiei CMOS, care, în urma perfecționărilor continue pe care le-a cunoscut, rămâne deocamdată tehnologia prezentului și a viitorului apropiat.

Tehnologia BICMOS este o combinație între tehnologia bipolară și tehnologia CMOS, care îmbină avantajele de viteză și curent mare de ieșire al tehnologiei bipolare cu avantajele consumului redus și a densității mari de integrare oferite de tehnologia CMOS. Familiile Low Voltage sunt proiectate pentru tensiuni de alimentare reduse (3,3V în loc de 5V), în scopul reducerii consumului de putere și a creșterii densității de integrare. Reprezentarea din figura 1.5 ilustrează tendințele de declin ale tehnologiei bipolare, de maturizare a tehnologiei CMOS și de creștere a tehnologiilor BICMOS și CMOS de tensiune redusă ([TI, 1997]).

Tehnologii alternative non-silicon sunt cele în arseniură de galiu, GaAs, sau cele cu joncțiuni Josephson. Dispozitivele în GaAs au timpi de propagare sub 200ps și consum mediu de putere pe poartă de circa 0,2 mW, iar dispozitivele Josephson oferă timpi de propagare cuprinși între 1 și 10 ps, la puteri disipate cuprinse între 1 și 10 mW / poartă, dar funcționează la temperaturi foarte scăzute ([Toacșe, 1996]).

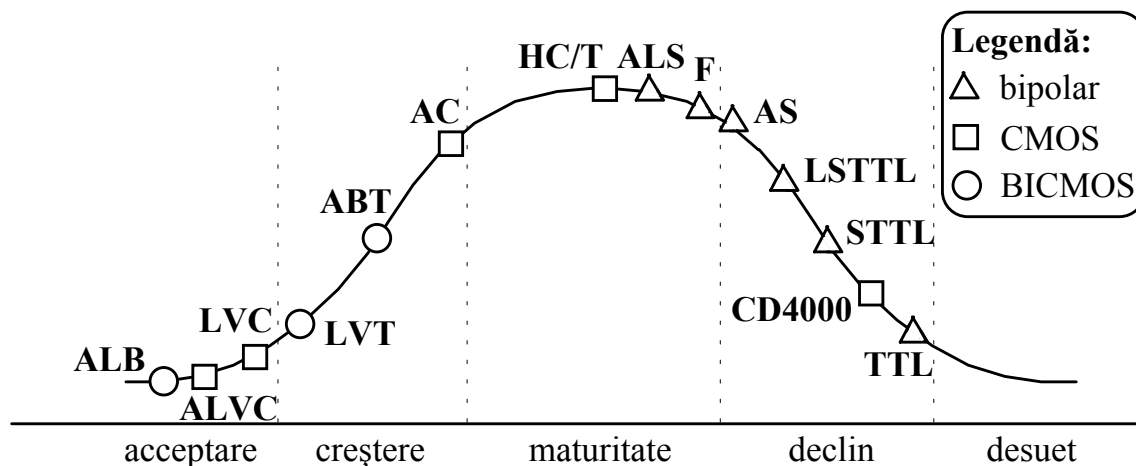


Fig. 1.5 Ciclul de viață a principalelor familii logice

O clasificare după **gradul de specificitate** al circuitelor folosite într-o aplicație este propusă în tabelul 1.3. Din acest punct de vedere se pot evidenția 3 categorii mari de circuite: circuite standard sau de uz general, circuite semidedicatate și circuite dedicate.

Circuitele **standard** acoperă întreaga gamă de complexitate de la SSI la ULSI, aria de siliciu este utilizată eficient deoarece aproape toate porțile logice existente pe cip sunt folosite în schema electrică a circuitului, iar eficiența economică depinde în mare măsură de aplicație. Când vorbim de eficiență economică ne gândim nu numai la costurile de fabricație, dependente și de numărul de circuite integrate din schemă, ci și la costurile de întreținere, dependente de consum, fiabilitate sau gabarit. Durata realizării sistemului depinde de complexitatea schemei, de tehnologia disponibilă pentru realizarea circuitului imprimat etc.

Circuitele **semidedicatate** includ structurile logice programabile, ariile de porți logice și celulele standard. **Structurile logice programabile** acoperă întreaga gamă de complexitate de la SSI la ULSI, dar utilizarea ariei de siliciu nu este de obicei eficientă. Acest aparent dezavantaj asigură însă pe de altă parte o arhitectură redundantă, care permite realizarea unor **structuri reconfigurabile**. Toate etapele de proiectare, analiză prin simulare logică și programare ale circuitului se fac pe calculator, folosind un suport **CAD** (**C**omputer **A**ided **D**esign) corespunzător. Timpul necesar pentru realizarea fizică a unui sistem numeric se reduce astfel la cel mult câteva zile. Exemple reprezentative sunt structurile **PAL**

Tabelul 1.3

Denumire		Comple - xitate	Utilizare resurse	Eficiență economică	Durata realizării	Exemple
circuite standard		SSI - ULSI	≈ 100 %	relativă	variabilă	porți, registre, memorii, μP
circuite semi - dedicate	logica program.	SSI - ULSI	1 - 100 %	foarte mare	ore, zile	PAL, GAL, FPGA
	arii de porți	MSI, LSI	40 – 60%	mare pt. serie mare	săpt., luni	MONOCIP
	celule standard	LSI - ULSI	100 %	mare pt. serie mare	luni	ZODIAC
circuite dedicate		LSI - ULSI	100 %	mare pt. serie mare	> 1 an	ceasuri, calculatoare

(**P**rogrammable **A**rray **L**ogic), **GAL** (**G**eneric **A**rray **L**ogic) sau **FPGA** (**F**ield **P**rogrammable **G**ate **A**rray). **Ariile de porți** logice conțin sute sau mii de porți care se interconectează conform dorinței beneficiarului în procesul de fabricație a circuitului integrat. MONOCIP este o arie de porți realizată în 1981 la ICCE București (conține sub 500 componente interconectate). **Celulele standard** sunt construite pe baza unei biblioteci de celule în care sunt definite o mulțime de blocuri funcționale SSI, MSI și LSI. Biblioteca de celule standard a sistemului ZODIAC conține 34 funcții logice (porți, bistabile, decodoare, numărătoare, registre, interfețe) și analogice (referințe, amplificatoare operaționale, oscilatoare, convertoare D/A de 6 biți etc.). Prin utilizarea software-ului pentru interconectarea acestor celule standard se poate defini funcționarea circuitului. La terminarea procesului, suportul CAD generează măștile pentru fabricarea circuitului. Aria de siliciu este folosită eficient, fiecare celulă fiind un modul proiectat separat, iar aria ocupată de canalele de interconectare este redusă la minimum. Cheltuielile sunt însă mai mari, deoarece trebuie proiectat un set complet de măști. Eficiența acestor structuri depinde de seria de fabricație. Pentru cantități foarte mari ele pot deveni mai ieftine decât ariile de porți. ([Olteanu, 1991]).

Circuitele dedicate (full custom) sunt proiectate practic de la zero și necesită cele mai mari cheltuieli și cel mai lung interval de timp pentru proiectare și producție. Asemenea abordare este eficientă numai în cazul producției de masă, în cantități foarte mari. Aria de siliciu este folosită în totalitate, iar prețul pe integrat poate fi destul de scăzut. Aplicațiile tipice includ ceasurile digitale, calculatoarele de buzunar sau industria de automobile.

Circuitele dedicate și cele semidedicat formează domeniul circuitelor **ASIC** (**A**pplication **S**pecific **I**ntegrated **C**ircuit), un domeniu aflat în plină expansiune. Dacă în 1980 circuitele ASIC nu reprezentau nici 20% din piața circuitelor integrate digitale, restul fiind ocupat de circuitele standard, în 1990 ele reprezintă peste 50% din piață, iar în viitorul apropiat vor avea peste 80 - 90% din totalul vânzărilor. Motivele acestei alegeri au fost deja prezentate: consum de putere, gabarit, fiabilitate, cost, viteză de procesare. Logica programabilă asigură și durata minimă de fabricație pentru produsul final.

Abordarea funcțională clasică în domeniul sistemelor numerice a impus o clasificare simplă a lor în **sisteme combinaționale** și **sisteme secvențiale**. **Sistemele combinaționale** se caracterizează prin faptul că fiecărei configurații binare de intrare îi corespunde o configurație binară de ieșire. Valorile logice ale ieșirilor sunt date numai de combinația binară aplicată pe intrări, iar variabila timp nu intervine practic în analiza circuitului. **Sistemele secvențiale** se caracterizează prin faptul că ieșirile depind de ordinea în care sunt aplicate la intrare configurațiile binare. Pentru aceeași configurație binară la intrare, ieșirile pot fi diferite, în funcție de secvența aplicată anterior intrării. Aceste circuite sunt circuite cu **memorie**, iar fiecare moment de timp este caracterizat de o **stare internă** a circuitului. Variabila timp, fie ea discretă sau continuă, intervine de această dată în funcționarea circuitului.

Exemplul 1.1 ([Ștefan, 2000])

Apăsarea tastei “=” a unui calculator de buzunar va produce un efect, pe sistemul de afișaj al calculatorului, care depinde de secvența unor tastări anterioare. Dacă anterior am tastat “2”, “+”, “4”, atunci efectul apăsării tastei “=” va fi afișarea cifrei 6. Dacă secvența de tastări anterioare a fost “7”, “-”, “4”, atunci efectul apăsării tastei “=” va fi afișarea cifrei 3. Deci calculatorul a răspuns unei *secvențe* date de comenzi, fiind așadar un sistem numeric secvențial.

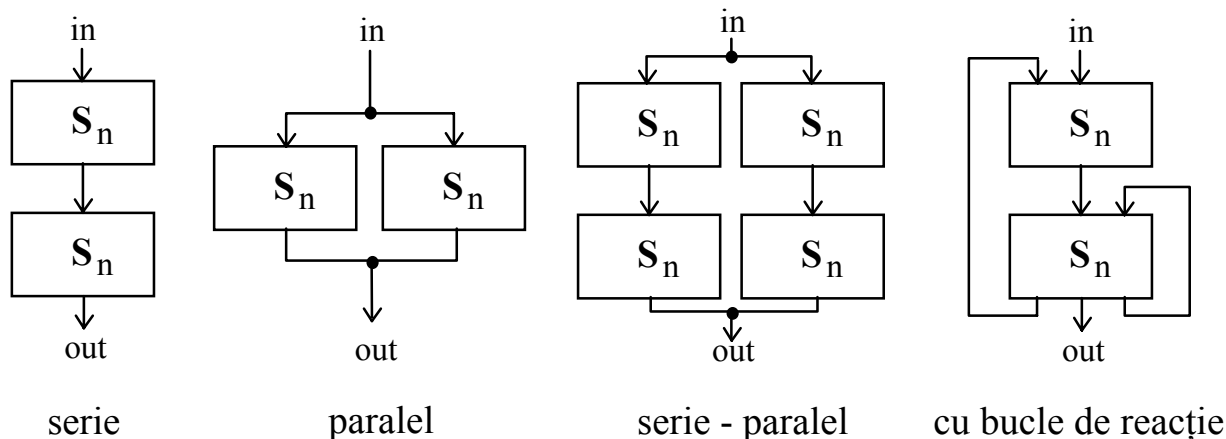


Fig. 1.7 Diferite cuplaje posibile pentru sistemele numerice

Clasificarea sistemelor numerice în combinaționale și secvențiale nu mai este satisfăcătoare la ora actuală, deoarece se dovedește a fi prea puțin nuanțată, criteriile utilizate fiind prea generale pentru un domeniu în care procesele de structurare au căpătat o mare amploare. Creșterea complexității funcționale a atras după sine nuanțări care presupun evidențierea unui **principiu de creștere structurală** în virtutea căruia vom obține o clasificare mai consistentă ([Ștefan, 1993]).

Clasificarea structurală a sistemelor numerice se face prin împărțirea lor în **ordine**, fiecărui ordin fiindu-i asociată o anumită clasă de funcții. Numărul de ordine nu este precis delimitat, limita superioară fiind dată de posibilitățile tehnologice de implementare și de modul în care va evolua gândirea funcțională.

Criteriul de clasificare propus presupune două aserțiuni:

- un sistem de ordin $n + 1$, notat prin S_{n+1} , poate fi generat prin interconectarea unor sisteme, dintre care cel puțin unul este de ordin n , notat prin S_n , într-o configurație ce presupune o buclă de reacție.
- sistemul de ordin zero, notat prin S_0 , este reprezentat de un circuit logic combinațional elementar cu o singură ieșire (poartă logică).

Posibilitățile de cuplare a două sau mai multe sisteme numerice sunt ilustrate în figura 1.7. Cuplarea serie, paralel sau mixtă a unor sisteme numerice, dintre care cel puțin unul este de ordin n , generează tot un sistem de ordin n . Introducerea unei bucle de reacție crește cu o unitate ordinul sistemului. În exemplul din figura 1.7 există două bucle de reacție care se includ, adică peste sistemul extins serie, format din cel puțin un sistem cu o buclă, se mai închide o buclă, care o include pe prima. Sistemul astfel obținut este un sistem de ordinul $n + 2$. Putem afirma că un sistem este de ordinul n , dacă în interiorul lui pot fi puse în evidență n bucle ierarhizate prin incluziune.

Majoritatea sistemelor numerice cunoscute până în prezent pot fi incluse în 6 ordine. Tabelul 1.4 ilustrează diversificarea acestora în funcție de tipurile posibile de cuplaje. Dacă clasificarea sistemelor de ordin inferior este relativ simplă, constatăm că încadrarea sistemelor în ordinele 4 și 5 este mai dificilă și discutabilă. Observăm că toate sistemele combinaționale sunt sisteme de ordinul 0, celelalte fiind secvențiale. Clasificarea sistemelor secvențiale este acum mult mai nuanțată decât înainte. Elementele de memorie, de la cele mai simple **latch**-uri și până la memoria RAM, sunt sisteme de ordinul 1, iar automatele cu stări finite sunt sisteme de ordinul 2. Structurile de ordin superior nu fac obiectul acestui cârți.

Tabel 1.4

Sisteme numerice	Ordinul sistemului	Exemple	Tipuri de cuplaje
combinaționale	0	porți logice	elementar
		multiplexoare	serie
		demultiplexoare	paralel
		memorie ROM, PLA	serie - paralel
de memorare	1	latch cu ceas	elementar
		registre serie	serie
		latch adresabil	paralel
		memoria RAM	serie - paralel
automate cu stări finite	2	bistabil JK, numărător	elementar
		unități PIPE LINE	serie
		RALU, CROM	paralel
		structuri PIPE LINE	serie - paralel
microprogramabile	3	procesorul	elementar
		rețele de procesare	serie, paralel și mixt
programabile	4	calculatorul	elementar
		rețele de calculatoare	serie, paralel și mixt
rețele complexe de procesare	5	multiprocesor INTERNET	elementar serie, paralel și mixt

Semnificația denumirilor prescurtate pentru unele circuite date ca exemple în tabelul 1.4 este următoarea: memorie **ROM** (**R**ead **O**nly **M**emory) - memorie care permite numai citirea informației binare stocate anterior în structura circuitului, memorie **RAM** (**R**andom **A**ccess **M**emory) - memorie cu acces aleator, care permite citirea și scrierea informației din/în memorie, structură **PLA** (**P**rogrammable **L**ogic **A**rray) - arie logică programabilă, automat de tip **CROM** (**C**ontroler and **ROM**) - o structură de ordin 2 care conține o memorie ROM și circuite aferente, automat de tip **RALU** (**R**egisters and **ALU** - **A**rithmetic **L**ogic **U**nit) - o structură de ordin 2 care conține registre și o unitate aritmetico-logică, structuri **PIPE LINE** - rețele care prelucrează simultan mai multe secvențe de date în același timp. Mai multe informații despre toate aceste structuri numerice vor fi prezentate în capitolele următoare.

Se observă că această clasificare structurală este alcătuită după criterii valabile, deoarece include toate tipurile de sisteme cunoscute până în prezent. În plus, este o clasificare deschisă, oferind posibilități de extindere către sisteme numerice de ordin superior celor existente. Rămâne ca aceste posibilități să fie confirmate în viitor.