

Capitolul 1. Integrarea pe Scara Foarte Mare.

Integrarea pe Scara Foarte Mare a circuitelor electronice reprezinta una dintre tehnologiile de varf ale industriei moderne. Cunoscuta in engleza sub prescurtarea VLSI (Very Large Scale Integration) aceasta tehnologie asigura componentele de baza si structurile functionale necesare realizarii unei game extrem de largi de produse si sisteme, pentru cele mai diverse aplicatii, incepand cu cele de uz casnic si terminand cu cele pentru industria aerospaciala.

Principalele avantaje ale produselor realizate in tehnologia ISFM se refera la implementarea unor sisteme cu o mare complexitate functionala in capsule de mici dimensiuni, in conditiile unui consum mic de putere si a unei fiabilitati extrem de ridicata.

Fara utilizarea tehnologiei VLSI nu ar fi de conceput echipamentele intalnite in bunurile de larg consum, intre care se pot mentiona:

- masinile de spalat cu comanda programata, cuptoarele cu microunde, frigiderele;
- echipamentele audio de mare fidelitate, televizoarele, aparatele de fotografiat, ceasurile electronice, sisteme de securitate pentru locuinte;
- calculatoarele personale, calculatoarele personale ultramobile, IPod-urile calculatoarele de buzunar, jucariile electronice;
- echipamentele medicale pentru masurarea tensiunii arteriale, echipamente portabile pentru masurarea si inregistrarea tensiunii, pulsului, electrocardiogramelor, echipamentele pentru asigurarea unei bune conditii fizice;
- telefoanele mobile, pager-ele etc;
- injectia electronica pentru automobile, calculatoare de bord, sensori pentru centurile de siguranta si pentru presiunea in anvelope, sisteme de alarmare etc.

Incepand cu inventarea tranzistorului, in anul 1947, tehnologia dispozitivelor semiconductoare a evoluat continuu. Din punctul de vedere al complexitatii, circuitele integrate s-au dezvoltat exponential. Spre exemplu, primul microprocesor, pe 4 biti, aparut in anul 1971, avea circa 1700 de tranzistoare, iar in anul 1990 microprocesoarele pe 32 de biti aveau deja peste 150.000 de tranzistoare. Procesoarele moderne utilizeaza

peste zeci si sute de milioane de tranzistoare (Fig.1.). Toate aceste exemple demonstreaza viabilitatea legii lui Moore, care apreciaza ca numarul de tranzistoare plasate pe o singura pastila se dubleaza la circa 18 luni.

Name	Date	Transistors	Microns	Clock speed	Data width	MIPS
8080	1974	6,000	6	2 MHz	8 bits	0.64
8088	1979	29,000	3	5 MHz	16 bits 8-bit bus	0.33
80286	1982	134,000	1.5	6 MHz	16 bits	1
80386	1985	275,000	1.5	16 MHz	32 bits	5
80486	1989	1,200,000	1	25 MHz	32 bits	20
Pentium	1993	3,100,000	0.8	60 MHz	32 bits 64-bit bus	100
Pentium II	1997	7,500,000	0.35	233 MHz	32 bits 64-bit bus	~300
Pentium III	1999	9,500,000	0.25	450 MHz	32 bits 64-bit bus	~510
Pentium 4	2000	42,000,000	0.18	1.5 GHz	32 bits 64-bit bus	~1,700
Pentium 4 "Prescott"	2004	125,000,000	0.09	3.6 GHz	32 bits 64-bit bus	~7,000

Fig. 1. Evolutia in timp a numarului de tranzistoare pe pastila pentru cateva procesoare Intel

Perfectionarea proceselor tehnologice in domeniul circuitelor integrate a permis, de asemenea, reducerea dimensiunilor dispozitivelor, ceea ce se poate exemplifica prin reducerea lungimii canalului tranzistorului elementar de la 5 μm , in 1985, la 0,35 μm , in 1997 si la 0,70 nm in 2005. In acelasi timp au crescut dimensiunile discurilor din siliciul monocristalin, care reprezinta suportul pe care se realizeaza structurile larg integrate. Evolutia in timp a unor elemente definitorii pentru circuitele integrate se poate urmari in tabelul de mai jos.

Intarzierea in propagarea semnalelor s-a redus cu trei ordine de marime in ultimii 20 de ani, ceea ce se reflecta in cresterea frecventei ceasului microprocesoarelor de la circa

1MHz in 1975 la pesete 1 GHz in anul 2000. In acelasi timp s-au redus in mod continuu costurile de fabricatie. Astfel, in cazul memoriilor RAM, costul pe bit s-a micsorat de la circa 1 cent, in 1970, la 10^{-4} - 10^{-5} centi, in prezent.

Actualmente, pentru circuitele integrate folosite in calculatoarele electronice, se folosesc numeroase tehnologii, care se pot grupa in *tehnologii bipolare si tehnologii MOS*

Tehnologii bipolare:

- TTL (Transistor Transistor Logic):
 - TTL-S (Schottky TTL),
 - TTL-LS (Low-Power Schottky TTL)
 - TTL-AS (Advanced Schottky TTL),
 - TTL-ALS (Advanced Low-power Schottky TTL),
 - FAST (Fairchild Advanced Schottky TTL).
- ECL (Emitter Coupled Logic).
- I2L (Integrated Injection Logic).

Tehnologii MOS:

- PMOS (MOS canal P).
- NMOS (MOS canal N):
 - HMOS (High performance MOS).
- CMOS (Complementary MOS):
 - HCMOS (High density CMOS),
 - ACL (Advanced CMOS Logic).
- MNOS (Metal Nitride Oxide Semiconductor):
 - FAMOS (Floating gate Avalanche injection MOS),
 - FLOTOX (FLOating gate Tunnel Oxide).

Circuitele integrate care se folosesc in constructia calculatoarelor se plaseaza in categoriile: *standard*, *specifice aplicatiilor* (ASIC - Application Specific Integrated Circuits) si *programabile/configurabile*.

La randul lor circuitele ASIC se impart in:

- Circuit personalizate la cerere (Semi-Custom):
 - Circuit predifuzate (Gate Arrays).
- Circuit realizate la comanda (Custom):
 - Circuit precaracterizate (Standard Cells),
 - Circuit realizate complet la cerere (Full Custom).

Se aminteste ca tranzistorul a fost inventat in anul 1947 si ca primele exemplare ocupau o suprafata de $3,5 \text{ mm}^2$. La sfirsitul anilor 50 a aparut circuitul integrat care, grupand pe aceeasi pastila mai multe tranzistoare, a avut o evolutie spectaculoasa in sensul dublarii numarului de componente pe pastila, la fiecare 18 luni. Aceasta s-a datorat in primul rand numeroaselor perfectionari ale proceselor tehnologice, care au permis rezolutii de ordinul a $2,5\mu\text{m}$ – $0,09 \mu\text{m}$. In continuare se vor da unele date privind tehnologiile circuitelor VLSI, in general, evolutia memoriilor si a procesoarelor

Evolutia tehnicilor de fabricatie a circuitelor integrate este unica in istoria industriei moderne. Tendintele privind cresterea vitezei, marirea densitatii, cat si reducerea costului circuitelor integrate s-au mentinut in mod constant, pe parcursul ultimilor 30 de ani. In continuare sunt ilustrate tendintele de scalare a tehnologiei.

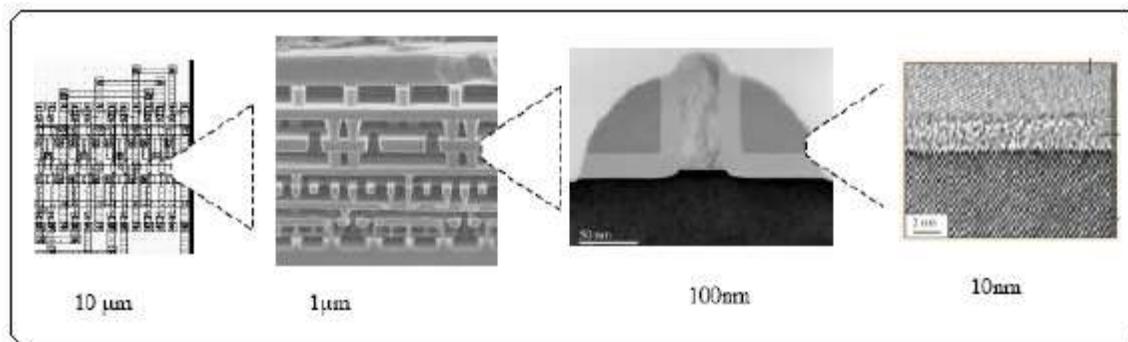


Fig. 2. Structuri reprezentative pentru un circuit integrat la diversele niveluri de detaliere de la $10\mu\text{m}$ la 1nm . (IBM, Fujitsu).

Mai jos se prezinta evolutia in timp a complexitatii procesoarelor, ca numar de dispozitive pe un circuit integrat. Pentium IV, care se producea in 2003, avea circa 50.000.000 tranzistoare MOS, pe o pastila de 2x2 cm².

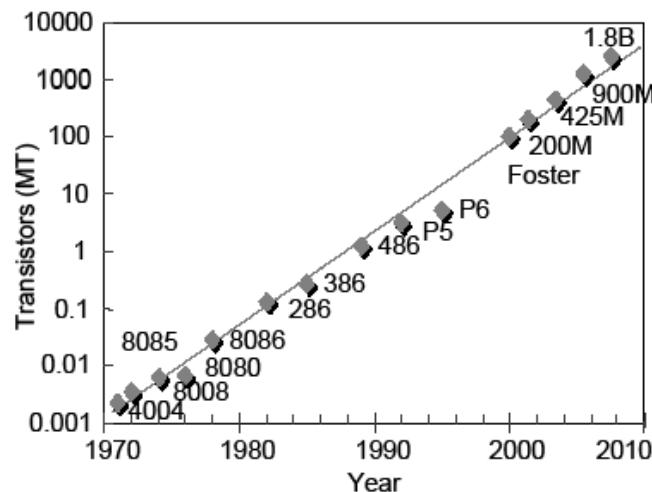


Fig.3.Complexitatea dispozitivelor din punctul de vedere al numarului de tranzistoare.

Incepand cu memoria de 1Kb, realizata de catre Intel, in 1971, memoriiile semiconductoare au avut o evolutie sustinuta in termeni de capacitate si performanta (timp de acces): 256Mb in anul 2000, 1Gb in anul 2004, cu tinta de 16Gb, in 2008, conform ITRS (International Technology Roadmap for Semiconductor Technology).

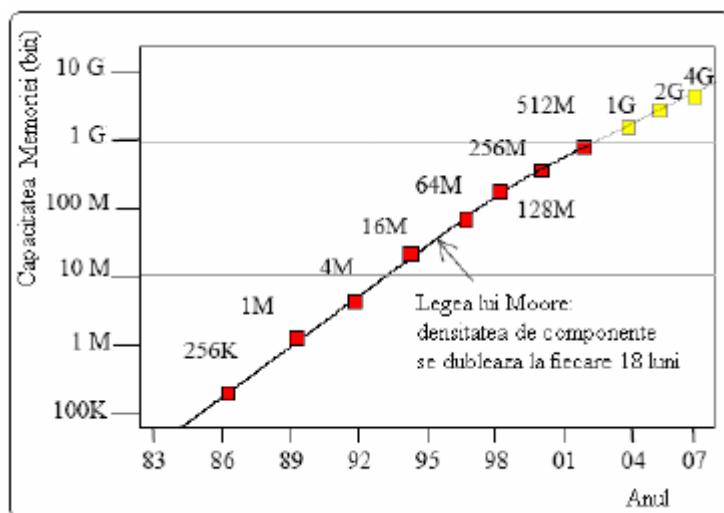


Fig. 4. Evolutia capacitatiilor in biti a circuitelor de memorie (ITRS)

Organizarea la nivelul planului de amplasare a blocurilor componente ale unui microcontrolor industrial destinat aplicatiilor in industria automobilelor este prezentata mai jos. Pe langa unitatea de prelucrare (procesor) microcontrolorul mai poseda diverse tipuri de memorii: EPROM, FLASH si RAM.

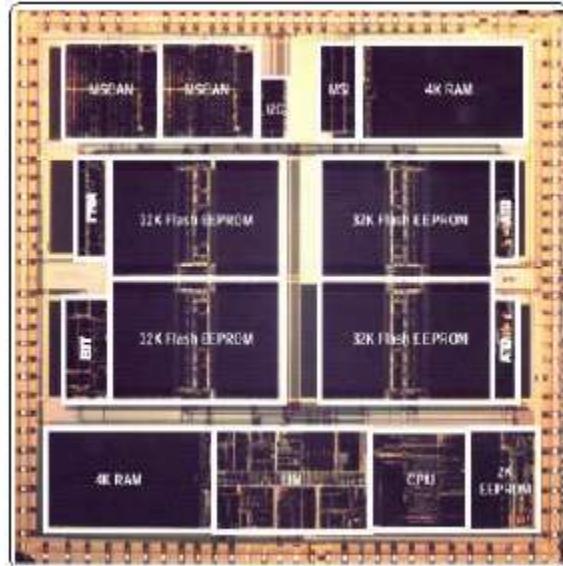


Fig.5. Planul de amplasare a blocurilor componente ale unui microcontrolor industrial.

In ceea ce priveste reducerea dimensiunilor, se vor considera patru generatii de tehnologii pentru circuitele integrate la nivelul de:

- micrometru;
- submicrometru, 1990 - tehnologie $0,8 \mu\text{m}$;
- adanc submicrometru (deep submicron), 1995 – tehnologie $0,3 \mu\text{m}$;
- ultra-adanc submicrometru (ultra deep submicron) – tehnologie $0,1 \mu\text{m}$.

Conform figurii de mai jos cercetarea se afla cu circa 5 ani inaintea productie de masa, in ceea ce priveste tehnologia. In anul 2007 procesele litografice au coborat sub $0,07 \mu\text{m}$. Litografia, exprimata in μm , corespunde celor mai mici forme, care pot fi realizate pe suprafata unui circuit integrat.

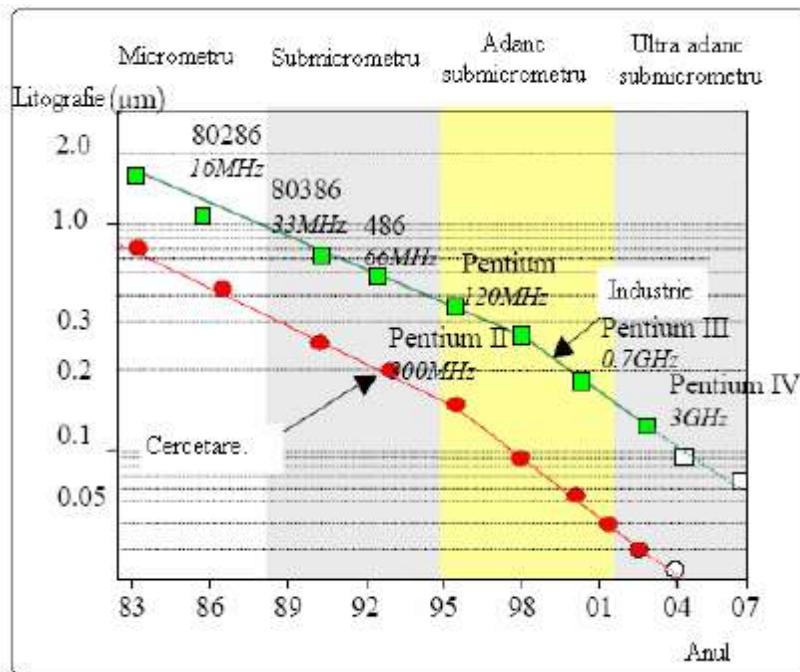


Fig. 6. Evolutia tehnologiilor proceselor litografice.

Evolutia frecventei ceasului pentru microprocesoarele si microcontroloarele performante a fost, de asemenea, influentata de reducerea dimensiunilor dispozitivelor integrate. Examinarea microprocesoarele Intel, destinate calculatoarelor personale (PC), si de la microcontroloarele Motorola, dedicate aplicatiilor din industria de automobile, pune in evidenta doua tendinte. Industria PC-urilor necesita procesoare extrem de rapide, care se caracterizeaza printr-o putere dissipata mare (30-100 W), in timp ce industria automobilelor solicita controloare incorporate, cu functii numeroase si sofisticate, cu memorii de diverse tipuri si circuite de interfata capabile sa asigure diferite protocoale de comunicatii. Tendintele evolutiei frecventelor de lucru in cele doua situatii sunt asemanatoare.

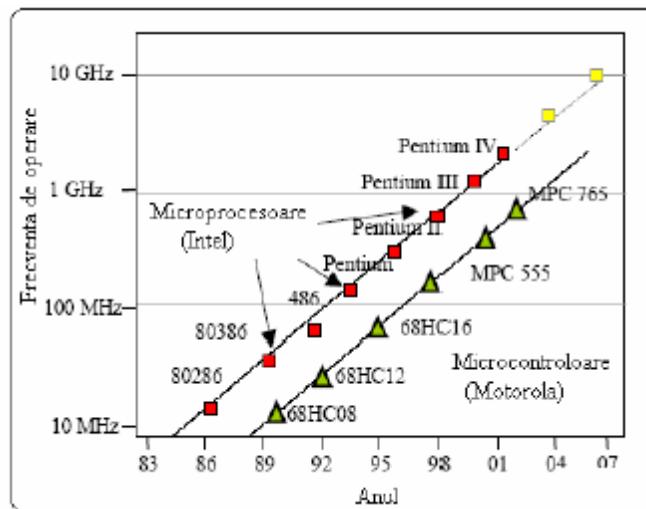


Fig. 7. Evolutia frecvențelor de operare pentru microprocesoare si pentru microcontroloare.

Tabela de mai jos prezinta parametrii mai importanți și evolutia lor odata cu perfectionarea tehnologiilor. Trebuie menționate creșterea numărului de straturi de metal, pentru interconectari, reducerea tensiunii de alimentare VDD, micsorarea grosimii stratului de oxid al portii, pana la dimensiuni atomice. Se remarcă, de asemenea, creșterea dimensiunilor pastilei, cat și marirea numărului de ploturi de I/E, disponibile pe o singura pastila.

Litografia	Anul	Straturi de metal	Tensiunea de alimentare (V)	Grosimea Oxidului (nm)	Aria Circutului mm x mm	Ploturi de I/E	Fisierul de reguli Microwind2
1.2µm	1986	2	5.0	25	5x5	250	Cmos12.nul
0.7µm	1988	2	5.0	20	7x7	350	Cmos08.nul
0.5µm	1992	3	3.3	12	10x10	600	Cmos06.nul
0.35µm	1994	5	3.3	7	15x15	800	Cmos035.nul
0.25µm	1996	6	2.5	5	17x17	1000	Cmos025.nul
0.18µm	1998	6	1.8	3	20x20	1500	Cmos018.nul
0.12µm	2001	6-8	1.2	2	22x20	1800	Cmos012.nul
90nm	2003	6-10	1.0	1.8	25x20	2000	Cmos90n.nul
70nm	2005	6-12	0.8	1.6	27x20	3000	Cmos70n.nul

Fig. 8. Parametrii mai importanți și evolutia lor odata cu perfectionarea tehnologiilor.

Procesul CMOS de $1,2\mu\text{m}$ specifica dispozitive NMOS si PMOS cu lungimea minima a canalului de $0,8\mu\text{m}$. Pachetul de proiectare *Microwind* [15] poate fi configurat in tehnologia CMOS $1,2\mu\text{m}$ folosind comanda: "File-> Select Foundry", si selectand din lista regulile de proiectare "cmos12.rul". Liniile de metal au latimea de $2\mu\text{m}$, iar adancimea zonelor de difuzie este de circa $1\mu\text{m}$.

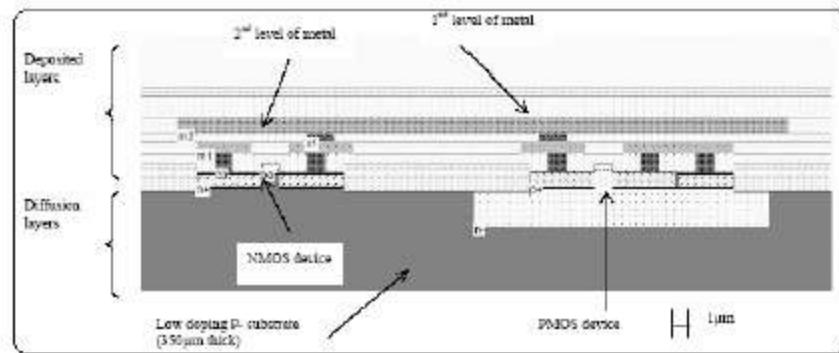


Fig. 9. Secțiune transversală printr-un dispozitiv CMOS în tehnologia $1,2\mu\text{m}$ (CMOS.MSK).

Tehnologia CMOS $0,35\mu\text{m}$ dispune de 5 straturi de metal si de dispozitive MOS cu lungimea canalului de $0,35\mu\text{m}$. Dispozitivul MOS include difuzii laterale ale drenei, cu izolatii de oxid insant putin adanc. Pachetul de proiectare Microwind poate fi configurat in tehnologia CMOS $0,35\mu\text{m}$ folosind comanda: "File-> Select Foundry" si selectand "cmos035.rul", din lista regulile de proiectare. Latimile traseelor de metal sunt mai mici decat $1\mu\text{m}$. Adancimea zonelor de difuzie este mai mica decat $0,5\mu\text{m}$

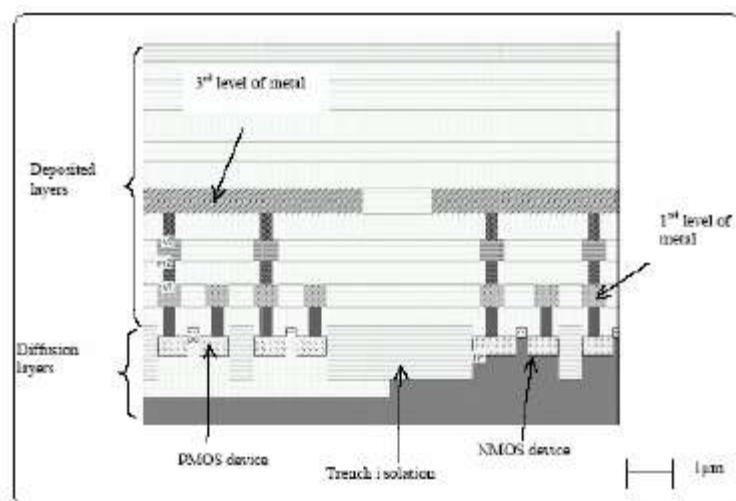


Fig. 10. Secțiune transversală printr-o structură CMOS în tehnologia $0,35\mu\text{m}$ (INV3.MSK).

Ca o consecinta a perfectionarii procesului litografic, pe aceeasi arie de siliciu se pot implementa mai multe functii. Cresterea numarului de straturi de metal, pentru interconectari, a condus la o utilizare mai eficienta a ariei de siliciu, ca si pentru circuitul imprimat. De asemenea, dispozitivele MOS pot fi plasate la distante mai mici unul fata de celalalt.

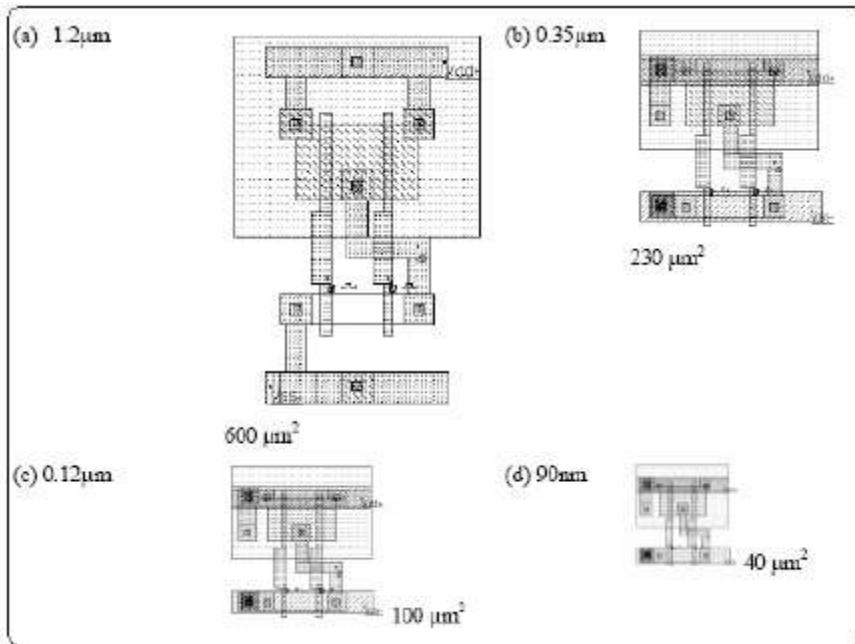


Fig. 11. Evolutia ariei de Si utilizata pentru implementarea portii NAND, poarta care reprezinta circa 20% din portile utilizate in ASIC.

Cresterea densitatii conduce la reducerea ariei si la micsorarea capacitatiilor parazite ale jonctiunilor si interconexiunilor, avand ca efect cresterea vitezei de operare. In acelasi timp, dimensiunile mai mici ale dispozitivelor permit, in continuare, sporirea vitezei de lucru, respectiv, cresterea frecventei ceasului.

Dimensiunile discurilor (wafers) de Si au crescut in mod continuu. Un diametru mai mare al discului inseamna mai multe structuri produse in acelasi timp, dar necesita echipamente ultra-performante pentru manipularea si prelucrarea acestora cu precizie la scara atomica. Aceasta tendinta este prezentata in figura de mai jos.

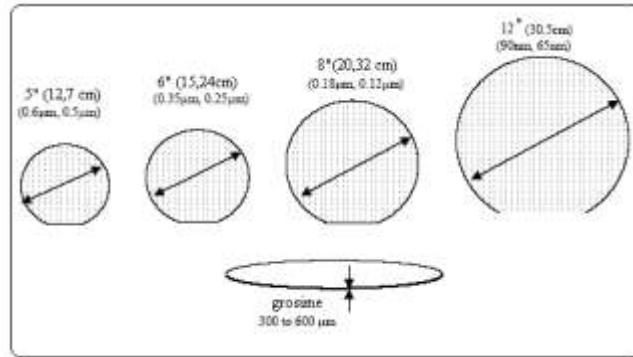


Fig. 12. Evolutia dimensiunilor discurilor (wafers) de Si.

1.1 Familii de circuite integrate.

In functie de tehnologiile utilizate, circuitele integrate pot fi clasificate in mai multe familii, dupa cum se poate observa in diagrama din figura 13.

Fiecare tehnologie se caracterizeaza prin cost, performanta, timp de proiectare, avantaje si dezavantaje. Intrucat tehnologia MOS este cea mai raspandita actualmente se va insista asupra acesteia in continuare.

Tehnologia MOS este utilizata in circuitele integrate pe sacra larga, de la ariile de porti simple pana la microprocesoare.

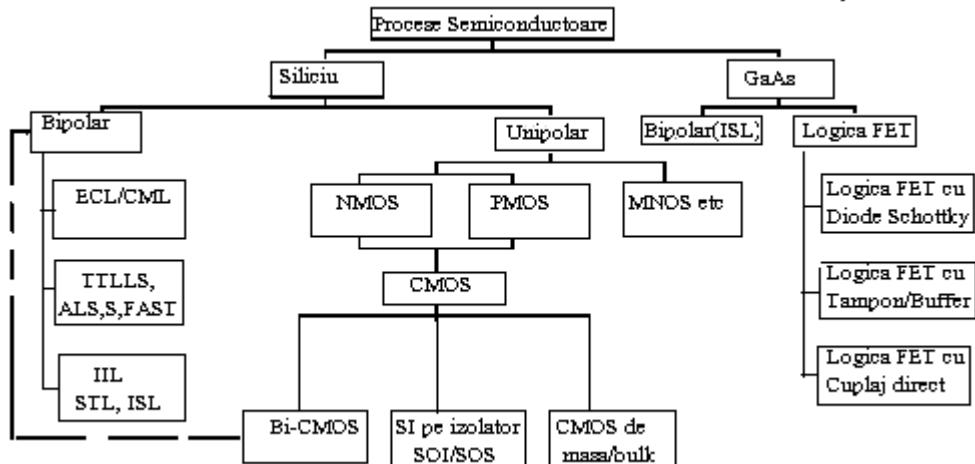


Fig. 13. Procese pentru dispozitive semiconductoare.

Dintre caracteristicile tranzistorului MOS se mentioneaza urmatoarele:

- realizarea unei densitati mari de tranzistoare, deoarece dispozitivele MOS consuma o putere mai mica decat dispozitivele TTL;

- nivelurile de la iesirea circuitelor sunt fie VDD, fie GND, ceea ce corespunde logicii cu restaurare, intrucat semnalele logice corespund nivelurilor maxime/minime de tensiune.

1.1.1. MOS

Tranzistorul MOS tipic este prezentat in figura 14. El consta intr-un substrat de siliciu monocristalin, regiunile de difuzie sursa si drena, oxidul izolator si poarta din siliciu policristalin. In functie de substrat si de difuzii exista doua tipuri de tranzistoare: NMOS si PMOS.

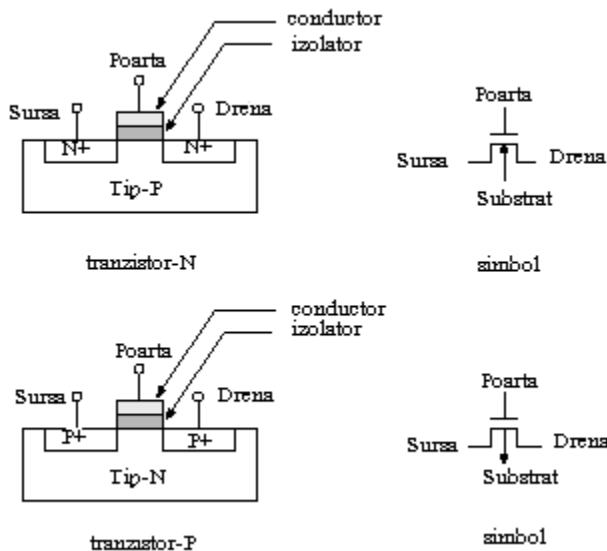


Fig14. Tipuri de tranzistoare MOS

Tranzistorul de tip N (NMOS) are un substrat dopat P , cu impuritati de tip acceptor: B, In, Ga, in timp ce regiunile de difuzie sunt dopate N^+ cu impuritati de tip donor: P, As. Tranzistorul de tip P (PMOS) este realizat pe un substrat de tip N , si cu regiuni de difuzie de tip P^+ . Zonele de difuzie se caracterizeaza printr-o rezistivitate mai coborata pentru a realiza un bun contact cu stratul de metal.

Poarta, de regula, este realizata printr-un proces de depunere chimica din siliciu policristalin dopat N , pentru a-i micsora rezistivitatea.

Elementul izolant al portii este SiO_2 sau o varianta a acestuia. Dioxodul de Si are o grosime mai mica de 1000 Å si o rezistivitate $\rho \approx 10^{16} \Omega\text{cm}$. Firele de legatura sau conexiunile se realizeaza

prin trei tipuri de materiale conductoare: metal, siliciu policristalin si difuzie. Zonele constituite din metal, siliciu policristalin si difuzie sunt separate prin material izolator SiO_2 .

Procesele moderne presupun straturi multiple de metal pentru transmisia semnalelor. Foarte rar se folosesc trasee de siliciu policristalin pentru transmiterea semnalelor electrice. Contactele intre straturile cu proprietati electrice diferite se realizeaza prin taieturi in stratul izolant. Atunci cand sunt separate printr-un strat izolant, un traseu dintr-un material dat se poate intersecta cu un traseu dintr-un alt material, fara a se constata efecte majore, cu singura exceptie a suprapunerii unui traseu de siliciu policristalin cu un traseu de difuzie. In acest caz se realizeaza un tranzistor.

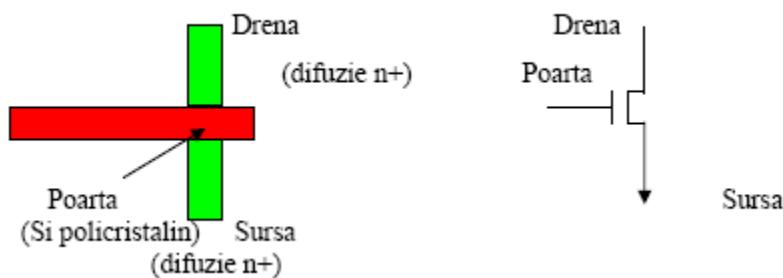


Fig. 15. Realizarea unui tranzistor NMOS.

Detalii privind procesele de fabricatie vor fi prezentate intr-un capitol special.

Tranzistoarele MOS sunt unipolare, in sensul ca functionarea lor se bazeaza pe un singur tip de purtatori: electroni (dispozitivele NMOS) si goluri (dispozitivele PMOS). Intrucat mobilitatea electronilor este mai mare decat cea a golurilor, dispozitivele NMOS sunt mult mai raspandite.

1.1.1.1. Comutatoare realizate cu tranzistoare MOS.

Intr-o maniera simplificata tranzistoarele MOS pot fi examineate ca simple comutatoare bipozitionale: inchise/deschise. Operarea comutatorului este asigurata prin tensiunea aplicata pe poarta, caracterizata prin nivelul ridicat sau coborat. Tranzistorul MOS realizeaza o cale inchisa sau deschisa, intre sursa si drena, cand este conectat intr-un circuit.

Comutatorul are o rezistenta interna, care poate influenta capacitatatile acestuia de a transmite semnalul nealterat.

Codificand cu valorile logice “0” si “1” nivelurile de tensiune 0V/GND si +5V/VDD, operarea tranzistorului MOS, in calitate de comutator, poate fi examinata din punctul de vedere al algebrei

logicii, al algebrei Booleene. Operarea tranzistoarelor NMOS si PMOS in calitate de comutatoare este ilustrata in figura16:

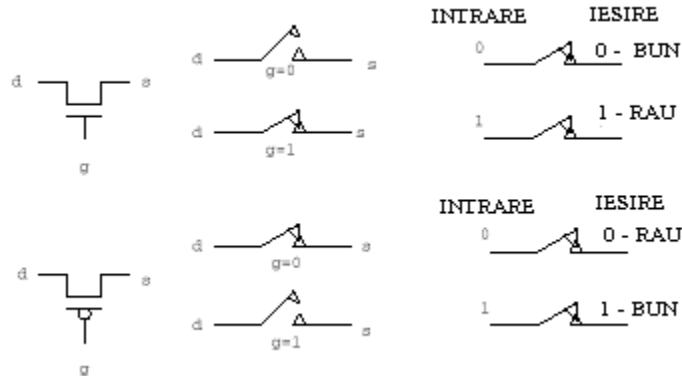


Fig. 16. Operarea tranzistoarelor la nivelul comutatoarelor.

La un tranzistor/comutator NMOS, cand poarta se afla la nivel logic "1", comutatorul este inchis/conduce, drena si sursa sunt conectate, iar curentul curge de la drena la sursa. In cazul in care poarta se afla la nivel logic "0", comutatorul este deschis/nu conduce, drena si sursa nu sunt conectate, fluxul curentului intre drena si sursa este intrerupt.

Comutatorul PMOS poseda proprietati complementare in raport cu comutatorul NMOS. Astfel, atunci cand poarta se afla la "1" logic, comutatorul este deschis curentul fiind intrerupt intre sursa si drena, iar cand poarta se afla la "0" logic, comutatorul este inchis si curentul curge intre sursa si drena.

Comutatorul NMOS transmite foarte bine "0" logic si mai putin bine "1" logic. Pe de alta parte, comutatorul PMOS conduce foarte bine "1" logic si mai putin bine "0" logic. Astfel, o combinatie de comutatoare NMOS si PMOS in paralel, controlate pe porti cu semnale de comanda in antifaza/ complementare, va transmite la fel de bine "0" logic si "1" logic. Aceasta idee se afla la baza conceptului de dispozitive CMOS (MOS Complementar). In figura 6 se prezinta schema unui astfel de comutator/ poarta de transfer (T-gate) CMOS.

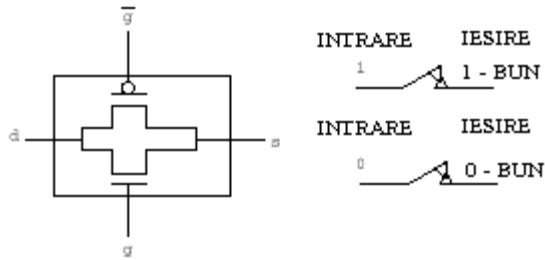


Fig.17. Poarta de transfer.

1.1.2 CMOS

Dupa cum s-a mentionat anterior, dispozitivele CMOS preiau avantajele dispozitivelor NMOS si PMOS. Unul din avantajele majore, fata de utilizarea exclusiva a tranzistoarelor NMOS sau PMOS, consta in aceea ca dispozitivele CMOS au un consum redus de putere. CMOS reprezinta actualmente tehnologia cea mai raspandita pentru realizarea structurilor numerice integrate pe scara larga..

In sectiunile care urmeaza se vor prezenta combinatii de comutatoare pentru realizarea portilor logice de baza, intalnite in sistemele numerice.

1.1.2.1 Inversorul NOT/NU.

Componenta fundamentala a unui sistem numeric o reprezinta inversorul. In tehnologia CMOS, un inversor este realizat prin legarea in serie a unui tranzistor PMOS si a unui tranzistor NMOS. In timpul operarii este inchis fie tranzistorul NMOS, fie tranzistorul PMOS, in timp ce celalalt tranzistor este deschis. Astfel, iesirea este fortata fie la VDD, de catre dispozitivul PMOS, fie la Vss, de catre dispozitivul NMOS. In ambele cazuri nu va curge nici un curent intre VDD si Vss/GND, deoarece unul dintre tranzistoare va fi deschis. Astfel, nu va exista un curent permanent care sa curga de la VDD la Vss/GND si, in consecinta, nici o putere dissipata in curent continuu. Aceasta proprietate recomanda utilizarea circuitelor CMOS in aplicatiile in care se impune un consum mic de putere.

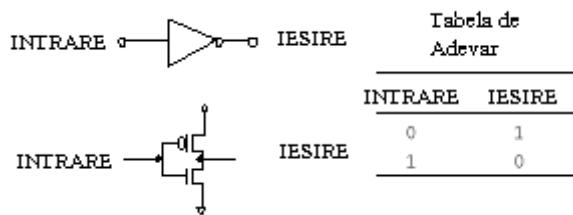


Fig.18. Inversorul CMOS

1.1.2.2 Functia AND/SI.

Pentru realizarea unei porti AND/SI se pot utiliza fie doua comutatoare NMOS, fie doua comutatoare PMOS, in serie (fig19). constituite din porti AND-NMOS si AND-PMOS in paralel, controlate pe porti cu semnale in antifaza/complementare.

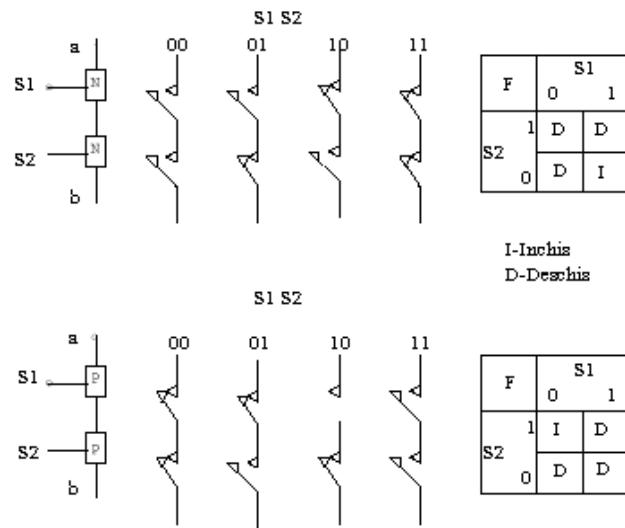


Fig. 19. Implementarea functiei SI.

Conexiunea intre punctele a si b este realizata ,in cazul in care portile tranzistoarelor NMOS sunt comandate cu “1” logic, in timp ce, in situatia utilizarii tranzistoarelor PMOS, portile acestora trebuie sa fie comandate cu “0” logic. Pentru a conduce la fel de bine nivelurile logice “1” si “0” se pot imagina structuri.

Adesea functia AND/SI este realizata prin conectarea in cascada a unui circuit NAND/SI-NU si a unui inversor. Desi solutia ar putea ocupa un spatiu mai mare pe pastila de Si, ea are avantajul ca nivelurile semnalelor, care corespund valorilor logice “1” si “0”, se apropie de valorile VDD si VSS.

1.1.2.3. Poarta NAND/SI-NU.

Poarta NAND se obtine prin conectarea in serie intre VDD si VSS a doua structuri constand in doua tranzistoare PMOS, in paralel, si doua tranzistoare NMOS, in serie, ca in figura20.

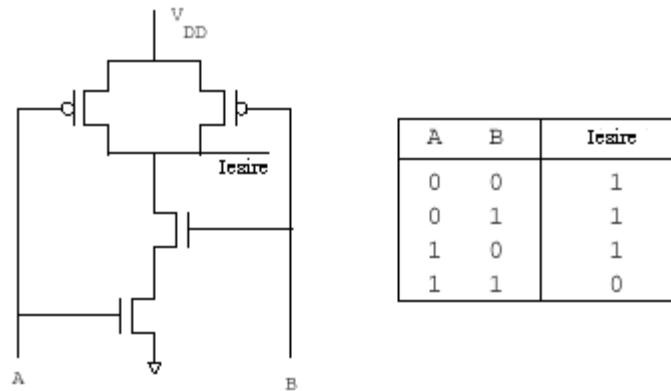


Fig.20. Poarta NAND/SI-NU

Iesirea se obtine de la nodul la care converg cele doua structuri. Structura serie “trage jos”, formata din tranzistoare NMOS reprezinta duala structurii paralele “trage sus”, constituita din circuite PMOS. Nivelurile de semnal obtinute la iesire se apropie de valorile VDD si VSS.

Pe langa avantajele oferite de circuitele CMOS, acestea prezinta o serie de probleme de care trebuie sa se tina seama in proiectare si operare. Dintre aceste se mentioneaza: partajarea sarcinii si “efectul de corp”, care vor fi studiate intr-unul din capitolele urmatoare.

1.2. Modalitati de reprezentare.

In procesul de proiectare, circuitele numerice integrate pot fi reprezentate sub aspect comportamental, structural si fizic. In cele ce urmeaza se va face o scurta descriere a acestor modalitati de descriere a circuitelor. Proiectantii de circuite integrate incearcă să utilizeze cat mai mult reprezentari, care fac abstractie de nivelul fizic si de tehnologie, intrucat aceasta din urma evolueaza rapid. Descrierea proiectului se realizeaza la un nivel de abstractizare cat mai inalt, in cadrul caruia se pot evita si corecta eventualele erori. Tranzitaia catre reprezentarile pe niveluri mai joase, mai apropiate de tehnologie, se realizeaza cu ajutorul uneltelor de proiectare asistata de calculator (CAD – Computer Aided Design), care fiind automatizate nu mai pot introduce erori.

1.2.1. Reprezentarea comportamentală.

Reprezentarea comportamentală se referă la modul în care un sistem numeric dat reacționează la un set de stimuli aplicati la intrare. Comportamentul poate fi specificat cu ajutorul ecuațiilor Booleene, al tabelelor valorilor de intrare si iesire sau al algoritmilor pe care ii implementeaza.

Acestia din urma pot fi descrisi in limbaje de programare de nivel inalt sau in limbaje specializate pentru descrierea hardware-lui: VHDL (Very High Speed Integrated Circuit Hardware Description Language), Verilog, ELLA etc.

Scopul urmarit de catre diversele sisteme moderne de proiectare consta in transformarea specificatiilor de operare a sistemului, date la nivelul cel mai inalt posibil de descriere, intr-un proiect viabil, in timpul cel mai scurt.

Pentru ilustrare se va considera un sumator binar pentru numere cu n ranguri. Acesta se obtine prin legarea in cascada a n sumatoare de cate un bit. Un sumator pentru numere de cate un bit are ca intrari operanzii A, B si transportul C, iar ca iesiri operanzii: S (suma) si Co (transportul catre rangul superior).

Sumatorul de un bit se poate descrie cu ajutorul tabelei de adevar de mai jos:

A	B	C	Co	S
0	0	0	0	
0	0	1	1	
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

sau cu ajutorul urmatoarelor ecuatii Booleene, care definesc ca functii pe S si Co:

$$S = \overline{A} \overline{B} C \cup \overline{A} B \overline{C} \cup A \overline{B} \overline{C} \cup ABC$$

$$Co = AB \cup AC \cup BC$$

Descrierea la nivel algoritmic a functiei Co, in limbajul Verilog are urmatoarea forma:

```
module carry (co, a, b, c );
output co;
input a, b, c;
assign co = (a+b) | (a+c) | (b+c);
endmodule
```

Circuitul care implementeaza functia Co poate fi specificat comportamental sub aspect Boolean, ca primitiva, in urmatoarea maniera:

```

primitive carry (co, a, b, c);
output co;
input a,b,c;
table
// a b c co
1 1 ? : 1 ;
1 ? 1 : 1 ;
? 1 1 : 1 ;
? 0 0 : 0 ;
0 ? 0 : 0 ;
0 0 ? : 0 ;
endtable
endprimitive

```

unde ? specifica o valoare 0 sau 1 (don't care/indiferenta).

Descrierea este independenta de tehnologie, fiind realizata la nivel logic. Nu se specifica modul de implementare si nici performantele privind intarzierile/viteza de operare. Uneori descrierea comportamentală implica specificare duratelor fronturilor semnalelor manipulate de poarta/circuitul in cauza. De exemplu, daca se doreste sa se specifica faptul ca semnalul co se modifica cu o intarziere de 10 unitati de timp, dupa modificare semnalelor a sau b sau c, descrierea comportamentală poate lua urmatorul aspect:

```

module carry (co, a, b, c) ;
output co;
input a, b, c;
wire #10 assign co = (a+b) | (a+c) | (b+c);
endmodule

```

Spre deosebire de limbajele de nivel inalt C, Pascal, FORTRAN etc., folosite pentru dezvoltarea de software, limbajele de tip HDL permit descrierea concurentei, a intarzierilor, a dimensiunii cuvantului si a vectorilor binari intr-o maniera convenabila, ceea ce face ca ele sa fie extrem de raspandite in proiectarea sistemelor VLSI.

Reprezentarile comportamentale sunt utilizate pentru a capta algoritmul. Ele pot avea aspecte diferite, de la exemplele simple de mai sus, pana la descrierile unor procesoare complexe de

semnal. Avantajul lor consta in aceea ca permit descrierea si verificarea proiectului la nivel functional. Dezavantajul major se refera la faptul ca descrierea comportamentala nu se poate implementa direct si eficient in hardware.

1.2.2. Reprezentarea structurala.

Reprezentarea structurala a unui sistem numeric prezinta modul in care sunt interconectate componentele sistemului in vederea realizarii unei functii date sau a unui anumit comportament. Descrierea structurala consta intr-o lista de module si de interconexiuni ale acestora.

Nivelurile abstracte ierarhice, ale descrierii structurale se refera la: *module*, *porti*, *comutatoare* si *circuite*. Pe masura parcurgerii ierarhiei de niveluri se evidentaiza detalii privind implementarea. Pentru exemplificare se va considera cazul unui sumator pe 4 biti, constituit prin conectarea in cascada a patru sumatoare de cate *un* bit. Descrierea este realizata in Verilog HDL.

```
module add4 (s,c4,ci,a,b) ;
input [3:0]a,b;
input ci;
output [3:0]s;
output c4;
wire [2:0]co;
add a0 (co[0],s[0],a[0],b[0],ci);
add a1 (co[1],s[1],a[1],b[1],c[0]);
add a2 (co[2],s[2],a[2],b[2],c[1]);
add a3 (c4,s[3],a[3],b[3],co[2]);
endmodule
```

Dupa declararea modulului add4, in urmatoarele patru linii sunt definite intrarile si iesirile, iar in linia a cincea se specifica vectorul binar intern, pe trei biti, co. In continuare se apeleaza de patru ori modulul add, a carui descriere se da mai jos:

```
module add (co,s,a,b,c) ;
input a,b,c;
output s,co;
sum s1(s,a,b,c);
carry c1(co,a,b,c);
endmodule
```

Modulul de mai sus specifica un sumator de *un* bit, care este constituit, la randul sau, din doua module pentru calculul sumei (sum) si al transportului (carry).

```
module carry (co,a,b,c) ;
input a,b,c;
output co;
wire x,y,z;
and g1 (x,a,b);
and g2 (y,a,c);
and g3 (z,b,c);
or g4 (co,x,y,z);
endmodule
```

Descrierea de mai sus este independenta de tehnologie deoarece au fost utilizate porti generice, fara a se specifica implementarea lor.

In cazul unei implementari in tehnologie CMOS, modulul carry poate fi descris la nivelul tranzistoarelor componente.

Tranzistoarele sunt descrise prin: tip, nume si conexiunile drenei, sursei si portii:

Tip-tranzistor|Nume|Drena(lesire)|Sursa(Data)|Poarta(comanda)

nmos n1 i1 vss a

Descrierea modulului carry, implementat in tehnologie CMOS,
are urmatorul aspect:

```
module carry (co,a,b,c) ;
input a,b,c;
output co;
wire i1,i2,i3,i4,cn;
nmos n1 (i1,vss,a);
nmos n2 (i1,vss,b);
nmos n3 (cn,i1,c);
nmos n4 (i2,vss,b);
nmos n5 (cn,i2,a);
pmos p1 (i3,vdd,b);
pmos p2 (cn,i3,a);
pmos p3 (cn,i4,c);
pmos p4 (i4,vdd,b);
```

```
pmos p5 (i4,vdd,a);  
pmos p6 (co,vdd,cn);  
nmos n6 (co,vss,cn);  
endmodule
```

In comparatie cu descrierea comportamentalala, descrierea structurala contine detalii referitoare la nodurile interne, la conexiunile intre componente primitive sau elementele de comutatie folosite pentru implementare. La nivelurile superioare de descriere ale modulului aceste conexiuni nu sunt relevante.

Descrierile de mai sus nu furnizeaza informatii referitoare la comportarea temporala a modulului carry, deoarece ele sunt realizate la nivelul portilor, la nivelul circuitelor de comutatie.

Unul din limbajele de descriere structurala care surprinde, printre altele, si comportarea temporala a modulelor este limbajul SPICE.

In SPICE tranzistoarele sunt specificate prin inregistrari, care contin urmatoarele campuri:

Mnume drena poarta sursa substrat tip W = latime L = lungime AD = aria drenei

AS = aria sursei

Numele tranzistoarelor incep cu majuscula M. Tipul specifica daca este un tranzistor nmos sau pmos.

Capacitatatile sunt descrise astfel:

Cnume nodul-1 nodul-2 valoare

De exemplu un NAND cu doua intrari poate fi descris in SPICE dupa cum urmeaza:

```
.SUBCKT NANDVDD VSS A B OUT  
MN1 I1 A VSS VSS NFET  
MN2 OUT B I1 VSS NFET  
MP1 OUT A VDD VDD PFET  
MP2 OUT B VDD VDD PFET  
.ENDS
```

unde A si B reprezinta terminalele de intrare, iar OUT este terminalul de iesire.

Reprezentarea structurala permite introducerea unor parametrii suplimentari si a dimensiunilor tranzistoarelor.

Astfel, specificarea unei porti NAND are aspectul de mai jos:

```

.SUBCKT NAND VDD VSS A B OUT
MN1 I1 A VSS VSS NFET W=8U L=4U AD=64P AS=64P
MN2 OUT B I1 VSS VSS NFET W=8U L=4U AD=64P AS=64P
MP1 OUT A VDD VDD PFET W=16U L=4U AD=128P AS=128P
MP2 OUT B VDD VDD PFET W=16U L=4U AS=128P AS=128P
CA A VSS 50fF
CB B VSS 50fF
COUT OUT VSS 100fF
.ENDS

```

Simulatorul SPICE calculeaza capacitatile parazite interne ale tranzistoarelor MOS, folosind modele adecvate, pe baza dimensiunilor specificate pentru dispozitive.

Pentru a introduce si influenta capacitatilor traseelor interne, prin care se conecteaza dispozitivele, se evalueaza valorile acestor capacitatii ce se conecteaza la nodurile corespunzatoare. Astfel, modulul poate fi caracterizat sub aspectul vitezei de operare, al puterii disipate si al conectivitatii. Rezultatele obtinute in urma simularii: intarzieri, duratele fronturilor crescatoare si cazatoare etc, pot fi furnizate descrierilor la nivel logic sub forma de intarzieri.

1.2.3. Reprezentarea fizica.

Descrierea fizica a sistemelor numerice furnizeaza informatii privind modul de constructie al unui circuit particular, care va avea o structura si o comportare date.

Intr-un proces de realizare a circuitelor integrate, specificarea fizica cu nivelul cel mai coborat o reprezinta descrierea geometrica a mastilor fotografice, pentru fiecare etapa a procesului tehnologic. Un exemplu de descriere geometrica este dat, la nivelul mastilor pentru tranzistoare NMOS si PMOS, in figura 21.

Descrierea fizica comporta, de asemenea, mai multe niveluri de abstractie. La nivel de modul, planul fizic pentru un sumator cu patru biti poate fi definit ca un dreptunghi sau un poligon, care specifica limitele externe pentru toata geometria sumatorului, un set de chemari de submodule si o colectie de porturi. Fiecare port corespunde unei conexiuni de I/E in descrierea structurala a sumatorului. Pentru fiecare port se specifica pozitia, stratul, numele, si latimea.

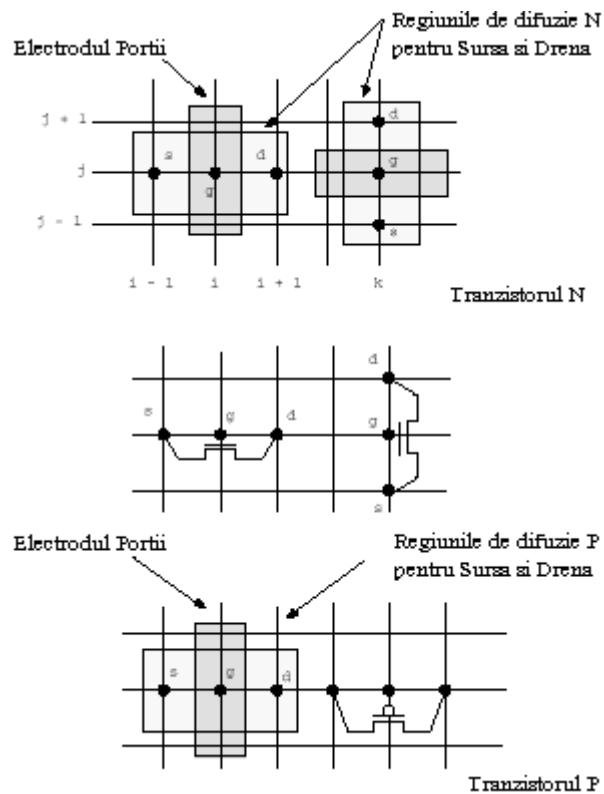


Fig. 21. Descrierea tranzistoarelor la nivelul mastilor.

Pentru exemplificare, în cele ce urmează se prezintă descrierea fizică incompleta a unui sumator pe patru biti, într-un limbaj de descriere fizică ad hoc.

```

module add4 ;
input a[3:0],b[3:0];
input ci;
output s[3:0];
output c4;
boundary [0,0,100,400];
port a[0] aluminium width=1 origin=[0,25];
port b[0] aluminium width=1 origin=[0,75];
port ci polysilicon width=1 origin=[50,100];
port s[0] aluminium width=1 origin=[100,50];
.
add a0 origin = [0,0];
add a1 origin = [0,100];
.
endmodule

```

Porturile sunt indicate prin cuvantul cheie port, iar chemarile submodulelor ce reprezinta sumatoare pe un bit sunt specificate prin cuvintul cheie add.

La cel mai jos nivel de descriere fizica se fac chemari la tranzistoare, fire si la conexiuni.

Acestea specifica dimensiunile dreptunghiurilor, care se implementeaza pe diverse straturi ale procesului CMOS. Aici nu se va intra in aceste detalii, poarta CMOS fiind tratata ca un dreptunghi cu frontiere date si cu porturile necesare. Fiecare port are o pozitie, un strat de conectare, o latime si un nume. Aceste informatii pot fi utilizate de catre un program automat de trasare, care va asigura interconectarea acestor module cu alte proiecte.

Un alt exemplu de reprezentare fizica o reprezinta Forma Intermediara Caltech (CIF), propusa de catre Carver Mead, in 1980. In limbajul CIF un circuit este reprezentat sub forma de straturi. Scopul principal al descrierii CIF este acela de a oferi o reprezentare standard, care poate fi citita de catre calculator. Pornind de la fisiere CIF se pot genera fisiere specifice diverselor dispozitive de iesire: display-uri, plotere, imprimante, echipamente pentru generarea mastilor sub forma de clisee/placi fotografice. Ca exemplu, se prezinta mai jos, un fragment din reprezentarea CIF a unui inversor.

DS 101 1 1;

9 inv{lay};

0V 1050 5500 -1050 5500 -1050 6700 1050 6700 2163883 2169080;

94 out 500 1650;

94 Vdd -1200 5850;

94 Gnd -1200 -5150;

94 in -1100 450;

L CM2;

P -50 1100 -50 2200 1050 2200 1050 1100;

P -1650 -100 -1650 1000 -550 1000 -550 -100;

...

L CNP;

P -850 -6000 -850 -4800 850 -4800 850 -6000;

20

P -1650 1800 -1650 5500 1450 5500 1450 1800;

DF;

C 101;

E

Se intlege de la sine ca un proiectant nu va fi interesat sa desfasoare activitatea de proiectare la acest nivel de reprezentare/abstractizare. El va prefera descrierile la nivel inalt ale sistemelor ce urmeaza a fi implementate. Aceste descrieri, dupa simulari si verificari exhaustive, vor fi compilate in fisiere de tip CIF, in vederea obtinerii mastilor si a unor simulari mai detaliate, avand in vedere atat aspectele geometrice, cat si cele privitoare la procesele tehnologice utilizate.

1.3. Etapele proiectarii.

Proiectarea unui circuit VLSI CMOS impliva mai multe etape, care sunt prezentate in diagrama din figura 22, de mai jos. Procesul incepe cu proiectarea conceptuala si se termina cu testarea. Procesul de proiectare necesita cunostinte de: fizica si de circuite, de metodologii de proiectare, in faza de proiectare conceptuala, si de performanta a circuitelor, in etapele de proiectarea a celulelor si de simulare.

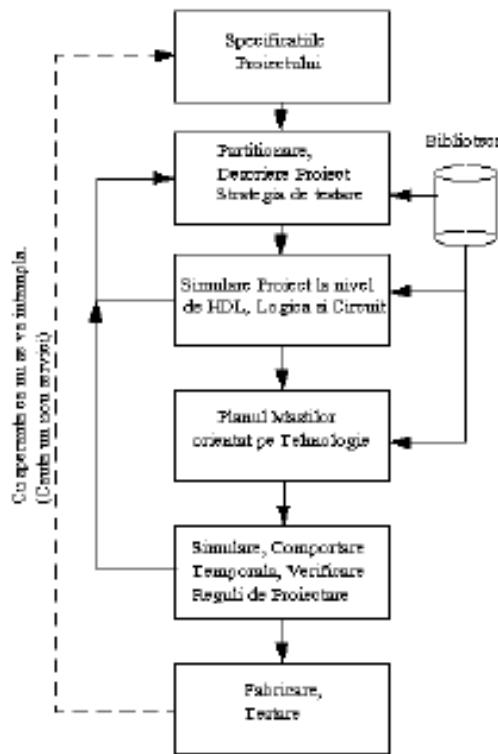


Fig. 22. Etapele obtinerii unui circuit integrat.

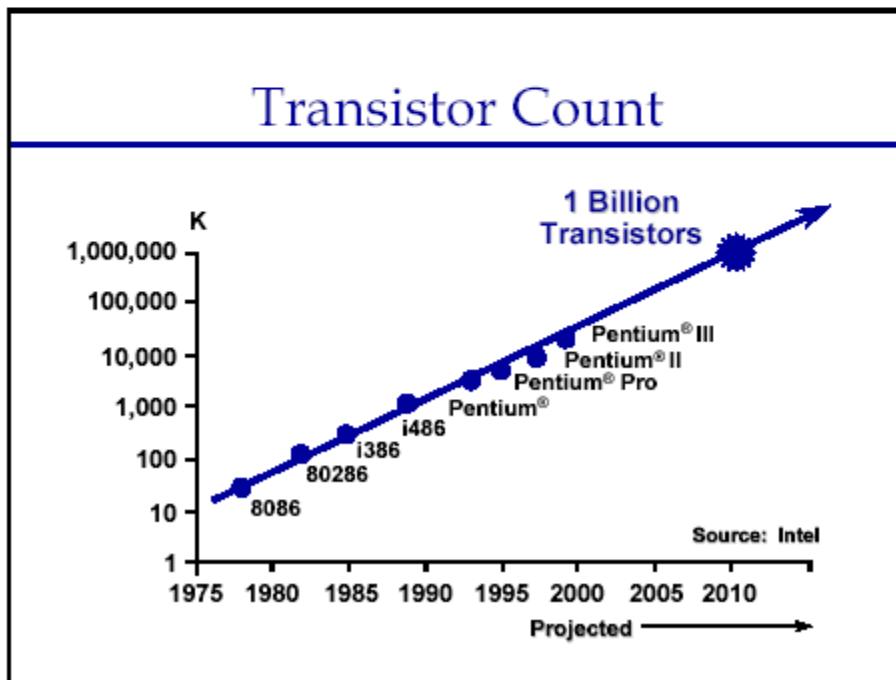
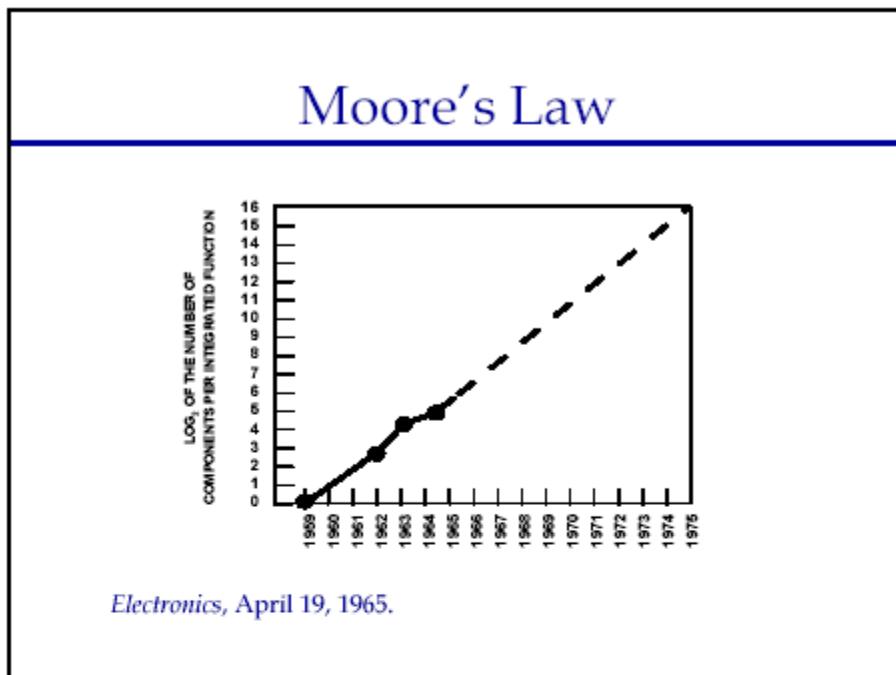
Secventa etapelor de proiectare este afectata, de asemenea, de nivelul de abstractie si de etapa la care proiectul este transferat catre producatorul de circuite integrate.

Captarea specificarii proiectului reprezinta una dintre cele mai dificile sarcini. De cele mai multe ori proiectul este specificat in limbaj natural, ceea ce poate induce un anumit grad de imprecizie, fara a mai mentiona imposibilitatea executiei acestei descrieri, in sensul simularii comportamentale. O specificare executabila permite simularea si verificarea functionalitatii.

Secventa de proiectare implica utilizarea unei biblioteci, care va contine modele functionale descrise la nivel inalt, modele de simulare corecte, cat si modelele unor circuite integrate reale. Standardizarea si utilizarea bibliotecilor, permit reutilizarea unor proiecte sau a unor parti de proiect, cat si reducerea timpului de proiectare.

Secventa de proiectare include numeroase bucle de reactie. De exemplu, dupa proiectare schemelor logice ale circuitelor se efectueaza simulari. Daca simularea pune in evidenta o eroare logica, proiectantul va reveni la schemele logice si va corecta eroare, dupa care va efectua din nou simularea. Secventa de proiectare descrisa mai sus poate sa capete aspecte usor diferite, in cadrul unor companii diferite.

Anexa 1. International Technology Roadmap for Semiconductors.



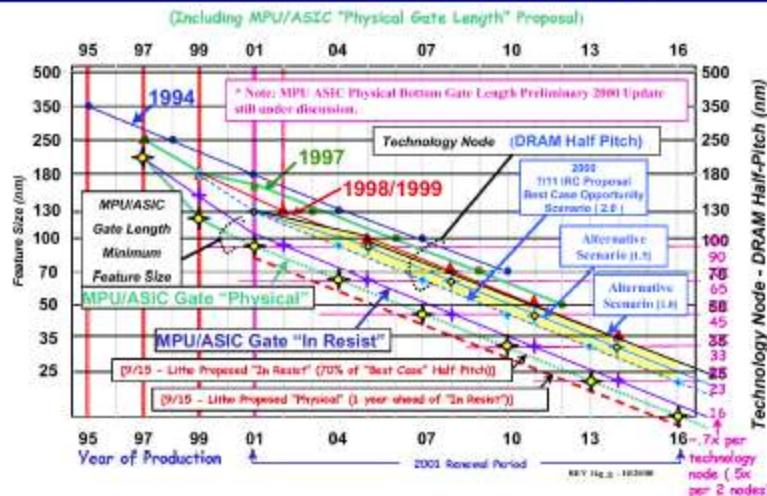
Technology Evolution (1997 data)							
International Technology Roadmap for Semiconductors							
Year of Introduction	1997	1999	2002	2005	2008	2011	2014
Channel length [nm]	200	140	100	70	50	35	25
Supply [V]	1.8-2.5	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	0.5-0.6	0.4
Metal layers	6	6-7	7	7-8	8-9	9	10
Max frequency [MHz], Local	750	1250	2100	3500	6000	10000	17000
Max µP power [W]	70	90	130	160	170	175	183

<http://www.sematech.org>, or <http://public.itrs.net>

Technology Evolution (2000 data)							
International Technology Roadmap for Semiconductors							
Year of Introduction	1999	2000	2001	2004	2008	2011	2014
Technology node [nm]	180		130	90	60	40	30
Supply [V]	1.5-1.8	1.5-1.8	1.2-1.5	0.9-1.2	0.6-0.9	0.5-0.6	0.3-0.6
Wiring levels	6-7	6-7	7	8	9	9-10	10
Max frequency [GHz], Local-Global	1.2	1.6-1.4	2.1-1.6	3.5-2	7.1-2.5	11-3	14.9-3.6
Max µP power [W]	90	106	130	160	171	177	186
Bat. power [W]	1.4	1.7	2.0	2.4	2.1	2.3	2.5

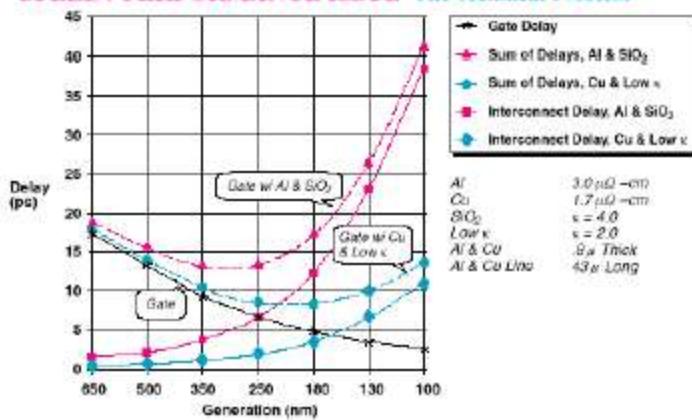
Node years: 2007/65nm, 2010/45nm, 2013/33nm, 2016/23nm

ITRS Technology Roadmap Acceleration Continues



Some Other Scares

SPEED / PERFORMANCE ISSUE *The Technical Problem*

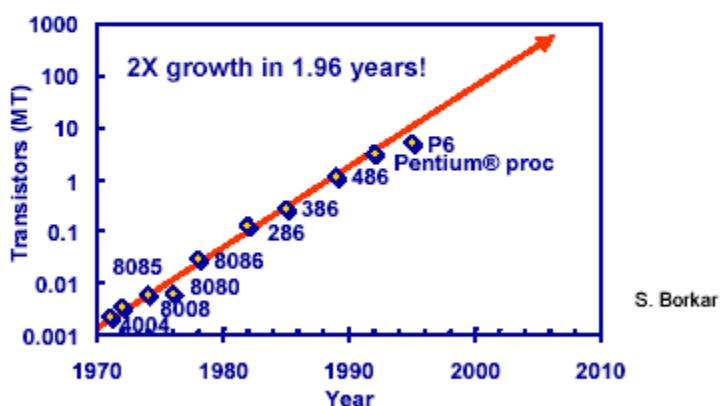


Technology Scaling

- Goals of scaling the dimensions by 30%:
 - » Reduce gate delay by 30% (increase operating frequency by 43%)
 - » Double transistor density
 - » Reduce energy per transition by 65% (50% power savings @ 43% increase in frequency)
- Technology generation spans 2-3 years, but μ P speed doubles every generation (not increased only by 43%)

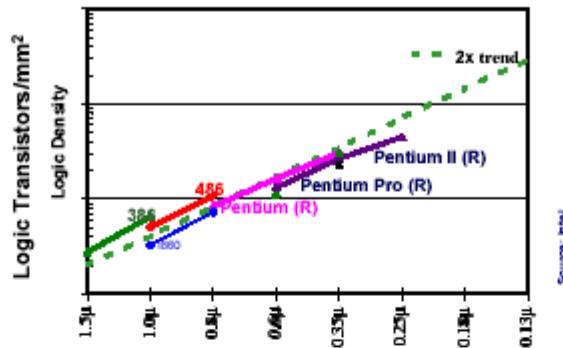
S. Borkar, IEEE Micro, July 1999.

Moore's law in Microprocessors



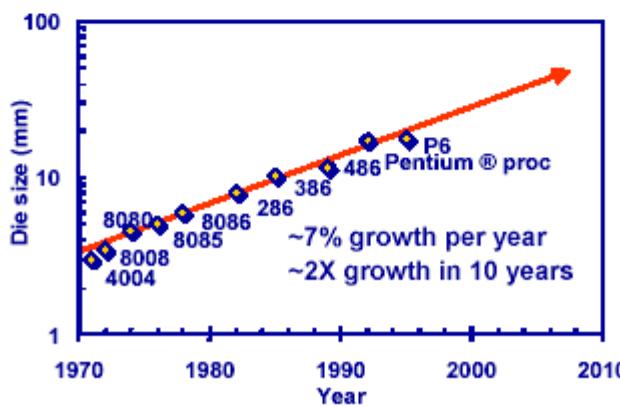
Transistors on Lead Microprocessors double every 2 years

Moore's Law - Logic Density



- ① Shrinks and compactions meet density goals
- ② New micro-architectures drop density

Die Size Growth

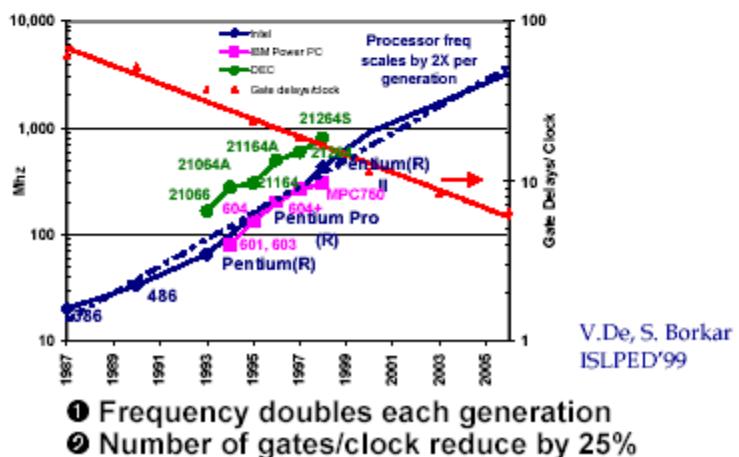


Die size grows by 14% to satisfy Moore's Law

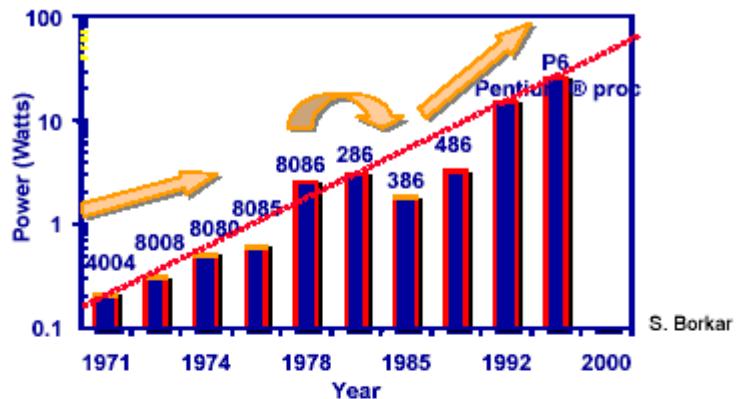
Frequency



Processor Frequency Trend

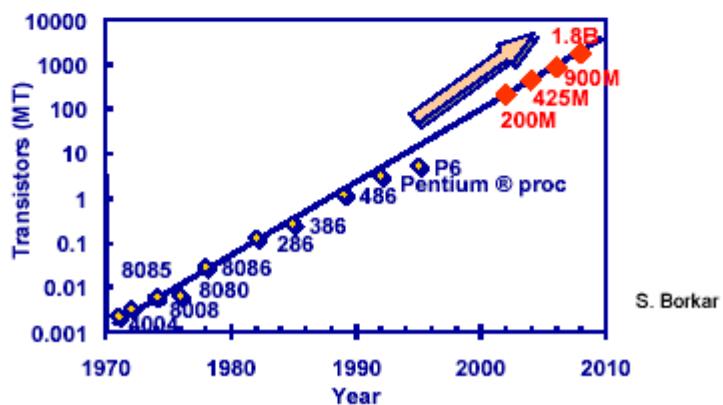


Power



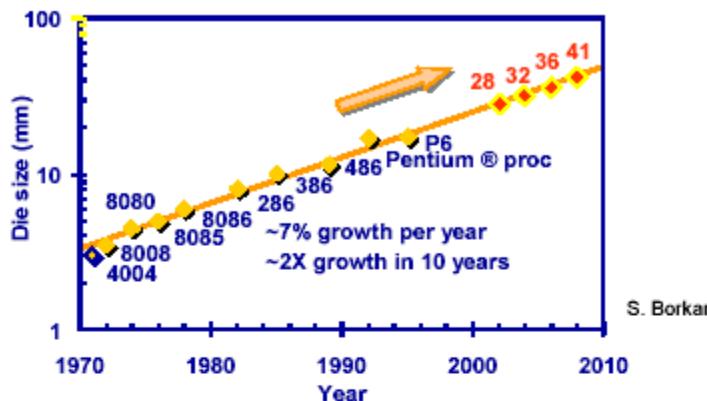
Lead Microprocessors power continues to increase

Obeying Moore's Law...



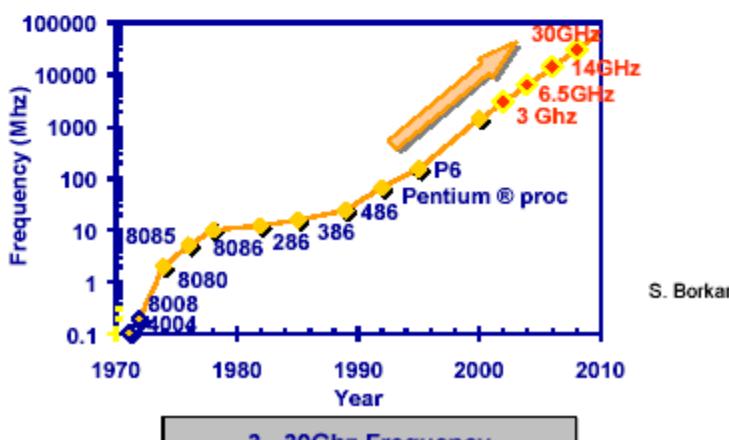
200M--1.8B transistors on the Lead Microprocessor

If die size increases



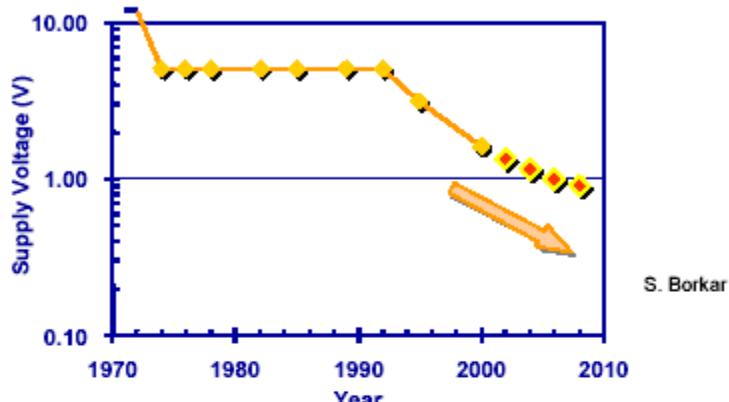
Die size will have to grow to 30 - 40mm

Frequency will increase

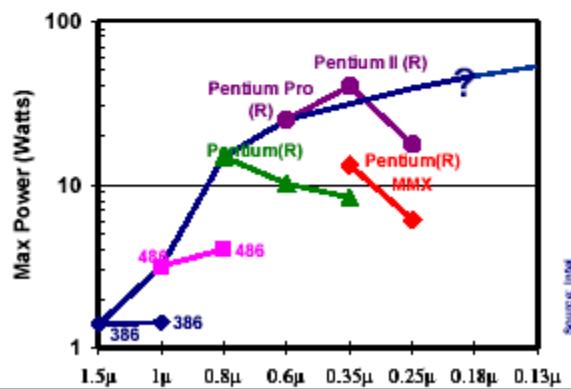


3 - 30Ghz Frequency

Supply voltage will continue to reduce



Processor Power



- ① Lead processor power increases every generation
- ② Compactions provide higher performance at lower power

Active power scaling

1. If $V_{cc} = 0.7$, and $\text{Freq} = (\frac{1}{0.7})$,

$$\text{Power} = CV^2f = (\frac{1}{0.7} \times 1.14^2) \times (0.7^2) \times (\frac{1}{0.7}) = 1.3$$

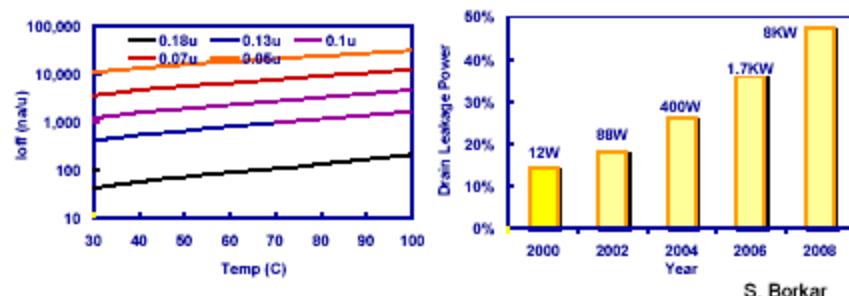
2. If $V_{cc} = 0.7$, and $\text{Freq} = 2$,

$$\text{Power} = CV^2f = (\frac{1}{0.7} \times 1.14^2) \times (0.7^2) \times (2) = 1.8$$

3. If $V_{cc} = 0.85$, and $\text{Freq} = 2$,

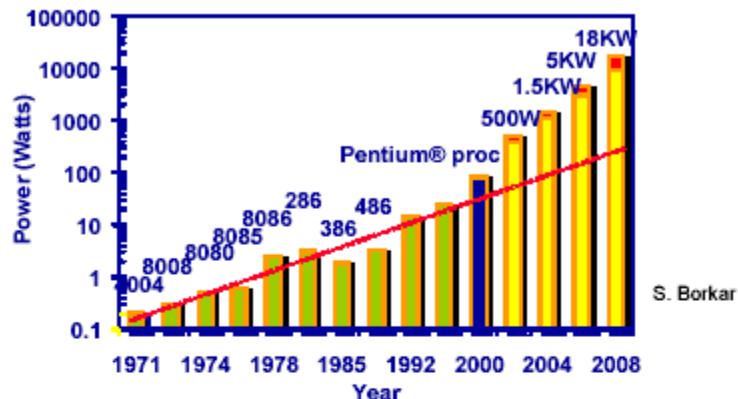
$$\text{Power} = CV^2f = (\frac{1}{0.7} \times 1.14^2) \times (0.85^2) \times (2) = 2.7$$

Leakage power increases



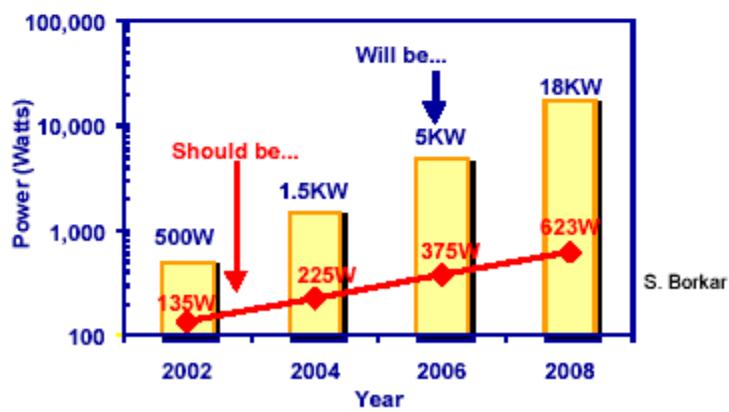
Drain leakage will have to increase to meet freq demand
Results in excessive leakage power

Power will be a problem

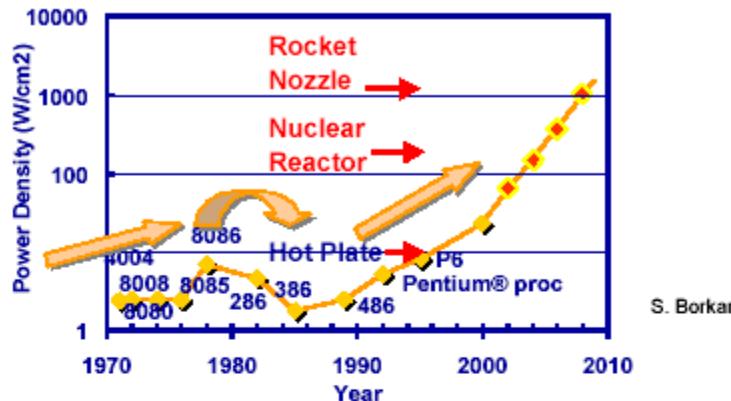


Power delivery and dissipation will be prohibitive

A closer look at the power

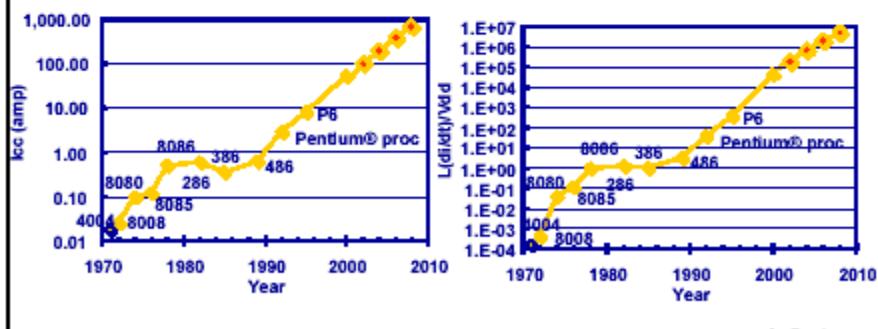


Power density will increase



Power density too high to keep junctions at low temp

Power delivery challenges

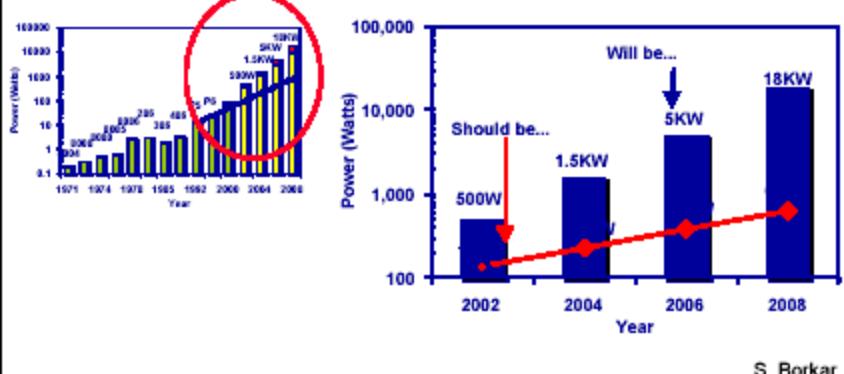


High supply currents at low voltage:
Challenges: IR drop and L(di/dt) noise

Moore's law challenge

- Double transistors every two years
» (Obey Moore's Law)
- Stay within the expected power trend
- Still deliver the expected performance

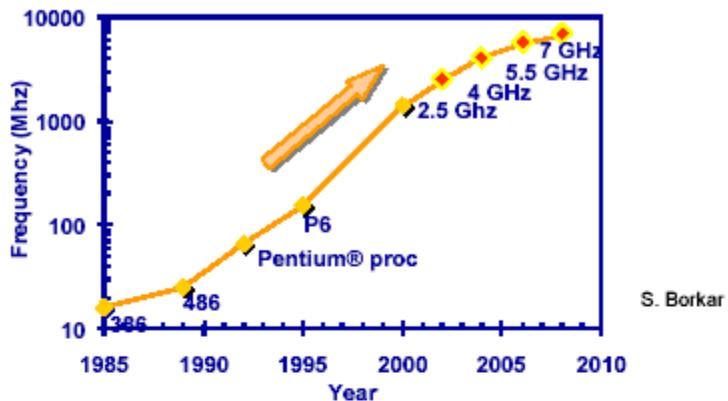
Expected power trend



S. Borkar

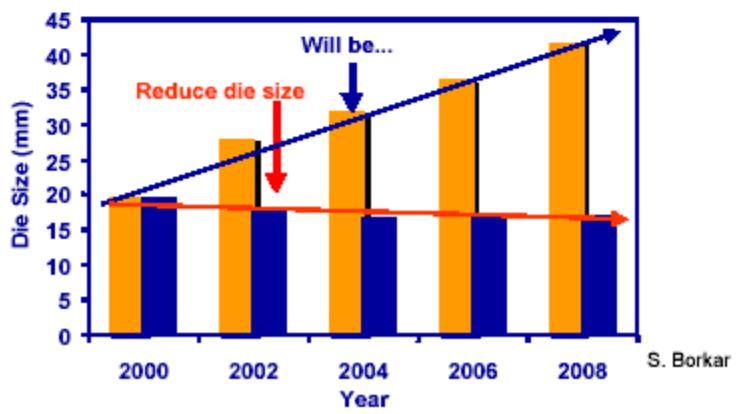
Goal: Restrict power to the expected trend

Restrict transistor leakage



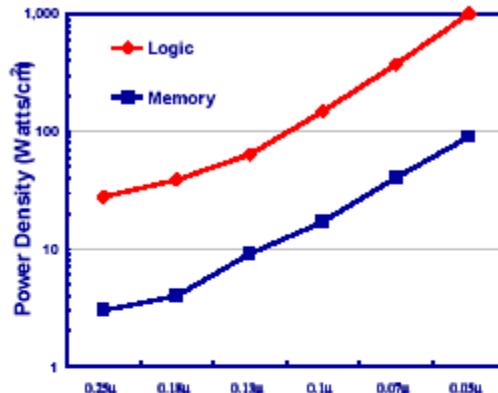
Reduce leakage \Rightarrow Frequency will not double every 2 years

Do not increase the die size



Restrict die size to ~ 20 mm

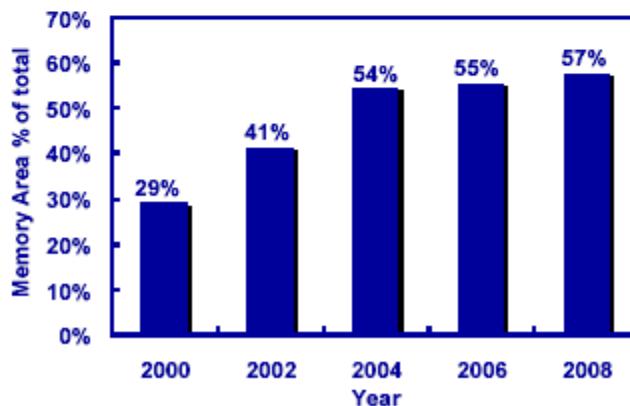
Memory has lower power density



S. Borkar

Exploit memory !

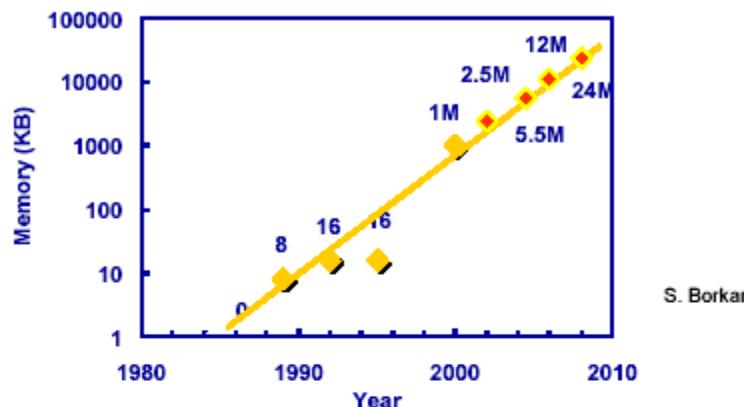
Increase memory area



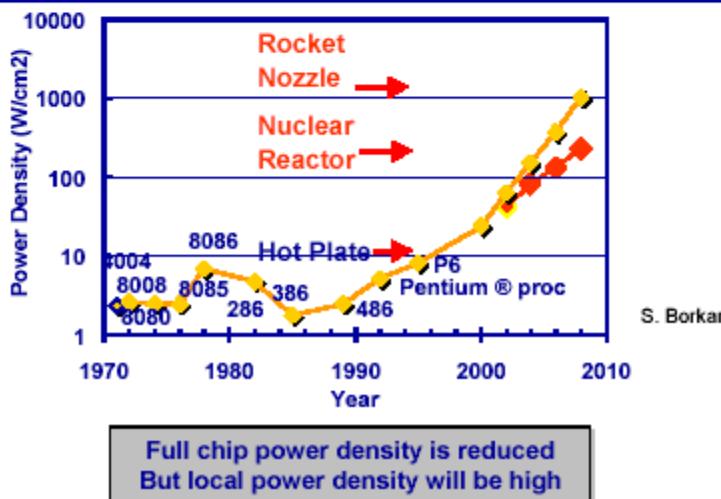
S. Borkar

Use > 50% die area in memory

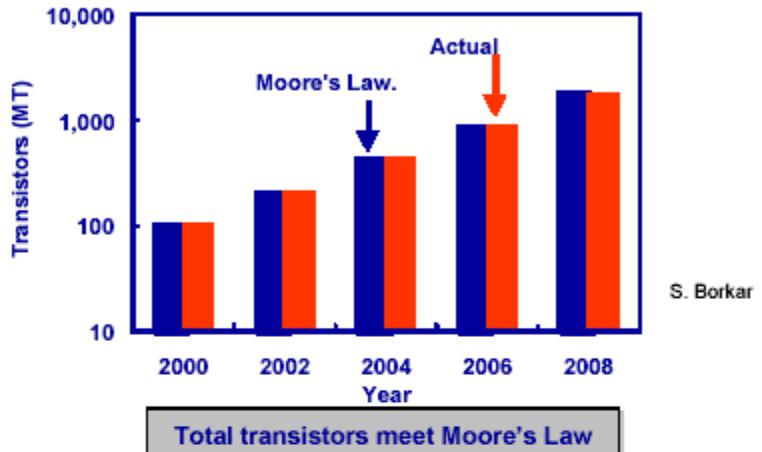
Total memory meets trend



Power density is reduced



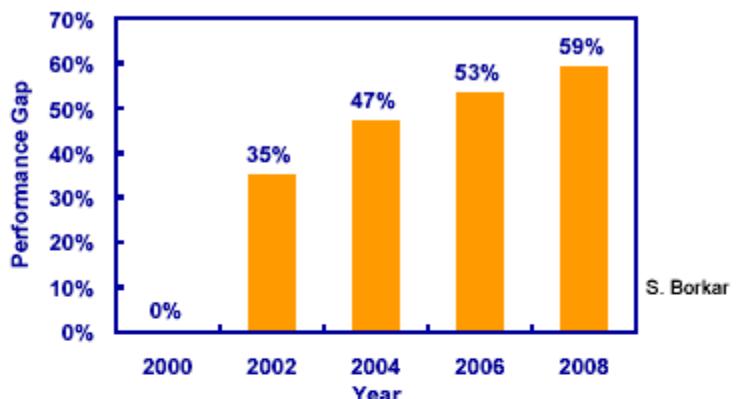
Still obey Moore's Law!



Too good to be true...

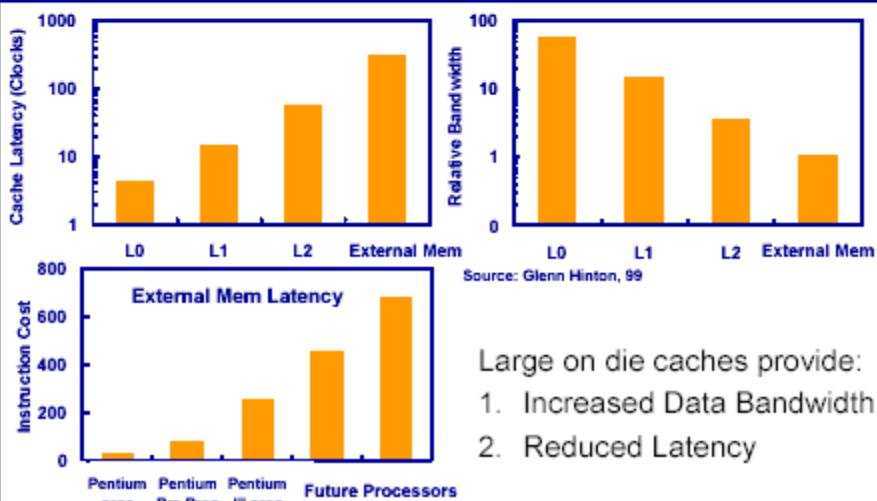
- Reduced transistor leakage
- Reduced frequency
- Decreased die size
- Increased memory, but reduced logic
- Does it deliver expected performance?

Reduced die size causes “Performance gap”

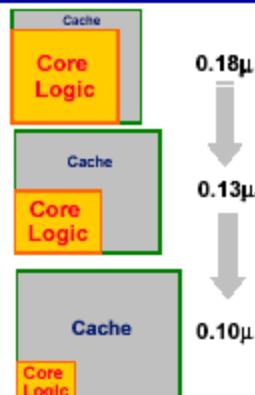


30-60% performance loss even after meeting Moore's Law

Large caches could improve performance

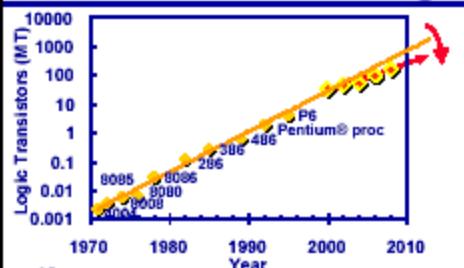


Cache size trend

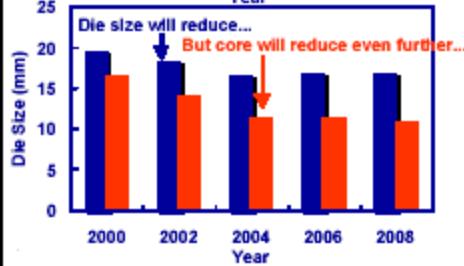


Cache memory area will dominate

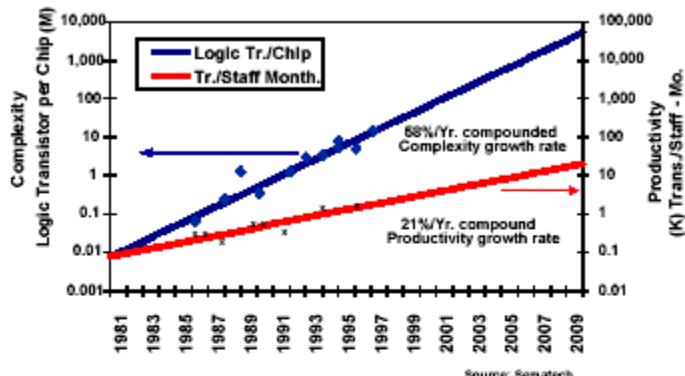
Other design challenges



- Modest increase in Logic transistors
- “Logic Core” size will decrease
- Tools/methodology for memories
- Interconnect RC may not be that big an issue



Productivity Trends



Complexity outpaces design productivity

Summary

- Moore's Law will be obeyed
- Barrier: Power delivery, dissipation, and density
- Exploit lower power density of memory-- creates performance gap
- Huge on die caches will help maintain performance trend
- Design challenges are different--not what we think they are!

ANEXA. 2 Exemple de procesoare modern Intel.

Intel® Pentium® M Processor

Processor	Clock Speed(s)	Intro Date(s)	Mfg. Process/ Transistors	Cache	Bus Speed	Core Voltage	Thermal Design Power (TDP)	Typical Use
Intel® Pentium® M Processor 770	2.13 GHz 2 GHz	Jan. 19, 2005 770	90 nm 140 million	2 MB L2 cache	533 MHz	1.260-1.3 72 V Max Perf. Mode 0.988V Battery Optimized Mode	27 W	Full-size and thin & light mobile PCs
760	1.86 GHz	760						
750	1.73 GHz	750						
740	1.60 GHz	740						
730		730						
Intel® Pentium® M Processor 765	2.10 GHz 2 GHz	Oct. 20, 2004 765	90 nm 140 million	2 MB L2 cache	400 MHz	1.276-1.340V Max Perf. Mode 0.988V Battery Optimized Mode	21 W	Full-size and thin & light mobile PCs
755	1.80 GHz							
745	1.70 GHz	June 23, 2004 725						
735	1.60 GHz							
725	1.50 GHz	715						
715								
Intel® Pentium® M Processor	1.70 GHz 1.60 GHz 1.50 GHz 1.40 GHz 1.30 GHz	June 2, 2003 1.70 GHz Mar. 12, 2003 1.60 GHz 1.50 GHz 1.40 GHz 1.30 GHz	0.13-micron 77 million	1 MB L2 cache	400 MHz	1.484V in Max. Perf. Mode 0.956V Battery Optimized Mode (1.40-1.70 GHz) 1.39V in Max Perf. Mode 0.96V in Battery Optimized Mode (1.30 GHz)	24.5 W	Full-size and thin & light mobile PCs

Marking of modern Intel processors

Just to warm up - a few words on the marking of Intel's modern processors. I presume, the pinout of the new processors is clear - it is the well-known and familiar LGA775. All the more or less modern Intel CPU models - both high-end and mainstream - have long migrated to LGA775.

But the marking of CPUs has undergone serious changes and now is made up of 5 elements - a combination of the letter prefix followed by a 4-digit numerical index. It makes sense decoding the 5-digit marking

of Intel CPUs with the letter index which identifies the TDP of the processor, without any relation to the form factor.

Letter indices in modern 5-digit marking of Intel's CPUs	
X	TDP over 75W
E	TDP 50W and higher
T	TDP within 25W to 49W
L	TDP within 15W to 24W
U	TDP about 14W and less

In its turn, the 4-digit index also has the following meaning: the higher the 4-digit number in the CPU marking, the higher performance and power consumption it offers. At the same time, the first digit means the belonging of the chip to a certain product family, and the second one stands for the respective ranking within the family. Therefore, the greater the figure is, the higher is the performance of the chip.

Here are a few examples of how the marking of modern processors looks and what stands behind the notation:

- Core 2 Extreme X6800 – 2.93 GHz, 4 MB L2 cache, 1066 MHz FSB
- Core 2 Duo E6600 – 2.4 GHz, 4 MB L2 cache, 1066 MHz FSB
- Core 2 Duo E6400 – 2.13 GHz, 2 MB L2 cache, 1066 MHz FSB
- Core Duo T2500 – 2 GHz, 2 MB L2 cache, 667 MHz FSB
- Core Duo U2500 – 1.06 GHz, 2 MB L2 cache, 533 MHz FSB

Then, there follows a list of key connectors and contact systems for Intel processors. Incomplete, of course. In future, it is planned to add all the types of contact systems for Intel CPUs, including the options for onboard soldering.

Key connectors/contact systems of Intel CPUs	
Socket-W	Socket 423 (for desktop PCs. Taken out of production)
Socket-N	Socket 478 (for desktop PCs. Taken out of production)
Socket-F	Socket 603 (for server systems)
Socket-T	LGA775 (Land Grid Array. T - Tejas)

Socket-C

The socket was originally planned for use with Cedar Mill. Cancelled.

Marking of modern Intel processors

The table collects the so-called "working names" of Intel CPUs for desktop PCs which are used unofficially before the date of announcement, together with the matching names of Intel's micro-architectures and the retail names of the chips.

Processors for desktop PCs	
NetBurst micro-architecture (Pentium 4)	
CedarMill	65 nm Pentium 4
CedarMill-V	65 nm Celeron
Smithfield	90 nm Pentium D 8xx
Presler	65 nm Pentium D 9xx
Core (Conroe) micro-architecture	
Conroe	65 nm Core (Merom)
Allendale	Advanced Conroe core
Millville	Next generation of Conroe core
Wolfdale	Next generation of Conroe core
Nehalem	New generation of the Conroe architecture with support for EM64T

The same - working names of Intel CPUs with the matching names of micro-architectures and the retail names of the chips, for mobile PCs.

Intel CPUs for mobile and economy systems	
Banias micro-architecture	
Banias	0.13 mk Pentium M/Celeron M
Dothan	90 nm generation of Pentium M/Celeron M
Yonah-2P	65 nm generation dual-core CPUs
Core (Yonah, Merom) micro-architecture	
Yonah-DC (Yonah-2P)	Dual-core 65 nm version of Core Duo

Yonah-SC (Yonah-1P)	Single-core 65 nm version of Core Solo CPUs
Yonah-2D	Alternative name Yonah-DC (Dothan Core)
Merom	New 65 nm generation of Core micro-architecture
Penryn	45 nm version of Merom core
Gilo	A new generation of the Core micro-architecture expected after the Merom

For reference - the same regarding the names Intel CPUs for server systems. Here is a complete list of CPUs of the recent years, including chips of the IA64 Itanium architecture.

Intel CPUs for server systems and workstations

Pentium 4 architecture

Foster	0.18 mk Xeon (based on the Willamette core)
Foster MP	0.18 mk Xeon MP with L3 cache
Gallatin	0.13 mk Xeon MP with L3 cache
Prestonia	0.13 mk Xeon DP
Nocona	90 nm Xeon DP (based on the Prescott core)
Irwindale	90 nm Xeon DP with 2 MB L2 cache
Cranford	90 nm Xeon MP (based on the Nocona core) with 1 MB L2 cache
Potomac	90 nm Xeon MP with L3 cache
Jayhawk	Xeon DP based on Tejas (cancelled)
Paxville MP	Dual-core 90 nm Xeon MP
Paxville DP	Dual-core 90 nm Xeon DP
Dempsey	Dual-core 90 nm Xeon DP (2 cores per chip)
Tulsa	65 nm version of dual-core server CPUs

Conroe/Merom/Yonah architecture

Sossaman	Dual-core 65 nm DP Xeon LV (on the base of the Yonah DC core)
Woodcrest	Dual-core 65 nm Xeon DP
Whitefield	Generic design of 65 nm 4-core Xeon MP CPU (cancelled)
Tigerton	Quad-core 65 nm Xeon MP

Dunnington	Quad-core 45 nm Xeon MP
Clovertown	Quad-core 65 nm CPU
IA-64 architecture	
Merced	First generation of the Itanium architecture
McKinley	0.18 mk Itanium 2
Madison	0.13 nm Itanium 2
Deerfield	0.13 nm LV Itanium 2
Madison 9M	0.13 nm Itanium 2 with 9 MB L3 cache
Fanwood	0.13 nm Itanium 2 with 4 MB L3 cache
LV Fanwood	0.13 nm LV Itanium 2
Chivano	New generation of the IA-64 architecture based on the Madison (cancelled)
Montecito	Dual-core 90 nm version of Itanium 2 MP/DP
Millington	Dual-core 90 nm version of IA-64 DP (cancelled)
LV Millington	90 nm LV DP IA-64 (cancelled)
Montvale	Dual-core 65 nm IA-64
Tukwila	Quad-core 65 nm MP IA-64
Tanglewood	Updated design of Tukwila
Dimona	65 nm version of DP IA-64
Poulson	8-core 45 nm MP IA-64

In view of Intel's new marketing strategy aimed primarily at the promotion of comprehensive platform solutions (and not specific components as before), it is indeed topical to include a list of working names of platforms within which Intel is generating hardware-software systems for desktop, mobile, and server systems.

Intel's platform technologies	
Desktop platforms	
Lyndon	Corporate platform of year 2005
Anchor Creek	Entertainment household platform of year 2005
Averill	Corporate platform of year 2006

Bridge Creek	Entertainment household platform of year 2006
Intel Viiv	Entertainment household platform
Intel vPro	Corporate platform
Mobile system platforms	
Carmel	Centrino, based on Odem/Montara
Sonoma	Centrino-2005, based on the Alviso chipset
Napa	Centrino-2006, based on the Calistoga chipset
Napa SC	Centrino-2006, single-core design based on the Calistoga
Napa DC	Centrino Duo - 2006, dual-core version based on Calistoga
Napa 64	Napa version based on the Merom core
Santa Rosa	Version of Centrino Duo (Centrino Pro?) based on Crestline, 2007
Montevina	Improved economy version Santa Rosa, plus SFF, DX9, HDCP for HDMI, DVI and UDI; Robson 2.0, VT, and Intel Trusted Execution Technology; support for HD DVD and Blu-ray; Penryn CPUs, chipsets Cantiga GM and PM. Second half of 2008
Platforms for server systems and workstations	
Truland	Platform IA-32 MP for servers based on the NetBurst architecture, year 2005
Gallaway	Platform IA-32 UP for workstations, year 2005
Bensley	Platform IA-32 DP for servers, year 2006
Bensley-VS	Platform IA-32 DP for servers, year 2006
Glidewell	Platform IA-32 DP for workstations, year 2006
Kaylo	Platform IA-32 UP for servers, year 2006
Wyloway	Platform IA-32 UP for workstations, year 2006
Reidland	Platform IA-32 MP for servers, year 2007
Richford	Platform IA-64 MP for servers based on the Tukwila, year 2008
Thurley	Platform for mainstream IA-32 MP servers based on Gainestown CPUs with integrated 6-channel DDR3 800/1066/1333 controller, with support for 42 PCI Express lanes (36 of them - PCI E 2.0), based on the Tylersburg-DP chipset with support for six SAS/SATA 3 GB/s ports with the hardware RAID 5 (Sunrise Lake), Dual GbE (Zoar/Adorami) and 10 GbE (Oplin), iAMT 3.0. Second half of 2008

It is not easy to make a decision regarding Intel's chipsets which are of no real interest to the modern buyer. The modern buyer is heterogeneous.

Intel chipsets	
For Intel Pentium 4/Celeron	
Tehama	Intel 850
Brookdale-SDRAM	Version of Intel 845 with support for SDRAM
Brookdale-DDR	Version of Intel 845 with support for DDR SDRAM
Brookdale-E	Intel 845E
Brookdale-G	Intel 845G
Brookdale-GL	Intel 845GL
Tehama-E	Intel 850E
Tulloch	Chipset with support for RDRAM (cancelled)
Canterwood	Intel 875P
Springdale-PE	Intel 865PE
Springdale-G	Intel 865G
Springdale-P	Intel 865P
Alderwood	Intel 925X
Alderwood 1066	Intel 925XE (with support for 1066 MHz FSB)
Grantsdale-P	Intel 915P
Grantsdale-G	Intel 915G
Grantsdale-GV	Intel 915GV
Grantsdale-GL	Intel 910GL
Glenwood	Intel 955X
Wyloway	Intel 975X/XE Express for high-end PC
Lakeport-P	Intel 945P
Lakeport-G	Intel 945G
Broadwater GC	Integrated Intel G965 for household PC
Broadwater P	Discrete Intel P965 for household PC
Broadwater G	Integrated Intel Q965 for corporate PC

Broadwater GF	Integrated Intel Q963 for corporate PC
Bearlake	Future model of year 2007 (with support for DDR3)
For mobile systems	
Banister	Intel 440MX
Almador-M	Intel 830M (for Tualatin M)
Greendale	Mobile chipset with support for RDRAM (cancelled)
Brookdale-M	Intel 845MP
Brookdale-MZ	Intel 845MZ
Odem	Intel 855PM
Montara-GM	Intel 855GM
Montara-GM+	Intel 855GME
Montara-GML	Intel 852GM
Montara-GT	Intel 852GME
Montara-P	Intel 852PM
Montara-GML+	Intel 852GMV
Alviso-GM	Intel 915GM
Alviso-PM	Intel 915PM
Alviso-GMS	Intel 915GMS
Alviso-GML	Intel 910GML
Calistoga-GM	Intel 945GM
Calistoga-PM	Intel 945PM
Calistoga-GMS	Intel 945GM
Calistoga-GML	Intel 945GML
Crestline	Chipset of year 2006 for the Merom core
Calexico	Intel PRO/Wireless 2100 adapter
Calexico2	Intel PRO/Wireless 2200 adapter
Golan	Intel PRO/Wireless 3945ABG adapter
Gaston	Wireless adapter of year 2005 (cancelled)
For server systems and workstations	
Carmel	Intel 840

Colusa	Intel 860
Canterwood-ES	Intel E7210 (for UP servers)
Copper River	Intel E7221 (for UP servers with support for PCI Express)
Granite Bay	Intel E7205 (for UP workstations)
Plumas	Intel E7500/7501 (for DP systems)
Placer	Intel E7505 (for DP workstations)
Lindenhurst	Intel E7520 (for DP workstations with support for PCI Express)
Lindenhurst-VS	Intel E7320 (for DP systems with support for PCI Express)
Twincastle	Intel E8500 (for MP servers with support for PCI Express)
Tumwater	Intel E7525 (for DP workstations with support for PCI Express)
Blackford	Intel 5000P (chipset for DP servers with support for FB-DIMM)
Blackford-VS	Intel 5000V (chipset for DP servers with support for FB-DIMM)
Greencreek	Intel 5000X (chipset for DP workstations with support for FB-DIMM)
Mukilteo	Intel E7230 (chipset for UP servers with support for PCI Express)
Mukilteo-2/P	Year 2006 version for UP servers with support for PCI Express

Intel processors for desktop PCs

Quad-core Intel Core 2 Quad CPUs for desktop PCs

Intel Core 2 Quad CPUs (working name Kentsfield) are made on the base of the 4-core Intel Core architecture, and from the technical viewpoint are two dual-core Conroe cores interlinked on a single substrate. Core 2 Extreme QX6xxx chips offer 8 MB of total L2 cache (4 MB of distributed L2 cache per each pair of cores), which provides support for four physically independent threads and in theory allows for additional performance boost if there is support from the software part of the platform. Core 2 Extreme QX6xxx CPUs are compatible to motherboards based on the i975X and other chipsets which offer a power supply module and meet the VRM 11 specifications, as well as have the matching BIOS firmware.

Core2 Quad CPUs offer all the key advantages of the new Intel Core micro architecture, including support for Enhanced Intel SpeedStep

Technology, Intel Wide Dynamic Execution, Intel Intelligent Power Capability, Intel Advanced Smart Cache, Intel Advanced Digital Media Boost, and Intel Smart Memory Access. For details of novelties in the Intel Core micro architecture, read our article [Evolution of the multi-core Intel Core processor architecture: Conroe, Kentsfield....](#)

4-core Intel CPUs- Kentsfield core									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
4-core Intel Core 2 Quad CPUs									
Q6700	2.66 GHz	1066 MHz	4 MB x2 (distr.)	65 nm	+	-	+	+	LGA775
Q6600	2.40 GHz	1066 MHz	4 MB x2 (distr.)	65 nm	+	-	+	+	LGA775

Multi-core Intel Core 2 Extreme CPUs for desktop PCs

The Intel Core 2 Extreme series includes both the dual-core (X6800) and quad-core version (QX6700). In fact, Intel Core 2 Extreme X6800 is different from the higher-end models of Intel Core 2 Duo E6xxx having the Conroe core in only higher clock speeds, which is caused by the high TDP - over 75 W, and automatically moved this processor to a class of chips with the letter index X.

From the technical viewpoint, the 4-core Core 2 Extreme QX6700 appears to be two dual-core chips of the Core 2 Duo E6700 (Conroe) class integrated within a single package. Therefore, the overall L2 cache size makes 8 MB, or 4 MB per each pair of cores, with the remaining key specifications being similar to Core 2 Duo E6700.

2-core Intel CPUs- Conroe core

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
2-core Intel Core 2 Extreme									
QX6700	2.66 GHz	1066 MHz	4 MB (distr.)	65 nm	+	-	+	+	LGA775
X6800	2.93 GHz	1066 MHz	4 MB (distr.)	65 nm	+	-	+	+	LGA775

Dual-core Intel Core 2 Duo CPUs for desktop PCs

The Intel Core 2 Duo series based on the modern Intel Core micro architecture is a family of Intel's most interesting processors in terms of performance per watt and efficiency.

Dual-core Intel Core 2 Duo chips for desktop PC (Conroe core) offer 4 MB or 2 MB of distributed L2 cache, FSB up to 1066 MHz, support Intel Wide Dynamic Execution, Intel Intelligent Power Capability, Intel Smart Memory Access, Intel Advanced Smart Cache, and Intel Advanced Digital Media Boost.

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
2-core Intel CPUs- Conroe core									
E6700	2.67 GHz	1066 MHz	4 MB (distr.)	65 nm	+	-	+	+	LGA775
E6600	2.40 GHz	1066 MHz	4 MB (distr.)	65 nm	+	-	+	+	LGA775
E6400	2.13 GHz	1066 MHz	2 MB (distr.)	65 nm	+	-	+	+	LGA775

E6300	1.86 GHz	1066 MHz	2 MB (distr.)	65 nm	+	-	+	+	LGA775
E4300	1.80 GHz	800 MHz	2 MB (distr.)	65 nm	-	-	+	+	LGA775

Dual-core Intel Pentium Extreme Edition for desktop PCs

Pentium Extreme Edition CPUs (or Pentium XE) became the first 2-core chips for desktop platforms made by Intel. Intel Pentium Extreme Edition CPUs based on the Smithfield core offer support for the 1066 MHz system bus, are equipped with 4 MB or 2 MB of L2 cache, 2 MB or 1 MB per each core. Pentium XE processors support the Intel Extended Memory 64 technology (support for 32-bit and 64-bit addressing), Hyper-Threading (up to four software threads simultaneously), Execute Disable Bit (if supported by the operating systems, protects against viruses that exploit memory buffer overflow errors).

Originally, Pentium XE processors were positioned for use with systems based on the Intel 955X Express chipset with support for up to 8 GB 2-channel ECC DDR2-667/533 memory with error correction (ECC).

2-core Intel CPUs- Smithfield core

2-core Intel Pentium Extreme Edition									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Pentium XE 965	3.73 GHz	1066 MHz	2 MB x2	65 nm	+	+	+	+	LGA775
Pentium XE 955	3.46 GHz	1066 MHz	2 MB x2	65 nm	+	+	+	+	LGA775
Pentium XE 840	3.20 GHz	800 MHz	1 MB x 2	90 nm	-	+	+	+	LGA775

Dual-core Intel Pentium D for desktop PCs

First announced in May 2005, the 2-core Intel Pentium D processors are currently presented with models made following the 65-nm and 90-nm process technologies. The chips are equipped with 2 MB or 1 MB L2 cache per each core (4 MB altogether), 2 x 16 K of L1 cache per each core, and 2 Execution Trace Caches which are able storing up to 12K of decoded micro-ops. Intel Pentium D CPUs support the Intel Extended Memory 64 (Intel EM64T), Execute Disable Bit, Enhanced Intel SpeedStep (in a number of chips), and the SSE3 instruction set. Essentially, Intel Pentium D processors are chips based on the NetBurst architecture - two Pentium 4 per chip.

The main distinction of 2-core Pentium D processors from the Pentium XE series in the lack of support for the Hyper-Threading Technology.

2-core Intel CPUs- Smithfield core									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
2-core Intel Pentium D									
Pentium D 960	3.60 GHz	800 MHz	2 MB x2	65 nm	+	-	0	0	LGA775
Pentium D 950	3.40 GHz	800 MHz	2 MB x2	65 nm	+	-	0	0	LGA775
Pentium D 945	3.40 GHz	800 MHz	2 MB x2	65 nm	-	-	+	+	LGA775
Pentium D 940	3.20 GHz	800 MHz	2 MB x2	65 nm	+	-	+	+	LGA775
Pentium D 930	3 GHz	800 MHz	2 MB x2	65 nm	+	-	+	+	LGA775
Pentium D 925	3 GHz	800 MHz	2 MB x2	65 nm	-	-	+	+	LGA775

Pentium D 920	2.80 GHz	800 MHz	2 MB x2	65 nm	+	-	+	+	LGA775
Pentium D 915	2.80 GHz	800 MHz	2 MB x2	65 nm	-	-	+	+	LGA775
Pentium D 840	3.20 GHz	800 MHz	1 MB x 2	90 nm	-	-	+	+	LGA775
Pentium D 830	3 GHz	800 MHz	1 MB x 2	90 nm	-	-	+	+	LGA775
Pentium D 820	2.80 GHz	800 MHz	1 MB x 2	90 nm	-	-	+	+	LGA775
Pentium D 805	2.66 GHz	533 MHz	1 MB x 2	90 nm	-	-	+	+	LGA775

Intel Pentium 4 6xx for desktop PCs

Intel Pentium 4 600 (Prescott 2M) CPUs offer support for the Hyper-Threading and 2 MB L2 cache. The series includes CPUs of clock speeds within 3.0 GHz to 3.80 GHz. The declared TDP of Intel Pentium 4 660 and 670 processors is 115 W, others - 84 W.

Intel Pentium 4 series 600 processors are based on the Intel NetBurst architecture, offer support for the FSB 800 MHz, Execute Disable Bit, Enhanced Intel Speedstep Technology (EIST), Intel EM64T. The main distinction of Pentium 4 series 600 processors (Prescott 2M core) from the already phased-out Pentium 4 500 series (Prescott core) is in the doubled capacity of L2 cache at lower clock speeds.

Intel CPUs - Prescott 2M core									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Intel Pentium 4 6xx									
Pentium 4,672	3.80 GHz	800 MHz	2 MB	90 nm	+	+	+	+	LGA775

Pentium 4,670	3.80 GHz	800 MHz	2 MB	90 nm	-	+	+	+	LGA775
Pentium 4,662	3.60 GHz	800 MHz	2 MB	90 nm	+	+	+	+	LGA775
Pentium 4,661	3.60 GHz	800 MHz	2 MB	65 nm	-	+	+	+	LGA775
Pentium 4,660	3.60 GHz	800 MHz	2 MB	90 nm	-	+	+	+	LGA775
Pentium 4,651	3.40 GHz	800 MHz	2 MB	65 nm	-	+	+	+	LGA775
Pentium 4,650	3.40 GHz	800 MHz	2 MB	90 nm	-	+	+	+	LGA775
Pentium 4,641	3.20 GHz	800 MHz	2 MB	65 nm	-	+	+	+	LGA775
Pentium 4,640	3.20 GHz	800 MHz	2 MB	90 nm	-	+	+	+	LGA775
Pentium 4,631	3 GHz	800 MHz	2 MB	65 nm	-	+	+	+	LGA775
Pentium 4,630	3 GHz	800 MHz	2 MB	90 nm	-	+	+	+	LGA775

Intel Pentium 4 5xx for desktop PCs

The main distinction of Pentium 4 series 500 series (Prescott core) from the higher-end Pentium 4 600 series (Prescott 2M core) is in the L2 cache size twice as little. Intel Pentium 4 series 500 processors are made following the norms of the 90 nm process technology in LGA 775 packages, contain about 125 mln transistors. The declared TDP of the Pentium 4 550 and higher processors is 115 W, other chips - 84 W.

Intel CPUs - Prescott core

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket

Intel Pentium 4 5xx									
Pentium 4,571	3.80 GHz	800 MHz	1 Mb	90 nm	-	+	+	+	LGA775
Pentium 4,570J	3.80 GHz	800 MHz	1 Mb	90 nm	-	+	-	+	LGA775
Pentium 4,561	3.60 GHz	800 MHz	1 Mb	90 nm	-	+	+	+	LGA775
Pentium 4,560J	3.60 GHz	800 MHz	1 Mb	90 nm	-	+	-	+	LGA775
Pentium 4,560	3.60 GHz	800 MHz	1 Mb	90 nm	-	+	-	-	LGA775
Pentium 4,551	3.40 GHz	800 MHz	1 Mb	90 nm	-	+	+	+	LGA775
Pentium 4,550J	3.40 GHz	800 MHz	1 Mb	90 nm	-	+	-	+	LGA775
Pentium 4,550	3.40 GHz	800 MHz	1 Mb	90 nm	-	+	-	-	LGA775
Pentium 4,541	3.20 GHz	800 MHz	1 Mb	90 nm	-	+	+	+	LGA775
Pentium 4,540J	3.20 GHz	800 MHz	1 Mb	90 nm	-	+	-	+	LGA775
Pentium 4,540	3.20 GHz	800 MHz	1 Mb	90 nm	-	+	-	-	LGA775
Pentium 4,531	3 GHz	800 MHz	1 Mb	90 nm	-	+	+	+	LGA775
Pentium 4,530J	3 GHz	800 MHz	1 Mb	90 nm	-	+	-	+	LGA775
Pentium 4,530	3 GHz	800 MHz	1 Mb	90 nm	-	+	-	-	LGA775
Pentium 4,521	2.80 GHz	800 MHz	1 Mb	90 nm	-	+	+	+	LGA775
Pentium 4,520J	2.80 GHz	800 MHz	1 Mb	90 nm	-	+	-	+	LGA775

Pentium 4,520	2.80 GHz	800 MHz	1 Mb	90 nm	-	+	-	-	LGA775
Pentium 4,519K	3.06 GHz	533 MHz	1 Mb	90 nm	-	-	+	+	LGA775
Pentium 4 519J	3.06 GHz	533 MHz	1 Mb	90 nm	-	-	-	+	LGA775
Pentium 4,516	2.93 GHz	533 MHz	1 Mb	90 nm	-	-	+	+	LGA775
Pentium 4,515	2.93 GHz	533 MHz	1 Mb	90 nm	-	-	-	-	LGA775
Pentium 4,511	2.80 GHz	533 MHz	1 Mb	90 nm	-	-	+	+	LGA775
Pentium 4,506	2.66 GHz	533 MHz	1 Mb	90 nm	-	-	+	+	LGA775
Pentium 4 505	2.66 GHz	533 MHz	1 Mb	90 nm	-	-	-	-	LGA775

Intel Celeron D for desktop PCs

Intel Celeron D processors of clock speeds up to 3.36 GHz (Celeron D 360) offer the options typical for most Prescott processors, with the difference in lower FSB=533 MHz supported. The L2 cache size of "old" 90 nm models is 256 K, whereas new models made following the 90-nm norms - 512 K.

Intel Celeron D series processors are produced in two makes of the housing design - LGA775 and mPGA478. The whole line Celeron D supports the SSE3 instruction set, and a number of models support the EM64T.

Intel CPUs - Prescott core

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket

Intel Celeron D									
Celeron D 360	3.46 GHz	533 MHz	512 K	65 nm	-	-	+	+	LGA775
Celeron D 356	3.33 GHz	533 MHz	512 K	65 nm	-	-	+	+	LGA775
Celeron D 355	3.33 GHz	533 MHz	256 K	90 nm	-	-	+	+	LGA775
Celeron D 352	3.20 GHz	533 MHz	512 K	65 nm	-	-	+	+	LGA775
Celeron D 351	3.20 GHz	533 MHz	256 K	90 nm	-	-	+	+	LGA775
Celeron D 350	3.20 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478
Celeron D 347	3.06 GHz	533 MHz	256 K	65 nm	-	-	+	+	LGA775
Celeron D 346	3.06 GHz	533 MHz	256 K	90 nm	-	-	+	+	LGA775
Celeron D 345J	3.06 GHz	533 MHz	256 K	90 nm	-	-	-	+	LGA775
Celeron D 345	3.06 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478
Celeron D 341	2.93 GHz	533 MHz	256 K	90 nm	-	-	+	+	LGA775
Celeron D 340J	2.93 GHz	533 MHz	256 K	90 nm	-	-	-	+	LGA775
Celeron D 340	2.93 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478
Celeron D 336	2.80 GHz	533 MHz	256 K	90 nm	-	-	+	+	LGA775
Celeron D 335J	2.80 GHz	533 MHz	256 K	90 nm	-	-	-	+	LGA775
Celeron D 335	2.80 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478

Celeron D 331	2.66 GHz	533 MHz	256 K	90 nm	-	-	+	+	LGA775
Celeron D 330J	2.66 GHz	533 MHz	256 K	90 nm	-	-	-	+	LGA775
Celeron D 330	2.66 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478
Celeron D 326	2.53 GHz	533 MHz	256 K	90 nm	-	-	+	+	LGA775
Celeron D 325J	2.53 GHz	533 MHz	256 K	90 nm	-	-	-	+	LGA775
Celeron D 325	2.53 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478
Celeron D 320	2.40 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478
Celeron D 315	2.26 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478
Celeron D 310	2.13 GHz	533 MHz	256 K	90 nm	-	-	-	-	Socket 478

Intel CPUs for mobile and compact PCs

Intel processors for mobile PCs - Pentium M or Celeron M - are inseparably linked with the integrated Centrino platform for mobile PCs promoted by the company for several years, which includes Intel's best technologies for notebook PCs.

Dual-core Intel Core 2 Duo CPUs for mobile and compact PCs

From a certain time, the new Intel Core architecture has been topical for all the market sectors - mobile, desktop, and server systems. Having inherited the philosophy of efficient power consumption first implemented in Intel Pentium M for mobile PCs with the working name Banias, owing to new technologies and implementation of some developments of the new NetBurst architecture, the new generation

architecture has improved the ratings. Here are the key innovations of Intel's new CPU architecture:

- The **Intel Wide Dynamic Execution** technology is to provide a greater number of instructions executed per cycle, thus improving the efficiency of running applications and reducing the power consumption. Each core of the processor that supports this technology is now able executing up to four instructions simultaneously using the 14-stage pipeline.
- The **Intel Intelligent Power Capability** that enables specific components of the chip only when needed allows to achieve a substantial reduction in the power consumption of the system on the whole.
- The **Intel Advanced Smart Cache** technology implies using a unified L2 cache memory common for all the cores, whose joint use allows to cut down the power consumption and raise the performance. At the same time, one of the processor cores may use up the whole volume of the cache memory whenever needed, with the other core disabled dynamically.
- The **Intel Smart Memory Access** technology increases the system performance due to the reduced memory response time and thus optimized bandwidth of the memory subsystem.
- The **Intel Advanced Digital Media Boost** technology allows processing all the 128-bit SSE, SSE2, and SSE3 commands widely used in multimedia and graphic applications in one cycle, which increased their speed of execution.

Intel Core 2 Duo with the 2-core Merom design aimed at mobile and economy PCs are represented by the T series whose TDP is rated within 25 to 49 W.

Mobile 2-core Intel CPUs- Merom core

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
2-core mobile Intel Core 2 Duo									

Core 2 Duo T7600	2.33 GHz	667 MHz	4 MB (distr.)	65 nm	+	-	+	+	-
Core 2 Duo T7400	2.16 GHz	667 MHz	4 MB (distr.)	65 nm	+	-	+	+	-
Core 2 Duo T7200	2 GHz	667 MHz	4 MB (distr.)	65 nm	+	-	+	+	-
Core 2 Duo T5600	1.83 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	+	+	-
Core 2 Duo T5500	1.66 GHz	667 MHz	2 MB (distr.)	65 nm	-	-	+	+	-
Core 2 Duo T5300	1.73 GHz	533 MHz	2 MB (distr.)	65 nm	-	-	+	+	-
Core 2 Duo T5200	1.60 GHz	533 MHz	2 MB (distr.)	65 nm	-	-	+	+	-

Dual-core Intel Core 2 Duo Low Voltage CPUs for mobile and compact PCs

Dual-core Intel Core 2 Duo for mobile PCs are also represented by the economy Low Voltage series with the index L, which means the TDP is within 15 to 24 W.

Mobile 2-core Intel CPUs- Merom core									
2-core mobile Intel Core 2 Duo Low Voltage (LV)									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Core 2 Duo L7400	1.50 GHz	667 MHz	4 MB (distr.)	65 nm	+	-	+	+	-
Core 2 Duo L7200	1.33 GHz	667 MHz	4 MB (distr.)	65 nm	+	-	+	+	-

Dual-core Intel Core Duo CPUs for mobile and compact PCs

Dual-core Intel Core Duo for mobile PCs which are a component of the mobile platform Intel Centrino Duo are made on the base of the Yonah architecture following the norms of the 65-nm process technology.

The processors are remarkable for parallel execution of task branches on two cores with distributed CPU resources, offer 32 K L1 cache for instructions and L2 Intel Smart Cache of 2 MB capacity with support for the Advanced Transfer Cache architecture that provides efficient usage of cache memory and the CPU bus to improve the performance of the 2-core system and reduce the power consumption; the Intel Digital Media Boost technology that optimizes fetching of Streaming SIMD Extensions 2 (SSE2) and Streaming SIMD Extensions 3 (SSE3).

Intel Core Duo processors support the power-consumption optimized FSB = 667 MHz, as well as Intel Dynamic Power Coordination with the Dynamic Bus Parking feature to coordinate the performances of the cores @on demand". The Intel Dynamic Power Coordination allows each core to switch to the Halt, Stop Clock, and Deep Sleep states dynamically, and in the 2-core mode – synchronously to the Deeper and Enhanced Deeper Sleep modes. The distributed logic of the chip's power consumption control coordinates operation of the Enhanced Intel SpeedStep mode as well as switching between the C-states, which results in low supply voltage operation for Core Duo chips and minimum heat dissipation in the active state

Mobile 2-core Intel CPUs- Yonah core

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
2-core mobile Intel Core Duo									
Core Duo T2700	2.33 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-
Core Duo T2600	2.16 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-

Core Duo T2500	2 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-
Core Duo T2400	1.83 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-
Core Duo T2300	1.66 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-
Core Duo T2300E	1.66 GHz	667 MHz	2 MB (distr.)	65 nm	-	-	-	+	-
Core Duo T2250	1.73 GHz	533 MHz	2 MB (distr.)	65 nm	-	-	-	+	-
Core Duo T2050	1.60 GHz	533 MHz	2 MB (distr.)	65 nm	-	-	-	+	-
Core Duo L2500	1.83 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-

Dual-core Intel Core Duo Low Voltage CPUs for mobile PCs

Dual-core Intel Core Duo for mobile PCs are represented by the economy Low Voltage series with the index L, which means the TDP is within 15 to 24 W.

Mobile 2-core Intel CPUs- Yonah core									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
2-core mobile Intel Core Duo Low Voltage (LV)									
Core Duo L2400	1.66 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-
Core Duo L2300	1.50 GHz	667 MHz	2 MB (distr.)	65 nm	+	-	-	+	-

Dual-core Intel Core Duo Ultra Low Voltage CPUs for mobile PCs

Another ultra-saving series of dual-core Intel Core Duo for mobile PCs - the Ultra Low Voltage with the U index, which means the TDP is below 14 W.

Mobile 2-core Intel CPUs- Yonah core									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
2-core mobile Intel Core Duo Ultra Low Voltage (ULV)									
Core Duo U2500	1.20 GHz	533 MHz	2 MB (distr.)	65 nm	+	-	-	+	-
Core Duo U2400	1.06 GHz	533 MHz	2 MB (distr.)	65 nm	+	-	-	+	-

Intel Core Solo for mobile PCs

Single-core version of Yonah DC - Intel Core Solo chips on the Yonah-SC core (or Yonah-1P).

Mobile Intel CPUs - Yonah-1P core									
Mobile Intel Core Solo									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Core Solo T1400	1.83 GHz	667 MHz	2 MB	65 nm	-	-	-	+	-
Core Solo T1350	1.86 GHz	533 MHz	2 MB	65 nm	-	-	-	+	-

Core Solo T1300	1.66 GHz	667 MHz	2 MB	65 nm	-	-	-	+	-
------------------------	----------	---------	------	-------	---	---	---	---	---

Intel Core Solo Ultra Low Voltage CPUs for mobile PCs

Super-economy version of single-core Intel Core Solo - Ultra Low Voltage series with the U index and TDP below 14 W.

Mobile Intel CPUs - Yonah-1P core									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Mobile Intel Core Solo Ultra Low Voltage (ULV)									
Core Solo U1400	1.20 GHz	533 MHz	2 MB	65 nm	+	-	-	+	-
Core Solo U1300	1.06 GHz	533 MHz	2 MB	65 nm	+	-	-	+	-

Intel Pentium M for mobile PCs

Previous generation of Centrino platform, Dothan CPUs. Rated TDP of the chips - about 27 W.

Mobile Intel CPUs - Dothan core									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Mobile Intel Pentium M									
Pentium M 780	2.26 GHz	533 MHz	2 MB	90 nm	-	-	-	+	Socket 479

Pentium M 770	2.13 GHz	533 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 765	2.10 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 760	2 GHz	533 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 755	2 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 750	1.86 GHz	533 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 745A	1.80 GHz	533 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 745	1.80 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 740	1.73 GHz	533 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 735	1.70 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 730	1.60 GHz	533 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 725	1.60 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 715	1.50 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 705	1.50 GHz	400 MHz	1 Mb	0.13 mk	-	-	-	-	Socket 479

Intel Pentium M Low Voltage for mobile PCs

Economy Low Voltage series of Intel Pentium M based on the Dothan core, TDP - about 10 W, ULV versions of Dothan - about 5.5 W.

Mobile Intel CPUs - Dothan core

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Mobile Intel Pentium M Low Voltage (LV)									
Pentium M 778	1.60 GHz	400 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 758	1.50 GHz	400 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 738	1.40 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 718	1.30 GHz	400 MHz	1 Mb	0.13 mk	-	-	-	-	Socket 479

Intel Pentium M Ultra Low Voltage for mobile PCs

Super-economy Ultra Low Voltage (ULV) series of Intel Pentium M based on the Dothan core, TDP - about 5.5 W.

CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Mobile Intel CPUs - Dothan core									
Pentium M 773	1.30 GHz	400 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 753	1.20 GHz	400 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 733J	1.10 GHz	400 MHz	2 MB	90 nm	-	-	-	+	Socket 479
Pentium M 733	1.10 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479

Pentium M 723	1 GHz	400 MHz	2 MB	90 nm	-	-	-	-	Socket 479
Pentium M 713	1.10 GHz	400 MHz	1 Mb	0.13 mk	-	-	-	-	Socket 479

Intel Celeron M for mobile PCs

Perhaps, a motley "company" of Intel's mobile processors which includes a number of generations (from 0.13 mk to 65 nm) of value versions of Pentium M chips, normally with reduced L2 cache size, lower FSB and clock speeds.

Mobile Intel Celeron M									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Celeron M 450	2 GHz	533 MHz	1 Mb	65 nm	-	-	-	+	-
Celeron M 440	1.86 GHz	533 MHz	1 Mb	65 nm	-	-	-	+	-
Celeron M 430	1.73 GHz	533 MHz	1 Mb	65 nm	-	-	-	+	-
Celeron M 420	1.60 GHz	533 MHz	1 Mb	65 nm	-	-	-	+	-
Celeron M 410	1.46 GHz	533 MHz	1 Mb	65 nm	-	-	-	+	-
Celeron M 390	1.70 GHz	400 MHz	1 Mb	90 nm	-	-	-	+	Socket 479
Celeron M 380	1.60 GHz	400 MHz	1 Mb	90 nm	-	-	-	+	Socket 479
Celeron M 370	1.50 GHz	400 MHz	1 Mb	90 nm	-	-	-	+	Socket 479

Celeron M 360J	1.40 GHz	400 MHz	1 Mb	90 nm	-	-	-	+	Socket 479
Celeron M 360	1.40 GHz	400 MHz	1 Mb	90 nm	-	-	-	-	Socket 479
Celeron M 350J	1.30 GHz	400 MHz	1 Mb	90 nm	-	-	-	+	Socket 479
Celeron M 350	1.30 GHz	400 MHz	1 Mb	90 nm	-	-	-	-	Socket 479
Celeron M 340	1.50 GHz	400 MHz	512 K	0.13 mk	-	-	-	-	Socket 479
Celeron M 330	1.40 GHz	400 MHz	512 K	0.13 mk	-	-	-	-	Socket 479
Celeron M 320	1.30 GHz	400 MHz	512 K	0.13 mk	-	-	-	-	Socket 479
Celeron M 310	1.20 GHz	400 MHz	512 K	0.13 mk	-	-	-	-	Socket 479

Intel Celeron M Ultra Low Voltage for mobile PCs

Economy versions of value mobile Intel Celeron M.

Mobile Intel Celeron M ULV									
CPU	Clock speed	FSB	L2	Process technology	VT	HT	64-bit	XD	Socket
Mobile Intel Celeron M Ultra Low Voltage (ULV)									
Celeron M 423	1.06 GHz	533 MHz	1 Mb	65 nm	-	-	-	+	-
Celeron M 383	1 GHz	400 MHz	1 Mb	90 nm	-	-	-	+	Socket 479
Celeron M 373	1 GHz	400 MHz	512 K	90 nm	-	-	-	+	Socket 479

Celeron M 353	900 MHz	400 MHz	512 K	90 nm	-	-	-	-	Socket 479
Celeron M 333	900 MHz	400 MHz	512 K	0.13 mk	-	-	-	-	Socket 479

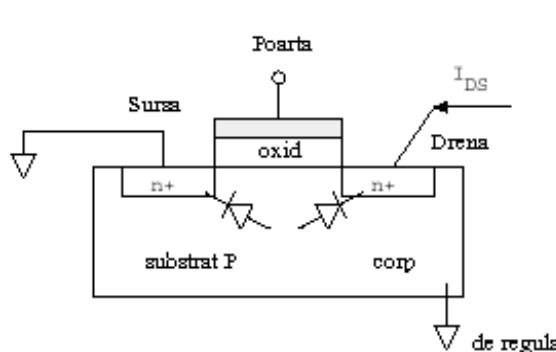
Capitolul 2. Functionarea tranzistorului.

Circuitele integrate MOS au fost realizate la inceput in tehnologia PMOS, datorita predictibilitatii tensiunii de prag pentru tranzistoarele PMOS. Pe masura perfectionarii tehnologiei prin recurgerea la implantare ionica, tranzistorul NMOS a devenit tot mai raspandit. Evolutia tehnologiei a condus, in continuare, la aparitia dispozitivelor CMOS. Acest capitol este consacrat tranzistoarelor NMOS si PMOS din punctul de vedere al structurii si operarii.

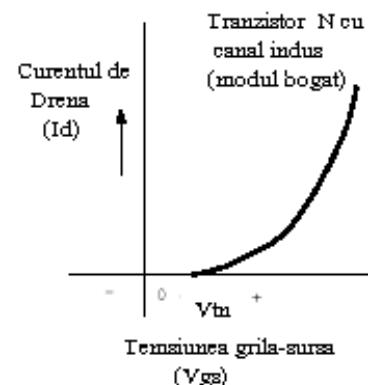
2.1. Tranzistorul NMOS in modul bogat (enhancement mode).

Structura de baza a tranzistorului NMOS in modul bogat este arataza mai jos. Ea consta inre-un substrat de tip p , sursa si drena de tip $n+$. Intre sursa si drena se afla canalul acoperit cu SiO_2 . Acest strat de oxid mai poarta numele de oxid de poarta. Peste stratul de oxid al portii se afla un strat de siliciu policristalin, care reprezinta poarta. Atunci cand $V_{GS} = 0$, sursa si drena tranzistorului sunt separate de o regiune de tip p . Currentul de la drena la sursa, I_{DS} , este foarte mic si limitat de currentul invers de polarizare din diode.

Pe masura ce V_{GS} creste electronii din zonele sursei si drenei vor fi atrasi de regiunea aflata sub poarta. Aceasta va conduce la cresterea conductibilitatii intre drena si sursa. In momentul in care V_{GS} depaseste tensiunea de prag V_{th} , se formeaza/se induce un canal (regiune de tip n) intre sursa si drena permitand curgerea unui current de la drena la sursa. O sectiune prin structura tranzistorului NMOS, cat si relatia intre V_{GS} si I_{DS} sunt prezentate in figurile (a) si (b), de mai jos.



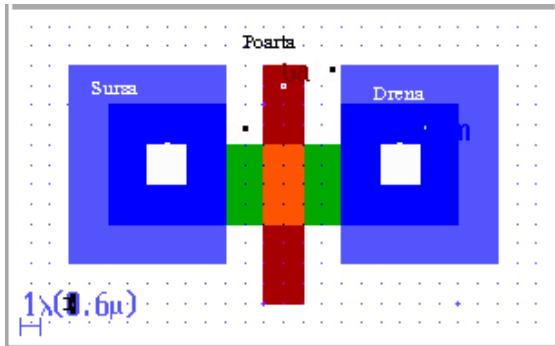
(a)



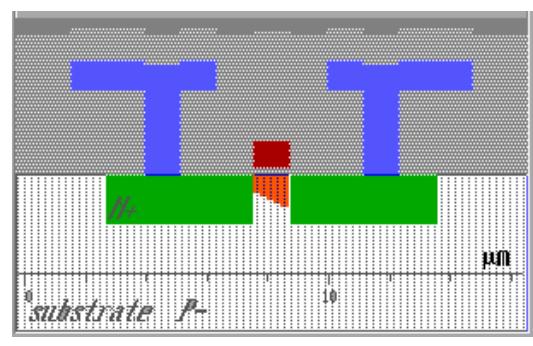
(b)

Intrucat pe poarta tranzistorului NMOS nu se constata prezenta unui curent sau a unei tensiuni, acesta in mod normal este blocat. Se spune ca tranzistorul este in modul bogat deoarece se impune o tensiune pe poarta pentru a imbogati canalul in vederea realizarii acestuia.

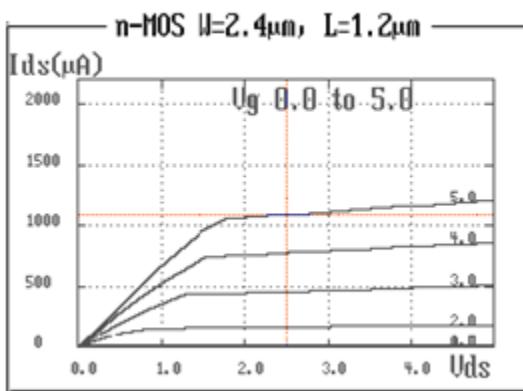
In figura de mai jos se prezinta: (a) mastile pentru un tranzistor NMOS, (b) o sectiune transversala prin structura tranzistorului si familiile de caracteristici (c) $I_{DS} = f(V_{DS}, V_{GS})$, (d) $I_{DS} = f(V_{GS}, V_B)$, pentru $V_{DS} = \text{const}$. Tensiune V_B este tensiunea de polarizare a substratului. Modificarea acestei tensiuni de polarizare conduce la modificarea tensiunii de prag.



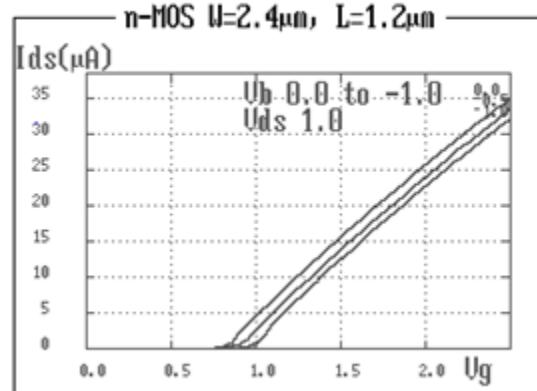
(a)



(b)



(c)



(d)

2.2. Tranzistorul NMOS in modul sarac (depletion mode).

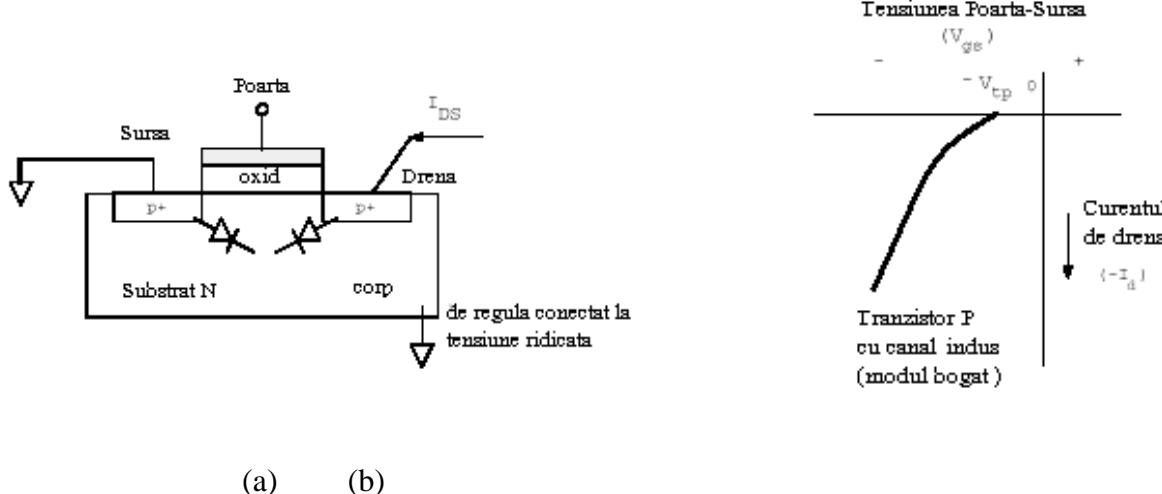
La tranzistorul NMOS in modul sarac canalul este prezent in absenta unor surse externe. Canalul este natural, tranzistorul fiind in conductie/deschis, ceea ce se manifesta prin $I_{DS} > 0$ pentru conditia: $V_{GS} = 0$.

Pentru blocarea canalului trebuie sa se aplice pe poarta o tensiune negativa in raport cu sursa.

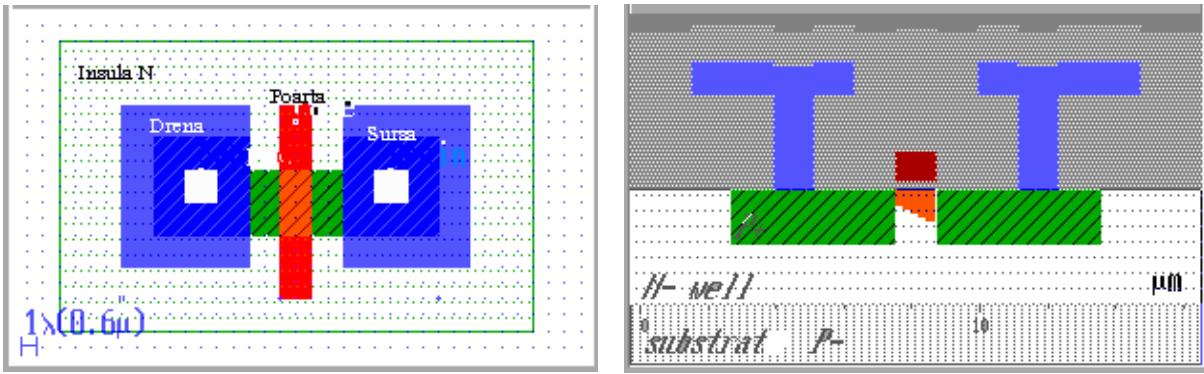
2.3. Tranzistorul PMOS in modul bogat.

Ca si tranzistorul NMOS in modul bogat, tranzistorul PMOS in modul bogat nu poseda canal natural, canalul fiind indus atunci cand pe poarta se aplică o tensiune. Astfel, in mod normal este blocat. Caracteristica curent-tensiune a tranzistorului PMOS in mod bogat este inversa in raport cu cea a tranzistorului NMOS in modul bogat. Pentru a aduce tranzistorul in stare de conductie trebuie sa se aplice o tensiune negativa intre poarta si sursa. Mai jos se prezinta o sectiune prin:

(a) structura tranzistorului si (b) caracteristicile $I_{DS} = f(V_{DS})$.

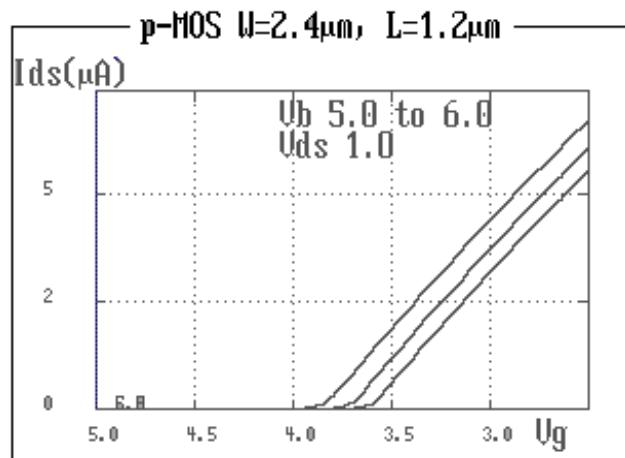


In figura de mai jos se prezinta: (c) mastile pentru un tranzistor PMOS, (d) o sectiune transversala prin structura tranzistorului si (e) familia de caracteristici $I_{DS} = f(V_{GS}, V_B)$, pentru $V_{DS} = \text{const}$. Tensiune V_B este tensiunea de polarizare a substratului. Modificarea acestei tensiuni de polarizare conduce la modificarea tensiunii de prag



(c)

(d)



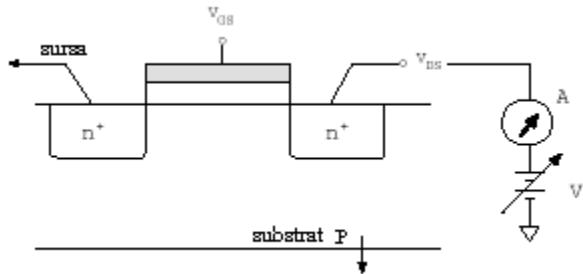
(e)

2.4. Operarea tranzistorului.

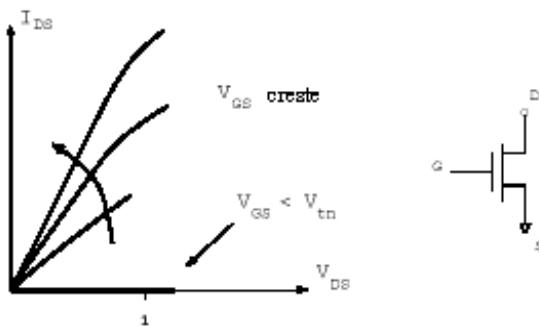
Operarea tranzistorului poate fi examinata in cadrul a doua regiuni: regiunea liniara si regiunea de saturatie (pinch-off). In cele ce urmeaza va fi examinata functionarea tranzistorului in cele doua regiuni.

2.4.1. Regiunea liniara.

Considerand dispozitivul NMOS, cu canal indus, tensiunile de drena si poarta au ca referinta tensiunea sursei. In cazul in care V_{GS} este mai mare decat tensiunea de prag se va forma canalul n .



Pentru valori mici (sub 1V) ale tensiunii V_{DS} , caracteristica $I_{DS} - V_{DS}$ este liniara, ca in figura de mai jos, si ramane liniara pentru valori ale lui V_{DS} mai mici decat V_{GS} . Rezistenta intre sursa si drena este controlata de catre tensiunea pe poarta. Aceasta este regiunea liniara. O tensiune pozitiva mica aplicata intre poarta si sursa va induce o sarcina negativa in zona de sub poarta. Aceasta sarcina indusa este formata din acceptori ionizati, care asigura o regiune saraca, ca urmare a repulsiei golurilor. Cresterea in continuarea a tensiunii face ca purtatorii minoritari (electronii) sa fie atrasi catre canal de la sursa si de la drena. La o valoare data a tensiunii pe poarta, concentratia, la suprafata, a purtatorilor minoritari (electronii) sa depaseasca densitatea golurilor in material, ceea ce corespunde unei inversii de suprafata. Pentru valori mici ale lui V_{DS} , stratul de inversie se intinde de la sursa la drena.

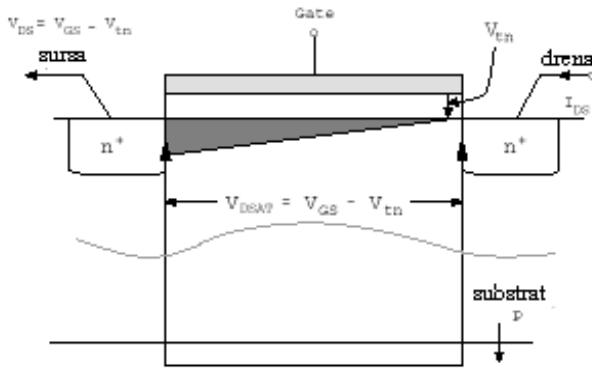


2.4. Regiunea de saturatie.

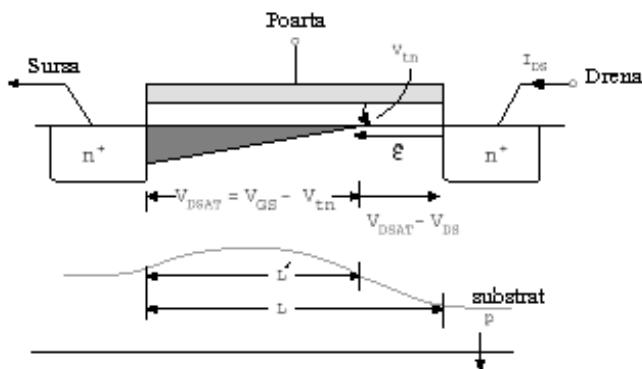
Se considera situatia in care tensiunea de grila depaseste tensiunea de prag, dispozitivul fiind in conductie, cu canalul format. Pentru valori mici ale lui V_{DS} , atunci cand $V_{DS} < V_{GS} - V_m$, stratul de inversie se intinde de la sursa la drena. Dispozitivul lucreaza ca un rezistor variabil controlat de catre o tensiune.

Pe masura ce V_{DS} creste, tensiunea intre poarta si drena scade, dar I_{DS} inca are o crestere.

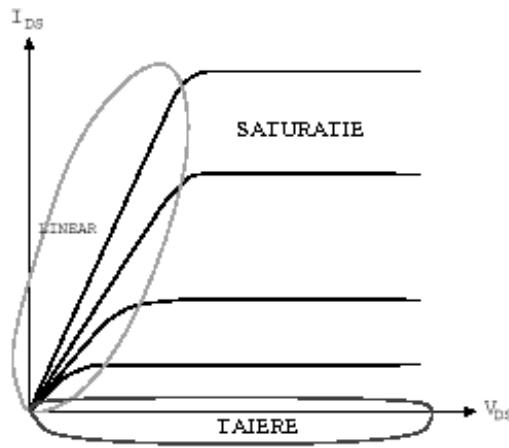
Cand $V_{DS} = V_{GS} - V_{tn}$, campul electric E present la capatul dinspre drena si dielectricul portii se reduce la o asemenea valoare incat stratul de inversie nu mai este asigurat. In aceasta situatie se spune ca a avut loc taierea (pinche-off) canalului. Aceasta situatie este prezentata in figura de mai jos. Currentul de la drena la sursa este saturat si notat cu I_{DSAT} .



O crestere in continuare a tensiunii V_{DS} fata de V_{DSAT} conduce la crearea unei zone fara inversie, ca in figura de mai jos. Datorita proprietatii de continuitate I_{DS} este egal cu I_{DSAT} . Zona formata, disponand de un camp puternic, accelereaza electronii, care ajung la punctul de taiere, pentru a trece spre drena. Currentul ramane constant si egal cu cel de saturatie. Se poate insa constata ca lungimea canalului s-a micsorat de la L la L' . Acest fenomen poarta numele de modulare a canalului si are ca rezultat o usoara crestere a curentului in regiunea de saturatie.



Caracteristica I-V a tranzistorului NMOS, cu canal indus, este data mai jos.



2.5. Ecuatia curentului I_{DS} .

In cele ce urmeaza se vor deduce o serie de ecuatii pentru curentul I_{DS} , pe baza modelului simplificat al controlului prin sarcina. Pentru regiunea liniara, unde $V_{GS} > V_m$ si $V_D < V_G - V_m$, se poate presupune o distributie uniforma a sarcinii, ceea ce face ca urmatoarea ecuatie sa fie valida:

$$Q_c \sim -C_{ox}(V_{GS} - V_m)$$

unde Q_c este densitatea de sarcina in canal (coulombi/unitatea de arie) si C_{ox} este capacitatea portii pe unitatea de arie. Se poate observa ca semnul este negativ datorita faptului ca sarcina este constituita din electroni.

Timpul de tranzit al sarcinii prin canal este egal cu lungimea canalului L impartita la viteza:

$$\tau \sim L/viteza$$

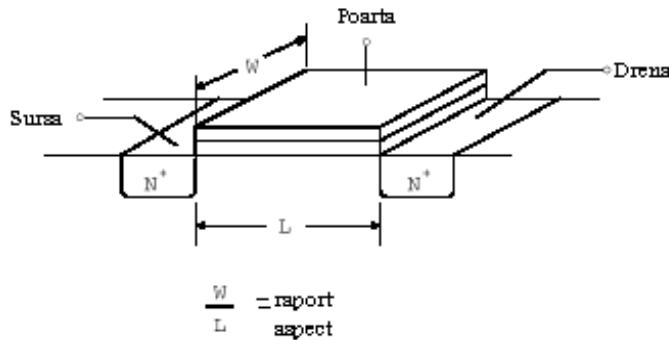
Viteza electronilor este proportionala cu mobilitatea μ_n inmultita cu intensitatea campului electric E , $viteza = \mu_n \cdot E$, ceea ce face ca:

$$\tau \sim L/(\mu_n \cdot E)$$

Pe de alta parte intensitatea campului electric E este egala cu V_{DS}/L ($E = V_{DS}/L$), ceea ce face ca:

$$\tau \sim L^2/(\mu_n \cdot V_{DS})$$

Elementele geometrice ce caracterizeaza canalul L – lungime, W - latime sunt prezentate in figura urmatoare:



Raportul W/L poarta numele de *raport- aspect*.

Pe baza celor de mai sus se poate scrie:

$$I_{SD} \sim Q/\tau, \text{ sarcina totala in canal/timpul de tranzit sau}$$

$$I_{SD} \sim -Cox(V_{GS} - V_{tn}).W.L/(L^2/(\mu_n.V_{DS}))$$

Dar, $I_{DS} = -I_{SD}$, astfel, ecuatia curentului I_{DS} devine:

$$I_{DS} \sim \mu_n.Cox.(W/L).(V_{GS} - V_{tn}).V_{DS}$$

Parametrul W/L poate fi controlat de catre proiectant.

Observatii:

- $\mu_n.Cox.(W/L)$ reprezinta *factorul de amplificare al tranzistorului* (β);
- $\mu_n.Cox$ este tipic $(20-30) \times 10^{-6} A/V^2$
- μ_n este tipic $(6-9) \times 10^{-2} m^2/V.s$
- Cox este tipic $(3-4) \times 10^{-4} F/m^2$

In continuare se face o comparatie intre ecuatia simplificata si o ecuatie mai exacta pentru I_{DS} , in regiunea liniara:

- ecuatia simplificata: $I_{DS} \sim \mu_n.Cox.(W/L).(V_{GS} - V_{tn}).V_{DS}$
- ecuatia mai exacta: $I_{DS} \sim \mu_n.Cox.(W/L).[(V_{GS} - V_{tn}).V_{DS} - V_{DS}^2/2]$

Cand V_{DS} este mic, modelul bazat pe controlul prin sarcina asigura o precizie rezonabila. In cazul reducerii dimensiunilor dispozitivului, aceste ecuatii de proiectare devin mai putin precise.

Chiar fara termenul la patrat I_{DSAT} poate fi calculat pe baza modelului controlat prin sarcina prin inlocuirea $V_{DS} = V_{DSAT} = V_{GS} - V_{tn}$, ceea ce va conduce la:

$$I_{DSAT} \sim \mu_n \cdot Cox \cdot (W/L) \cdot (V_{GS} - V_{tn})^2$$

Aceastei estimari ii lipseste termenul $V_{DS}^2/2$

Intr-o maniera similara se poate examina tranzistorul PMOS.

2.6. Tensiunea de prag (V_{tn} sau V_m).

Tensiunea de prag poate fi definita ca tensiunea intre poarta si sursa unui dispozitiv MOS sub valoarea careia curentul I_{DS} devine zero.

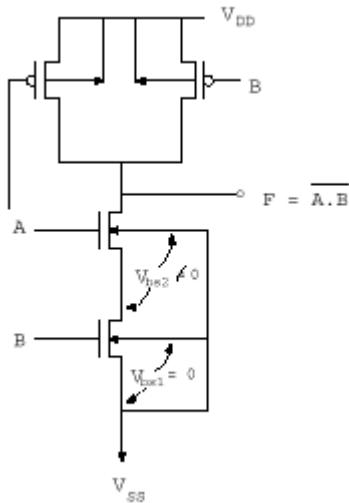
Considerand o valoare mica pentru V_{GS} se pot face urmatoarele constatari:

- sarcina initiala in canal se datoreaza numai acceptorilor ionizati din regiunea saraca
- concentratia de purtatori minoritari de la suprafata devine egala cu concentratia purtatorilor majoritari din substrat;
- cresterea in continuarea a lui V_{GS} induce un puternic strat de inversie;
- tensiunea de prag V_{tn}/ V_{tp} poate fi controlata prin modificarea grosimii stratului de dioxid de siliciu t_{ox} , care nu constituie un parametru de proiectare, fiind controlat de specialistul in procesul de fabricatie.

2.7. Efectul de corp

Sursa unui dispozitiv de tip N este de regula conectata la o tensiune coborata, ca si structura de baza/corpul, astfel incat tensiunea intre ele $V_{BS} = 0$. In aceste conditii tensiunea de polarizare a substratului/corpului va afecta tensiunea de prag in mod direct.

Dispozitivele MOS sunt construite pe un substrat comun, ceea ce face ca tensiunea de polarizare a substratului sa fie aceeasi. Exista unele combinatii, dupa cum se vede in figura de mai jos, in care tensiunea intre sursa si substrat sa nu fie egala cu zero ($V_{BS1} = 0, V_{BS2} \neq 0$).



Cand creste tensiunea intre sursa si substrat creste, de asemenea, si latimea stratului sarac canal-substrat.

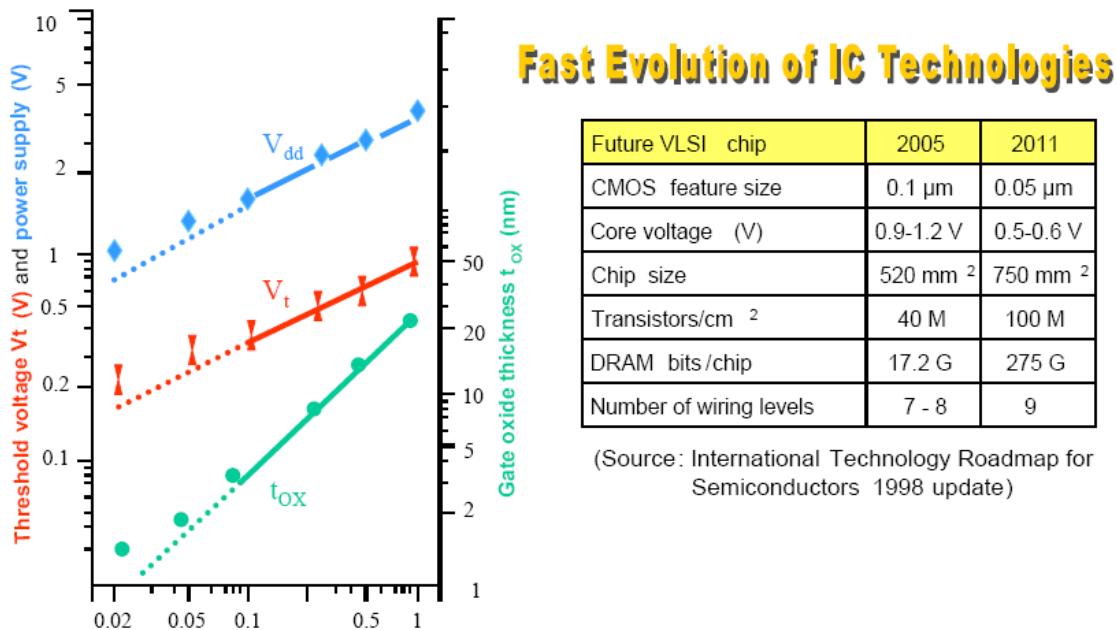
Ca rezultat, creste numarul acceptorilor ionizati din stratul sarac. Acesta trebuie compensat cu o sarcina egala si de semn opus pe poarta inainte de formarea stratului de inversie, ceea ce conduce la cresterea tensiunii de prag. O aproximatie de ordinul 0, pentru tensiunea de prag, este urmatoarea:

$$V_{tn} = V_{tn(0)} + \gamma \cdot V^{1/2}_{BS}$$

unde $V_{tn(0)}$ este tensiune de prag atunci cand $V_{BS1}=0$, iar γ poate lua valori de la $0.1 \text{ V}^{1/2}$ la $1 \text{ V}^{1/2}$.

In relatia de mai sus semnul este pozitiv pentru dispozitivele NMOS.

Status of Microelectronics Technology



Productivity Gap: Technology vs. CAD

Need to increase Designers Productivity in order to make use of new Technologies

SIA Roadmap for the Design Technology Requirements (near term)

YEAR TECHNOLOGY NODE	1999 180 nm	2000	2001	2002 130 nm	2003	2004	2005 100 nm
MPU new design cycle (months)	36	36	36	32	32	32	30
MPU transistors per designer-month (300-person team) (thousand)	2	3	4	7	10	15	20
ASIC new design cycle (months)	12	12	12	12	12	12	12
ASIC transistors per designer-month (50-person team) (million)	0.3	0.4	0.5	0.7	1.0	1.3	1.8
Portion of verification by formal methods	15%	15%	15%	20%	20%	20%	30%
Portion of test covered by BIST	20%	20%	20%	30%	30%	30%	40%

BIST= Built In Self Test

Capitolul 3. Inversorul.

In primul capitol au fost prezentate mai multe exemple de porti logice, realizate cu ajutorul tranzistoarelor NMOS si PMOS. Tranzistorul NMOS, in calitate de comutator, conduce bine semnalele de nivel coborat si mai putin bine pe cele de nivel ridicat, in timp ce tranzistorul PMOS are o comportare complementara.

Capitolul de fata urmareste prezentarea functionarii portilor CMOS, avand ca exemplu inversorul, pentru analiza. De asemenea, inversorul va fi utilizat pentru prezentarea proiectarii mastilor necesare in procesul de realizare a diverselor dispozitive pe un substrat de siliciu monocristalin.

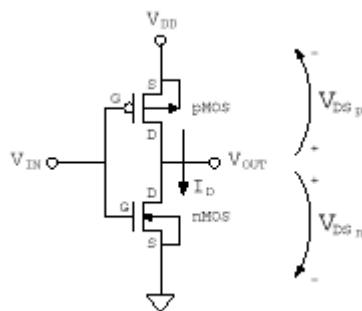
Vor fi examineate regulile de proiectare si scopul acestora. Se vor face exemplificari pentru un proces CMOS caracterizat printr-o rezolutie de $0,8 \mu\text{m}$, ceea ce corespunde unui exemplu didactic.

De asemenea, se vor trece in revista aspectele electronice de baza privitoare la poarta logica elementara.

3.1. Analiza functionarii.

3.3.1. Caracteristicile de curent continuu.

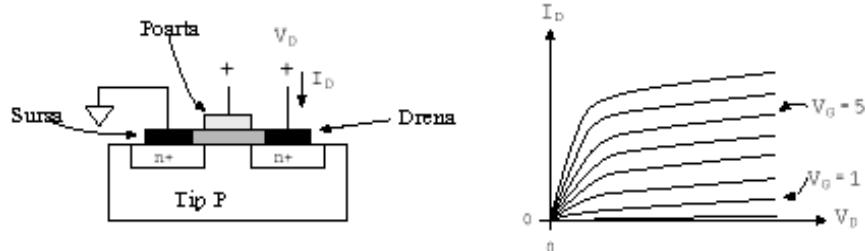
In figura de mai jos se prezinta schema unui inversor CMOS. Acesta este compus din doua tranzistoare cu canal indus PMOS si NMOS, conectate in serie intre V_{DD} si GND. Tranzistorul PMOS are conectatae: sursa la V_{DD} , drena la V_{OUT} si substratul la V_{DD} , in timp ce tranzistorul



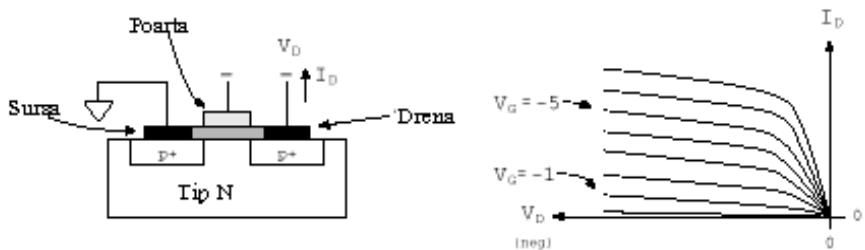
NMOS are conectate sursa si substratul la GND, si drena la V_{OUT} . Portile celor doua tranzistoare sunt conectate la V_{IN} .

Pentru analiza functionarii in curent continuu se vor examina caracteristicile curent-tensiune pentru cele doua tipuri de tranzistoare. Se considera ca la un tranzistor de tip N curentul intra prin drena, iar la un tranzistor P curentuliese prin drena.

NMOS



PMOS



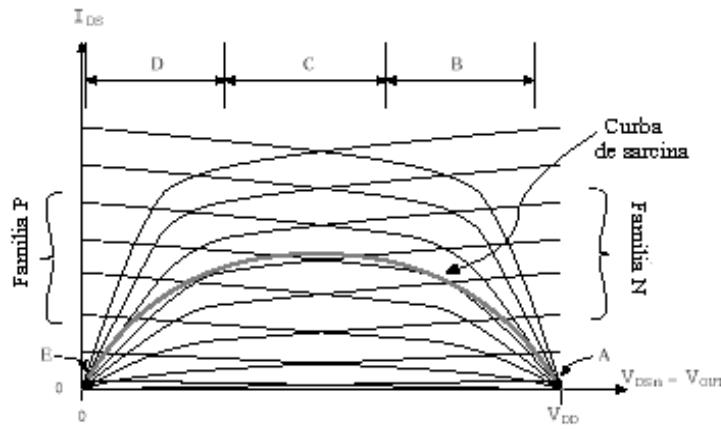
In continuare se vor suprapune caracteristicile I-V ale celor doua tipuri de dispozitive pentru a efectua o analiza a sarcinii. Plecand de la schema invesorului se constata ca acelasi curent ($I_{DSN} = I_{SDP}$) traverseaza, in acelasi sens, cele doua tranzistoare. Din acest motiv caracteristicile I-V ale celor doua tranzistoare pot folosi aceeasi axa pentru curenti. V_{DD} se poate scrie astfel:

$$V_{DD} = V_{DSN} + (-V_{DSP})$$

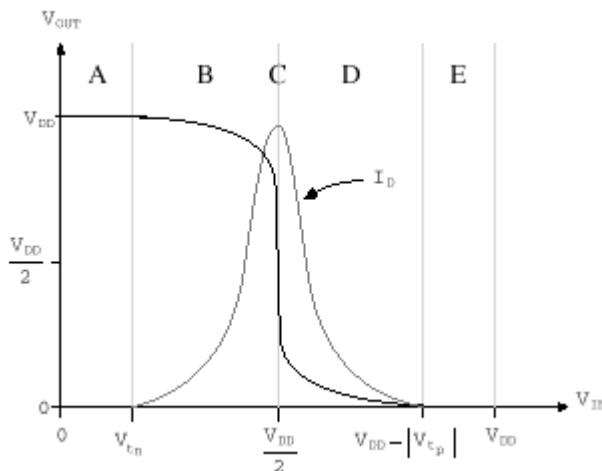
Rescriind ecuatia:

$$V_{DSP} = V_{DSN} - V_{DD}$$

Se observa ca tensiunea drenei tranzistorului PMOS difera de cea a tranzistorului NMOS cu o valoare egala cu V_{DD} , ceea ce permite utilizarea aceleiasi axe pentru tensiuni, insa deplasata cu V_{DD} . Cele doua familii de caracteristici I-V sunt prezentate mai jos:



In continuare se urmărește gasirea caracteristicii de transfer. Pe curba de sarcină vor fi evidențiate două cazuri. Primul caz apare cand $V_{IN} = V_{DD}$, care este marcat pe figura de mai jos cu E. În acest punct tranzistorul P este taiat, iar tranzistorul N funcționează în regiunea liniară, ceea ce va plasa inversorul în regiunea E. Al doilea caz apare atunci cand $V_{IN} = GND$, situație marcată cu A. În acest punct tranzistorul P funcționează în regiunea liniară, tranzistorul N este taiat, iar inversorul se găsește în regiunea A. Analiza poate fi continuată pentru toate punctele caracteristicii de transfer. Curba de sarcină va reprezenta soluția pentru familiile de caracteristici I-V ale celor două tipuri de tranzistoare. În aceasta analiză, în curent continuu, s-a neglijat curentul, care ar ieși sau intra prin terminalul notat cu V_{OUT} .



Caracteristica de transfer de mai sus are cinci regiuni distincte marcate prin A,..., E, care vor fi analizate în continuare.

Regiunea A: Tranzistorul N este taiat, tranzistorul P este în regiunea liniară:

$$V_{IN} < V_{tn}$$

$$V_{OUT} = V_{DD}$$

Regiunea B: Tranzistorul P ramane in regiunea liniara, iar tranzistorul N trece in saturatie. Astfel, primul tranzistor se comporta ca o rezistenta, iar cel de-al doilea opereaza ca o sursa de curent:

$$V_{tn} \leq V_{IN} < V_{DD}/2$$

$$V_{OUT} = (V_{IN} - V_{tp}) + [(V_{IN} - V_{tp})^2 - 2(V_{IN} - V_{DD}/2 - V_{tp}) V_{DD} - \beta_n / \beta_p (V_{IN} - V_{tn})^2]^{1/2}$$

Regiunea C: Expresia pentru V_{OUT} se poate stabili prin egalarea curentilor de drena pentru cele doua tranzistoare. Tranzistorul N ramane in saturatie, iar tranzistorul P trece in saturatie. Panta mare a caracteristicii face ca iesirea sa varieze foarte mult la o mica modificare a intrarii.

$$V_{IN} = V_{DD}/2$$

$$V_{IN} - V_{tn} < V_{OUT} < V_{IN} - V_{tp}$$

Regiunea D: Tranzistorul PMOS ramane in saturatie, jucand rolul unei surse de curent, iar tranzistorul NMOS trece in regiunea liniara, comportandu-se ca o rezistenta.

$$V_{DD}/2 < V_{IN} \leq V_{DD} - V_{tp}$$

$$V_{OUT} = (V_{IN} - V_{tn}) + [(V_{IN} - V_{tn})^2 - \beta_p / \beta_n (V_{IN} - V_{DD} - V_{tp})^2]^{1/2}$$

Regiunea E: Tranzistorul NMOS ramane in regiunea liniara, iar tranzistorul PMOS este taiat.

$$V_{IN} > V_{DD} + V_{tp}$$

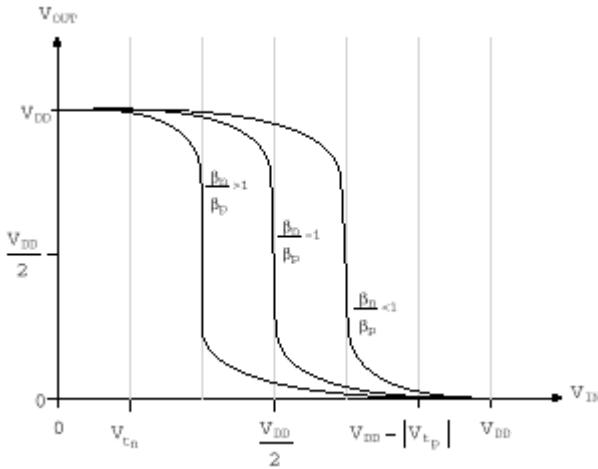
$$V_{OUT} = 0$$

3.1.2 Raportul β_n / β_p .

Atunci cand se proiecteaza un inversor este de dorit ca procesul de comutare sa aibe loc la o valoare egala cu jumatarea tensiunii de alimentare. Astfel, comutarea trebuie sa aibe loc la $V_{IN} = V_{DD}/2$. Punctul la care inversorul comuta este dependent de valorile amplificarilor tranzistoarelor (β). Valoarea lui β este calculata cu ajutorul expresiei:

$$\beta = (\mu \epsilon / tox) * W/L$$

unde: μ este mobilitatea, ϵ permitivitatea oxidului, tox este grosimea stratului de oxid, W este latimea canalului, iar L este lungimea canalului. Daca se considera raportul β_n / β_p , pentru diferite valori ale acestuia, caracteristica de transfer se va modifica conform figurii de mai jos.



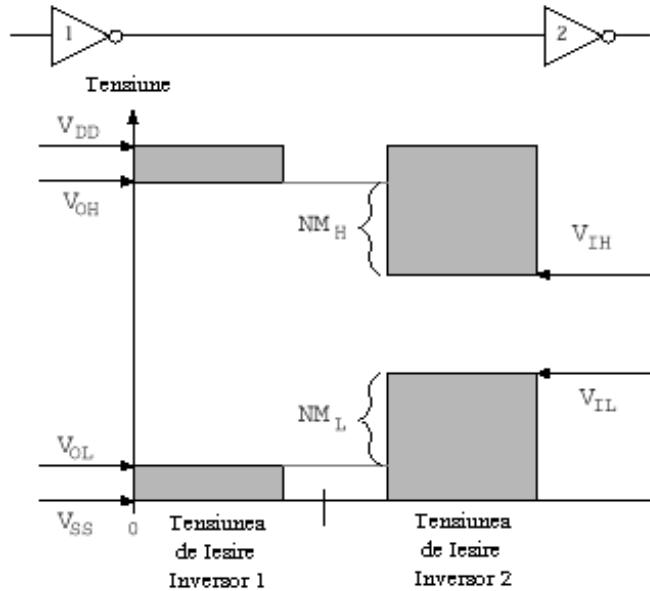
Caracteristica de transfer poate fi influentata de temperatura. Amplificarea β pentru un tranzistor este dependenta de temperatura conform relatiei:

$$\beta \propto T^{-3/2}$$

Aceasta se explica prin dependenta mobilitatii de temperatura. In realitate, variatia de temperatura nu afecteaza raportul β_n / β_p deoarece ambele amplificari sunt influentate cu aceeasi cantitate. O scadere a temperaturii va creste tensiunile de prag V_m si $|V_{tp}|$. In acest mod o crestere a temperaturii va cobora regiunea A si va ridica regiunea E, iar caracteristica de transfer va avea o panta mai mica in regiunea C.

3.1.3 Marginea de zgomot.

Marginea de zgomot a inversorului poate fi stabilita pe baza caracteristicii de transfer. Marginea de zgomot reprezinta abaterea maxima a unui semnal fata de valoarea normala, inainte ca el sa fie recunoscut ca un alt semnal. Pentru calculul marginii de zgomot se folosesc urmatoarele notatii, conform desenului de mai jos:



- V_{IL} – tensiunea cea mai mare, care poate fi considerata ca intrare de nivel coborat,
- V_{IH} – tensiunea cea mai coborata, care poate fi considerata intrare de nivel inalt,
- V_{OL} – tensiunea cea mai coborata, care poate fi generata ca iesire,
- V_{OH} – tensiunea cea mai ridicata, care poate fi generata ca iesire.

Valorile V_{IL} and V_{IH} apar acolo unde caracteristica de transfer are amplificarea egala cu unu. Acestea sunt punctele de pe caracteristica de transfer unde zgomotul are efect critic. Intrucat amplificarea este egala cu unu, in aceste doua puncte iesirea se modifica in aceeasi masura ca si intrarea. Daca tensiunea semnalului de zgomot este mai mica decat V_{IH} , pentru un semnal de nivel ridicat, iesirea va comuta de la nivel coborat la nivel ridicat. Daca tensiunea semnalului de zgomot este mai mare decat V_{IL} , pentru un semnal de nivel coborat, iesirea va comuta de la nivel ridicat la nivel coborat, datorita amplificarii supraunitare.

Marginile de zgomot pot fi definite astfel

- $NM_L = V_{IL} - V_{OL}$ – pentru valoare coborata.
- $NM_H = V_{oh} - V_{IH}$ – pentru valoare ridicata..

Valoarea lui NM_L si NM_H este de aproximativ 2V, pentru $\beta_n / \beta_p = 1$, $V_{tn} = |V_{tp}| = 1$ V si $-V_{DD} = 5$ V.

Este interesant de observat ca marginea de zgomot va descreste pe masura ce V_{DD} descreste. Acest fapt poate fi constatat in ecuatia ce descrie cele doua margini de zgomot. De asemenea, tensiunile de prag sunt deja foarte apropiate.

3.2 Proiectarea mastilor/sabloanelor.

3.2.1 Sabloanele de baza.

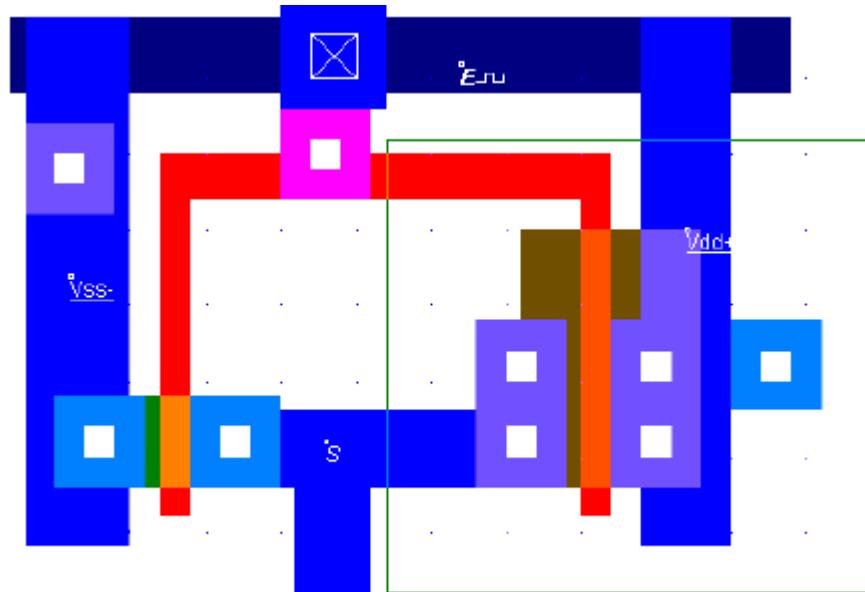
Schema unui inversor CMOS a fost prezentata intr-un paragraf anterior. In continuare se vor examina, din punct de vedere geometric, sabloanele necesare pentru realizarea unui inversor CMOS, cu ajutorul unui proces de fabricatie plecand de la un substrat de tip P. Fabricarea dispozitivelor VLSI implica crearea mai multor straturi din materiale cu diverse proprietati electrice, depuse unul peste celalalt. In figurile de mai jos se prezinta:

- codul colorilor folosit pentru identificarea diverselor straturi,
- geometria simplificata a unui inversor CMOS, la nivelul diverselor straturi,
- sectiuni transversale prin structura inversorului CMOS



Substratul este de tip P. Pentru crearea tranzistorului de tip P se realizeaza, printr-un proces de difuzie, o insula N (Nwell). Zonele in care vor fi plasate tranzistoarele de tip N si P vor fi dopate prin difuzie cu impuritati de tip N (N+ Diffusion) si P (P+ Diffusion). Separarea celor doua straturi de metal1 si metal2 se efectueaza printr-un strat de dioxid de siliciu. Zonele in care cele doua straturi se suprapun poarta numele de Via m1/m2.

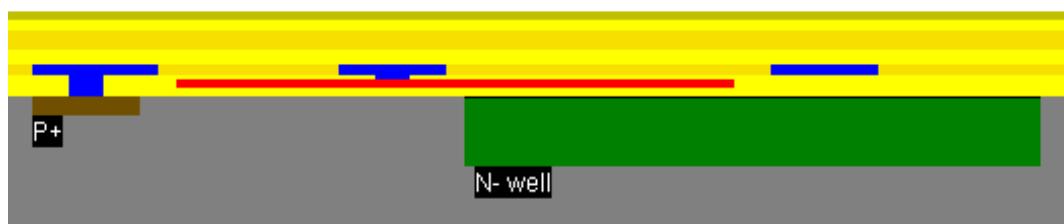
In figura de mai jos, tranzistorul din dreapta, de tip PMOS, este realizat pe o insula de tip N, in timp ce tranzistorul din stanga, de tip N, este creat pe substratul de tip P. La extrema dreapta se afla un strat de metal1, care asigura tensiunea de alimentare V_{DD} . Traseul de metal1, de la limita din stanga realizeaza conexiunea la masa, GND. Intrarea V_{IN} este realizata prin traseul din siliciu policristalin, care traverseaza zonele portilor celor doua tranzistoare si care este conectat la stratul de metal1, acesta din urma fiind conectat la stratul metal2.



Geometria inversorului CMOS

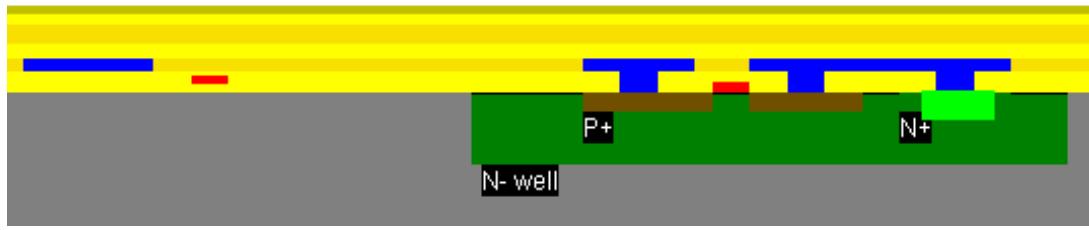


Sectiune transversala prin structura inversorului, la nivelul metal2.



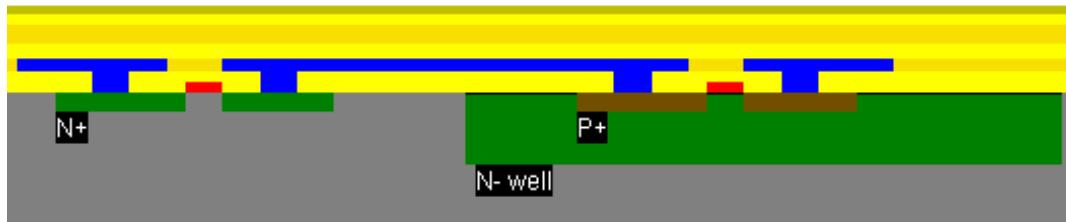
Sectiune transversala prin structura inversorului la nivelul contactului metal1-siliciu policristalin.

Se poate observa in stanga conexiunea substratului la GND prin zona de difuzie P+.



Sectiune transversala prin structura inversorului la nivelul canalului tranzistorului P.

Se poate observa in partea din dreapta conectarea insulei N, la traseul metal1 (V_{DD}) prin zona puternic difuzata N+.



Sectiune transversala prin structura inversorului la nivelul canalelor celor doua tranzistoare N si P.

Pentru realizarea unui inversor trebuie sa se creeze, pe substrat, mai multe straturi din materiale cu proprietati electrice diferite. Modurile in care vor fi depuse aceste materiale pe substrat vor fi discutate intr-un alt capitol. Deocamda se pleaca de la premisa ca ele pot fi depuse pe substrat, in zonele specificate de catre proiectant. Crearea unui inversor CMOS presupune urmatorii pasi:

- se foloseste un substrat de siliciu usor dopat P, pe care se vor depune celelalte straturi;
- se creste un strat de dioxid de siliciu si se creaza taieturile necesare in acesta;
- se creaza o insula N in substrat, prin doparea cu ioni de P sau As, insula N fiind necesara pentru realizarea tranzistorului P;
- se creste un strat subtire de dioxid de siliciu, avand rolul de izolator, in zonele in care se vor crea tranzistoarele P si N;
- se plaseaza un strat de siliciu policristalin peste stratul subtire de dioxid, pentru a forma portile celor doua tranzistoare;
- se implanteaza/difuzeaza o regiune N+ in substratul P, pentru a forma sursa si drena tranzistorului de tip N si, in mod asemănător, se implantă/difuzează o regiune P+ in insula N, pentru a forma sursa si drena tranzistorului P;

- peste intraga structura se creste un strat de oxid si se deschid in acesta taieturile de contact, pentru ca stratul de metal1, care se va depune, in continuare, sa realizeze conexiunile cu diversele straturi, in vederea: aplicarii tensiunii de alimentare, conectarii la masa, la intrare si iesire;
- in cazul in care procesul de fabricatie permite crearea celui de-al doile strat de metal se va depune un strat de oxid pe intreaga suprafata si se vor crea taieturi de contact in zonele (via) in care metal2 si metal1 trebuie sa vina in contact.

Pentru crearea formelor straturilor mentionate mai sus se utilizeaza procese litografice complexe, care vor fi discutate intr-un alt capitol.

3.2.2 | Reguli de proiectare.

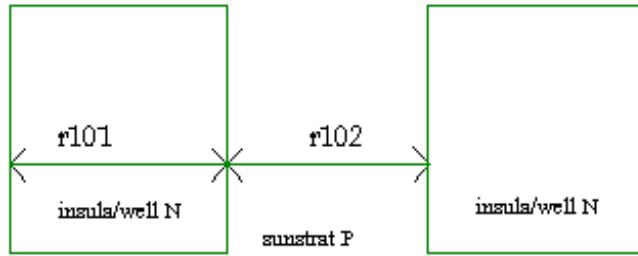
Cand se realizeaza desenele sabloanelor unor circuite integrate se urmarest ca acestea sa ocupe o suprafata cat mai mica. Procesele de fabricatie impun insa o serie de restrictii referitoare la dimensiunile minime ale unor trasee, cat si la distantele intre traseele din acelasi material sau din mteriale diferite, chiar daca se afla pe niveluri diferite.

Plecand de la elementele ce caracterizeaza diferite procese de fabricatie in termenii dimensiunilor si distantelor minime au fost generate o reguli de proiectare. Aceste regule reprezinta un ghid in proiectare, prin care se urmarest reducerea ariei ocupate de un circuit, garantandu-se insa functionarea corecta.

Specialistii au cautat sa lege regulile de proiectare de un factor ce poate caracteriza un procesele tehnologice din acest domeniu. Acest factor, care poarta numele de rezolutie a procesului, este notat cu λ si este influentat de o serie de factori legati de proces: precizia de aliniere a mastilor, precizia controlului de corodare s.a. De exemplu, in cazul procesului ATMEL-ES2 2-metal CMOS, rezolutia este egala cu $0,8\mu\text{m}$.

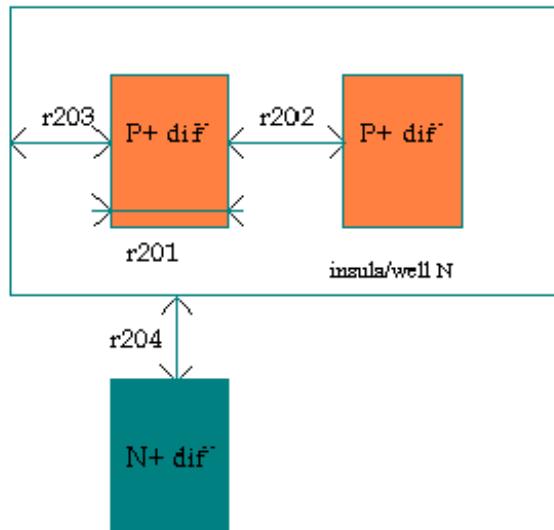
Regulile de proiectare pot fi exprimate in valori absolute (μm) sau sub forma relativa in raport cu rezolutia λ . Astfel, latimea minima a unui traseu de siliciu policristalin este de $1,6 \mu\text{m}$, in cazul procesului $0,8\mu\text{m}$ CMOS. In forma relativa, latimea minima a unui traseu de siliciu policristalin va fi egala cu 2λ . In continuare vor fi prezentate regulile de proiectare in forma relativa, pentru procesul ATMEL-ES2 2-metal $0,8\mu\text{m}$ CMOS.

Insula N



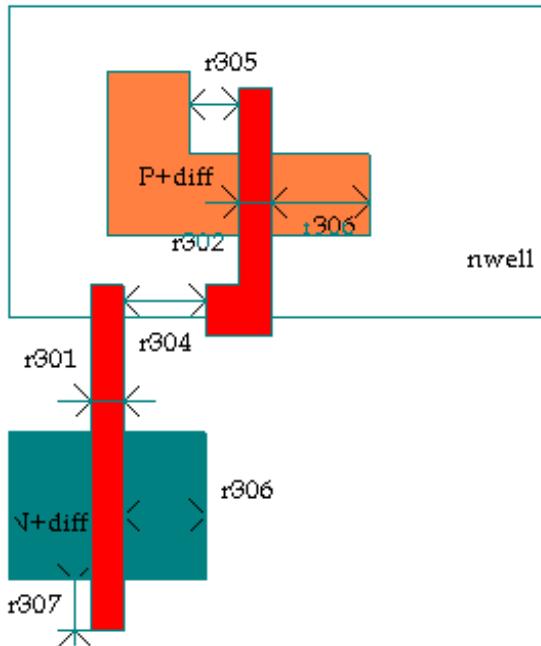
- r101 dimensiunea minima a insulei: 12
- r102 distanta minima intre insule: 12

Difuzie



- r201 dimensiunea minima a zonei de difuzie: 4
- r202 distanta minima intre doua zone de difuzie: 4
- r203 extensia insulei fata de difuzie: 6
- r204 distanta minima intre o zona de difuzie si o insula: 6

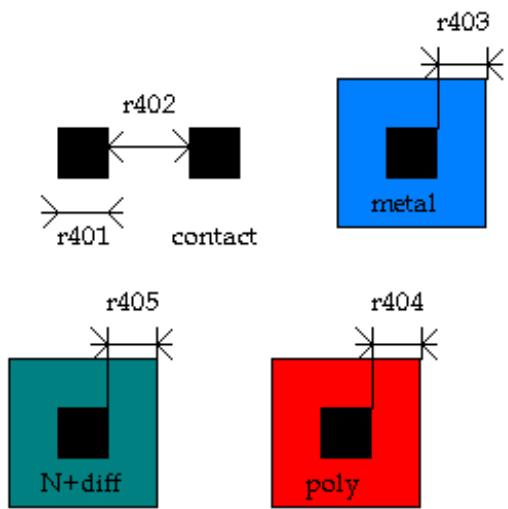
Siliciu policristalin



- r301 latimea traseului de siliciu policristalin:2
- r302 latimea portii de siliciu policristalin pe difuzie P+: 2
- r303 latimea portii de siliciu policristalin pe difuzie N+: 2
- r304 distanta minima intre doua trasee de siliciu policristalin: 3
- r305 distanta minima intre un traseu de siliciu policristalin si un strat de difuzie: 2
- r306 extensia unui strat de difuzie in raport cu stratul de siliciu policristalin: 4
- r306 extensia unui strat de siliciu policristalin in raport cu stratul de difuzie: 2

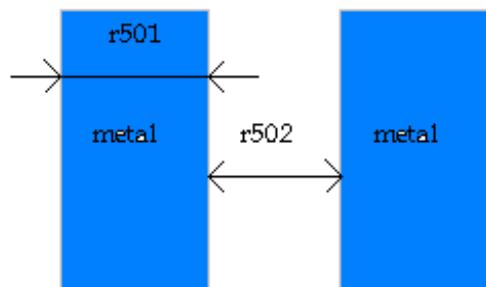
Contact.

- r401 latimea contactului: 2
- r402 distanta minima intre doua contacte: 3
- r403 extensia metalului fata de taietura de contact: 2
- r404 extensia siliciului policristalin fata de taietura de contact: 2
- r405 extensia difuziei fata de taietura de contact: 2



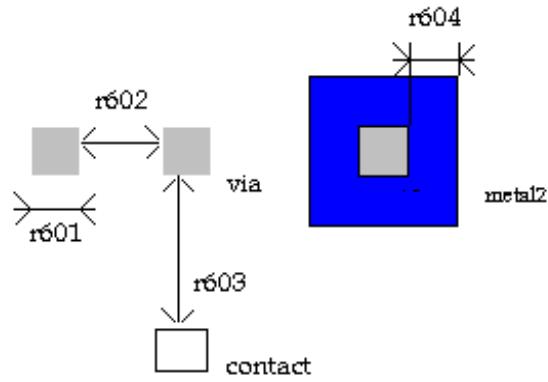
Metal1

- r501 dimensiunea minima a metalului1: 3
- r502 distanta minima intre doua zone de metal1: 3



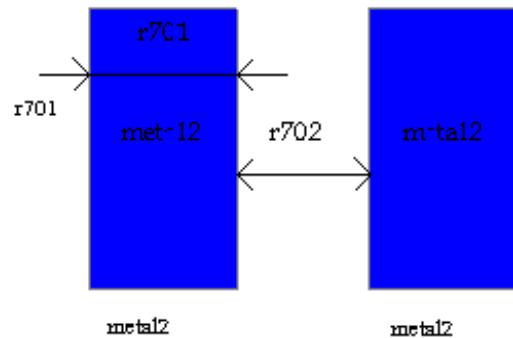
Via.

- r601 latimea zonei via: 3
- r602 distanta minima intre doua zone via: 3
- r603 distanta minima intre via si contact: 3
- r604 extensia metal1 peste via: 2
- r605 extensia metal2 peste via: 2



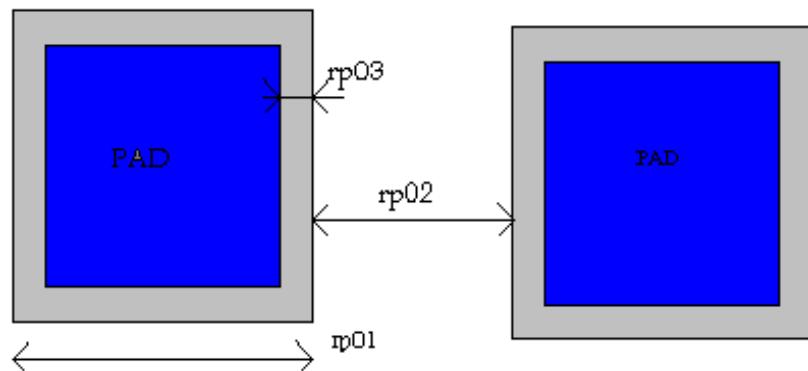
Metal2.

- r_{701} dimensiunea minima a zonei de metal2: 5
- r_{702} distanta minima intre doua zone de metal2: 5



Plotul

- rp_{01} dimensiunea minima 100 μm
- rp_{02} distanta minima intre doua ploturi: 100 μm

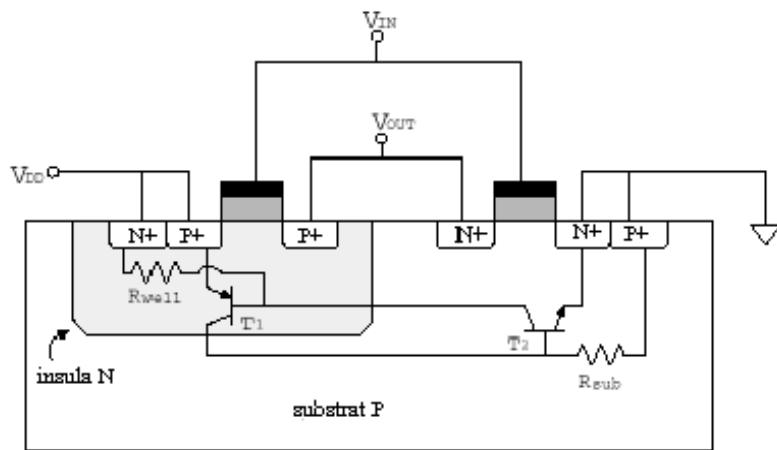


- rp03 deschiderea in pasivizare fata de via: 5 μm
- rp04 deschiderea in pasivizare fata de metale: 5 μm
- rp05 distanta minima intre plot si zone active fara legatura cu plotul: 20 μm

3.2.3 Efectul Latchup.

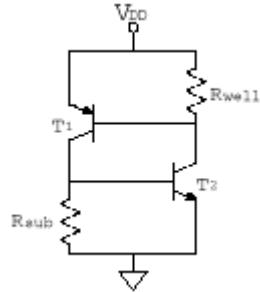
Primele circuite CMOS prezintau efectul latchup, care putea conduce la deteriorarea lor.

Dupa cum se poate observa din desenul de mai jos, in conditiile existentei unei perechi de tranzistoare PMOS si NMOS este inerenta aparitia unor tranzistoare bipolare parazite.



In mod normal circuitul format din aceste doua tranzistoare parazite nu functioneaza. O serie de fenomene cu caracter tranzitoriu pot aduce in conductie acest circuit. Intrucat acest circuit are o reactie pozitiva, dupa aducerea lui in stare de conductie, el va ramane in aceasta stare pana la deconectarea sursei de alimentare. Activarea acestor tranzistoare bipolare impiedica functionarea normala a circuitului CMOS si chiar il poate distruge. Fenomenul de latch-up apare cu precadere in apropierea structurilor de I/E, unde fenomenele tranzitorii sunt mai frecvente.

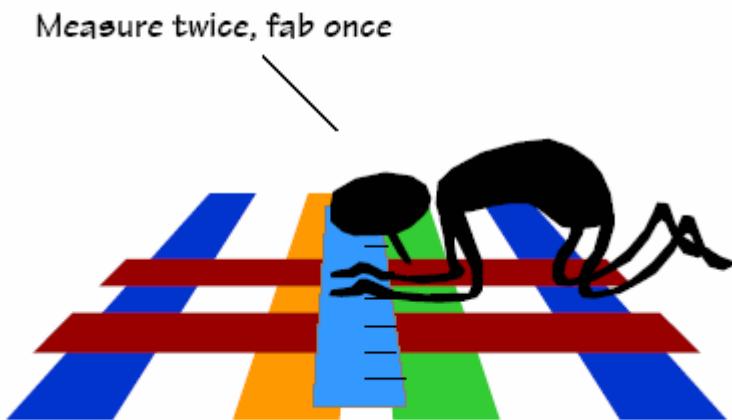
Dupa cum se poate observa, in figura de mai sus, sursa tranzistorului PMOS, insula N si substratul P formeaza un tranzistor PNP. De asemenea, un tranzistor lateral este format de catre insula N a tranzistorului P, substratul P si sursa tranzistorului NMOS. Analiza poate fi efectuata pe schema echivalenta de mai jos:



Un scenariu posibil este acela care ar conduce la deschiderea celor doua tranzistoare. Un curent de electroni este injectat in substratul P. Acesta va face ca T_2 sa fie polarizat in sensul conductiei de catre R_{sub} , care va deschide T_2 . Tranzistorul T_2 fiind deschis va crea o tensiune de polarizare pe R_{well} , care va deschide T_1 , ceea ce va duce la injectarea unui curent mai mare in substrat. Astfel, se va stabili o cale intre V_{DD} si GND. Pe masura ce curentul trece prin aceasta cale influenta circuitului parazit va creste.

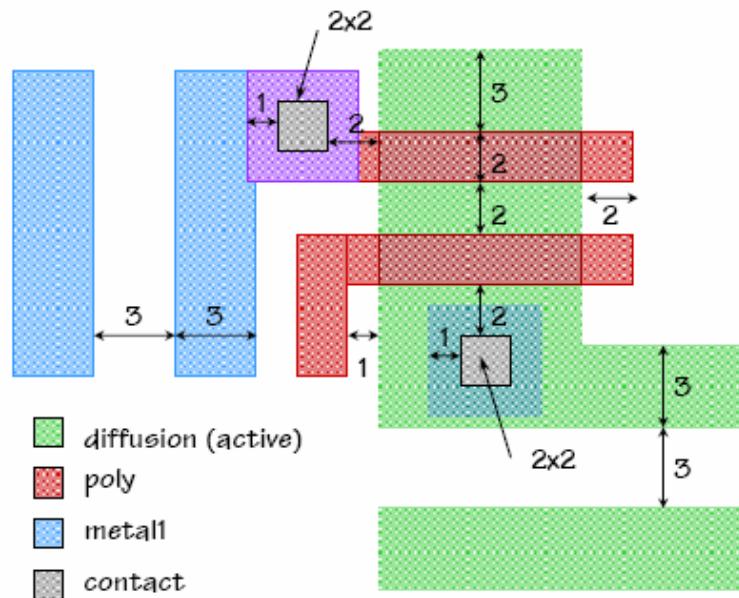
Exista mai multe metode pentru limitarea sau prevenirea acestui fenomen, care sunt, fie la indemana proiectantului de circuit, fie la indemana proiectantului de proces. Din punctul de vedere al proiectantului de circuit, se poate actiona prin cresterea numarului de contacte intre substrat si insula. Cea de-a doua metoda consta in reducerea valorii produsului amplificarilor $\beta_{pnp} \times \beta_{npn}$ la o valoare subunitara. Aceasta metoda nu este la indemana proiectantului de circuit. Cea de-a treia metoda consta in decuplarea perechii PNP – NPN, folosind tehnologia SOI (Silicon On Insulator).

CMOS Layout

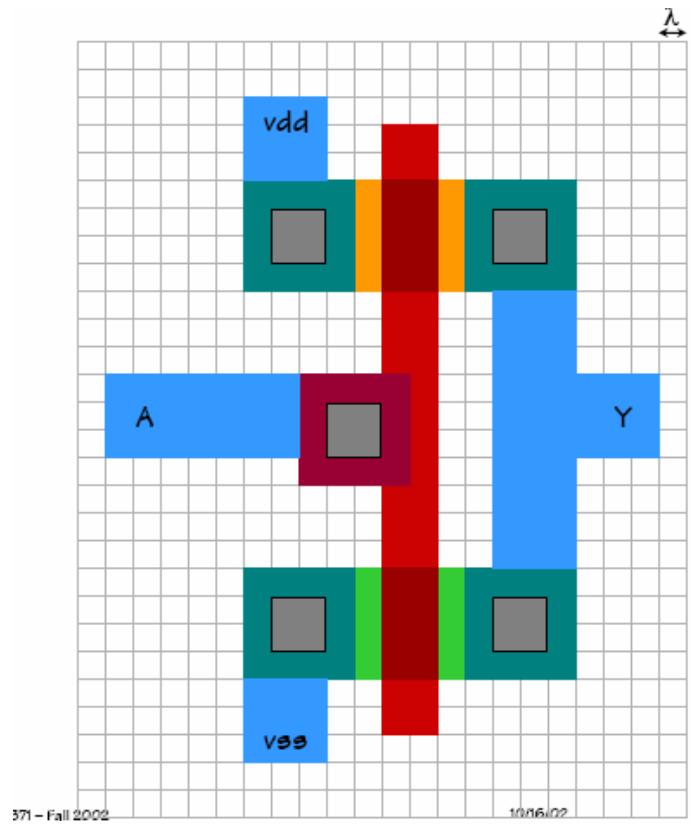


Lambda-based design rules

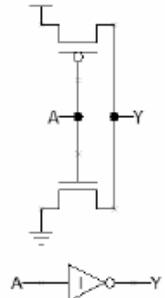
One lambda = one half of the “minimum” mask dimension, typically the length of a transistor channel. Usually all edges must be “on grid”, e.g., in the MOSIS scalable rules, all edges must be on a lambda grid.



More info at: <http://www.mosis.org/Technical/Designrules/scmos/scmos-main.html>



Sample “Lambda” Layout

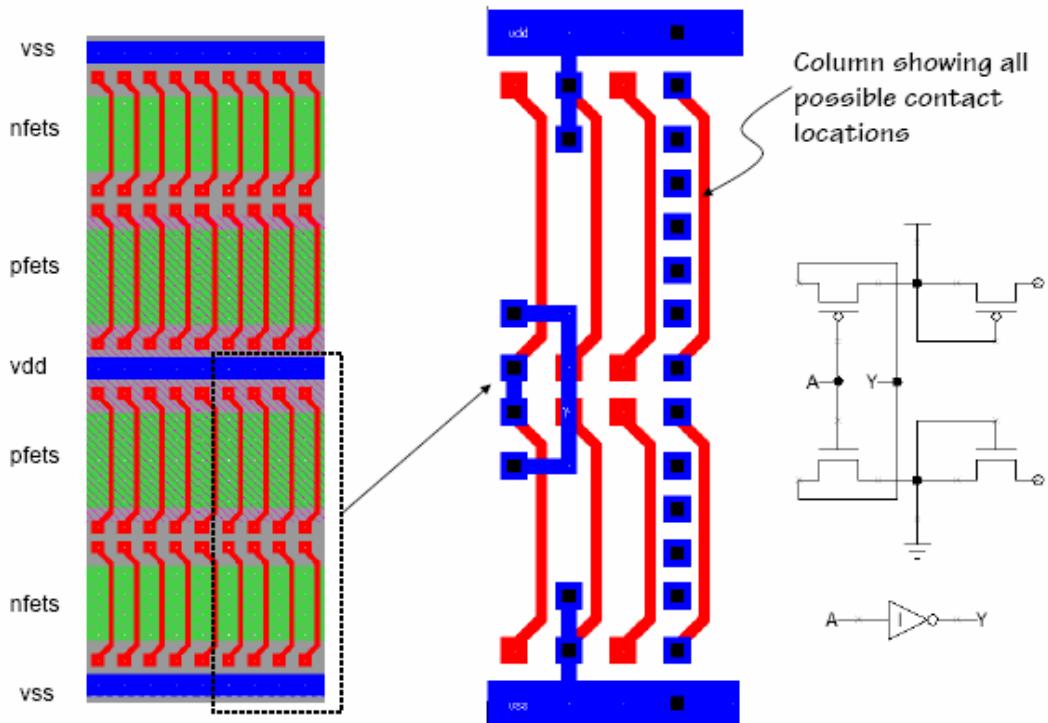


371 - Fall 2002

10/16/02

L12 – CMOS Layout

Sample Sea-of-Gates Layout



Lambda vs. Micron rules

Lambda-based design rules are based on the assumption that one can scale a design to the appropriate size before manufacture. The assumption is that **all manufacturing dimensions scale equally**, an assumption that "works" only over some modest span of time. For example: if a design is completed with a poly width of 2λ and a metal width of 3λ then minimum width metal wires will always be 50% wider than minimum width poly wires.

Consider the following data from Weste,

Table 3.2:

	lambda rule	lambda = 0.5μ	micron rule
contacted metal pitch	1λ	0.5μ	0.375μ
1/2 * contact size	1λ	0.5μ	0.5μ
contact surround	3λ	1.5μ	1.0μ
metal-to-metal spacing	1λ	0.5μ	0.5μ
contact surround	1λ	0.5μ	0.375μ
1/2 * contact size	8λ	4μ	2.75μ

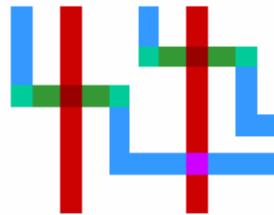
Scaled design is legal but much larger than it needs to be!

Retargetable Layouts

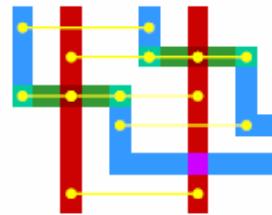
So, should one use lambda rules, or not?

- probably okay for retargeting between "similar" processes, e.g., when later process is a simple "shrink" of the earlier process. This often happens between generations as a mid-life kicker for a process. Some 0.35μ processes are shrinks of an earlier 0.5μ process. Can be useful for "fabless" semiconductor companies.
- most industrial designs use micron rules to get the extra space efficiency. Cost of retargeting by hand is acceptable for a successful product, but usually it's time for a redesign anyway.
- invent some way of entering a design symbolically but use a more sophisticated technique for producing the masks for a particular process. Insight: relative sizes may change but topological relationship between components does not. So, instead of shrinking a design, compact it! LED offers compaction for leaf cells...

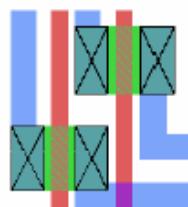
Sticks and Compaction



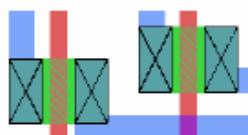
Stick diagram



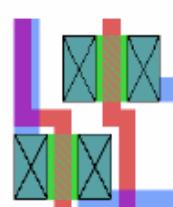
Horizontal constraints
for compaction in X



Compact X then Y

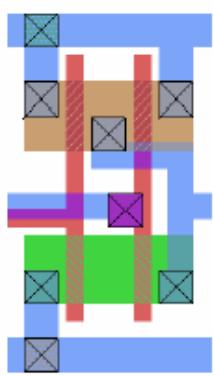


Compact Y then X



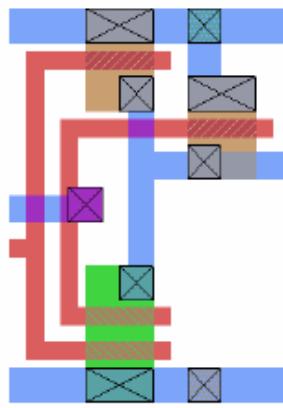
Compact X with jog
insertion, then Y

Choosing a “style”



Vertical Gates

Good for circuits where fet sizes are similar and each gate has limited fanout. Best choice for multiple input static gates and for datapaths.



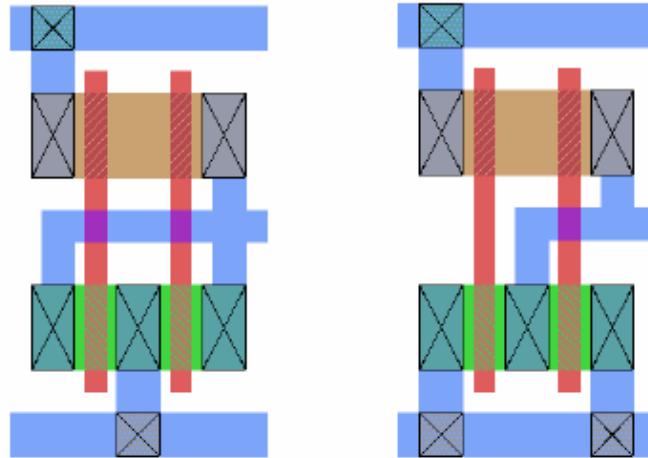
Horizontal Gates

Good for circuits where long and short fet's are needed or where nodes must control many fet's. Often used in multiple-output complex gates (e.g., sum/carry circuits).

What about routing signals between gates? Note that both layouts block metal/poly routing inside the cell. Choices: metal2 routing over the cell or routing above/below the cell.

- ♦ avoid long (> 50 squares) poly runs
- ♦ don't "capture" white space in a cell
- ♦ don't obsess over the layout, instead make a second pass, **optimizing where it counts**

Optimizing connections

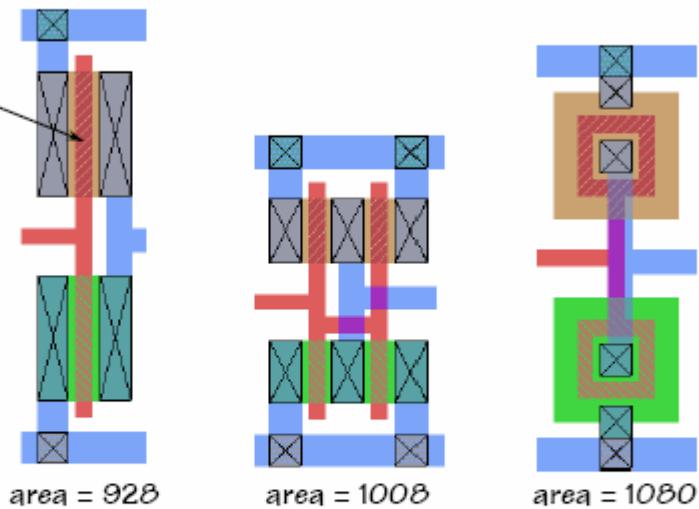


Which is the better gate layout?

- considering node capacitances?
- considering "composability" with neighboring gates?

can't make gates too long because of poly resistance! Eventually really large transistors have to be broken into smaller transistors in parallel.

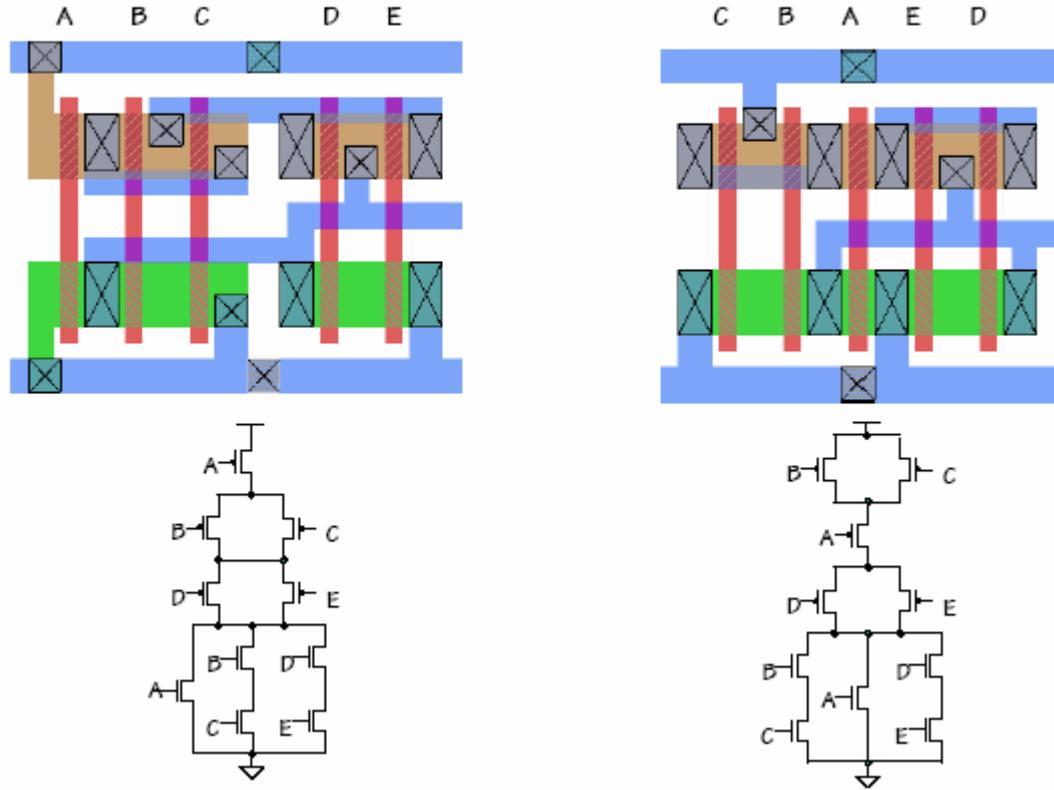
Big vs. Parallel



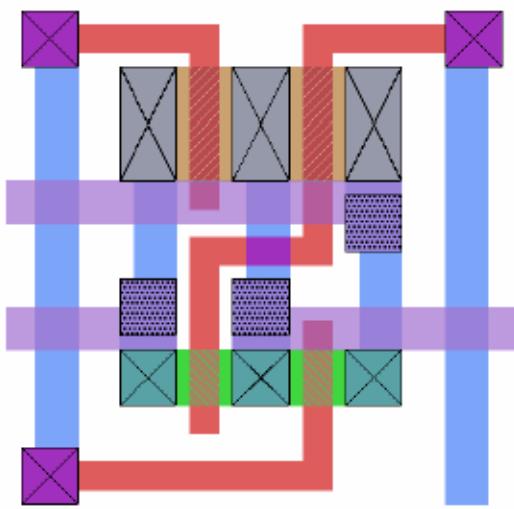
Which is the better gate layout?

- considering node capacitances?
- considering "composability" with neighboring gates?

Eliminating Gaps



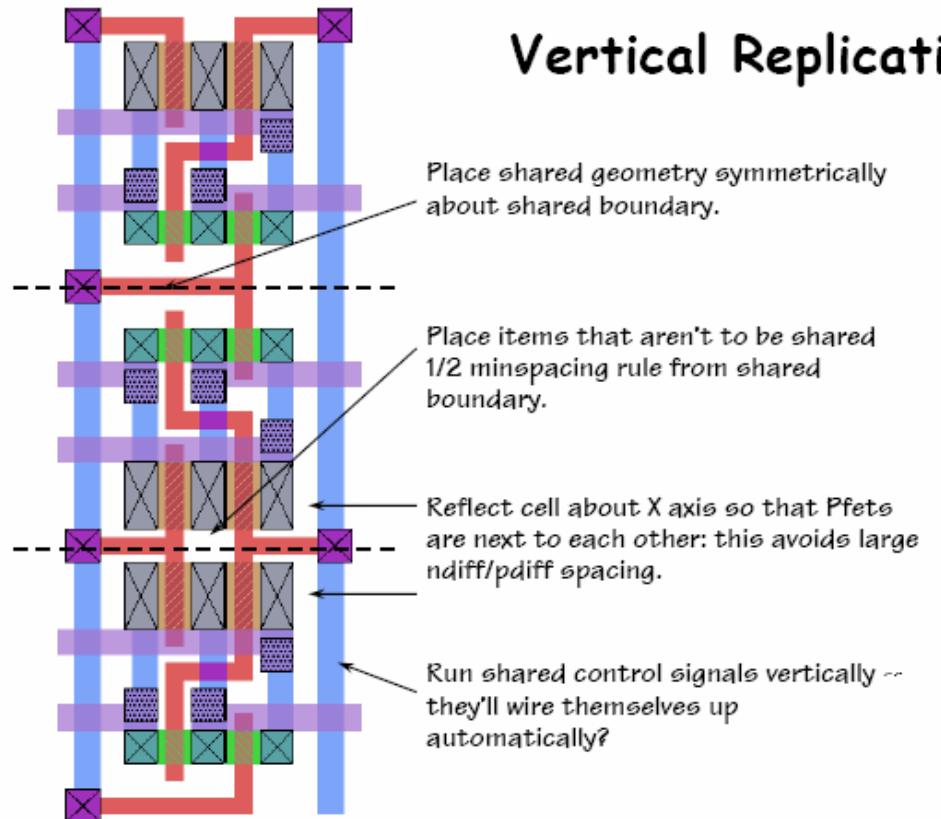
Replicating Cells



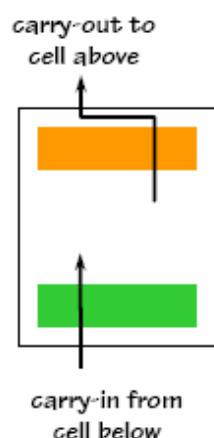
What does this cell do?

What if we want to replicate this cell vertically, i.e., make a stack of the cells, to process many bits in parallel?

- what nodes are shared among the cells?
- what nodes aren't shared?
- how should we arrange the cells vertically?

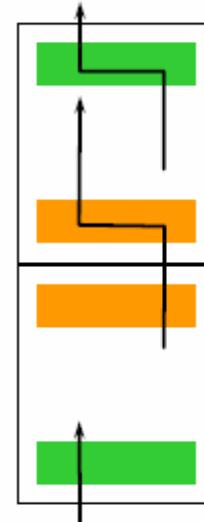


Vertical Intercell Routing



Suppose we have a signal that will run vertically from one cell to the next, e.g., the carry-out from one cell becomes the carry-in for the cell above.

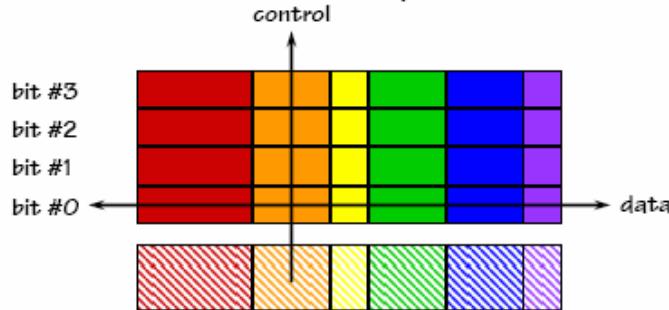
Looks okay until we reflect the cell when we do the vertical replication!



Solution: we have to do the routing for vertical intercell signals for a pair of cells, then replicate the pair (complete with routing) vertically.

Building a Datapath

It's often the case that we want to operate on many bits in parallel. A sensible way to arrange the layout of this sort of logic is as a **datapath** where data signals run horizontally between functional units and control signals run vertically to all the bits of a particular functional unit:



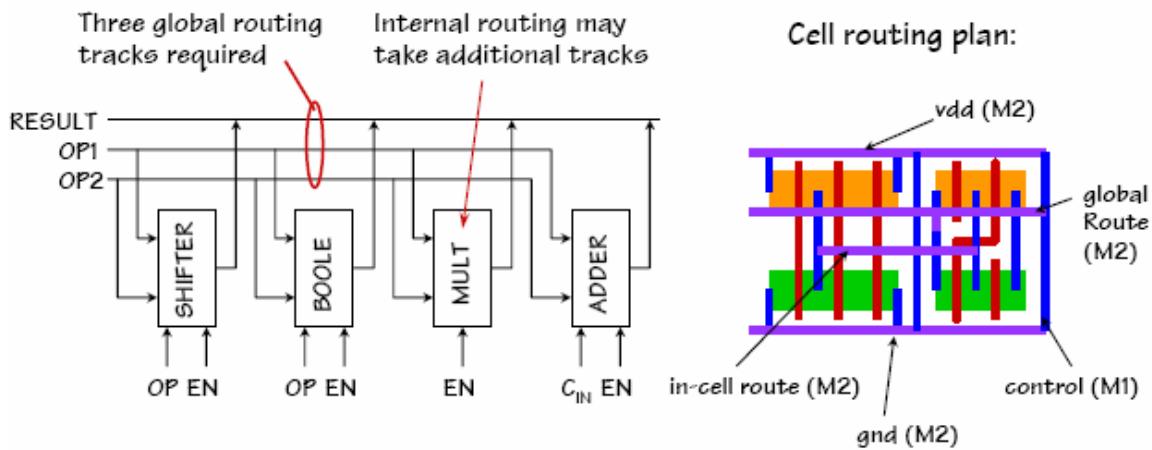
Logic that generates the control signals can be placed at the bottom of the datapath. If control logic is complicated or irregular, it might be placed in a separate standard cell block and only the control signal buffers placed just below the datapath. Although it's tempting to run control signals in poly (so they can control fets) this is unwise for tall datapaths because of poly resistance (e.g., 32 bits \times 20u/bit = 640u = ~1000 squares = ~20k ohms!)

Datapath Bit Pitch

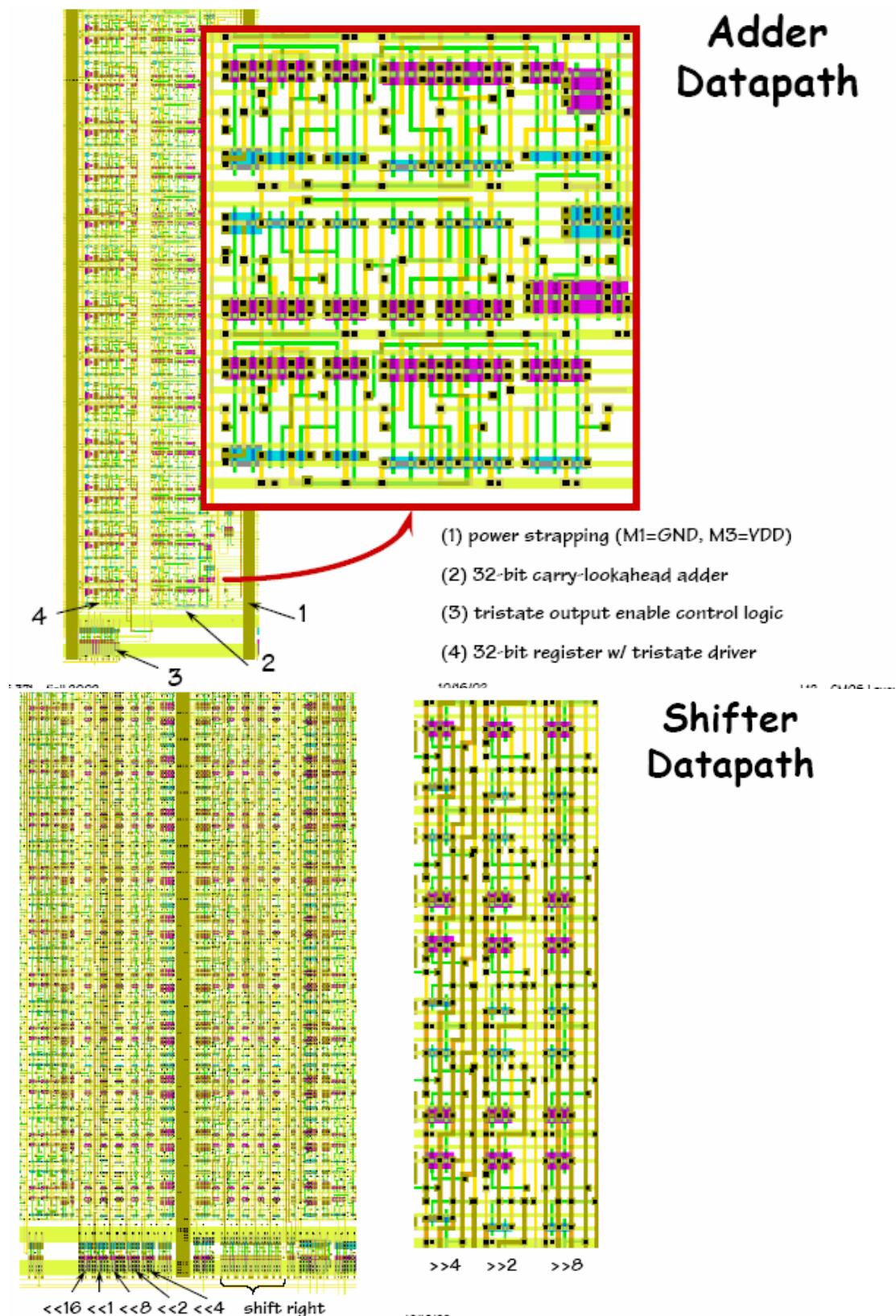
How tall should we make each bit of the datapath? That depends on

- ♦ the width of the nfets and pfets
- ♦ how much in-cell routing there is
- ♦ how much over-the-cell global routing there is

Global routes can be determined from datapath schematic:

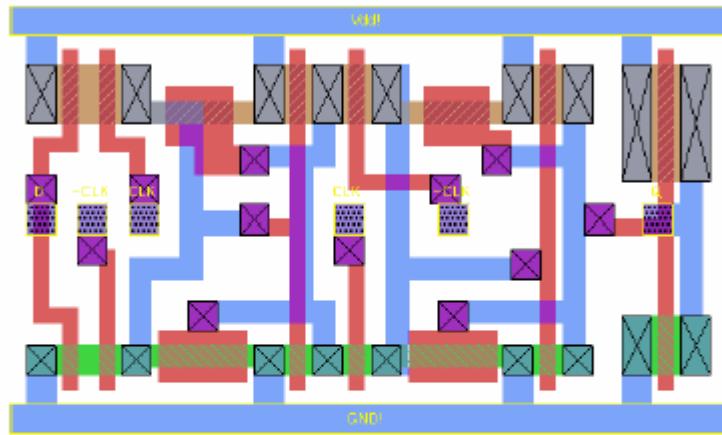


Adder Datapath



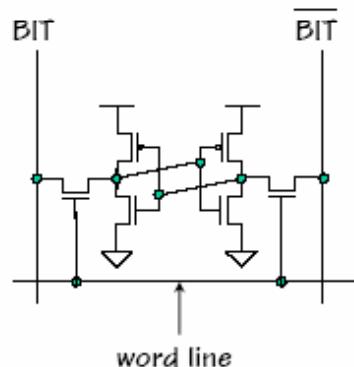
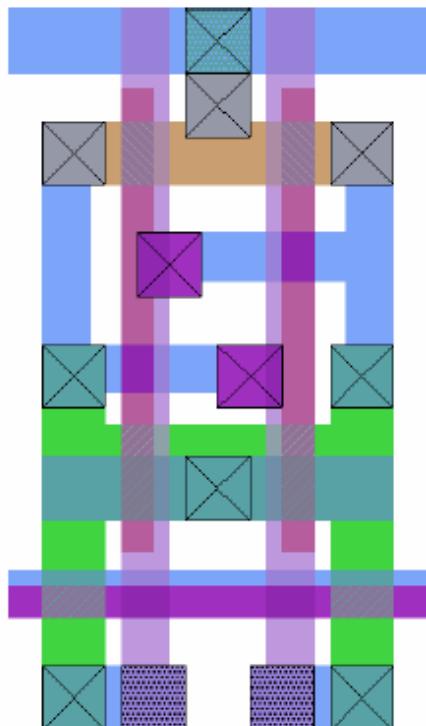
Shifter Datapath

Design for Reuse



- what's this cell do?
- what are the "fat" fets?
- Cell was designed for placement "under" a metal2/metal3 routing grid. How was the layout affected by this design requirement?

Think Globally



- How are neighboring cells placed?
- Isn't the word line a long poly wire?
- Where's the p-substrate contact?

Checking Layouts

Design Rule Checker (DRC). This is a program that checks each piece of the layout against the process design rules. This is a slow process:

- canonicalize layout into a set of leading and trailing non-overlapping mask edges. Some boolean mask operations may be needed.
- determine electrical connectivity and label each edge with the node it belongs to.
- test each edge end point against neighboring edges to check for spacing (leading edges) and width (trailing edges) violations.

Layout vs. Schematic (LVS). First a netlist is **extracted** from the layout. Use the electrical info generated by the DRC and then recognize transistors are juxtapositions of channel with diffusion. Then see if extracted netlist is **isomorphic** to the schematic netlist. This is done by a coloring algorithm:

- initialize all nodes to the same color
- compute a new color for each node as some hashing function involving the colors of connected (ie, thru a fet) nodes.
- nodes that have a unique color are isomorphic to similarly colored node in other network
- worry about parallel fets, ambiguous nodes

Example Sea-of-Gates Layout

Let's build the layout for an 8-bit ripple carry adder using the Ocean tools. First we have to set up a directory for our project:

```
athena% setup 6.371
Attaching 6.371 ...
Running commands in /mit/6.371/.attachrc ...

6.371% mk6371 rippleadder
----- creating fishbone project rippleadder -----
----- importing primitives -----
----- importing lib6371 -----
----- copying default config files -----

...done. Enjoy your new project "rippleadder" !!

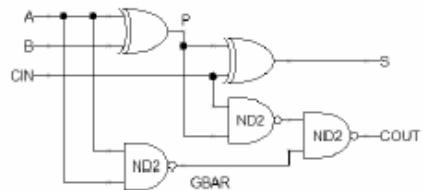
6.371% cd rippleadder
6.371%
```

Enter SLS netlists

```
#include "sls_prototypes/lib6371.ext"

network fa(terminal a,b,cin,s,cout,vss,vdd)
{
{xgbar} nand2(a,b,gbar,vss,vdd);
{xp} xor2(a,b,p,vss,vdd);
{xs} xor2(p,cin,s,vss,vdd);
{xcl} nand2(p,cin,c1,vss,vdd);
{xcout} nand2(c1,gbar,cout,vss,vdd);
}

network adder8(terminal a[0..7],b[0..7],cin,s[0..7],cout,vss,vdd)
{
{xfa0} fa(a[0],b[0],cin,s[0],c0,vss,vdd);
{xfa1} fa(a[1],b[1],c0,s[1],c1,vss,vdd);
{xfa2} fa(a[2],b[2],c1,s[2],c2,vss,vdd);
{xfa3} fa(a[3],b[3],c2,s[3],c3,vss,vdd);
{xfa4} fa(a[4],b[4],c3,s[4],c4,vss,vdd);
{xfa5} fa(a[5],b[5],c4,s[5],c5,vss,vdd);
{xfa6} fa(a[6],b[6],c5,s[6],c6,vss,vdd);
{xfa7} fa(a[7],b[7],c6,s[7],cout,vss,vdd);
}
```



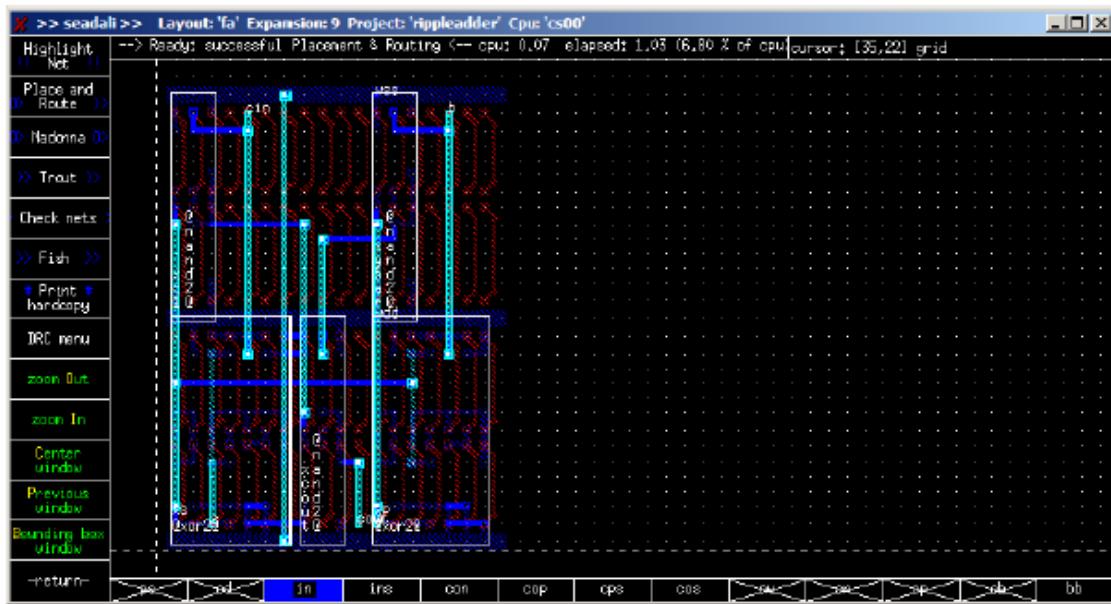
See /mit/6.371/examples/adder8.sls

6.371% cscls adder8.sls
File cells.sls:
Parsing network: fa
Parsing network: adder8
6.371% seadali & ← Ignore complaint about "Cannot allocate 3 bitplanes"



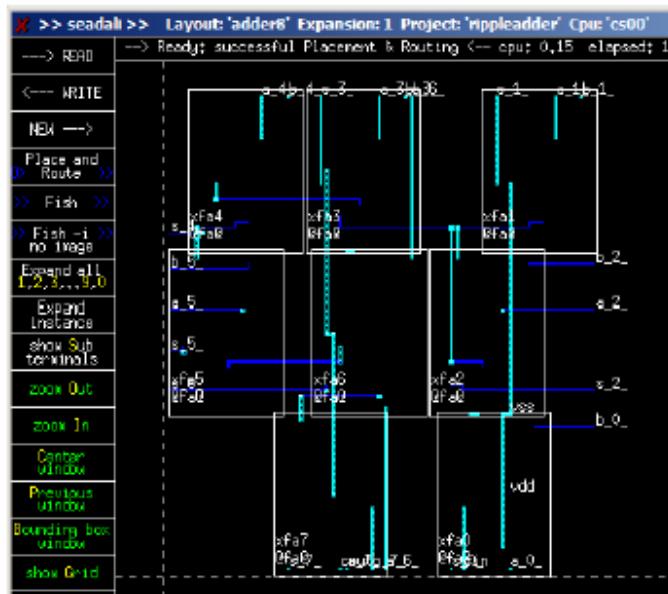
Try automated tools...

```
[1] click "automatic tools"
[2] click "Place and Route"
[3] Enter circuit: type "fa"
[4] click "*** DO IT! ***"
```



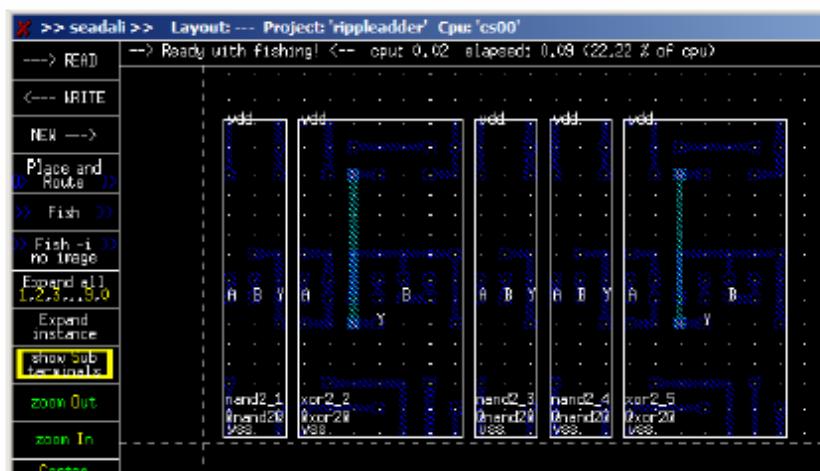
Save layout in database and layout ADDER8

```
[1] click "-return-"
[2] click "database"
[3] click "<--- WRITE"
[4] click "fa"
[5] click "Place and Route"
[6] type "adder8"
[7] click "*** DO IT! ***"
[8] click "<--- WRITE"
[9] click "adder8"
```



Now try manual layout

- [1] click "NEW --->"
- [2] click "-return-
- [3] click "instances"
- [4] click "ADD imported instance"
- [5] click "nand2", click in main screen to place
- [6] type "o" several times to zoom out, "c" to recenter screen
- [7] click "next", "xor2", click to place next to NAND2
- [8] continue, adding 2 more NAND2 and last XOR2
- [9] when done, click "-return-",
- [10] type "S" to show sub terminals
- [11] type "2" show interior routing



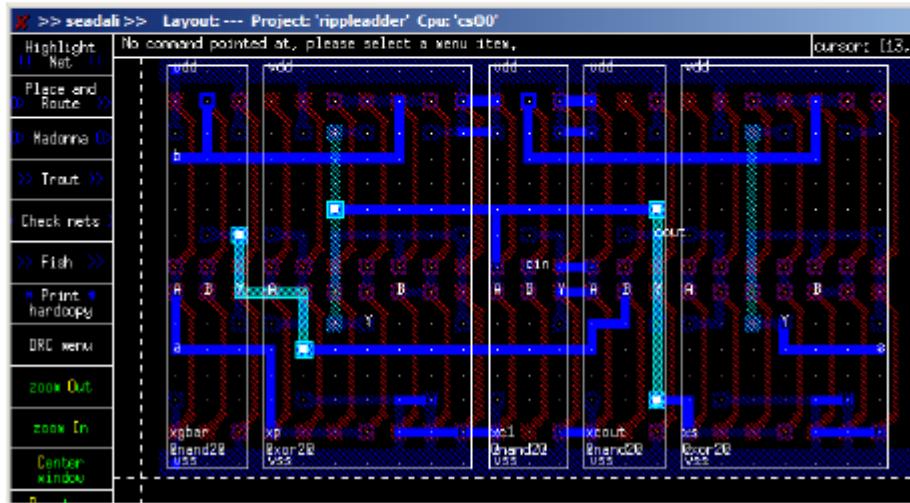
Prepare for autoroute

- [1] click "-return-
- [2] click "terminals"
- [3] click "ADD terminal"
- [4] use cursor to select grid point for terminal "a"
- [5] terminal name: type "a" (capitalization counts!)
- [6] position terminal "b" on left edge
- [7] position terminal "s" on right edge
- [8] position terminal "cout" on Y output of rightmost NAND2
- [9] position terminal "cin" on B input of middle NAND2



Autoroute

- [1] click "-return-"
- [2] click "automatic tools"
- [3] click ">> Trout >>"
- [4] enter circuit name: type "fa", return
- [5] click "border terminals" to deselect
- [6] click "*** DO IT! **"
- [7] dismiss various pop up screens
- [8] return to database screen, save layout for FA

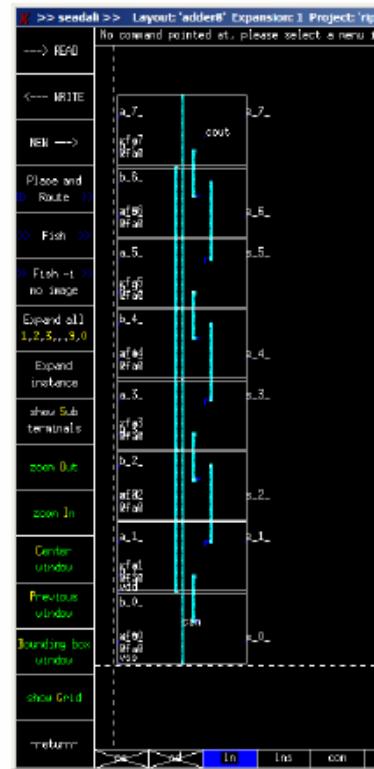


Place and Route ADDER8

- [1] start a "NEW --->" layout
- [2] use instance tools to make a stack of FAs
- [3] use "set instance name" to name them xfa0..xfa7
- [4] use ">> Trout >>" on automated tools screen to complete the route for ADDER8
- [5] save layout in database

Morals from this story:

- you have to help tools find the structure
- control randomness of tools
 - place instances based on global plan
 - place terminals based on global plan
 - automated routing okay
 - automated placement for random logic



Prepare Simulation Stimulus

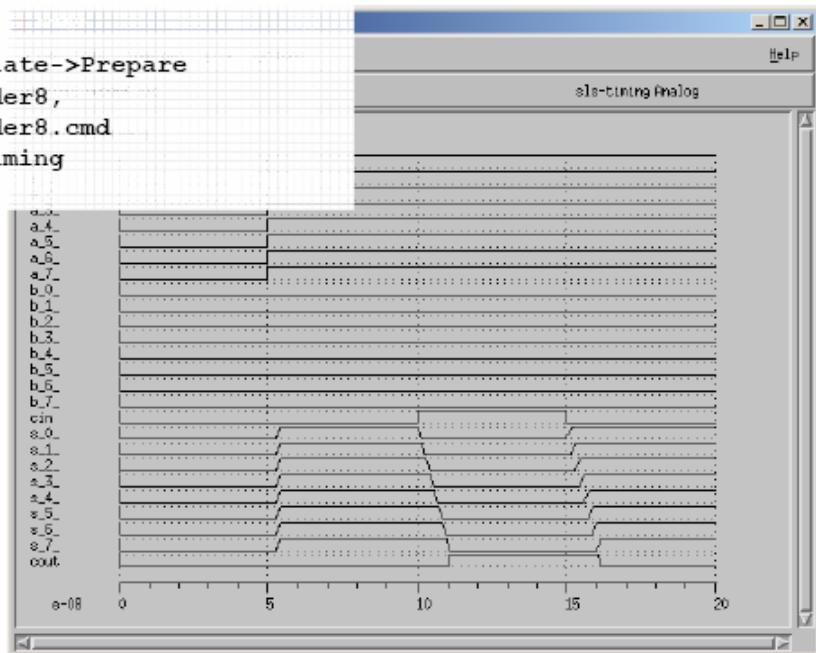
See /mit/6.371/examples/adder8.cmd
and /mit/6.371/examples/Adder8.cmd

```
plot a[0..7], b[0..7], cin, s[0..7], cout
option level = 3
option simperiod = 4
option sigunit = 50.000000e-09
option outacc = 10p
/*
 */
tstep 0.2n
trise 0.5n
tfall 0.5n
/*
*/
.options cptime=500
/*
*/
set vdd = h~
set vss = l~
set a[0] = l*1 h*1 h*1 h*1
set a[1] = l*1 h*1 h*1 h*1
set a[2] = l*1 h*1 h*1 h*1
set a[3] = l*1 h*1 h*1 h*1
set a[4] = l*1 h*1 h*1 h*1
set a[5] = l*1 h*1 h*1 h*1
set a[6] = l*1 h*1 h*1 h*1
set a[7] = l*1 h*1 h*1 h*1
set b[0] = l*1 l*1 l*1 l*1
set b[1] = l*1 l*1 l*1 l*1
set b[2] = l*1 l*1 l*1 l*1
set b[3] = l*1 l*1 l*1 l*1
set b[4] = l*1 l*1 l*1 l*1
set b[5] = l*1 l*1 l*1 l*1
set b[6] = l*1 l*1 l*1 l*1
set b[7] = l*1 l*1 l*1 l*1
set cin = l*1 l*1 h*1 l*1
```

Extract and Run Simulation

```
6.371% cp /mit/6.371/examples/*8.cmd .
6.371% space -c fa; ghoti Fa
6.371% space -c adder8; ghoti Adder8
6.371% simeye&
```

```
[1] select Simulate->Prepare
[2] Circuit: Adder8,
[3] Stimuli: Adder8.cmd
[4] Type: sls-timing
[5] click "Run"
```



Capitolul 4. Fabricarea circuitelor VLSI - CMOS.

4.1. Introducere.

Procesul fundamental de fabricatie a unui circuit integrat consta in formarea selectiva a unor paturi/straturi de tip semiconductoare, dielectric si metal, pe suprafata placetei de siliciu monocristalin. Dupa realizare, aceste materiale devin componente active si pasive ale circuitului integrat. In acest capitol vor fi examineate procesele fundamentale legate de materiale, litografia si procedeele pentru crearea unui circuit integrat. Se va avea in vedere tehnologia CMOS cu doua insule. Tratarea nu are un caracter exhaustiv, ci mai mult generic, ilustrand aspectele de detaliu comune celor mai multe procese CMOS.

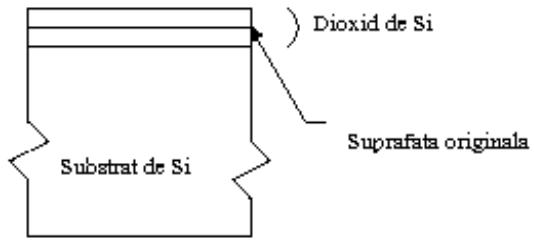
4.2. Tehnologia de procesare a materialului de baza.

4.2. Fabricarea placetelor.

In oricare proces VLSI placheta reprezinta materialul de la care se porneste. In tehnologia Si – CMOS, placheta se realizeaza dintr-un lingou de siliciu monocristalin. Lingoul se obtine printr-un procedeu de tragere dintr-un creuzet, in care se afla siliciu pur topit la o temperatura de circa 1475 °C. Cea mai frecventa metoda de obtinere a lingoului din siliciu monocristalin se datoreaza lui Czochralski. Lingoul reprezinta un monocristal de Si, aproape fara defecte, cu o lungime de mai multe zeci de centimetri si cu un diametru de circa 10 cm. Pentru a impiedeca aparitia altor impuritati, monocristalul este dopat N. Monocristalul este taiat sub forma unor discuri/plachete cu o grosime, din considerente mecanice, de circa 300 μm , intrucat structurile electrice nu depasec in grosime 10 μm . Placheta, de regula de tip N, este acoperita cu un strat epitaxial de Si aproape intrinsec, inainte de a incepe procesul de fabricare a circuitului integrat. Aceasta prelucrare initiala conduce la cresterea rezistentei *latch-up*. Partea posterioara a placetei poate beneficia de un proces de implantare ionica pentru reducerea rezistentei electrice de contact, la impachetarea finala.

4.2.2. Oxidarea.

Oxidarea se refera, de regula, la cresterea sau depozitarea SiO_2 pe suprafata placetei. La o temperatura inalta de 1000 °C si intr-o atmosfera de O_2 , o placheta expusa se va oxida. Aceasta structura de SiO_2 foloseste Si de pe placheta, fiind plasata, atat in materialul placetei, cat si pe suprafata acesteia, dupa cum se va vedea in desenul de mai jos.



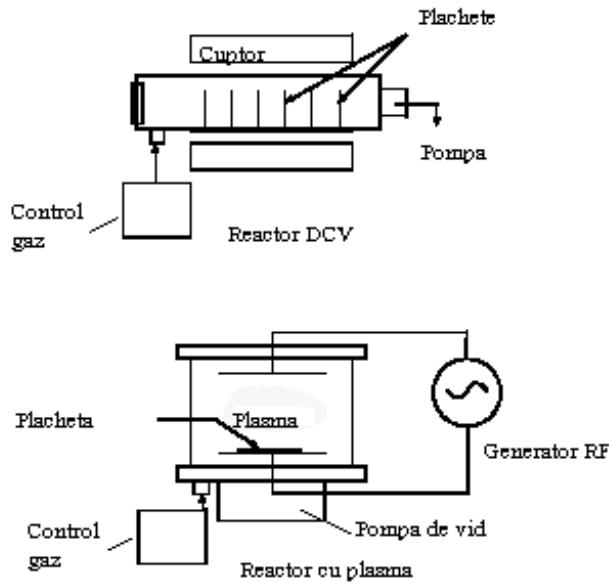
Acest proces creste un strat de SiO_2 de calitate, care poate fi folosit, fie ca dielectric izolator, fie ca oxid pentru poarta. In prezent, oxidul de poarta cu cele mai bune proprietati electrice este format prin oxidare termica. Aceasta impune ca procesele care vor urma sa aibe loc pe Si expus.

4.2.3. Depunerea de strat subtire.

O serie de materiale utilizate in fabricarea circuitelor integrate isi au originea in stare gazoasa. Aceste materiale sau straturi sunt constituite din siliciul policristalin, materialul principal pentru poarta tranzistorului MOS, cat si din izolatori dielectrici (SiO_2 si Si_3N_4), folositi intre diferitele straturi conductoare.

In practica se intalnesc numeroase tehnici de depunere, care au multe elemente comune. Cea mai simpla, Depunerea Chimica de Vapori (DCV) la presiune atmosferica, este folosita pentru formarea siliciului policristalin. Placheta este incalzita in cuptor in prezenta silanului (SiH_4), la o temperatura de circa $650\text{ }^{\circ}\text{C}$, care este suficienta pentru a descompune moleculea de SiH_4 si a depune un atom de Si pe suprafata placetei. Pe suprafata expusa a placetei se vor forma un strat de $1\text{--}10\text{ }\mu\text{m}$ din cristale de siliciu cu diverse orientari.

O alta tehnica, avand o importanta crescanda, este cea bazata pe depunere de materiale aflate in stare de plasma. Plasma reprezinta un gaz neutru cu un numar egal de electroni si atomi ionizati sau molecule ionizate. Un material poate fi adus in stare de plasma prin diverse procedee de excitare bazate pe campuri de radio-frecventa sau microunde. Plasma este generata in incinte vidate, la presiune joasa, ceea ce permite gazului sa se descompuna mai usor. In timpul depunerii gazele necesare sunt introduse in incinta vidata la debit si presiune constanta. Pentru a accelerata electronii, care provoaca coliziunile, excitarea si ionizarea se folosesc campuri de radio-frecventa. Generarea de radicali activi usureaza cerintele impuse pentru mentinerea placetei la temperatura inalta, in timpul procesarii. In figura de mai jos se prezinta o instalatie cu DCV si un reactor cu plasma.



4.2.4. Difuzia.

Difuzia reprezinta un proces termic de dopare N sau P a Si pentru a-i schimba caracteristicile electrice. Desi nu are loc o depunere sau o crestere de material, stratul care a suferit un proces de difuzie are un caracter critic in formarea dispozitivului CMOS. De exemplu, regiunile sursa si drena, ale unui tranzistor NMOS, sunt realizate in timpul unui proces de difuzie. In particular, atunci cand se proiecteaza o poarta de baza, sunt folosite in mod intensiv straturile de difuzie. Difuzia are loc, de cele mai multe ori, plecand de la o sursa chimica in stare de vaporii, la temperatura inalta. Mai recent, difuzia a fost cuplata cu implantarea ionica, pentru a obtine regiuni mai bine izolate ale dispozitivelor.

4.2.5. Implantarea ionica.

Implantarea ionica consta in introducerea fortata a unor atomi ionizati sau molecule ionizate intr-un material tinta, in conditiile in care li s-a imprimat o energie suficienta, pentru a penetra suprafata materialului. Sursa de ioni o constituie plasma, extractia acestora realizandu-se cu ajutorul unor campuri puternice de CC sau RF. Intrucat permite controlul multor parametri in timpul procesului, implantarea ionica este utilizata in mod extensiv in fabricarea VLSI. De exemplu, adancimea de patrundere a atomilor este controlata direct prin potentialul de accelerare, in timp ce doza este controlata prin produsul curent \times timp de expunere. Dozele

coborate sunt folosite pentru controlul tensiunii de prag, atat la tranzistoarelor active, cat si la cele parazite. Dozele mari sunt utilizate in formarea regiunilor de sursa si drena, prin autoaliniere. Se poate afirma ca implantarea ionica este direct responsabila pentru succesul CMOS fata de alte tehnologii. De exemplu, unul dintre motivele utilizarii dispozitivelor PMOS, la inceputul tehnologiei MOS, a fost controlul inadecvat al tensiunii de prag la fabricarea dispozitivelor NMOS. Implantarea ionica a facut ca procesul NMOS sa fie mult mai fiabil.

4.2.6. Corodarea.

Tehnicile examineate mai sus, cum ar fi DCV, pentru siliciul policristalin, acopera intrega placheta, in timp ce, pentru realizarea unui dispozitiv este nevoie de o forma, de un sablon. Una din metodele de transfer selectiv al formelor consta in inlaturarea portiunilor nemascate ale unui strat. Corodarea s-a realizat initial in baie chimica (acid) umeda. Cat timp dimensiunile, avute in vedere pentru forme, sunt de ordinul a $10 \mu\text{m}$ sau mai mult, aceasta opereaza satisfactor. In conditiile in care dimensiunile se reduc, tensiunea la suprafata materialului impiedica transferul efectiv al formei. De aceea corodarea uscata a devenit o metoda predominanta, care corespunde cerintelor VLSI. Corodarea uscata are loc din starea gazoasa, fiind ajutata de plasma. De exemplu, CF_4 si O_2 , combinate in plasma, pot genera un material extrem de eficient pentru corodarea Si , SiO_2 si a Si_3N_4 . Avantajele se refera la corodarea anisotropica, care asigura formarea unor pereti verticali, ceea ce reprezinta un element critic pentru DRAM-uri de mare densitate. Dificultatile asociate cu corodarea uscata se refera la selectivitate si la mentinerea uniformitatii pe toata suprafata plachetei. Corodarea uscata reprezinta un proces complicat, care depinde de o paleta larga de parametri.

4.2.7. Metalizarea.

Metalizarea reprezinta unul dintre cele mai bine cunoscute procese de fabricatie. Aluminul sau aliajele de aluminiu sunt, de regula, evaporate in vid sau imprastiate de pe o tinta de Al, in prezenta unei plasme. Pentru a crea traseele necesare de metal, trebuie sa aibe loc un transfer de forme. In functie de metal si strat, formarea este realizata printr-un proces litografic, folsind corodarea umeda sau uscata. In prezent se fac eforturi, care sa conduca la gasirea unor aliaje capabile sa asigure densitati mai mari de curent.

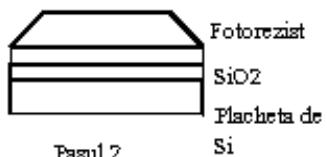
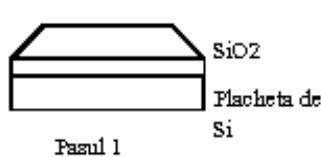
4.3. Litografia.

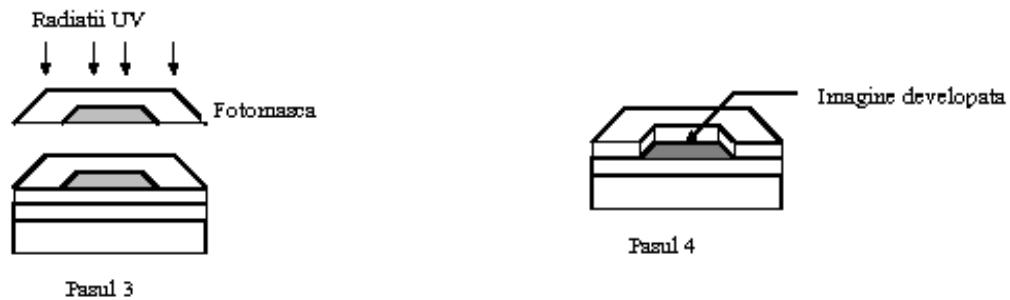
Litografia reprezinta etapa de baza in transferarea formelor geometrice de pe masti pe materialul de pe suprafata placetei. Formele materialului definesc ferestrele/taieturile de contact, interconexiunile intre diferitele straturi de metal, zonele de dielectric ale portilor etc. Intreaga reprezentare geometrica a circuitului este redusa la o structura pe niveluri. De exemplu, nivelurile de metalizare sau de difuzie poarta numele de niveluri de masti. Circuitul integrat este realizat prin transferul secvential al formelor de pe fiecare masca, nivel cu nivel, pe suprafata de Si. In functie de tehnologia utilizata, masca poate fi pentru intreaga placheta sau pentru un circuit/structura de pe placheta. Pentru tehnologiile, care necesita precizie si rezolutie mai mari, masca corespunde unui circuit/structura, iar placheta este baleiată sub forma de rastru, pentru transpunerea formelor, in procesul de multiplicare a circuitelor de pe placheta. Procesul litografic este esential pentru fabricatie si se caracterizeaza printr-un nivel coborat, in sensul ca, pe parcursul activitatii de proiectare a circuitului, detaliiile procesului nu sunt avute in vedere de catre proiectant, chiar daca este vorba de un circuit specializat. Astfel, procesul litografic poate fi gandit ca un macro sau, la nivelul limbajului de asmbalare, ca un microcod.

Trecerea in revista a etapelor de baza ale unui proces litografic.

- Pe suprafata placetei se aplica o pelicula dintr-un polimer fotosensibil, care se usuca si este apoi supus unei radiatii de ultraviolete printre-o masca fotografica, corespunzatoare formei dorite. Radiatiile ultraviolete se utilizeaza pentru a reduce fenomenul de difractie.
- In timpul expunerii polimerul reactioneaza cu radiatia ultravioleta, fie prin intarirea, fie prin slabirea lanturilor macromoleculare. Dupa expunere, suprafata placetei este developata, ceea ce face ca ea sa contina imaginea, la nivelul peliculei de polimer, pe materialul fotosensibil. In functie de polimer, in timpul developarii, se inlatura, fie suprafata expusa, fie cea neexpusa.
- Placheta este plasata intr-un mediu, care corodeaza zonele neprotejate de catre forma realizata din polimer. Din aceasta cauza, polimerul mai poarta numele de *rezist* sau *fotorezist*, in cazul in care se foloseste lumina in procesul litografic.

Detalii privitoare la procesul litografic sunt date mai jos:

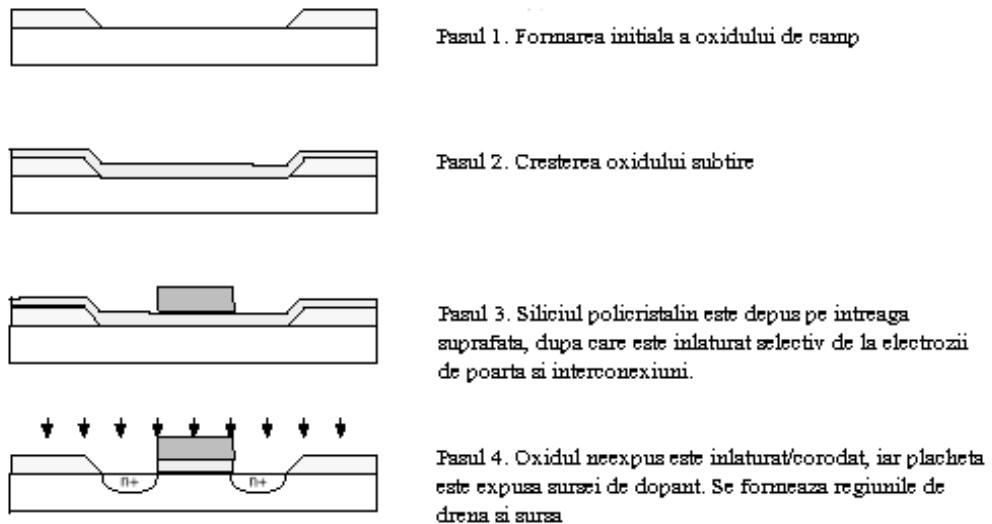


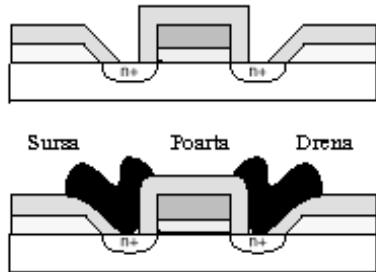


4.4. Procesul CMOS simplificat.

4.4.1 Un proces NMOS simplu.

Inainte de a examina un proces CMOS, este util sa se ilustreze pe scurt etapele fabricarii tranzistorului de tip NMOS, conform celor de mai jos.





Pasul 5. Intreaga structura este acoperita cu SiO_2 . Se corodeaza ferestrele de contact pentru a permite conexiunea cu straturile inferioare.

Pasul 6. Se depun traseele de metal si se corodeaza pentru a realiza interconexiunile cerute.

- Pasul 1. Se formeaza oxidul de camp SiO_2 initial. Deschiderile in oxid definesc regiunile active.
- Pasul 2. Se creste oxidul subtire, definit adesea ca “thinox”.
- Pasul 3. Se depune siliciu policristalin pe intreaga suprafata. Acesta este inlaturat selectiv din zonele electrozilor de poarta si ale interconexiunilor.
- Pasul 4. Oxidul subtire neexpus este indepartat/corodat, iar placeta este expusa sursei de dopare.
- Pasul 5. Intreaga structura este acoperita cu SiO_2 , iar ferestrele/taieturile de contact sunt corodate, pentru a permite efectuarea conexiunilor cu straturile inferioare.
- Pasul 6. Metalul, pentru interconexiuni, este evaporat si apoi corodat, pentru a realiza interconexiunile finale.

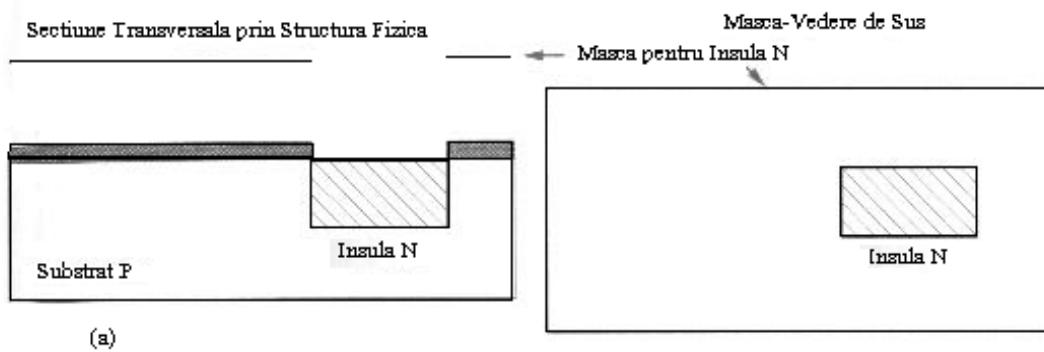
Este important sa se sublinieze faptul ca jonctiunile de difuzie sunt realizate numai in regiunile in care poarta de siliciu policristalin nu mascheaza substratul inferior. Acesta este un proces cu autoaliniere. In aceasta maniera, regiunile de drena si sursa nu se extind sub poarta, ceea ce reduce capacitatea parazita, care ar fi putut degrada performanta. Initial, dispozitivele MOS utilizau metal, in calitate de material pentru poarta. Aceste tehnologii, in mod inherent, erau mai lente, datorita capacitatiilor parazite. Trebuie, de asemenea, subliniat faptul ca, ori de cate ori, un traseu de siliciu policristalin intersecteaza un traseu de difuzie, se formeaza un tranzistor. Regiunile de drena si sursa sunt marcate cu N^+ sau $n+$, subliniind faptul ca ele sunt puternic dopate, pentru a imbunatati contactele ohmice cu metalul, pentru a reduce rezistentele parazite si pentru a transfera rapid sarcina in canal, la aplicarea tensiunii pe poarta. Un alt avantaj este acela al reducerii dependentei capacitati portii de tensiune.

4.4.2. Un proces CMOS cu insula N.

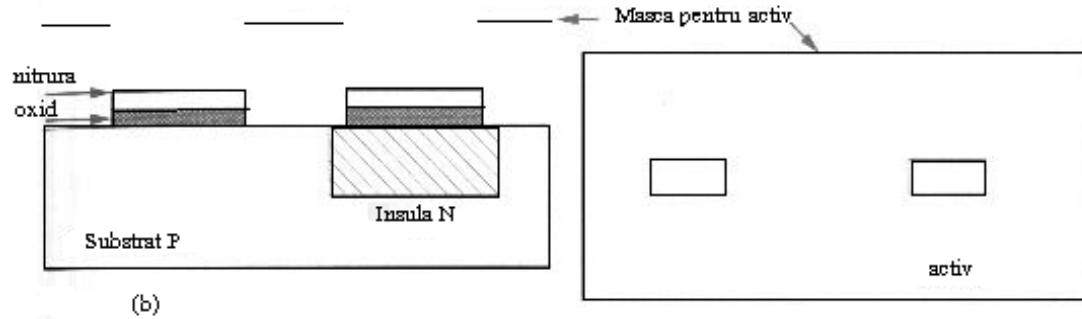
O abordare comună în fabricarea structurii CMOS, cu insula N, este aceea de a porni de la o placă dopată usor P și de a crea pe aceasta o insula N, pe care se formează un tranzistor de tip P. În zona nativă P, în substratul se realizează tranzistorul N.

În cele ce urmează vor fi prezentate etapele principale ale unui proces CMOS, cu insula de tip N. În realitate, etapele prelucrării placătăi sunt relativ complexe și depind de linia de fabricație. Fiecare etapă este ilustrată printr-o secțiune transversală în structura CMOS, cat și prin masă asociată etapei.

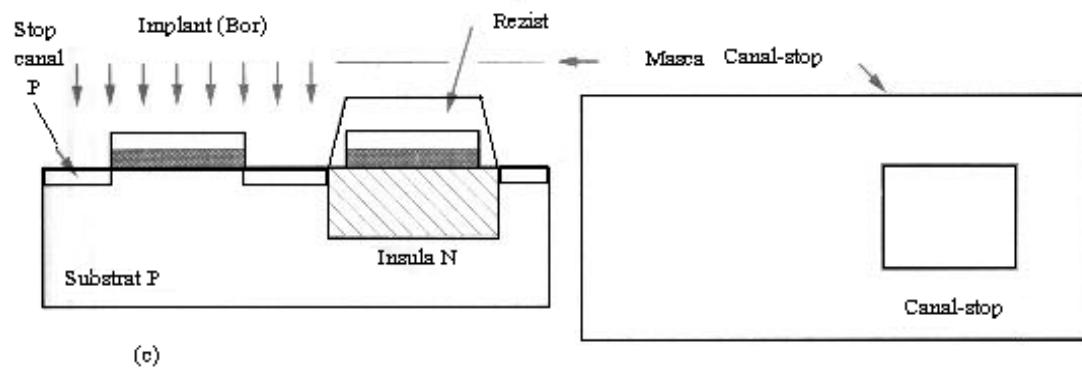
- În etapa (a) masă defineste insula N, în care se va realiza tranzistorul P. Insula P este realizată prin implantare ionica sau prin difuzie. Implantarea ionica are avantajul de a crea insule de mica grosime, ceea ce satisfă procesele care presupun dimensiuni mici, în timp ce difuzia, realizându-se în toate direcțiile, cu cat este mai profunda, cu atât se răspandeste lateral. Astfel, în cazul difuziei pot fi afectate structurile vecine, ceea ce impune zone mai mari de separare fata de acestea, având drept consecință reducerea densității componentelor pe structură.



- Masă următoare (b) poartă numele *activă*, întrucât ea definește zonele în care va fi prezent oxidul subtire, necesar realizării portilor tranzistorilor N și P, cat și a surselor și drenelor acestora, prin implantare ionica sau difuzie. Aceasta masă mai poartă numele de *masă pentru oxidul subtire* sau *mesa*. În cadrul acestei etape se crește un strat subtire de SiO_2 , care se acoperă cu SiN , pentru a forma un strat de mască, necesar următoarelor două etape.



- In aceasta etapa (c) se realizeaza implantarea pentru ceea ce se numeste canal-stop. Aceasta foloseste masca insula-P, care este complementara mastii insula-N, pentru a dopa P+ substratul in zonele in care nu se afla tranzistoare N, folosind o masca de fotorezist. Aceasta dopare, cat si stratul gros de oxid, ce va acoperi aceste zone, vor impiedeca realizarea unei conductii intre zonele drena/sursa ale unor tranzistoare, care nu au nici o legatura intre ele.

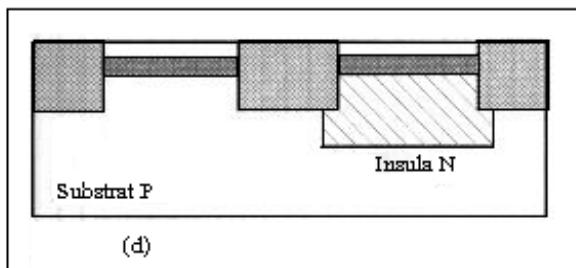


- Dupa implantarea canal-stop, masca din fotorezist este inlaturata, ceea ce permite definirea regiunilor active de catre structura SiO2/SiN mascata anterior. In continuare este format stratul gros de oxid, in zonele in care stratul de SiN este absent. Stratul de oxid se formeaza, atat in directie verticala, cat si pe orizontala/lateral, sub structura SiO2/SiN (d). Aceasta extindere laterală poartă numele de "cioc de pasare", datorită formei pe care o capătă, și are ca efect reducerea dimensiunilor zonelor active. Astfel, latimea canalului unui tranzistor va

fi mai mica decat cea presupusa la realizarea mastii. Aceasta tehnica, de realizare a stratului gros de oxid, poarta numele de LOCOS (Local Oxidation Of Silicon).

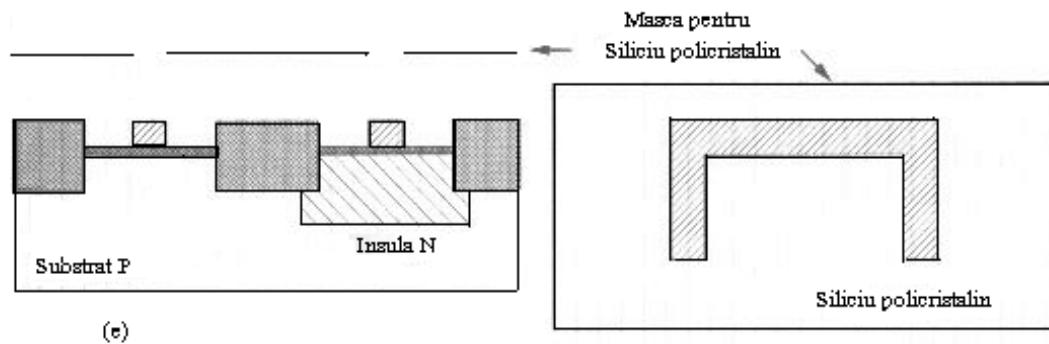
In scopul reducerii efectului mentionat mai sus, s-au propus diferite metode, printre care si cea numita SWAMI (Side Wall Masked Isolation)

Un alt aspect important este cel legat de planaritatea interfetei intre oxidul subtire, de poarta, si oxidul gros, de camp. In cazurile in care diferentele de cote intre cele doua straturi sunt mari, exista pericolul fisurarii traseelor de metal, care se depun in zonele de interfata intre cele doua straturi de oxid. Pentru a preintampina un asemenea fenomen, se recurge la o serie de tehnici de "planarizare". Una dintre tehnici consta in precorodarea placantei substrat, pe o adancime egala cu jumatea grosimii stratului gros de oxid, in zonele care vor fi acoperite de catre acesta. In continuare va fi format stratul de oxid LOCOS, in conditiile unei "planaritati" satisfacatoare.



- In aceasta etapa, se va efectua o ajustarea a tensiunii de prag a tranzistorului N, folosind masca de fotorezist pentru insula P. Procesele de fabricatie curente presupun o dopare N+ a siliciului policristalin. In conditiile proceselor corespunzatoare dispozitivelor de mici dimensiuni, concentratiile normale de dopare au ca rezultat tensiuni de prag de $0,5 \div 0,7$ V, pentru dispozitivele de tip N, si de $-1,5 \div 2,0$ V, pentru dispozitivele de tip P. Astfel, pentru dispozitivele de tip P ajustarea tensiunii de prag se va face intr-o masura mai mare decat cea a tensiunii de prag, pentru dispozitivele de tip N. Aceasta se realizeaza prin introducerea unui strat suplimentar, incarcat negativ, la interfata intre siliciu si oxid. In acest mod canalul de la interfata siliciu /oxid se deplaseaza in siliciu, in adancime, formand un dispozitiv cu "canal ingropat". Dupa aceasta este format oxidul de poarta.

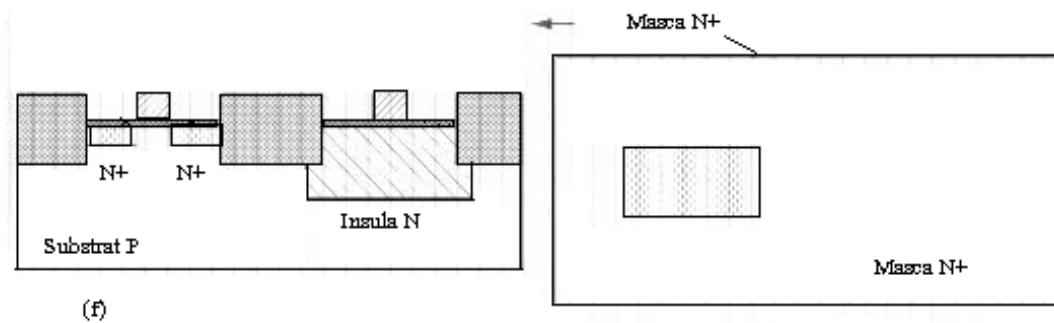
- Definirea portii implica acoperirea suprafetei cu siliciu policristalin, dupa care are loc indepartarea acestuia, prin corodare, pentru a obtine forma dorita, in cazul de fata un "U" intors (e). Dupa cum este cunoscut, in cazul portii din siliciu policristalin, are loc o autoaliniere a regiunilor drena-sursa.



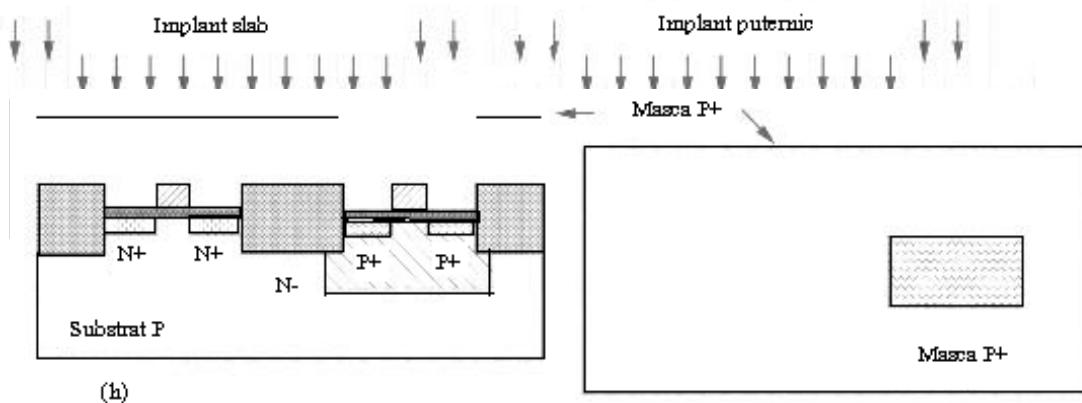
- In continuare se utilizeaza o masca N+, pentru a specifica zonele de difuzie si de siliciu policristalin, care vor fi implantate N+ (f). Daca zona N+ se afla pe substratul de tip P, atunci se formeaza un tranzistor cu canal N. In cazul cand zona N+ este plasata pe o insula de tip N, se va obtine un contact *ohmic* la insula. Aceasta are, evident, un caracter rezistiv. Aceasta masca mai poarta numele de *select*, deoarece selecteaza acele regiuni in care se formeaza tranzistoarele de tip N.

Dupa cum este cunoscut, in cazul proceselor, care presupun dimensiuni reduse pentru dispozitive, se face simtit efectul "electronilor fierbinti". Acestia, datorita energiei pe care o poseda, pot disloca goluri la drena, care sunt preluate de catre substratul incarcat negativ, ceea ce duce la aparitia unui curent de substrat. Electronii fierbinti au efecte negative asupra timpului de reimprospatare, la memoriile dinamice, asupra zgomotului si a fenomenului de "latc-up". In cazul cand acestia penetreaza oxidul portii, apare si un curent de poarta, care poate degrada tensiunea de prag, curentul de subprag si transconductanta. Pentru a evita asemenea fenomene se recurge la un proces de formare a zonelor surs/drena in doua etape. Mai intai se realizeaza, in aceste zone, o structura putin adanca, usor dopata N (LDD-Light Doped Drain structure), ca in figura (g), in zonele neacoperite de siliciul policristalin. In continuare se creste un oxid de "spatiere", peste traseul de siliciu policristalin al portii, dupa care are loc o implantare N+. Aceasta nu va afecta zona aflat la limita portii, sub oxid. Dupa

înlăturarea oxidului de "spatiere" rezulta o structură mult mai rezistentă la efectele electronilor fierbinti. Procesele curente de $0,25 \mu\text{m}$ nu vor utiliza tehnica LDD.

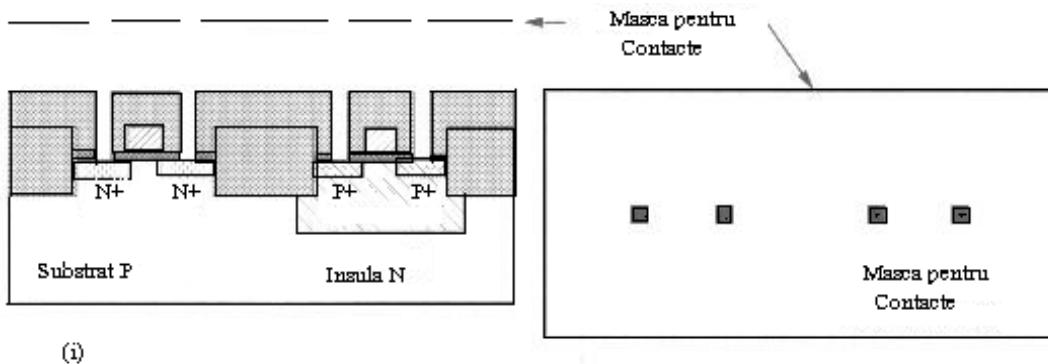


- Urmatoarea etapa utilizeaza complementul mastii N^+ , cat si o masca suplimentara. Absenta unei regiuni N^+ , in zonele acoperite de catre oxidul subtire, specifica faptul ca acestea vor fi zone de difuzie P^+ sau zone active Zona activa P , pe o insula N , defineste fie tranzistoare de tip P , fie fire (h). O difuzie de tip P^+ , pe un substrat P , permite realizarea unui contact *ohmic*. Dupa aceasta se depune un strat de SiO_2 . In cazul tranzistoarelor de tip P , nu se pune

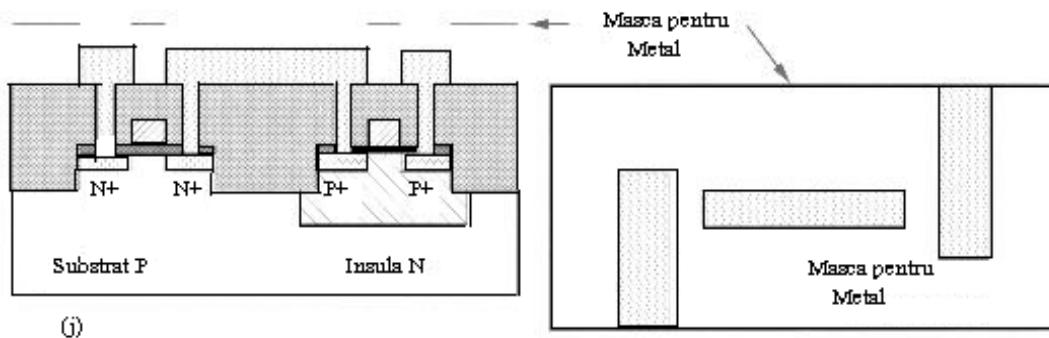


problema unei etape LDD, intrucat, in acest caz, purtatorii fierbinti nu au aceleasi efecte ca in cazul tranzistoarelor de tip N .

- In aceasta etapa se definesc tajeturile de contact, ceea ce presupune corodarea SiO₂ pana la stratul cu care trebuie sa se realizeze contactul (i). Aceasta permite, in cadrul urmatoarei etape, sa se efectueze contacte cu regiunile de difuzie sau siliciu policristalin.



- Pentru realizarea contactelor se face o metalizare a suprafetei, urmata de o corodare selectiva.



- Ultima etapa, neilustrata printr-un desen, se refera la *pasivizarea* intregii suprafete a siliciului, cu practicarea unor tajeturi, prin corodare, in zonele ploturilor de contact. Pasivizarea consta in acoperirea structurii, cu un strat de sticla, in scopul protejarii acesteia fata de eventualele contaminari, care ar modifica in mod nedorit comportarea circuitului.

Capitolul 4. Fabricarea circuitelor VLSI - CMOS.

4.1. Introducere.

Procesul fundamental de fabricatie a unui circuit integrat consta in formarea selectiva a unor paturi/straturi de tip semiconductoare, dielectric si metal, pe suprafata plachetei de siliciu monocristalin. Dupa realizare, aceste materiale devin componente active si pasive ale circuitului integrat. In acest capitol vor fi examineate procesele fundamentale legate de materiale, litografia si procedeele pentru crearea unui circuit integrat. Se va avea in vedere tehnologia CMOS cu doua insule. Tratarea nu are un caracter exhaustiv, ci mai mult generic, ilustrand aspectele de detaliu comune celor mai multe procese CMOS.

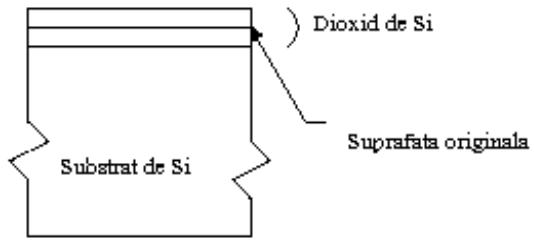
4.2. Tehnologia de procesare a materialului de baza.

4.2. Fabricarea plachetelor.

In oricare proces VLSI placheta reprezinta materialul de la care se porneste. In tehnologia Si – CMOS, placheta se realizeaza dintr-un lingou de siliciu monocristalin. Lingoul se obtine printr-un procedeu de tragere dintr-un creuzet, in care se afla siliciu pur topit la o temperatura de circa 1475 °C. Cea mai frecventa metoda de obtinere a lingoului din siliciu monocristalin se datoreaza lui Czochralski. Lingoul reprezinta un monocristal de Si, aproape fara defecte, cu o lungime de mai multe zeci de centimetri si cu un diametru de circa 10 cm. Pentru a impiedeca aparitia altor impuritati, monocristalul este dopat N. Monocristalul este taiat sub forma unor discuri/plachete cu o grosime, din considerente mecanice, de circa 300 μm , intrucat structurile electrice nu depasec in grosime 10 μm . Placheta, de regula de tip N, este acoperita cu un strat epitaxial de Si aproape intrinsec, inainte de a incepe procesul de fabricare a circuitului integrat. Aceasta prelucrare initiala conduce la cresterea rezistentei *latch-up*. Partea posterioara a plachetei poate beneficia de un proces de implantare ionica pentru reducerea rezistentei electrice de contact, la impachetarea finala.

4.2.2. Oxidarea.

Oxidarea se refera, de regula, la cresterea sau depozitarea SiO_2 pe suprafata plachetei. La o temperatura inalta de 1000 °C si intr-o atmosfera de O_2 , o placheta expusa se va oxida. Aceasta structura de SiO_2 foloseste Si de pe placheta, fiind plasata, atat in materialul plachetei, cat si pe suprafata acesteia, dupa cum se va vedea in desenul de mai jos.



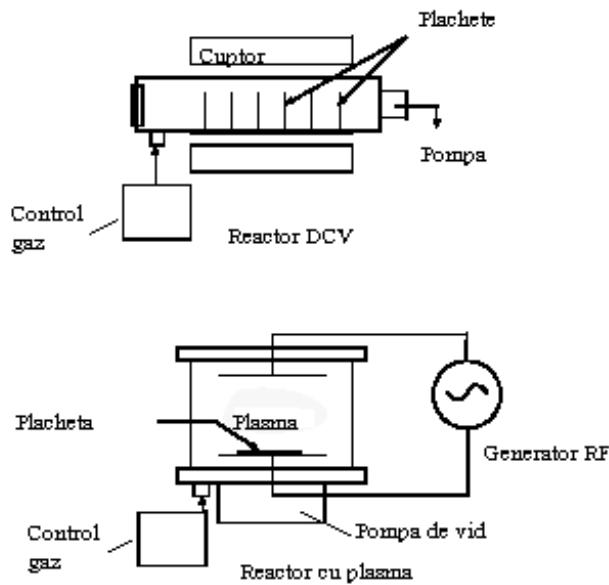
Acest proces creste un strat de SiO_2 de calitate, care poate fi folosit, fie ca dielectric izolator, fie ca oxid pentru poarta. In prezent, oxidul de poarta cu cele mai bune proprietati electrice este format prin oxidare termica. Aceasta impune ca procesele care vor urma sa aibe loc pe Si expus.

4.2.3. Depunerea de strat subtire.

O serie de materiale utilizate in fabricarea circuitelor integrate isi au originea in stare gazoasa. Aceste materiale sau straturi sunt constituite din siliciul policristalin, materialul principal pentru poarta tranzistorului MOS, cat si din izolatori dielectrici (SiO_2 si Si_3N_4), folositi intre diferitele straturi conductoare.

In practica se intalnesc numeroase tehnici de depunere, care au multe elemente comune. Cea mai simpla, Depunerea Chimica de Vapori (DCV) la presiune atmosferica, este folosita pentru formarea siliciului policristalin. Placheta este incalzita in cuptor in prezenta silanului (SiH_4), la o temperatura de circa $650\text{ }^{\circ}\text{C}$, care este suficienta pentru a descompune moleculea de SiH_4 si a depune un atom de Si pe suprafata placetei. Pe suprafata expusa a placetei se vor forma un strat de $1\text{--}10\text{ }\mu\text{m}$ din cristale de siliciu cu diverse orientari.

O alta tehnica, avand o importanta crescanda, este cea bazata pe depunere de materiale aflate in stare de plasma. Plasma reprezinta un gaz neutru cu un numar egal de electroni si atomi ionizati sau molecule ionizate. Un material poate fi adus in stare de plasma prin diverse procedee de excitare bazate pe campuri de radio-frecventa sau microunde. Plasma este generata in incinte vidate, la presiune joasa, ceea ce permite gazului sa se descompuna mai usor. In timpul depunerii gazele necesare sunt introduse in incinta vidata la debit si presiune constanta. Pentru a accelera electronii, care provoaca coliziunile, excitarea si ionizarea se folosesc campuri de radio-frecventa. Generarea de radicali activi usureaza cerintele impuse pentru mentinerea placetei la temperatura inalta, in timpul procesarii. In figura de mai jos se prezinta o instalatie cu DCV si un reactor cu plasma.



4.2.4. Difuzia.

Difuzia reprezinta un proces termic de dopare N sau P a Si pentru a-i schimba caracteristicile electrice. Desi nu are loc o depunere sau o crestere de material, stratul care a suferit un proces de difuzie are un caracter critic in formarea dispozitivului CMOS. De exemplu, regiunile sursa si drena, ale unui tranzistor NMOS, sunt realizate in timpul unui proces de difuzie. In particular, atunci cand se proiecteaza o poarta de baza, sunt folosite in mod intensiv straturile de difuzie. Difuzia are loc, de cele mai multe ori, plecand de la o sursa chimica in stare de vaporii, la temperatura inalta. Mai recent, difuzia a fost cuplata cu implantarea ionica, pentru a obtine regiuni mai bine izolate ale dispozitivelor.

4.2.5. Implantarea ionica.

Implantarea ionica consta in introducerea fortata a unor atomi ionizati sau molecule ionizate intr-un material tinta, in conditiile in care li s-a imprimat o energie suficienta, pentru a penetra suprafata materialului. Sursa de ioni o constituie plasma, extractia acestora realizandu-se cu ajutorul unor campuri puternice de CC sau RF. Intrucat permite controlul multor parametri in timpul procesului, implantarea ionica este utilizata in mod extensiv in fabricarea VLSI. De exemplu, adancimea de patrundere a atomilor este controlata direct prin potentialul de accelerare, in timp ce doza este controlata prin produsul curent \times timp de expunere. Dozele

coborate sunt folosite pentru controlul tensiunii de prag, atat la tranzistoarelor active, cat si la cele parazite. Dozele mari sunt utilizate in formarea regiunilor de sursa si drena, prin autoaliniere. Se poate afirma ca implantarea ionica este direct responsabila pentru succesul CMOS fata de alte tehnologii. De exemplu, unul dintre motivele utilizarii dispozitivelor PMOS, la inceputul tehnologiei MOS, a fost controlul inadecvat al tensiunii de prag la fabricarea dispozitivelor NMOS. Implantarea ionica a facut ca procesul NMOS sa fie mult mai fiabil.

4.2.6. Corodarea.

Tehnicile examineate mai sus, cum ar fi DCV, pentru siliciul policristalin, acopera intrega placheta, in timp ce, pentru realizarea unui dispozitiv este nevoie de o forma, de un sablon. Una din metodele de transfer selectiv al formelor consta in inlaturarea portiunilor nemascate ale unui strat. Corodarea s-a realizat initial in baie chimica (acid) umeda. Cat timp dimensiunile, avute in vedere pentru forme, sunt de ordinul a $10 \mu\text{m}$ sau mai mult, aceasta opereaza satisfactor. In conditiile in care dimensiunile se reduc, tensiunea la suprafata materialului impiedica transferul efectiv al formei. De aceea corodarea uscata a devenit o metoda predominanta, care corespunde cerintelor VLSI. Corodarea uscata are loc din starea gazoasa, fiind ajutata de plasma. De exemplu, CF_4 si O_2 , combinate in plasma, pot genera un material extrem de eficient pentru corodarea Si , SiO_2 si a Si_3N_4 . Avantajele se refera la corodarea anisotropica, care asigura formarea unor pereti verticali, ceea ce reprezinta un element critic pentru DRAM-uri de mare densitate. Dificultatile asociate cu corodarea uscata se refera la selectivitate si la mentinerea uniformitatii pe toata suprafata plachetei. Corodarea uscata reprezinta un proces complicat, care depinde de o paleta larga de parametri.

4.2.7. Metalizarea.

Metalizarea reprezinta unul dintre cele mai bine cunoscute procese de fabricatie. Aluminul sau aliajele de aluminiu sunt, de regula, evaporate in vid sau imprastiate de pe o tinta de Al, in prezenta unei plasme. Pentru a crea traseele necesare de metal, trebuie sa aibe loc un transfer de forme. In functie de metal si strat, formarea este realizata printr-un proces litografic, folsind corodarea umeda sau uscata. In prezent se fac eforturi, care sa conduca la gasirea unor aliaje capabile sa asigure densitati mai mari de curent.

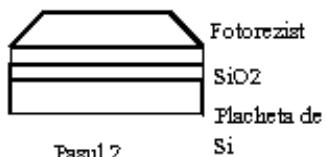
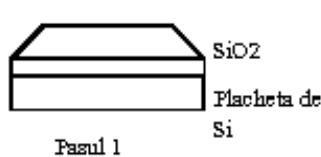
4.3. Litografia.

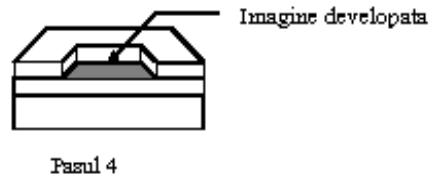
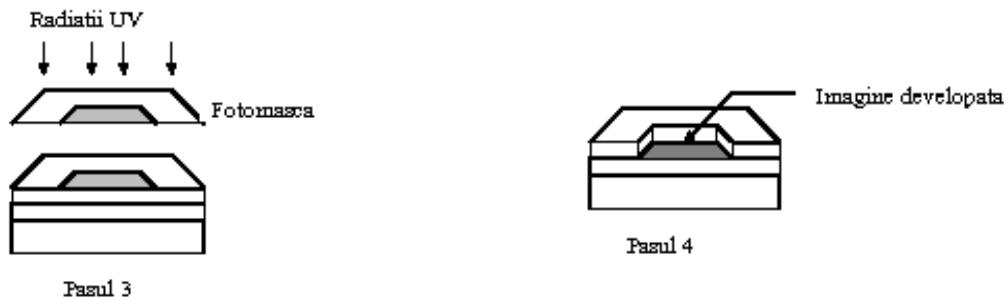
Litografia reprezinta etapa de baza in transferarea formelor geometrice de pe masti pe materialul de pe suprafata placetei. Formele materialului definesc ferestrele/taieturile de contact, interconexiunile intre diferitele straturi de metal, zonele de dielectric ale portilor etc. Intreaga reprezentare geometrica a circuitului este redusa la o structura pe niveluri. De exemplu, nivelurile de metalizare sau de difuzie poarta numele de niveluri de masti. Circuitul integrat este realizat prin transferul secvential al formelor de pe fiecare masca, nivel cu nivel, pe suprafata de Si. In functie de tehnologia utilizata, masca poate fi pentru intreaga placheta sau pentru un circuit/structura de pe placheta. Pentru tehnologiile, care necesita precizie si rezolutie mai mari, masca corespunde unui circuit/structura, iar placheta este baleiată sub forma de rastru, pentru transpunerea formelor, in procesul de multiplicare a circuitelor de pe placheta. Procesul litografic este esential pentru fabricatie si se caracterizeaza printr-un nivel coborat, in sensul ca, pe parcursul activitatii de proiectare a circuitului, detaliiile procesului nu sunt avute in vedere de catre proiectant, chiar daca este vorba de un circuit specializat. Astfel, procesul litografic poate fi gandit ca un macro sau, la nivelul limbajului de asmbalare, ca un microcod.

Trecerea in revista a etapelor de baza ale unui proces litografic.

- Pe suprafata placetei se aplica o pelicula dintr-un polimer fotosensibil, care se usuca si este apoi supus unei radiatii de ultraviolete printre-o masca fotografica, corespunzatoare formei dorite. Radiatiile ultraviolete se utilizeaza pentru a reduce fenomenul de difractie.
- In timpul expunerii polimerul reactioneaza cu radiatia ultravioleta, fie prin intarirea, fie prin slabirea lanturilor macromoleculare. Dupa expunere, suprafata placetei este developata, ceea ce face ca ea sa contina imaginea, la nivelul peliculei de polimer, pe materialul fotosensibil. In functie de polimer, in timpul developarii, se inlatura, fie suprafata expusa, fie cea neexpusa.
- Placheta este plasata intr-un mediu, care corodeaza zonele neprotejate de catre forma realizata din polimer. Din aceasta cauza, polimerul mai poarta numele de *rezist* sau *fotorezist*, in cazul in care se foloseste lumina in procesul litografic.

Detalii privitoare la procesul litografic sunt date mai jos:





4.4. Procesul CMOS simplificat.

4.4.1 Un proces NMOS simplu.

Inainte de a examina un proces CMOS, este util sa se ilustreze pe scurt etapele fabricarii tranzistorului de tip NMOS, conform celor de mai jos.



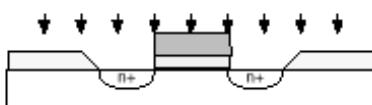
Pasul 1. Formarea initiala a oxidului de camp



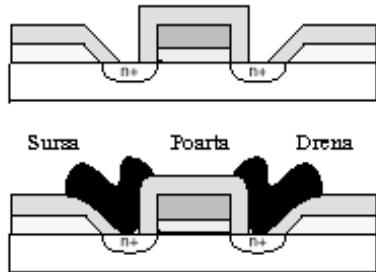
Pasul 2. Cresterea oxidului subtire



Pasul 3. Siliciul policristalin este depus pe intreaga suprafață, după care este intărit selectiv de la electrozi de poartă și interconexiuni.



Pasul 4. Oxidul neexpus este intărit/corodat, iar placă este expusă sursei de dopant. Se formează regiunile de drain și sursă



Pasul 5. Intreaga structura este acoperita cu SiO_2 . Se corodeaza ferestrele de contact pentru a permite conexiunea cu straturile inferioare.

Pasul 6. Se depun traseele de metal si se corodeaza pentru a realiza interconexiunile cerute.

- Pasul 1. Se formeaza oxidul de camp SiO_2 initial. Deschiderile in oxid definesc regiunile active.
- Pasul 2. Se creste oxidul subtire, definit adesea ca “thinox”.
- Pasul 3. Se depune siliciu policristalin pe intreaga suprafata. Acesta este inlaturat selectiv din zonele electrozilor de poarta si ale interconexiunilor.
- Pasul 4. Oxidul subtire neexpus este indepartat/corodat, iar placeta este expusa sursei de dopare.
- Pasul 5. Intreaga structura este acoperita cu SiO_2 , iar ferestrele/taieturile de contact sunt corodate, pentru a permite efectuarea conexiunilor cu straturile inferioare.
- Pasul 6. Metalul, pentru interconexiuni, este evaporat si apoi corodat, pentru a realiza interconexiunile finale.

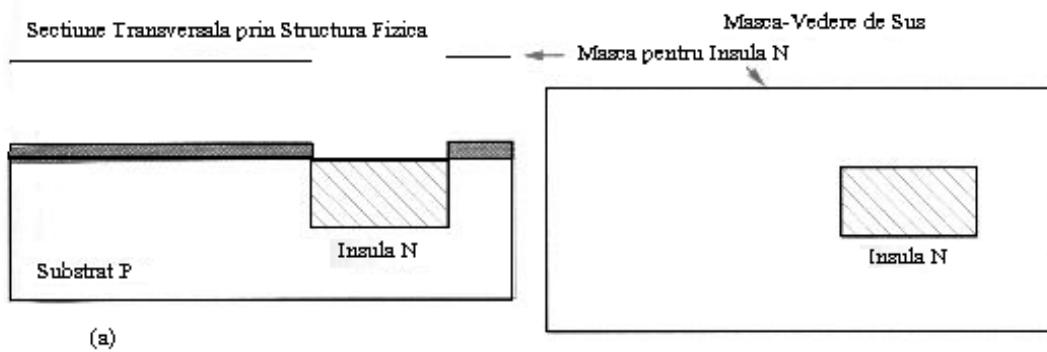
Este important sa se sublinieze faptul ca jonctiunile de difuzie sunt realizate numai in regiunile in care poarta de siliciu policristalin nu mascheaza substratul inferior. Acesta este un proces cu autoaliniere. In aceasta maniera, regiunile de drena si sursa nu se extind sub poarta, ceea ce reduce capacitatea parazita, care ar fi putut degrada performanta. Initial, dispozitivele MOS utilizau metal, in calitate de material pentru poarta. Aceste tehnologii, in mod inherent, erau mai lente, datorita capacitatiilor parazite. Trebuie, de asemenea, subliniat faptul ca, ori de cate ori, un traseu de siliciu policristalin intersecteaza un traseu de difuzie, se formeaza un tranzistor. Regiunile de drena si sursa sunt marcate cu N^+ sau $n+$, subliniind faptul ca ele sunt puternic dopate, pentru a imbunatati contactele ohmice cu metalul, pentru a reduce rezistentele parazite si pentru a transfera rapid sarcina in canal, la aplicarea tensiunii pe poarta. Un alt avantaj este acela al reducerii dependentei capacitatiilor portii de tensiune.

4.4.2. Un proces CMOS cu insula N.

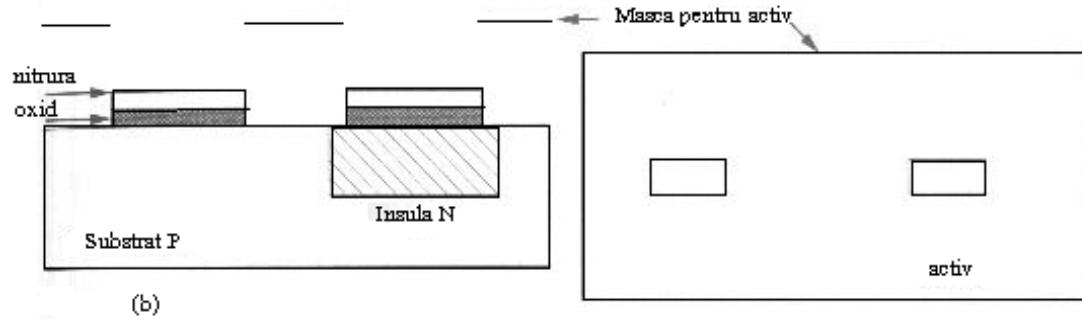
O abordare comună în fabricarea structurii CMOS, cu insula N, este aceea de a porni de la o placă dopată usor P și de a crea pe aceasta o insula N, pe care se formează un tranzistor de tip P. În zona nativă P, în substratul se realizează tranzistorul N.

În cele ce urmează vor fi prezentate etapele principale ale unui proces CMOS, cu insula de tip N. În realitate, etapele prelucrării placătăi sunt relativ complexe și depind de linia de fabricație. Fiecare etapă este ilustrată printr-o secțiune transversală în structura CMOS, cat și prin masă asociată etapei.

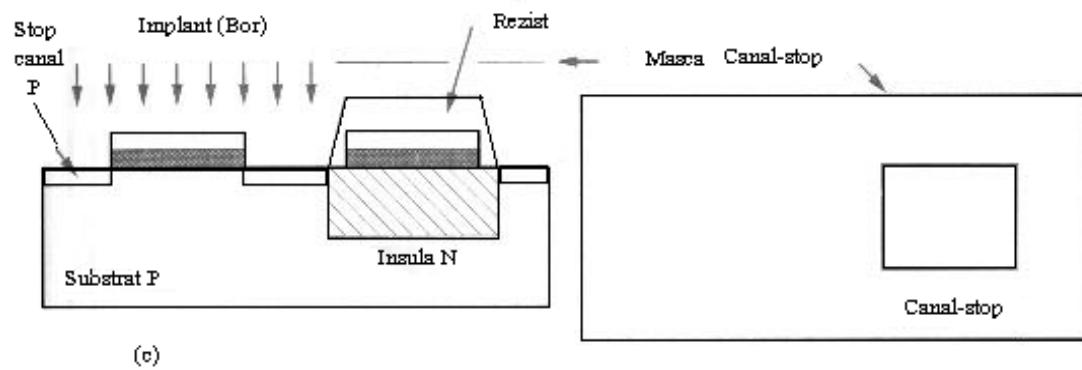
- În etapa (a) masă defineste insula N, în care se va realiza tranzistorul P. Insula P este realizată prin implantare ionica sau prin difuzie. Implantarea ionica are avantajul de a crea insule de mica grosime, ceea ce satisfă procesele care presupun dimensiuni mici, în timp ce difuzia, realizându-se în toate direcțiile, cu cat este mai profunda, cu atât se răspandeste lateral. Astfel, în cazul difuziei pot fi afectate structurile vecine, ceea ce impune zone mai mari de separare fata de acestea, având drept consecință reducerea densității componentelor pe structură.



- Masca următoare (b) poartă numele *activă*, întrucât ea definește zonele în care va fi prezent oxidul subțire, necesar realizării portilor tranzistorilor N și P, cat și a surselor și drenelor acestora, prin implantare ionica sau difuzie. Aceasta masă mai poartă numele de *masă pentru oxidul subțire* sau *mesa*. În cadrul acestei etape se crește un strat subțire de SiO_2 , care se acoperă cu SiN , pentru a forma un strat de mască, necesar următoarelor două etape.



- In aceasta etapa (c) se realizeaza implantarea pentru ceea ce se numeste canal-stop. Aceasta foloseste masca insula-P, care este complementara mastii insula-N, pentru a dopa P+ substratul in zonele in care nu se afla tranzistoare N, folosind o masca de fotorezist. Aceasta dopare, cat si stratul gros de oxid, ce va acoperi aceste zone, vor impiedeca realizarea unei conductii intre zonele drena/sursa ale unor tranzistoare, care nu au nici o legatura intre ele.

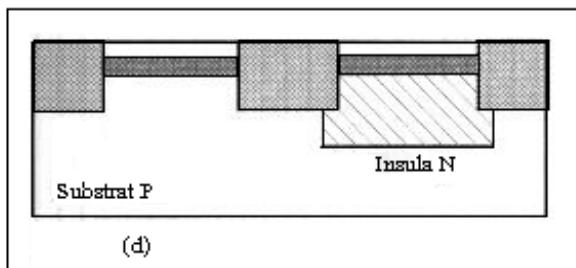


- Dupa implantarea canal-stop, masca din fotorezist este inlaturata, ceea ce permite definirea regiunilor active de catre structura SiO2/SiN mascata anterior. In continuare este format stratul gros de oxid, in zonele in care stratul de SiN este absent. Stratul de oxid se formeaza, atat in directie verticala, cat si pe orizontala/lateral, sub structura SiO2/SiN (d). Aceasta extindere laterală poarta numele de “cioc de pasare”, datorita formei pe care o capata, si are ca efect reducerea dimensiunilor zonelor active. Astfel, latimea canalului unui tranzistor va

fi mai mica decat cea presupusa la realizarea mastii. Aceasta tehnica, de realizare a stratului gros de oxid, poarta numele de LOCOS (Local Oxidation Of Silicon).

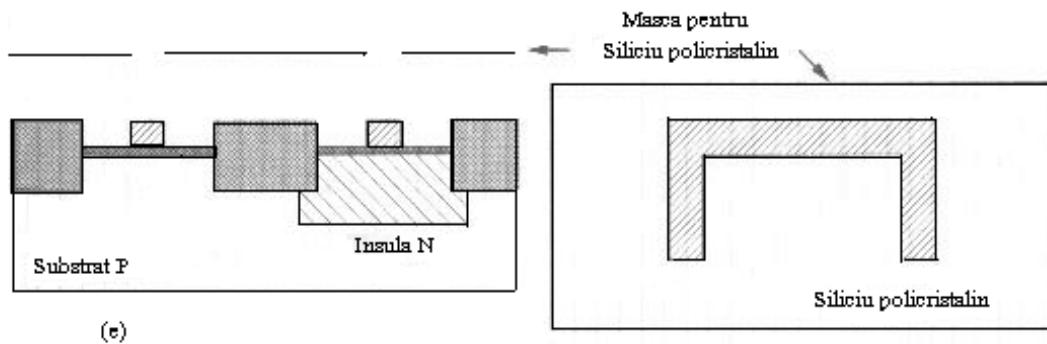
In scopul reducerii efectului mentionat mai sus, s-au propus diferite metode, printre care si cea numita SWAMI (Side Wall Masked Isolation)

Un alt aspect important este cel legat de planaritatea interfetei intre oxidul subtire, de poarta, si oxidul gros, de camp. In cazurile in care diferentele de cote intre cele doua straturi sunt mari, exista pericolul fisurarii traseelor de metal, care se depun in zonele de interfata intre cele doua straturi de oxid. Pentru a preintampina un asemenea fenomen, se recurge la o serie de tehnici de "planarizare". Una dintre tehnici consta in precorodarea placantei substrat, pe o adancime egala cu jumatea grosimii stratului gros de oxid, in zonele care vor fi acoperite de catre acesta. In continuare va fi format stratul de oxid LOCOS, in conditiile unei "planaritati" satisfacatoare.



- In aceasta etapa, se va efectua o ajustarea a tensiunii de prag a tranzistorului N, folosind masca de fotorezist pentru insula P. Procesele de fabricatie curente presupun o dopare N+ a siliciului policristalin. In conditiile proceselor corespunzatoare dispozitivelor de mici dimensiuni, concentratiile normale de dopare au ca rezultat tensiuni de prag de $0,5 \div 0,7$ V, pentru dispozitivele de tip N, si de $-1,5 \div 2,0$ V, pentru dispozitivele de tip P. Astfel, pentru dispozitivele de tip P ajustarea tensiunii de prag se va face intr-o masura mai mare decat cea a tensiunii de prag, pentru dispozitivele de tip N. Aceasta se realizeaza prin introducerea unui strat suplimentar, incarcat negativ, la interfata intre siliciu si oxid. In acest mod canalul de la interfata siliciu /oxid se deplaseaza in siliciu, in adancime, formand un dispozitiv cu "canal ingropat". Dupa aceasta este format oxidul de poarta.

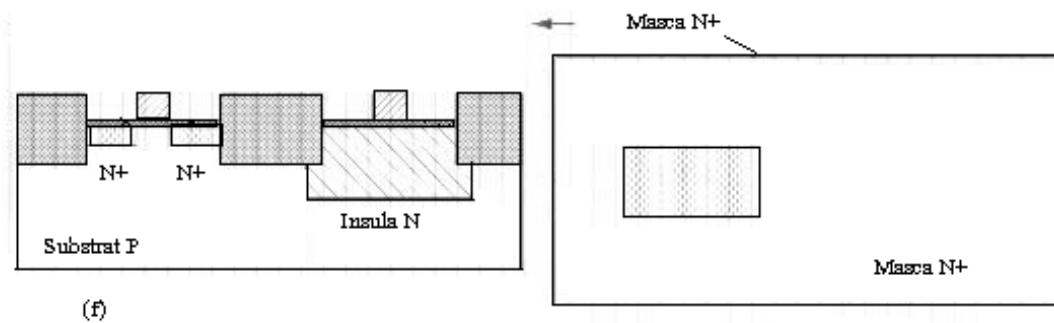
- Definirea portii implica acoperirea suprafetei cu siliciu policristalin, dupa care are loc indepartarea acestuia, prin corodare, pentru a obtine forma dorita, in cazul de fata un "U" intors (e). Dupa cum este cunoscut, in cazul portii din siliciu policristalin, are loc o autoaliniere a regiunilor drena-sursa.



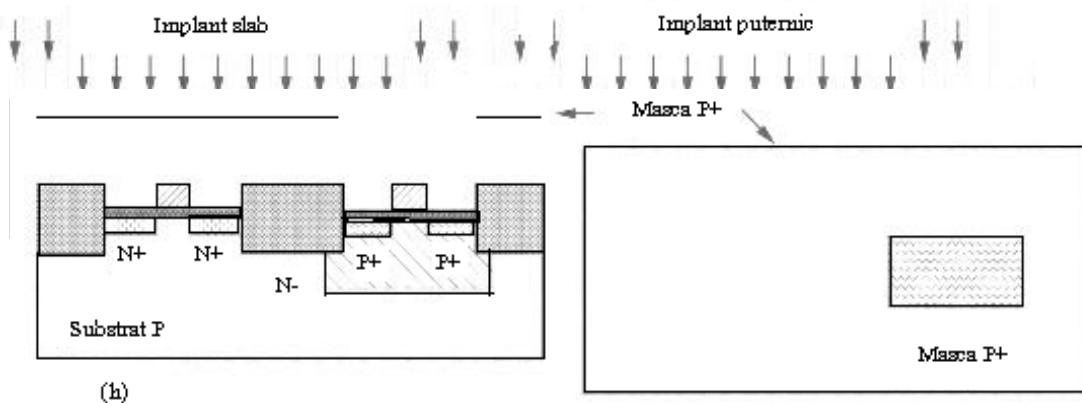
- In continuare se utilizeaza o masca N+, pentru a specifica zonele de difuzie si de siliciu policristalin, care vor fi implantate N+ (f). Daca zona N+ se afla pe substratul de tip P, atunci se formeaza un tranzistor cu canal N. In cazul cand zona N+ este plasata pe o insula de tip N, se va obtine un contact *ohmic* la insula. Aceasta are, evident, un caracter rezistiv. Aceasta masca mai poarta numele de *select*, deoarece selecteaza acele regiuni in care se formeaza tranzistoarele de tip N.

Dupa cum este cunoscut, in cazul proceselor, care presupun dimensiuni reduse pentru dispozitive, se face simtit efectul "electronilor fierbinti". Acestia, datorita energiei pe care o poseda, pot disloca goluri la drena, care sunt preluate de catre substratul incarcat negativ, ceea ce duce la aparitia unui curent de substrat. Electronii fierbinti au efecte negative asupra timpului de reimprospatare, la memoriile dinamice, asupra zgomotului si a fenomenului de "latc-up". In cazul cand acestia penetreaza oxidul portii, apare si un curent de poarta, care poate degrada tensiunea de prag, curentul de subprag si transconductanta. Pentru a evita asemenea fenomene se recurge la un proces de formare a zonelor surs/drena in doua etape. Mai intai se realizeaza, in aceste zone, o structura putin adanca, usor dopata N (LDD-Light Doped Drain structure), ca in figura (g), in zonele neacoperite de siliciul policristalin. In continuare se creste un oxid de "spatiere", peste traseul de siliciu policristalin al portii, dupa care are loc o implantare N+. Aceasta nu va afecta zona aflat la limita portii, sub oxid. Dupa

înlăturarea oxidului de "spatiere" rezulta o structură mult mai rezistentă la efectele electronilor fierbinti. Procesele curente de $0,25 \mu\text{m}$ nu vor utiliza tehnica LDD.

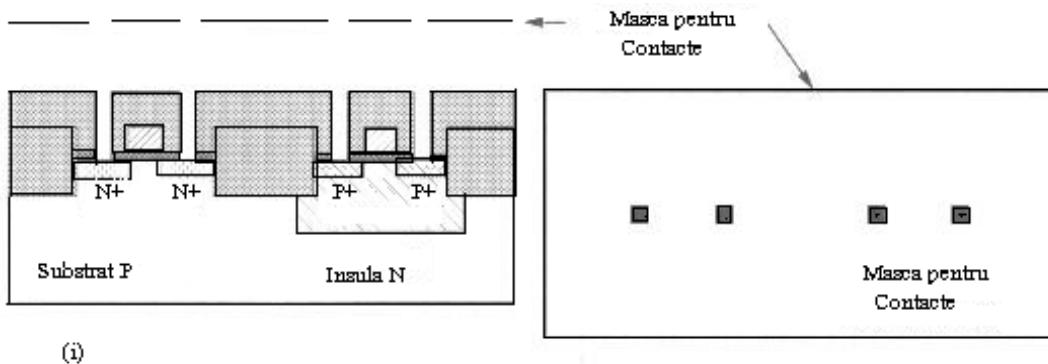


- Urmatoarea etapa utilizeaza complementul mastii N^+ , cat si o masca suplimentara. Absenta unei regiuni N^+ , in zonele acoperite de catre oxidul subtire, specifica faptul ca acestea vor fi zone de difuzie P^+ sau zone active Zona activa P , pe o insula N , defineste fie tranzistoare de tip P , fie fire (h). O difuzie de tip P^+ , pe un substrat P , permite realizarea unui contact *ohmic*. Dupa aceasta se depune un strat de SiO_2 . In cazul tranzistoarelor de tip P , nu se pune

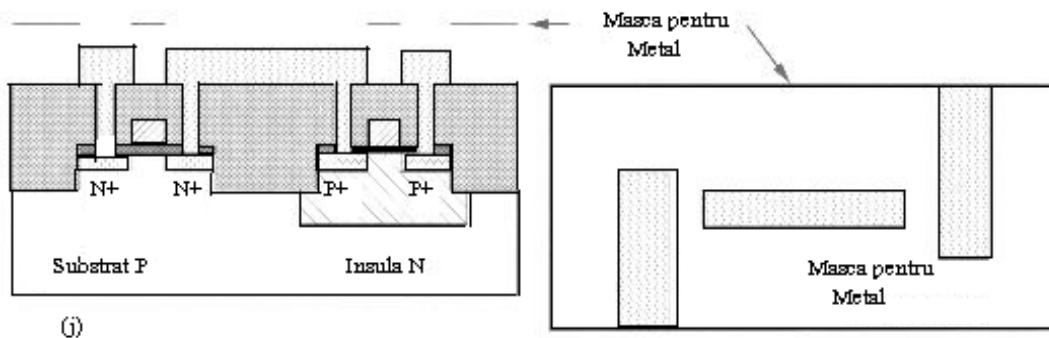


problema unei etape LDD, intrucat, in acest caz, purtatorii fierbinti nu au aceleasi efecte ca in cazul tranzistoarelor de tip N .

- In aceasta etapa se definesc tajeturile de contact, ceea ce presupune corodarea SiO₂ pana la stratul cu care trebuie sa se realizeze contactul (i). Aceasta permite, in cadrul urmatoarei etape, sa se efectueze contacte cu regiunile de difuzie sau siliciu policristalin.



- Pentru realizarea contactelor se face o metalizare a suprafetei, urmata de o corodare selectiva.



- Ultima etapa, neilustrata printr-un desen, se refera la *pasivizarea* intregii suprafete a siliciului, cu practicarea unor tajeturi, prin corodare, in zonele ploturilor de contact. Pasivizarea consta in acoperirea structurii, cu un strat de sticla, in scopul protejarii acestia fata de eventualele contaminari, care ar modifica in mod nedorit comportarea circuitului.

Capitolul 5. Performantele circuitelor VLSI.

5.1. Introducere.

Criteriul de performanta fundamental in oricare proiect VLSI este, de regula, viteza. Degradarea performantei apare ca rezultat al mai multor factori, care pot fi avuti in vedere in proiectare. Un exemplu comun este acela in care iesirea unui circuit CMOS, conectata la intrarea unui circuit similar, presupune o posibilitate de comanda (fan-out) nelimitata. Aceasta presupunere se poate verifica usor pe un numar redus de porti. Pana acum, nu a fost specificata nici o restrictie de viteza.

In capitolul de fata se vor face estimari pentru a putea evalua performantele circuitelor simple CMOS.

Plecand de la posibilitatile limitate de comanda in proiectarea circuitului se va putea constata ca acesta este in continuare lent. Prin inspectie se pot evidenta mai multe linii lungi de interconectare, care contribuie la degradarea performantei prin capacitatile parazite pe care le creeaza. Modelele de comportament al acestor trasee lungi conduc fie la constante (capacitati) concentrate, fie la constante (capacitati si rezistente) distribuite. Pentru a imbunatati proiectul se va mari posibilitatea de comanda a dispozitivului prin cresterea latimii canalului acestuia sau prin conectarea in cascada a unor dispozitive cu posibilitati mai mari de comanda. In aplicatii specifice pot fi luate in considerare solutii mixte BiCMOS sau solutii bazate pe dispozitive bipolare de mare viteza.

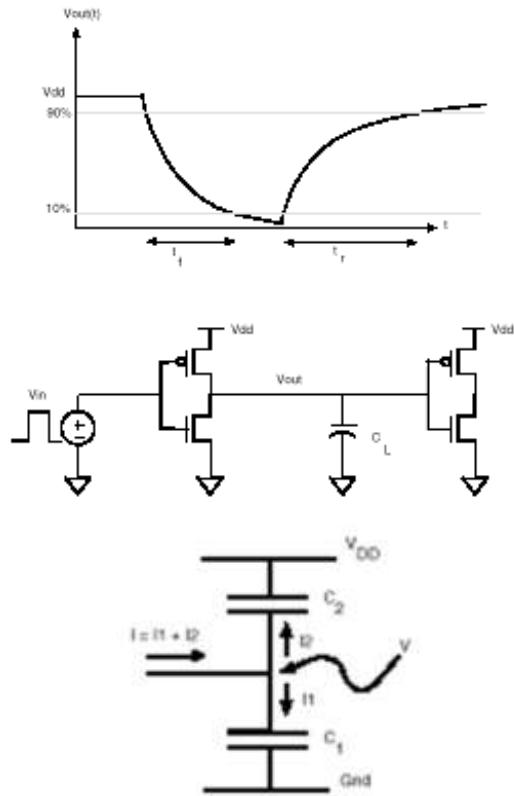
Trebuie sa se aibe in vedere faptul ca evaluările care vor fi efectuate au un caracter estimativ. Ele vor permite o mai buna cunoastere a functionarii dispozitivului. Estimarile nu urmaresc inlocuirea simularii detaliate, dar pot fi folosite pentru verificarea rapida a rezultatelor acesteia. Aspectele privitoare la performantele circuitelor VLSI sunt examinate pe larg in literatura de specialitate.

5.2. Caracteristicile de comutatie.

In vederea examinarii caracteristicilor de comutatie ale inversorului CMOS se va pleca de la schema de mai jos.

Se poate observa ca sarcina capacitiva este formata din capacitatea parazita a traseului in paralel cu capacitatea de intrare (capacitori MOS) a inversorului controlat. Avand in vedere potentialele GND si VDD la care sunt conectati electrozii inferioiri ai capacitatilor portilor

tranzistoarelor NMOS si PMOS, conectarea lor in paralel cu capacitatea parazita a traseului de legatura poate introduce unele confuzii.



Un calcul simplu bazat pe schema prezentata in continuare va face clarificarile necesare.

$$I = I1 + I2 = C_1 dV/dt + C_2 d(V - V_{DD})/dt = (C_1 + C_2) dV/dt.$$

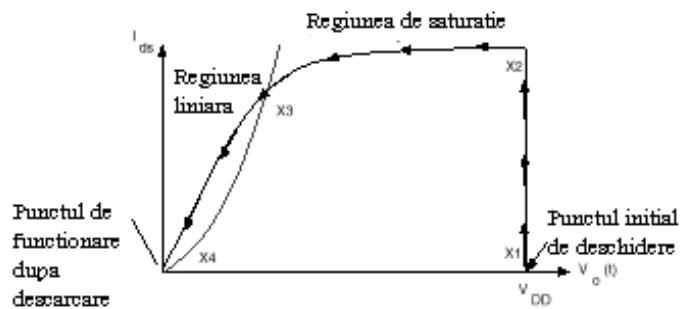
Intrucat V_{DD} este o constanta capacitorii C_1 si C_2 se aduna ca si cand ar fi conectati in paralel. Capacitorul este incarcat sau descarcat fie prin tranzistorul P, fie prin tranzistorul N ai inversorului. Intrucat aceste dispozitive prezinta o rezistenta atunci cand sunt in conductie ele vor contribui direct la intarzierea de tip RC a circuitului. Din cele arataate mai sus rezulta ca principala limitare o introduce viteza de comutare la incarcarea si descarcarea capacitorului de sarcina prin rezistenta dispozitivului.

Determinarea capacitatiei se face prin estimarea suprafetei unui strat dat inmultita cu capacitatea pe unitatea de suprafata. Pentru un proces dat, valoarea este tabelata si variaza de la un strat la altul. In cazul estimarii capacitatiei straturilor de difuzie trebuie sa se aibe in vedere si contributia peretilor laterali. Aceasta se realizeaza prin evaluarea perimetrului de difuzie si inmultirea

acestuia cu capacitatea pe unitatea de lungime. In cele mai multe cazuri R, din constanta de timp RC, rezulta direct din rezistenta canalelor tranzistoarelor in conductie prin care se efectueaza incarcarea si descarcarea sarcinii capacitive. Rezistenta parazita a liniilor lungi poate fi estimata prin evaluarea raportului Lungime/Latime si inmultirea acestuia cu rezistenta stratului exprimata in Ohm/patrat (Ω/\square), care este constanta pentru straturi patrate cu laturi de dimensiuni variabile.

5.2.1 Determinarea timpului de cadere (t_f).

Timpul de cadere, notat cu t_f , poate fi definit prin timpul necesar unei forme de unda de a ceda de la 90% la 10% din valoarea maxima. Traекторia curentului de drena al tranzistorului de tip N, ca functie de tensiunea de drena, este prezentata mai jos.

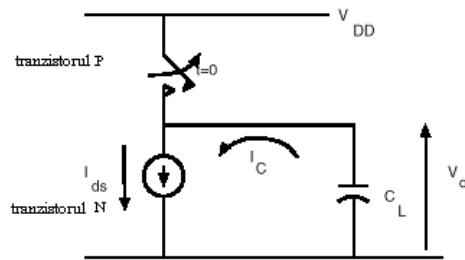


Aceasta traекторie ilustreaza parcursul prin caracteristicile I-V, corespunzatoare unui inversor care descarca o sarcina capacitiva C_L .

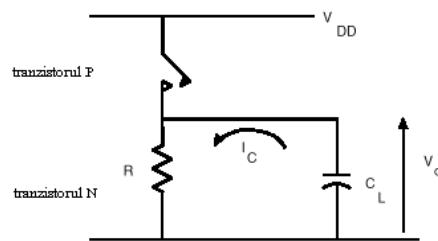
In continuare vor fi examinate cateva din punctele semnificative ale traectoriei.

- X_1 : inaintea descarcarii lui C_L , tranzistorul N (trage-jos sau pull-down) este blocat si C_L este mentinut la V_{DD} prin intermediul tranzistorului de tip P (trage-sus sau pull-up);
- X_2 : punctul corespunzator comutarii intrarii de la GND la V_{DD} . Tranzistorul N este in saturatie;
- X_3 : punctul in care tranzistorul N comuta de la saturatie la regiunea liniara;
- X_4 : punctul de operare dupa comutarea completa, cand sarcina capacitiva este descarcata.
- In timpul descarcarii lui C_L tranzistorul N este initial in saturatie intre punctele X_2 si X_3 , iar apoi in regiunea liniara de operare intre punctele X_3 si X_4 . Procesul este ilustrat in figura de mai jos si comentat astfel:

- a) in timpul descarcarii initiale a lui C_L , $V_{DS} > V_{GS} - V_m$, ceea ce face ca tranzistorul N sa fie in saturatie, ceea ce face ca C_L sa fie descarat, printr-o sursa de curent, cu o variație liniara a tensiunii, deoarece $Q=CV$ si $I=dQ/dt$.
- b) cand tensiunea de la iesire se reduce astfel incat $V_{DS} = V_o = V_{GS} - V_m$, tranzistorul N intra in regiunea liniara si descarca sarcina capacitivea C_L (cadere exponentiala a tensiunii).



a) saturatie: $V_o > V_{DD} - V_m$,



b) liniara: $0 < V_o < V_{DD} - V_m$

cea mai simpla metoda de a estima timpul de cadere consta in divizarea operarii tranzistorului N in regiunile de saturatie si liniara. Astfel, t_f va avea doua componente t_{f1} si t_{f2} definite dupa cum urmeaza:

t_{f1} timpul de cadere de la $0.9V_{DD}$ la $V_D - V_m$

t_{f2} timpul de cadere de la $V_{DD} - V_m$ la $0.1V_{DD}$

- t_{f1} : se egaleaza curentul care trece prin sarcina capacitivea cu cel care trece prin tranzistorul N(egali si de semn contrar)

$$C_L dV_o/dt = -\beta_n/2 \cdot (V_{DD} - V_m)^2$$

Termenul din stanga reprezinta curentul prin C_L , iar termenul din dreapta constituie curentul de saturatie (sursa de curent). Regiunea de interes va fi pentru $0.9V_{DD} \geq V_o \geq V_{DD} - V_m$.

- t_{f2} : aici se merge pe aceeasi idee ca mai sus, ceea ce va conduce la urmatoarea relatie:

$$C_L \frac{dV_o}{dt} = -\beta_n [(V_{DD} - V_{tn})V_o - V_o^2/2]$$

unde termenul din stanga reprezinta curentul prin sarcina capacativa, iar termenul din dreapta constituie curentul prin tranzistorul aflat in regiunea liniara. Regiunea de interes este cea corespunzatoare conditiei: $V_{DD} - V_{tn} \geq V_o \geq 0.1V_{DD}$.

Ambele ecuatii pot fi integrate pentru a obtine: $t_f = t_{f1} + t_{f2}$. Solutia pentru t_{f1} este directa:

$$\int_{V_{DD}}^{V_{DD}-V_{tn}} C_L dV_o = - \int_0^{t_{f1}} \beta_n / 2 (V_{DD} - V_m)^2 dt.$$

In mod similar t_{f2} poate fi gasit, printr-o integrare mai dificila. Astfel, se va obtine:

$$t_f = 2C_L/\beta_n (V_{DD} - V_{tn}) \times [(V_{tn} - 0.1V_{DD})/(V_{DD} - V_{tn}) + 1/2 \ln ((19V_{DD} - 20V_{tn})/V_{DD})]$$

Presupunand o valoare rezonabila $V_{tn} = 0.2V_{DD}$, se va obtine urmatoarea aproximatie utila pentru t_f

$$t_f \approx 4C_L/\beta_n V_{DD}.$$

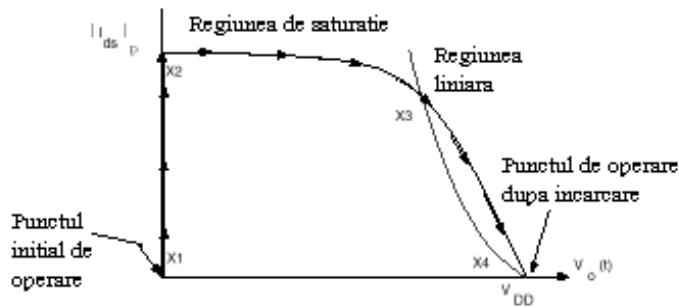
Aceasta reprezinta o aproximatie rezonabila care poate fi verificata prin simulare. De asemenea ea permite examinarea mai profunda a procesului de descarcare. Anterior s-a anticipat o descarcare in principal de tip RC. In expresia de mai sus numaratorul poate fi asimilat cu sarcina capacativa in timp ce inversul numitorul va corespunde aproximarii rezistentei canalului tranzistorului N, in regiunea liniara:

$$(\beta_n V_{DD})^{-1}$$

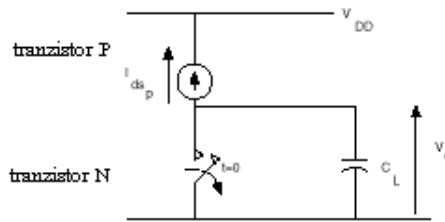
Pe aceasta baza se pot face diverse consideratii referitoare la faptul ca o crestere a amplificarii tranzistorului va conduce la cresterea vitezei, ca si cresterea lui V_{DD} . Aceasta din urma va antrena insa o crestere a puterii disipate si a purtatorilor fierbinti injectati, ca urmare a cresterii intensitatii campului electric. Practic viteza de operare poate fi crescuta prin reducerea lui V_{DD} simultan cu scurtarea lungimii canalului tranzistorului.

5.2.2. Determinarea timpului de crestere.

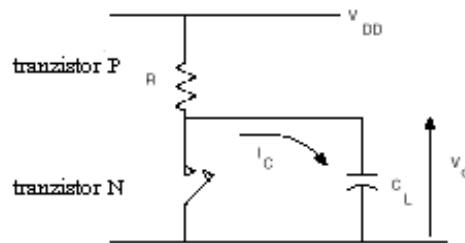
Timpul de crestere poate fi estimat in aceeasi maniera. Timpul de crestere, notat cu t_r , poate fi definit pe mai multe cai, in principiu echivalente. Una din definitii ar fi aceea a timpului necesar crestierii tensiunii de iesire V_o de la 10% la 90%, din valoarea maxima, conform unei figurii anterioare. Traекторia curentului de drena al tranzistorului P, ca functie de tensiunea de drena, este prezentata mai jos. Aceasta traекторie ilustreaza traseul prin caracteristicile I-V corespunzatoare unui inversor cu sarcina capacitiva C_L . In continuare se vor examina mai multe puncte plasate pe traectoria punctului de functionare.



- X_1 : Inainte de incarcarea lui C_L , tranzistorul de tip P este blocat iar capacitatea C_L este mentinuta la 0V de catre tranzistorul de tip N, aflat in conductie (tranzistoarele N si P se mai numesc “trage-jos” si “trage-sus”).
 - X_2 : Punctul corespunzator comutarii intrarii de la V_{DD} la GND. Tranzistorul P este in saturatie.
 - X_3 : Punctul de tranzitie unde tranzistorul P comuta de la regiunea de saturatie la regiunea liniara.
 - X_4 : Punctul de operare dupa terminarea comutarii. Sarcina capacitiva este incarcata la V_{DD} .
- In timpul incarcarii lui C_L tranzistorul P este initial in saturatie intre punctele X_2 si X_3 , iar apoi in regiunea liniara intre punctele X_3 si X_4 . Acest proces de incarcare este ilustrat in figura urmatoare.



a) saturatie: $V_o > V_{DD} - V_{tn}$



c) liniara: $0 < V_o < V_{DD} - V_{tn}$

- a) In timpul incarcarii initiale a lui C_L tranzistorul P se afla in saturatie. Currentul constant care il strabate va incarca C_L (crestere liniara a tensiunii deoarece $Q=CV$ si $I=dQ/dt$).
- b) Cand tensiunea de iesire creste astfel incat $|V_o - V_{DD}| = |V_{GSp} - V_{tp}|$, adica V_o creste peste $|V_{tp}|$, tranzistorul P intra in regiunea liniara, C_L incarcandu-se prin rezistenta acestuia la V_{DD} . t_r poate fi calculat ca si t_f , ceea ce va conduce, cu conditia $|V_{tp}| \approx 0.2 \cdot V_{DD}$, la:

$$t_r \approx 4C_L / \beta_p V_{DD}$$

Daca geometriile celor doua dispozitive N si P sunt identice (acelasi rapoarte W/L) $t_f \approx t_r/2$, datorita mobilitatii mai mari a electronilor in rapport cu goulurile ($\mu_n \approx 2\mu_p$). Astfel, pentru a avea timpii de crestere si cadere egali este necesar ca W_p sa fie de 2-3 mai mare decat W_n .

Un alt parametru comun este intarzirea pe poarta care poate fi definit din timpii de crestere si cadere dupa cum urmeaza:

$$\tau_{med} \approx (t_r + t_f)/4.$$

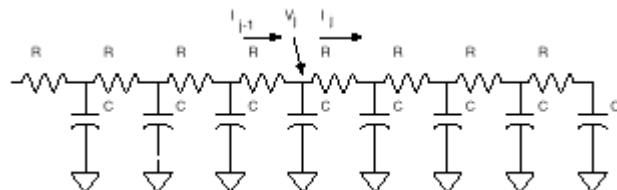
Acest parametru este important in simulatoarele care incorporeaza timpul si comportarea logica.

5.3. Efectele RC distribuite.

In paragraful anterior performanta unei porti simple a fost asociata cu incarcarea/descarcarea unei capacitatii prin rezistentele canalelor tranzistoarelor P/N. Desi in multe cazuri acest proces introduce limitari, uneori efectele liniilor de transmisie lungi nu pot fi neglijate. In aceste cazuri propagarea semnalului depinde de constantele distribuite R si C , de-a lungul liniei, cat si de impedanta interna a sursei de semnal si de impedanta de sarcina a liniei. In cazul liniilor lungi, realizate din siliciu policristalin, constantele RC distribuite reprezinta principala cauza a intarzierii semnalului. Chiar in cazul inlocuirii traseelor de siliciu policristalin cu un strat de metal (procesele in care se folosesc straturi multiple de metal au devenit frecvente), datorita dorintei de a obtine viteze ridicate constantele distribuite RC joaca un rol important, ca factor limitator. In figurile de mai jos se prezinta o linie lunga realizata din siliciu policristalin si schema ei electrica echivalenta, cu constante RC distribuite.



a) linie lunga din siliciu policristalin.



b) circuitul echivalent, cu constante RC distribuite

Pe baza primei teoreme a lui Kirchoff se obtine urmatoarea ecuatie:

$$C \frac{dV_j}{dt} = I_{j-1} - I_j \approx ((V_{j-1} - V_j)/R) - ((V_j - V_{j-1})/R),$$

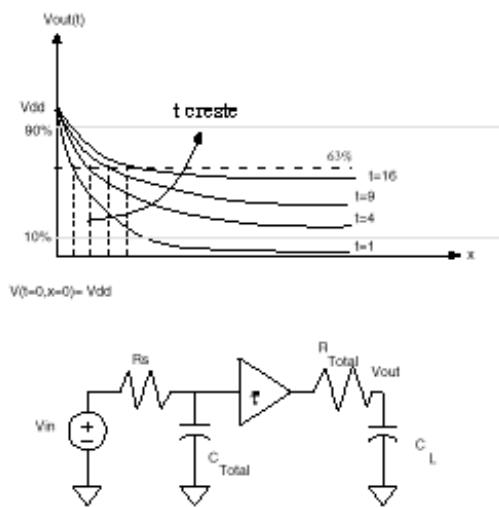
care, in cazul unui numar infinit de tronsoane (incare a fost descompusa linia lunga), devine ecuatie difuziei. De fapt expresia de mai sus reprezinta aproximarea prin diferente finite a ecuatiei Fourier, care guverneaza fenomenele de difuzie (transmisia caldurii s.a.). Daca se considera aplicata o treapta de tensiune la capatul din dreapta al liniei, propagarea acesteia va fi intarziata cu un timp t_x dat de expresia de mai jos:

$$t_x = kx^2.$$

cea ce arata ca intarzierea este proportionala cu patratul distantei. O solutie ar consta in sectionarea liniei si introducerea unor tampoane/amplificatoare, ceea ce ar putea conduce, in cel mai bun caz, la o dependenta logaritmica a intarzierii cu distanta.

La frecvente de ordinul GHz inductantele pot juca, de asemenea, un rol important. Pentru tehnologiile prezente efectul inductiv dominant este cel al conexiunilor la pastila.

In calculele care se vor efectua intarzierea va fi asociata cu momentul in care semnalul atinge valoarea de 63% din VDD, conform modelului de mai jos:



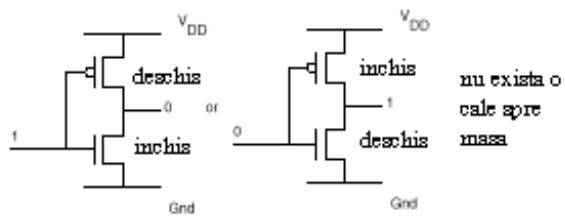
5.4. Puterea disipata.

In mod traditional circuitele CMOS au devenit atractive pentru consumul redus de putere. Principalul motiv este acela ca atunci cand poarta este blocata nu apare nici o cale, pentru curent, de la VDD la masa. Exista insa un curent de scurgere prin jonctiunile P-N inversate, cat si un curent de scurgere prin dielectric. Acesti curenti sunt neglijabili cat timp proiectul nu se bazeaza pe sarcin stocate, ca in cazul memoriei dinamice.

Puterea principala consumata este legata de procesul de comutare, adica de incarcarea si descarcarea sarcinii capacitive. Aceasta componenta devine semnificativa in cazul in care numarul dispozitivelor care comuta este mare, cat si in situatia in care frecventa de comutare este mare. Aceasta putere este consumata/disipata sub forma de caldura. Puterea disipata are o componenta dinamica si una statica. Cele mai multe proiecte de circuite integrate urmaresc reducerea puterii consumate. In aceasta sectiune se vor considera cateva metode pentru estimarea puterii consumate si pentru reducerea acestora.

5.4.1. Puterea disipata in regim static.

Atunci cand un circuit CMOS se afla in regim static (nu este in comutatie), avand la iesire 0 logic sau 1 logic, unul dintre tranzistoarele P sau N nu conduce, dupa cum se vede din figura de mai jos:



Aceasta implica inexistentia unei cai de la VDD la masa, ceea ce conduce la o putere disipata neglijabila. Motivul pentru care puterea disipata static nu este egala cu zero consta in prezenta jonctiunilor P-N, polarizate in sensul invers conductiei, pe tot cuprinsul pastilei. Aceste diode se caracterizeaza printr-o caracteristica IV de forma:

$$j_o = j_s (e^{qV/kT} - 1)$$

unde j_s este curentul invers de saturatie sau curentul de scurgere. O posibila estimare a puterii disipate static s-ar baza pe ideea de a considera intreaga pastila ca o jonctiune P-N polarizata invers. Evaluand densitatea curentul invers de saturatie si aria pastilei se va obtine puterea totala disipata. Aceasta ar fi ca si cum s-ar estima traficul telefonic intr-un oras mare folosind un singur telefon. In mod surprinzator aceasta abordare este destul de rezonabila. De exemplu, o pastila cu aria de $25 \cdot 10^{-6} m^2$ consuma circa 10 nW, in conditiile unei densitati a curentului invers de saturatie de $10^{-4} A/m^2$. Acest factor este dependent de temperatura putand atinge 100nW pentru un circuit , care opereaza la o temperatura cu $20 ^\circ C$ peste temperatura mediului ambiant. Astfel, puterea disipata static de catre un circuit CMOS este destul de mica, de cateva zeci de nW, intrucat nu exista o cale directa intre VDD si masa. Cu toate ca acest curent apparent nu are importanta, in prezent el se masoara in cadrul unui test denumit monitorizarea I_{DDQ} . Daca acest curent masurat este de ordinul micro sau miliamperilor inseamna ca exista un defect, care ofera o cale de la VDD la masa, ceea ce din punct de vedere logic nu ridica probleme, dar care poate crea probleme de fiabilitate.

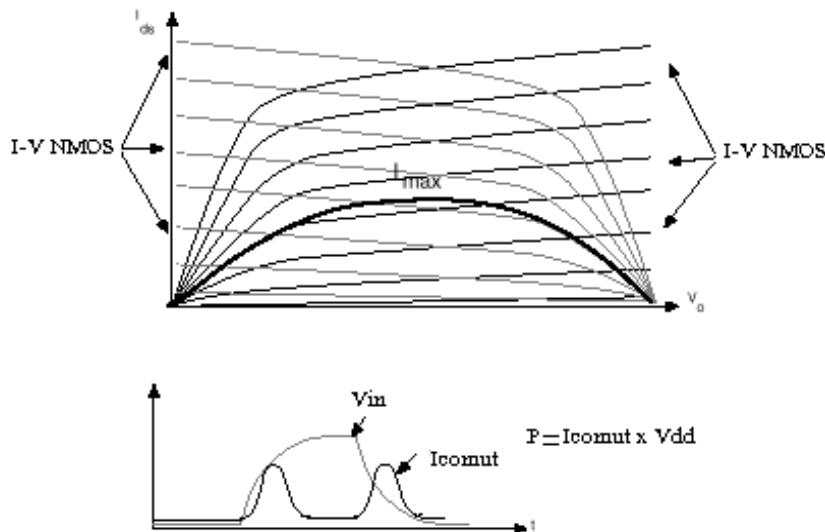
5.4.2. Puterea disipata in regim dinamic.

Puterea disipata in regim dinamic are doua componente principale:

- de tranzitie la comutatie, care apare in momentul realizarii unei cai de curent de la VDD la masa prin cele doua tranzistoare aflate temporar in conductie si
- de incarcare/descarcare a C_L , care se refera la pierderile de energie la incarcarea/descarcarea lui C_L , prin tranzistoarele P si N.

5.4.2.1. Componenta corespunzatoare tranzitiei la comutatie.

Pe durata tranzitiei nivelurilor logice de la $0 \rightarrow 1$ sau de la $1 \rightarrow 0$ exista momente in care cele doua tranzistoare conduc simultan, formand o cale pentru curent, de la VDD la masa, ceea ce da nastere unui varf de curent. Puterea disipata in tranzitia la comutatie poate fi evaluata cu ajutorul figurii de mai jos. Cazul cel mai defavorabil presupune ca circuitul opereaza la frecventa maxima/perioada minima ($t_f + t_r$). Se va presupune ca curentul maxim I_{max} curge conform traieroriei de tranzitie la comutatie de mai jos:



Considerand:

$$V_{in} = V_{DD}/2, V_m = |V_{tp}| = V_{DD}/5 \quad \text{si} \quad \beta_n = \beta_p$$

rezulta:

$$I_{max} \approx \beta_n / 2(V_{DD}/2 - V_{DD}/5)^2 \approx 1/20\beta_n V_{DD}^2.$$

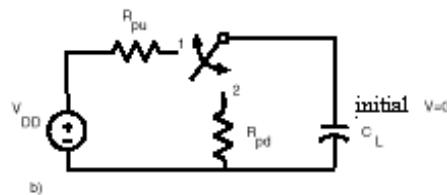
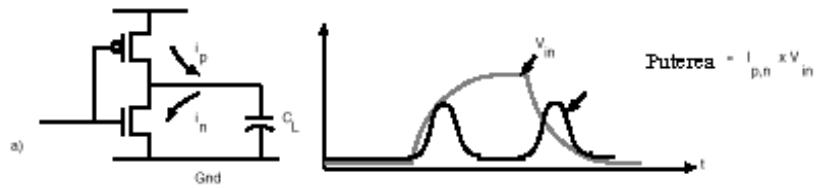
Limita superioara a energiei disipate pe durata unui ciclu devine:

$$E_{\text{dissipata/ciclu}} \leq \int_0^{t_f + tr} I_{\max} V_{DD} dt \leq 4/5 C_L V_{DD}^2.$$

Daca creste perioada puterea disipata nu va disparea. In cazul in care timpii de crestere si cadere cresc puterea disipata poate varia ca V_{DD}^3 . Circuitele care au tendinta de a consuma putere mare la comutare pot fi esantionate.

5.4.2.2. Incarcarea si descarcarea lui C_L .

Componenta cea mai mare a puterii disipate provine de la incarcarea/descarcarea sarcinii capacitive C_L , dupa cum se poate observa si din figura de mai jos:



Puterea disipata este:

$$P_d = 1/t_p \int_0^{tp/2} i_n(t) V_o dt + 1/t_p \int_{tp/2}^{tp} i_p(t) (V_{DD} - V_o) dt,$$

unde perioada de operare este inversa frecventei de operare f_p :

$$t_p = 1/f_p$$

Astfel, puterea disipata devine:

$$\begin{aligned} P_d &= C_L/t_p \int_{V_{DD}}^0 -V_o dV_o + C_L/t_p \int_0^{V_{DD}} -(V_{dd} - V_o) d(V_{DD} - V_o) \\ &= C_L V_{DD}^2 / t_p = f C_L V_{DD}^2 \end{aligned}$$

Expresia pentru energia disipata arata ca aceasta creste odata cu cresterea frecventei. O analiza simpla se poate efectua plecand de la figura b) de mai sus.

- Fie comutatorul trecut pe pozitia 1, care permite incarcarea lui C_L la V_{DD} . Cantitatea de sarcina furnizata este:

$$\int idt = Q = C_L V_{DD}.$$

Energia furnizat de sursa va fi:

$$\int V_{DD} idt = V_{DD} \int idt = C_L V_{DD}^2$$

Energia stocata de capacitor va fi:

$$1/2 C_L V_{DD}^2,$$

iar energia disipata in R_{pu} va fi:

$$1/2 C_L V_{DD}^2.$$

- Fie comutatorul trecut in pozitia 2, care asigura descarcarea lui C_L la masa. Energia initiala stocata in capacitor este:

$$1/2 C_L V_{DD}^2$$

iar energia finala este 0, ceea ce inseamna ca energia disipata in R_{pd} este:

$$1/2 C_L V_{DD}^2$$

Energia totala disipata la incarcarea/ descarcarea lui C_L este energia disipata in R_{pu} + energia disipata in R_{pd} :

$$E_{\text{dissipata/ciclu}} = 1/2C_L V_{DD}^2 + 1/2C_L V_{DD}^2 = C_L V_{DD}^2.$$

Puterea disipata la incarcarea/descarcarea lui C_L va fi:

$$= f C_L V_{DD}^2.$$

Puterea totala disipata de catre un circuit integrat reprezinta suma componentelor statica si dinamica, cea dinamica fiind dominanta. Puterea disipata variaza cu frecventa liniar si cu tensiunea de alimentare la patrat. Aceasta din urma reprezinta motivul pentru care s-a introdus tensiunea de alimentare "standard" de 3,3 V. Tensiunea redusa de alimentare va facilita aparitia dispozitivelor mai rapide, cu dimensiuni mai mici, in ciuda celor sugerate de catre aproximările pentru t_f si t_r .

5.5. Alternative in proiectare.

In cele ce urmeaza se vor prezenta solutii alternative pentru problemele legate de consumul de putere al circuitelor integrate CMOS.

- Circuitele care opereaza cu tensiune redusa. Pentru o tehnologie data, in vederea cresterii vitezei, se poate mari local tensiunea de alimentare a unor circuite critice. Una dintre dificultatile cu care se confrunta aceasta solutie se refera la efectele injectiei de purtatori fierbinti in substrat, ceea ce conduce la probleme de "latch-up" potential.
- O tehnologie alternativa. Combinarea dispozitivelor CMOS si Bipolare pe acelasi substrat, desi relativ costisitoare in prezent, ofera avantajele de viteza al dispozitivelor Bipolare si de putere consumata redusa al dispozitivelor CMOS. O alta tehnologie se bazeaza pe GaAs, care asigura viteze mari de lucru atat in aplicatiile numerice, cat si in cele analogice. Din pacate in cadrul acestei tehnologii nu exista dispozitive complementare. Tehnologia ECL asigura o viteza mare de lucru insa consuma o putere insemnata.
- Familii logice cu "bataie" redusa. Cresterea vitezei se poate realiza prin utilizare in unele parti ale circuitului a unor alternative de proiectare. Una dintre acestea srefera la

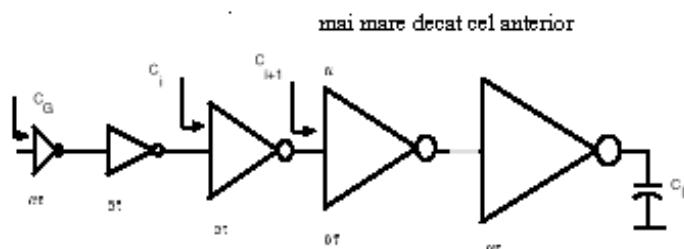
preincarcarea liniilor. Exemplul cel mai comun il reprezinta circuitele RAM, unde sarcina stocata pe capacitor este partajata cu o linie preincarcata si citita differential.

- Cresterea gradului de paralelism in prelucrarea informatiei. Un exemplu se refera la sumatorul paralel, care poate avea transportul succesiv/serial sau anticipat/paralel.

5.6. Exemple.

In acest paragraf se vor prezenta cateva exemple pentru a ilustra unele solutii privind performantele circuitelor CMOS si limitarile lor.

- Incarcarea sarcinilor capacitive mari. Adesea este necesara incarcarea unor sarcini capacitive mari cum sunt liniile lungi, ploturile de I/E si sarcinile capacitive conectate la terminele circuitelor. Se reaminteste ca valoarea capacitati unei porti poate fi de ordinul a 0.01 pF , in timp ce sarcina capacativa din afara circuitului integrat poate avea zeci de pF . Incercarea de a comanda o asemenea sarcina capacativa cu un inversor elementar ar necesita intervale de timp de ordinul *secundelor*. O solutie ar consta in conectarea in cascada a unor inversoare din ce in ce mai puternice, capabile sa conduca curenti din ce in ce mai mari, ca in figura de mai jos.



Intarzierea pentru un etaj inversor dat este proportionala cu sarcina capacativa vazuta de acel inversor si invers proportionala cu β etajului. Intarzierea totala este:

$$\tau_N \propto N\alpha\tau,$$

intrucat atat β , cat si sarcina capacativa vazuta de fiecare etaj se scaleaza cu α .

Deoarece,

$$C_{i+1} = \alpha C_i \rightarrow C_L = \alpha^N C_G \rightarrow \ln \alpha^N = \ln C_L / C_G,$$

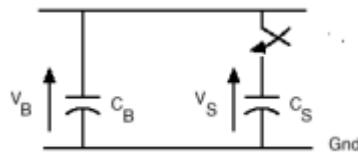
N poate fi extras si introdus in τ_N , pentru a obtine :

$$\tau_N = \tau \cdot \ln(C_L/C_G) \alpha / (\ln \alpha)$$

Aceasta expresie ia valoarea minima atunci cand $\alpha = e$, ceea ce implica utilizarea unor etaje succesive, adica 6-7 etaje, pentru cazul de mai sus.

$$N = \ln(C_L/C_G)$$

- Partajarea sarcinii. Multe dintre structurile circuitelor CMOS se comporta ca niste capacitori. De exemplu, o linie lunga sau o magistrala este tratata ca un capacitor concentrat C_B . Presupunand ca se doreste citire unei magistrale cu un capacitor C_S ca in figura de mai jos:



Cei doi capacitori vor fi in paralel la inchiderea comutatorului. Pentru ca V_B sa nu se modifice este necesar ca C_S sa fie mic. De exemplu, presupunand ca V_B este egal cu V_{DD} vom avea $Q = C_B$. V_{DD} , iar cand comutatorul este inchis, sarcina se redistribuie pe cei doi capacitori astfel ca: $Q = (C_B + C_S).V_{rez}$.

Deci, $V_{rez} = V_{DD} \cdot C_B / (C_B + C_S)$, ceea ce inseamna ca C_S nu trebuie sa depaseasca cateva zecimi din C_B .

- Calculul reimprospatarii pentru DRAM. Un calcul similar se refera la RAM dinamic. Intr-o celula DRAM sarcina este stocata pe un capacitor. O linie de bit preincarcata se conecteaza la capacitor, iar diferența de tensiune datorita partajarii sarcinii este citita pe linia de bit. Problema care apare se refera la capacitorul de memorare, care se descarca datorita curentului de scurgere prin dielectricul neideal si datorita existentei jonctiunii P-N polarizata in sensul invers conductiei. Pentru a pastra continutul corect in DRAM acesta trebuie reimprospatat periodic, la intervale de ms .

Capitolul 6.

Circuite fundamentale.

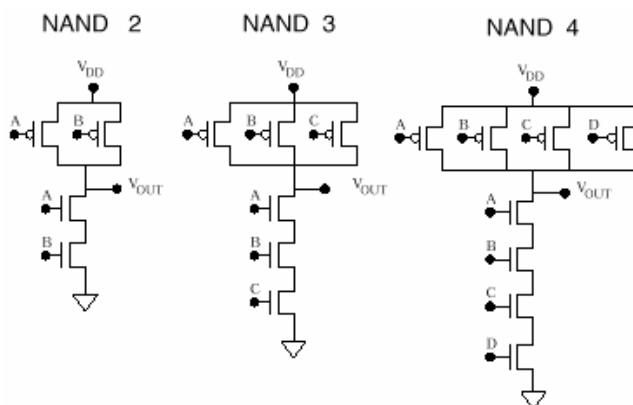
In acest capitol se vor prezenta circuitele standard CMOS sub forma de scheme si masti. Informatiile referitoare la mastile necesare in procesul de generare a diverselor straturi ale structurilor circuitelor CMOS numerice vor fi, intr-o mare masura, simplificate. Se va putea observa ca, in alcatauirea portilor de baza, se intalnesc microcelule elementare analoge limbajului de programare sau microcodului.

6.1. Portile standard.

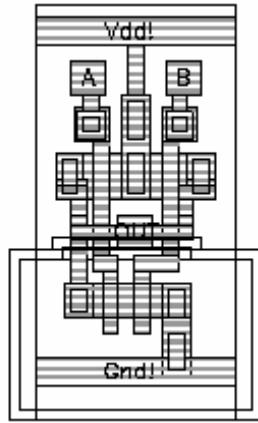
Cand se deseneaza un circuit complex, cea mai buna metoda consta intr-o abordare ierarhica. Spre exemplu, atunci cand se proiecteaza un microprocesor nu se va pleca de la nivelul tranzistoarelor, ci de la nivelul portilor de tipul NAND, NOR, si de la cel al bistabilelor. Pe aceasta baza se vor crea nivelurile superioare ale structurii: registrele, unitatile aritmetice-logice s.a. Aceasta abordare “de jos in sus” reprezinta un mijloc de a solutiona probleme cu caracter complex. In aceasta sectiune se vor prezenta cateva circuite standard si structuri de porti. Pe de alta parte, avand create aceste module sau megacelule, proiectele urmatoare se vor baza pe aceste structuri in cadrul unei abordari “de sus in jos”.

6.1.1. Circuitul NAND.

Schema pentru poarta NAND a fost prezentata in primul capitol. Figura de mai jos prezinta schemele pentru portile NAND cu doua, trei si patru intrari. Tranzistoarele PMOS au sursele conectate la V_{DD} si drenele la V_{OUT} . Tranzistoarele NMOS sunt conectate in serie, cel de la nivelul cel mai de jos avand sursa conectata la masa, iar cel de la nivelul cel mai de sus drena conectata la V_{OUT} .

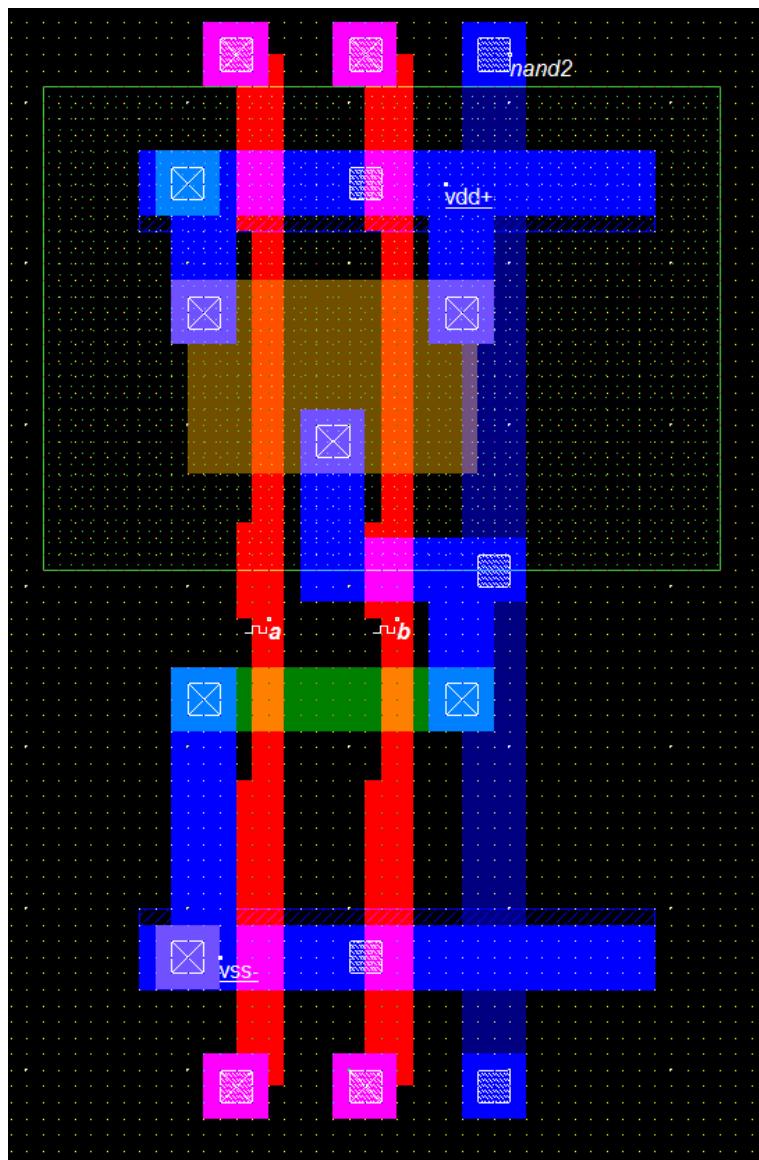


Desenul mastilor portii NAND cu doua intrari, in conditiile in care substratul este dopat N, este aratat mai jos. Aceasta celula posedă cateva trasaturi notabile. Mai intai, traseele VDD și GND formează o bară care se extinde transversal la nivelul superior și la nivelul inferior. Aceasta permite plasarea alăturată a unor asemenea celule. În acest mod se vor forma liniile de alimentare și de masa, care se extind de la stanga la dreapta, fără alte conexiuni externe. Ca rezultat, dimensiunea liniei și spațiile între ele trebuie standardizate.

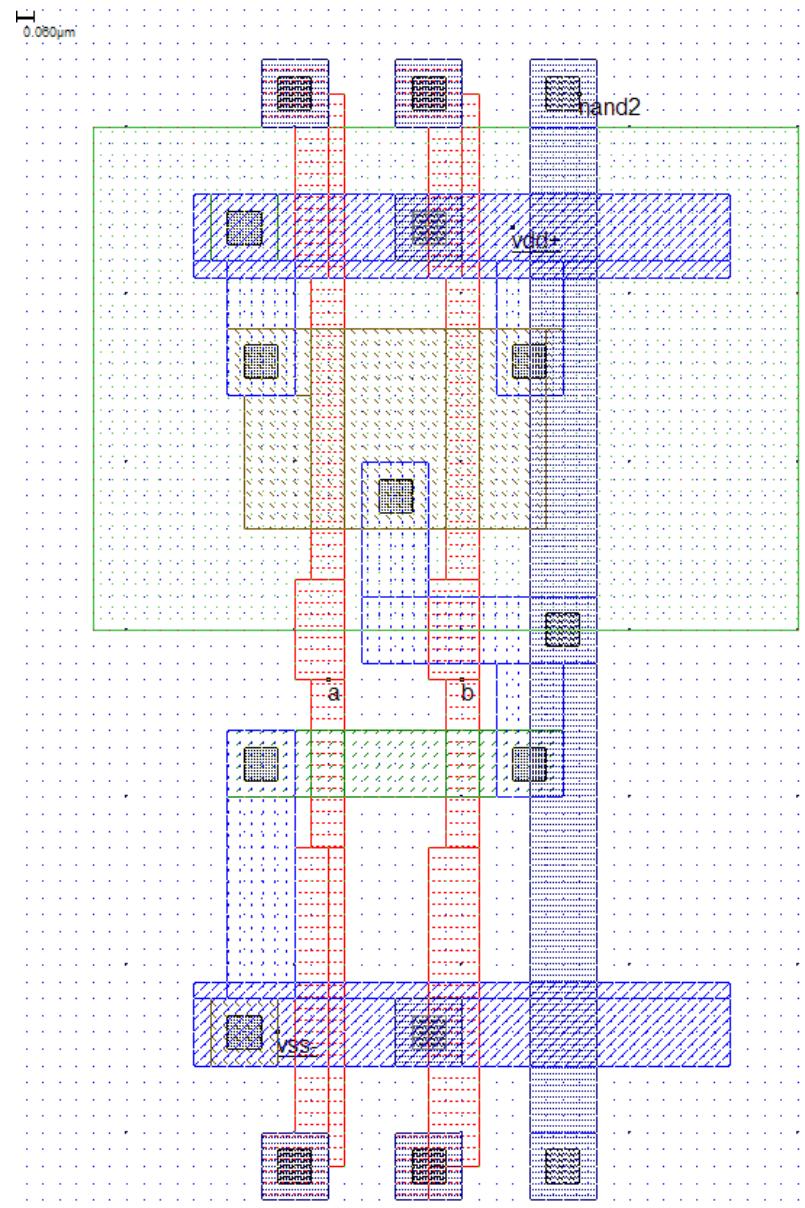


Acest tip de celula este folosit în programele de autorutare. Intrările A și B sunt conectate la trasee din siliciu policristalin, care formează tranzistoarele PMOS și NMOS. Cele două tranzistoare PMOS sunt adiacente la sursele lor, pentru a economisi spațiu și pentru a îmbunătăți performanța. O strategie similară de conectare va fi utilizată și pentru cele două tranzistoare NMOS. Se poate observa că dispozitivele NMOS se află pe aceeași insulă P, care are, de asemenea, un contact la GND.

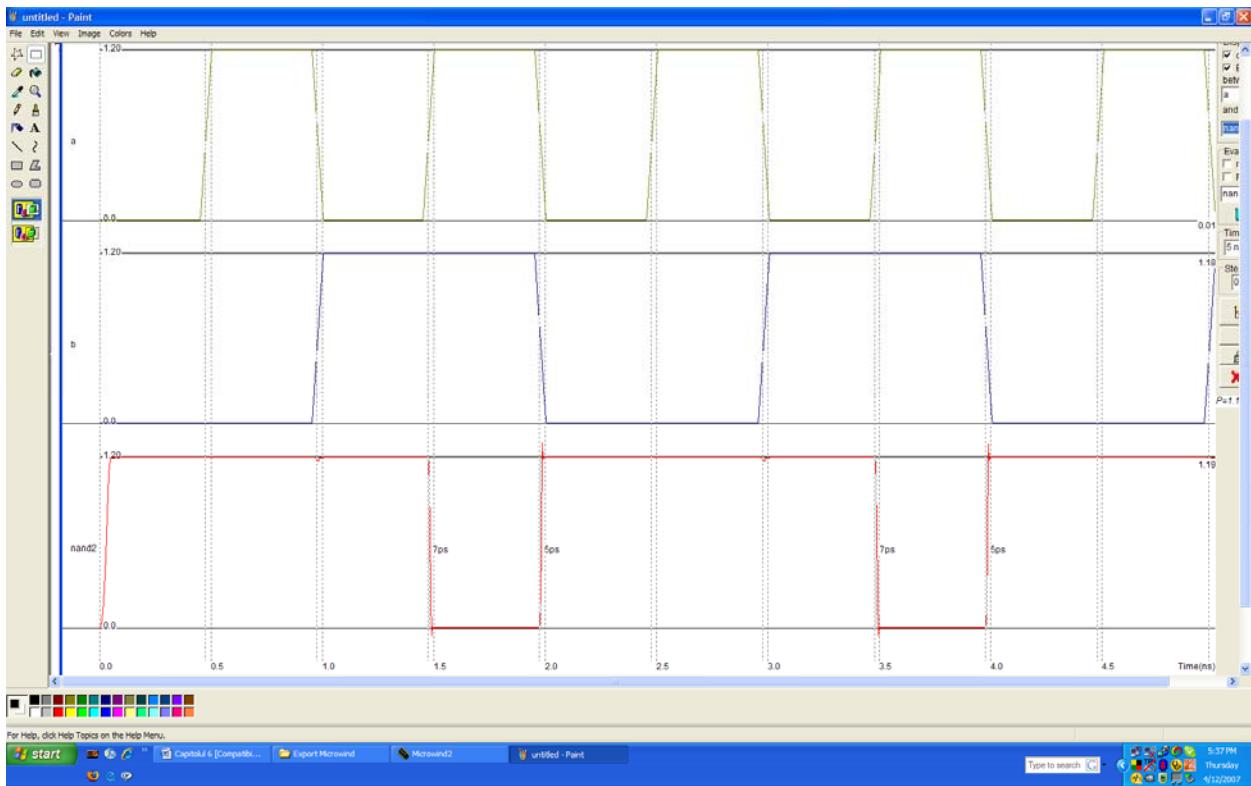
În desenul de mai jos, se prezintă mastile pentru o poartă NAND, cu 2 intrări, realizată pe un substrat dopat P.



Masti rezultate din compilarea expresiei: **nand2=~(a&b)**



Rezultatul simularii operarii portii NAND2

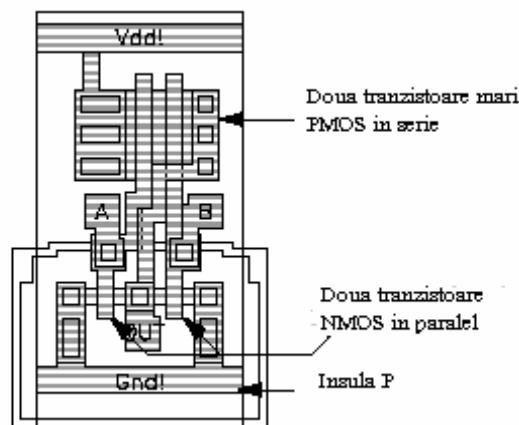
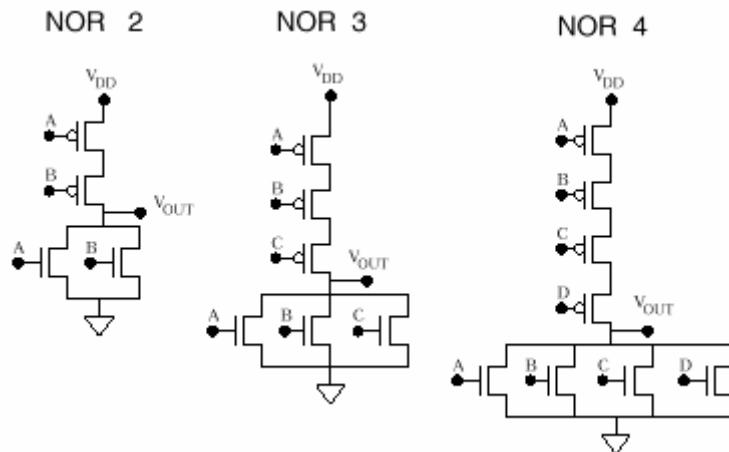


6.1.2. Circuitul NOR.

Schemele pentru portile NOR cu doua, trei si patru intrari sunt prezentate mai jos. Dupa cum se poate observa tranzistoarele PMOS sunt conectate in serie iar cele NMOS sunt conectate in paralel.

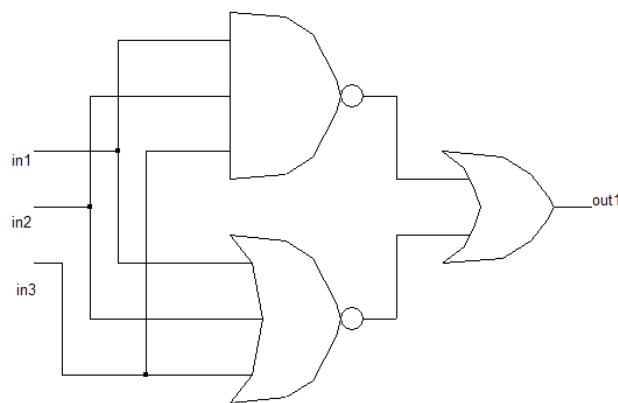
Desenul mastilor pentru poarta NOR cu doua intrari prezinta aceleasi caracteristici ca si cel pentru poarta NAND. Trebuie remarcat faptul ca tranzistoarele de tip P, fiind in serie, trebuie sa aibă dimensiuni mai mari decat tranzistoarele de tip N, pentru a asigura timpi de crestere si cadere egali. Explicatia acestei afirmatii s-a dat in Capitolul 5.

Un alt aspect se refera la faptul ca schema circuitului la nivelul tranzistoarelor reprezinta amplasarea relativa a acestora din urma. Desenele reale ale mastilor incearca, de regula, sa minimizeze aria ocupata. In acest scop se utilizeaza conexiuni si orientari ale tranzistoarelor cat mai convenabile.



Implementarea unei retele combinationale care realizeaza functia:

$$\text{out1} = \sim(\text{in1} \& \text{in2} \& \text{in3}) \mid \sim(\text{in1} \mid \text{in2} \mid \text{in3})$$



Se pleaca de la fisierul Verilog de mai jos

```
module nand3_or_nor3( in1,in2,in3,out1);
```

```
    input in1,in2,in3;
```

```
    output out1;
```

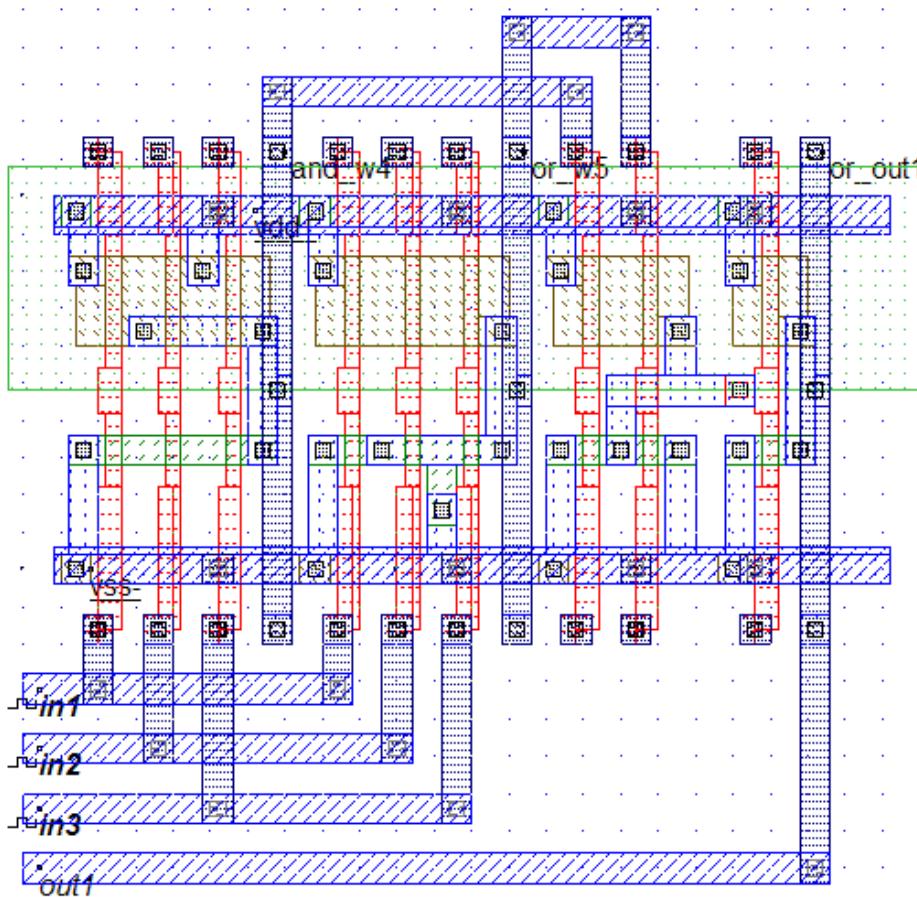
```
    wire w4, w5;
```

```
    nand #(13) and(w4,in1,in2,in3);
```

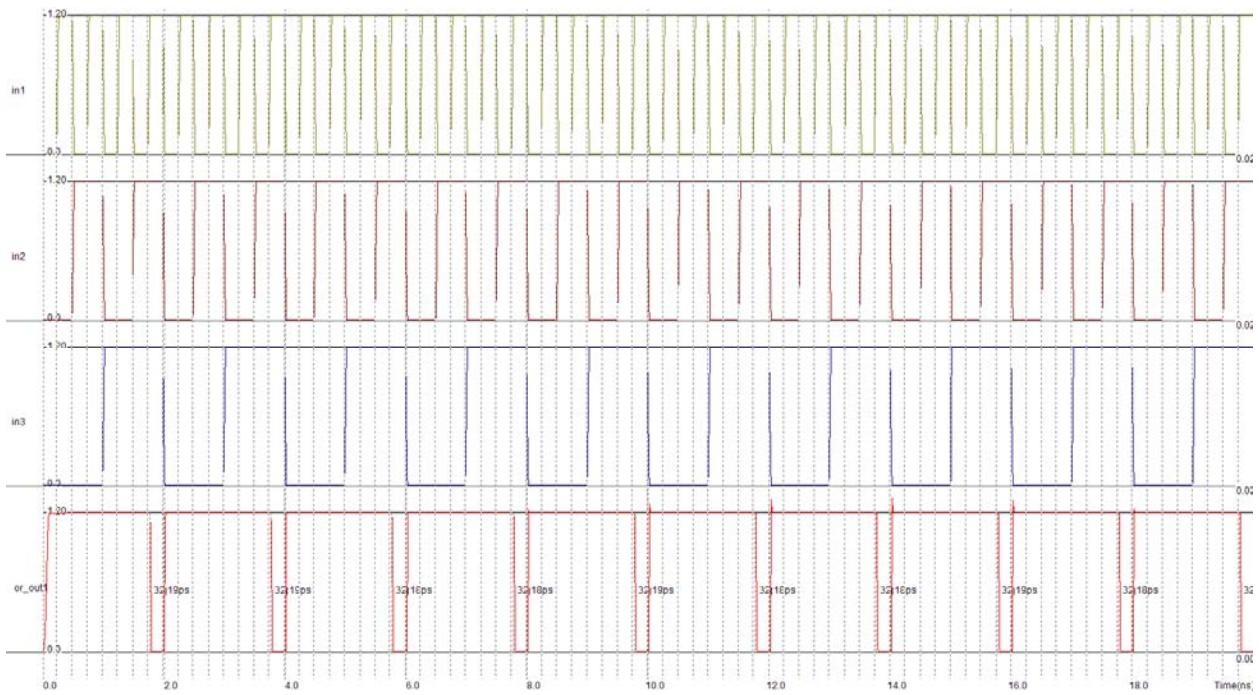
```
    nor #(13) or(w5,in1,in2,in3);
```

```
    or #(16) or(out1,w4,w5);
```

```
endmodule
```



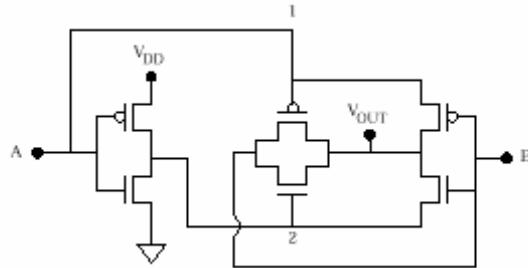
Rezultatul simularii:



6.3.1. SAU-Exclusiv (XOR).

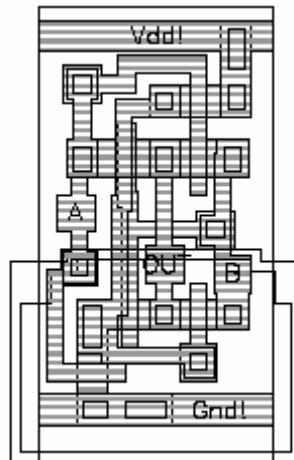
Poarta XOR are numeroase utilizari in proiectarea circuitelor numerice. Functia indeplinita de acest circuit nu se realizeaza direct, ca in cazul circuitului NAND. De aceea, vor fi prezentate doua versiuni ale circuitului XOR. Prima versiune se caracterizeaza printr-o arie ocupata mai mica, si prin folosirea portilor de transmisie. Intrucat puterea de comanda la iesire deriva din intrari, circuitul mai poarta numele de XOR pasiv. Cel de-a doua configuratie sde bazeaza pe porti statice CMOS. Aceasta solutie asigura un timp de raspuns mai rapid, dar utilizeaza mai multe tranzistoare in proiectare.

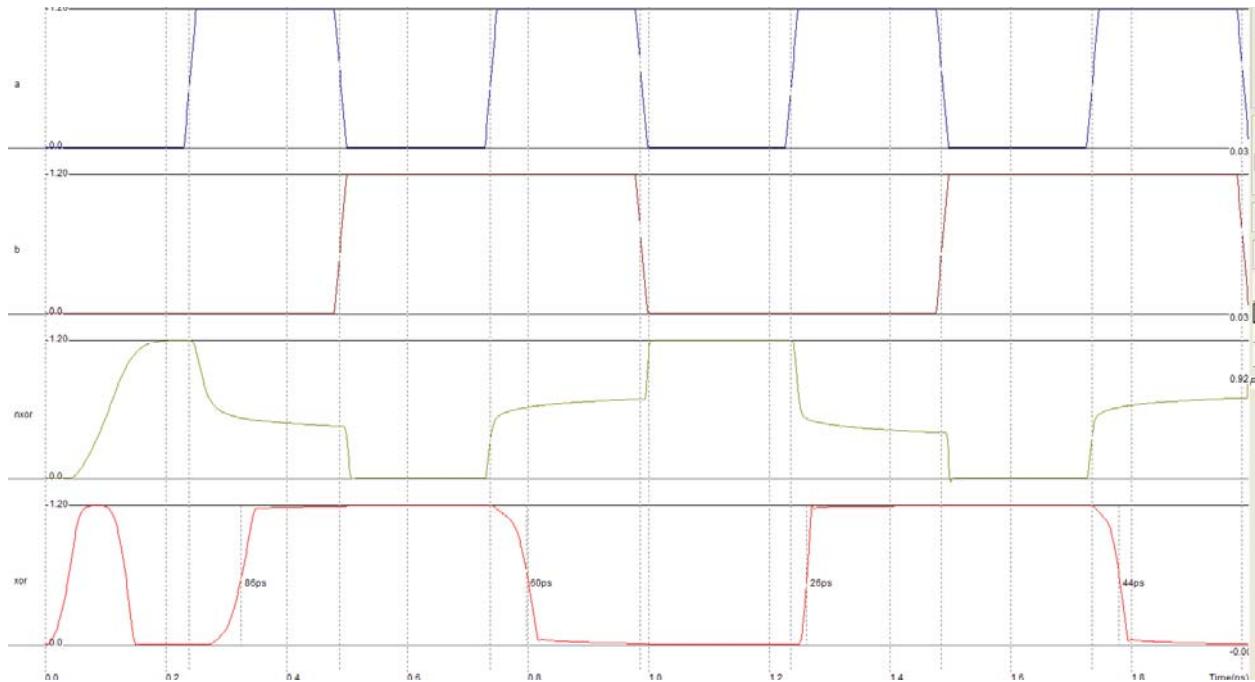
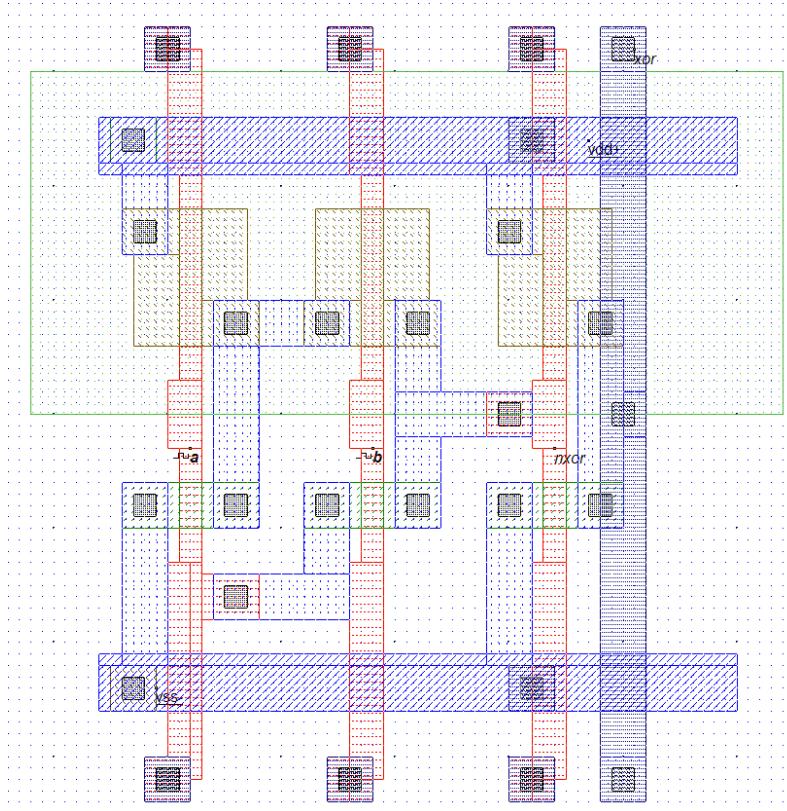
In figura de mai jos se prezinta schema pentru poarta XOR pasiva. Se poate observa inversorul de pe intrarea A, structura care seamna cu un inversor pe intrarea B si poarta de transmisie in centrul circuitului. In continuare se va face o analiza a celor patru cazuri posibile, pentru semnalele logice de la intrare.



- A=0, B=0. Aceasta face ca punctul 1 sa aibe valoarea logica 0 si punctul 2 – valoarea 1, ceea ce va face ca poarta de transmisie sa fie in conductie. Prin aceasta se creaza o cale de la B la V_{OUT} , prin poarta de transmisie. Intrucat B este la 0 sau la masa, $V_{OUT}=0$
- A=0, B=1. Din nou punctele 1 si 2 au valorile 0 si respectiv 1. Poarta de transmisie conduce ceea ce face ca intrarea 1, de la B, sa se propage catre V_{OUT} . Astfel, $V_{OUT} = 1$.
- A=1, B=0. Punctele 1 si 2 au valorile 1 si respectiv 0. Poarta de transmisie este blocata, iar inversorul corespunzator intrarii B va fi operational. Deoarece B=0 si A=1, iesirea V_{OUT} va avea valoarea 1.
- A=1, B=1. Acest caz este similar cu cel de mai sus, cu mentiunea ca inversorul corespunzator intrarii B va face ca $V_{OUT}=0$.

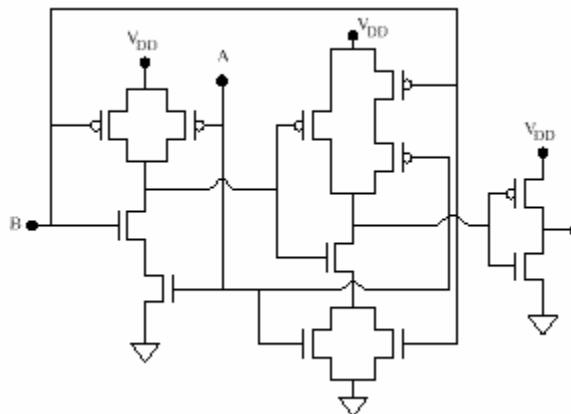
Aceasta poarta utilizeaza numai 6 tranzistoare, dupa cum se poate vedea din figura de mai jos.



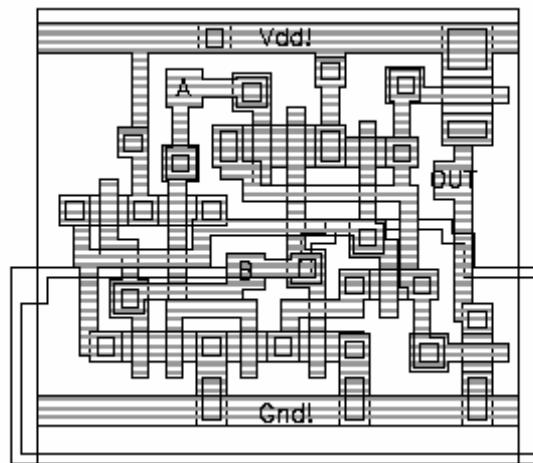


Cealalta implementarea, care utilizeaza 12 tranzistoare, are o performanta mai buna, dar este

mai costisitoare, ca hardware. Prima sectiune a schemei poate fi recunoscuta ca o poarta NAND. Sectiunea din mijloc a schemei implementeaza functia NOT((A \cup B).F) unde F=NOT(A.B). Ultima sectiune reprezinta un inversor. Aceasta structura permite ultimului inversor sa realizeze o comanda mai puternica decat in cazul proiectului anterior, bazat pe poarta de transmisie. In figurile de mai jos se prezinta schema si mastile acestui circuit. In acest caz curentul de comanda de la iesire este furnizat de sursa de alimentare, in timp ce pentru prima solutie acesta era asigurat de intrarile A si B.



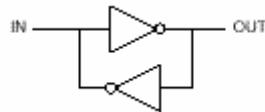
a) schema circuitului XOR activ



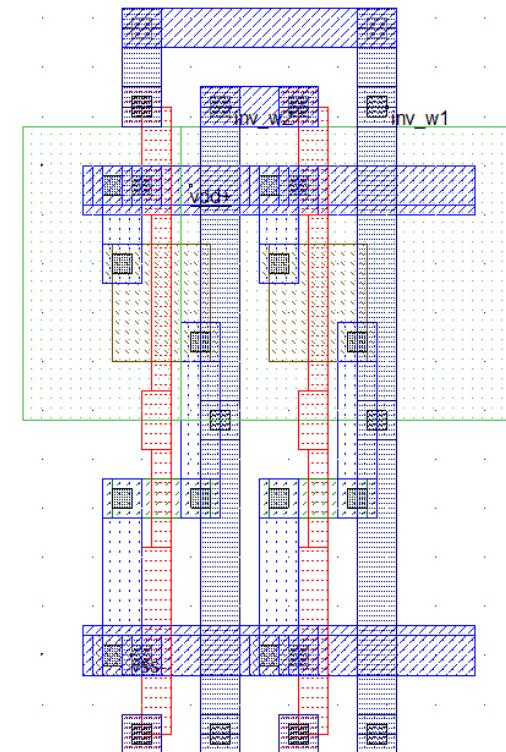
b) planul mastilor circuitului XOR active

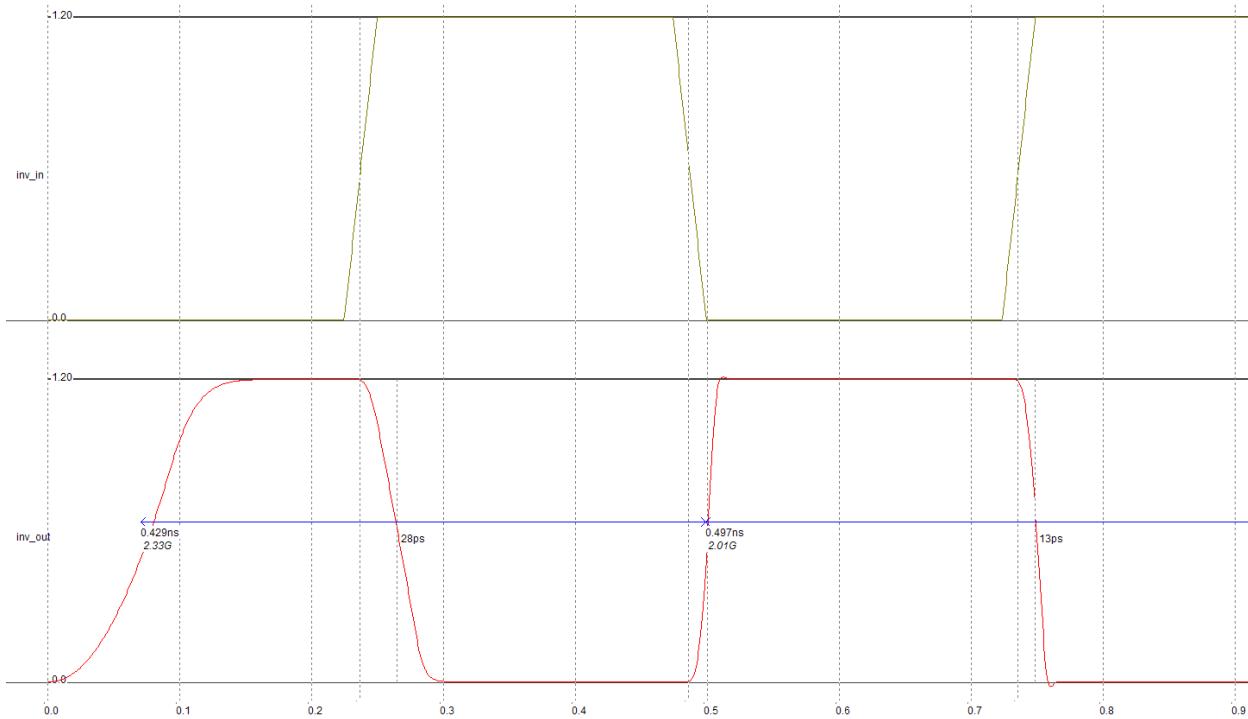
6.2. Elemente de memorare, registre de deplasare, bistabile.

La implementarea logicii se eventuale sunt necesare elemente de memorare pentru a stoca rezultatele. Aceste elemente stocheaza un bit si pot fi organizate pentru a forma structuri mai mari. Cel mai simplu element de memorare este latch-ul format din doua inversoare, ca in desenul de mai jos, unde iesirea OUT va avea o valoare constanta, fie 1, fie 0. Pentru a forta o noua informatie intrarea trebuie sa depaseasca reactia latch-ului. Semnalul de reactie trebuie sa fie mai slab, in ceea ce priveste comanda, decat semnalul de intrare.

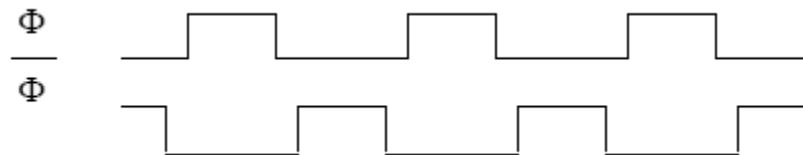


```
module Cel_mem( );
not #(10) inv(w2,w1);
not #(10) inv(w1,w2);
endmodule
```

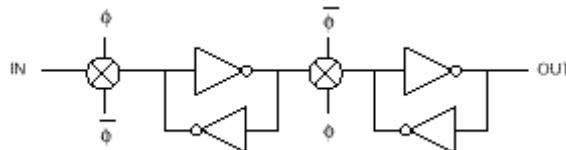




In figura de mai jos se arata modul in care poate fi realizat un registru de deplasare, folosind latch-uri. Un element de memorare se poate realiza cu doua latch-uri si doua porti de transmisie. Portile de transmisie vor fi controlate cu impulsuri de ceas complementare Φ si $\sim\Phi$. Pentru efectuarea deplasarii cu un rang trebuie aplicat semnalul de ceas.

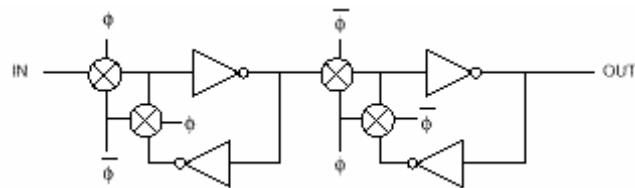


a) semnale de ceas bifazic, fara suprapunere.



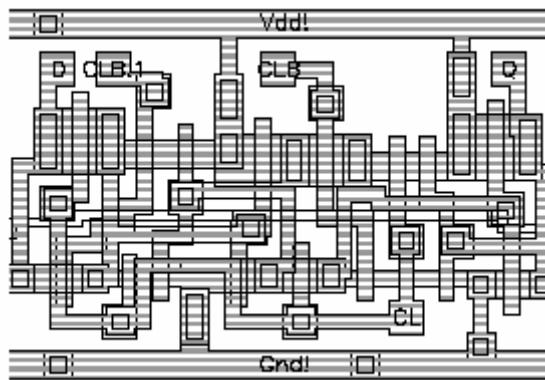
b) element de memorare cu latch-uri si porti de transmisie.

In figura urmatoare se prezinta schema unui bistabil de tip D realizat cu doua latch-uri si doua porti de transmisie controlate cu semnale de ceas bifazic.



Cand semnalul de ceas este pe nivel coborat , poarta de transfer este blocata, calea de reactie e ramane activa, iar latch-ul pastreaza valoarea memorata. Cand semnalul de ceas este pe nivel ridicat, calea de reactie este blocata, iar calea de intrare este activa, astfel ca, o noua valoare a semnalului poate fi stocata. Pentru a realiza un registru de deplasare se pot conecta in cascada astfel de celule.

In figura de mai jos se prezinta planul mastilor pentru un asemenea bistabil, de tip D.



6.3 Ploturi de Intrare/Iesire.

6.3.1. Intrare.

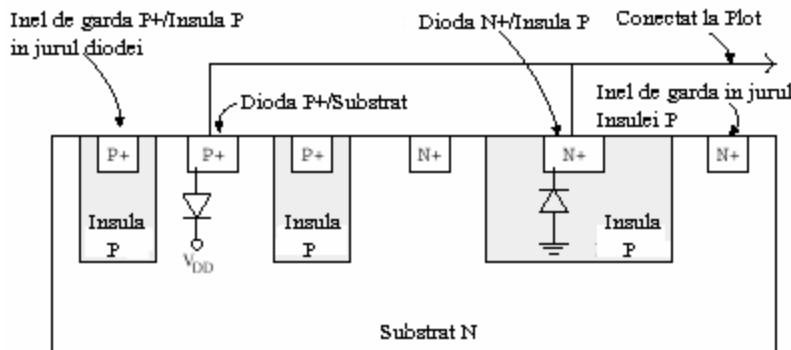
Atunci cand se realizeaza un plot de intrare trebuie sa se aibe in vedere protectia circuitului.

Variatiile de tensiune pot conduce la defectarea circuitului. Pentru a preveni asemenea situatii trebuie utilizata o schema de protectie ca in figura de mai jos.

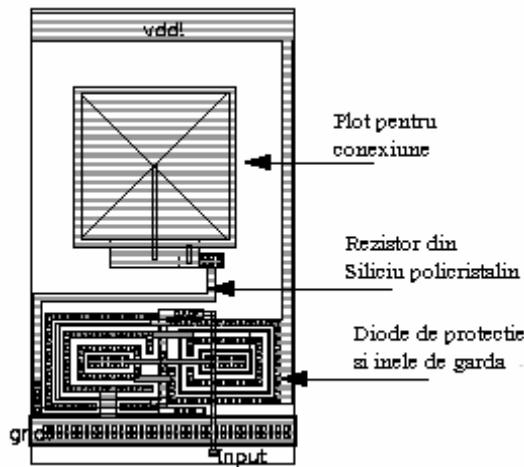
Rezistorul este realizat din siliciu policristalin si are o valoare a rezistentei de $500 - 3000 \Omega$.

Daca tensiunea de intrare depaseste V_{DD} , dioda superioara va conduce la limitarea tensiunii de intrare a portii la valoarea V_{DD} . In mod asemanator, daca tensiunea de intrare scade sub V_{SS} ,

dioda inferioara va conduce limitand inferior tensiunea de intrare a portii la Vss. In cazul unui proces cu insula P, dioda superioara este creata prin plasarea unei regiuni P₊ pe un substrat de tip N. Dioda inferioara este realizata prin formarea unei regiuni N₊ pe o insula P. Pentru a realiza protectia fata de curentii din substrat se creaza un inel de protectie in jurul diodelor. In figura de mai jos se prezinta o sectiune transversala prin structura de protectie.



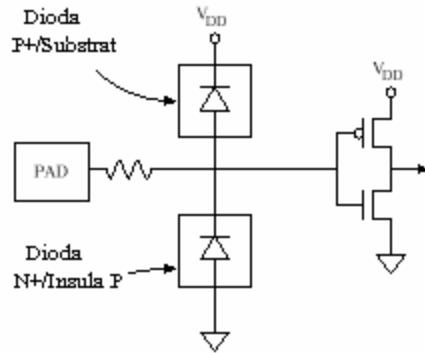
In figura urmatoare se prezinta planul mastilor unui plot de intrare. Dioda polarizata la VDD se afla in stanga jos, iar dioda polarizata de Vss este plasata in dreapta sus.



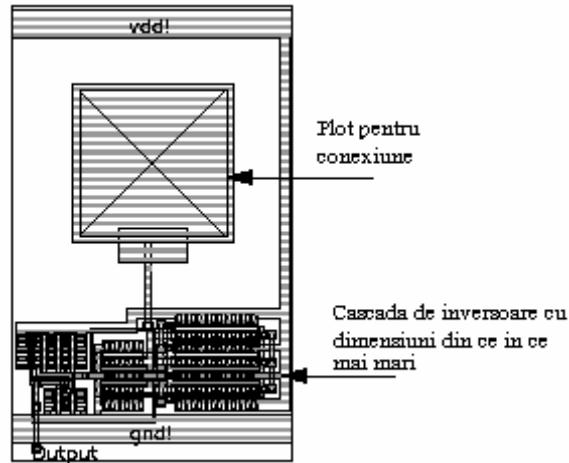
6.3.2. Iesire.

Un plot de iesire trebuie sa aibe o capacitate de comanda a unei sarcini externe astfel incat timpii de crestere si de cadere sa nu fie excesiv de mari. Configuratia uzuala se bazeaza pe mai

multe inversoare, cu dimensiuni din ce in ce mai mari, conectate in cascada. Raportul optim pentru dimensiuni este de circa 2,7. Planul mastilor pentru un plot de iesire este dat mai jos.

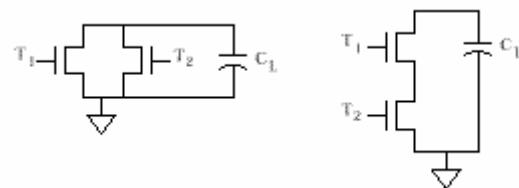


In stanga jos se afla unul dintre cele doua inversoare ale acestui plot de iesire. In dreapta jos este plasat un alt inversor cu dimensiuni mai mari decat cel din stanga.



6.4. Performanta.

Proiectarea portilor logice trebuie sa aibe in vedere si performanta. In exemplele de porti NAND si NOR, tranzistoarele erau conectate in serie si in paralel. In figura de mai jos se prezinta doua tranzistoare NMOS conectate in paralel si in serie, prin care trebuie sa se descarce sarcina capacitiva C_L .



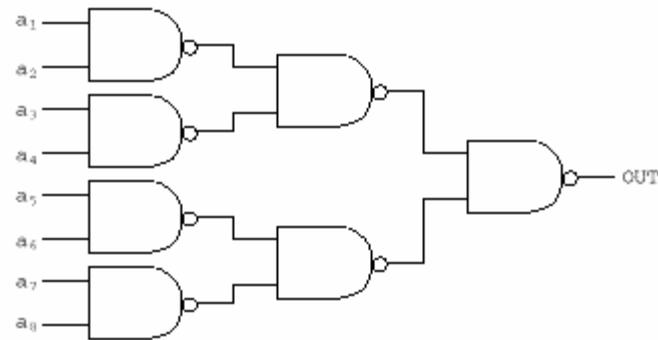
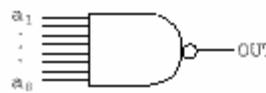
Cand ambele tranzistoare sunt deschise ele vor opera in regiunea liniara si se vor comporta ca

niste rezistoare. In cazul tranzistoarelor in serie descarcarea va fi mai lenta deoarece rezistentele celor doua tranzistoare se aduna. In cazul legarii in paralel rezistenta va fi mai mica decat cea a unui tranzistor. Astfel, timpul necesar descarcarii unui capacitor va fi dat de

$$T_{fSerie} \sim m \cdot T_f$$

$$T_{fParalel} \sim T_f/m$$

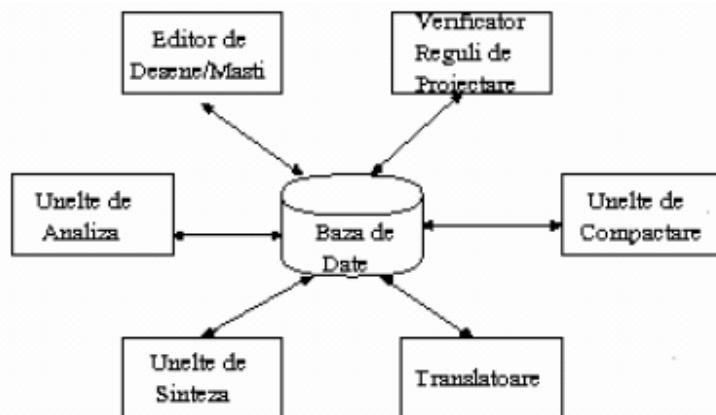
Unde m este numarul de tranzistoare. Trebuie aratat ca relatiile se pastreaza daca tranzistoarele comuta simultan.



Capitolul 7. Unelte de proiectare.

Perfectionarea tehniciilor de fabricatie, cat si reducerea dimensiunilor dispozitivelor au condus la realizarea unor sisteme numerice mari si complexe pe o singura pastila. Complexitatea crescandă a circuitelor integrate comerciale curente se datoreaza parțial si existentei uneltelor de Proiectare Asistata de Calculator (PAC). In acest mod proiectantul nu mai are de-a face cu lucrările/task-urile de nivel coborat, putandu-se concentra pe aspectele de nivel inalt ale proiectarii circuitelor integrate. In acest capitol se vor prezenta pe scurt cateva din uneltele disponibile pentru PAC, cat si algoritmii utilizati.. Pentru cea mai mare parte a uneltelor PAC discutate in acest capitol se vor examina aspectele de implementare de nivel coborat.

Un mediu idealizat de proiectare este aratat mai jos. Fiecare unealta folosita in proiectarea circuitelor integrate partajeaza o baza comună de date. Interfatarea cu alte unelte de proiectare, care utilizeaza o baza de date diferita se realizeaza prin translatarea informatiei de proiectare in formate de interschimb, care reprezinta limbaje standard suportate de catre cei mai multi ofertanti de unelte PAC.



Motivul pentru care schema de sus este idealizată este acela că, de regulă, o parte dintre ofertanții de unelte sunt angajați pentru anumite aspecte ale proiectării. În acest caz, adesea, ieșirea oferită de către o unealtă către alta o reprezintă o listă a componentelor/conexiunilor.

7.1. Descrierea proiectului.

Cele mai importante unelte folosite în mediul de proiectare de circuite integrate sunt uneltele de descriere și bazele lor de date asociate. Aceste unelte sunt utilizate pentru crearea, editarea și întreținerea informației de proiectare, indiferent că este vorba de masti, de conectivitate sau informație despre comportament.

In aceasta secțiune se vor examina câteva programe pentru descrierea proiectului, cat si tehnici referitoare la bazele de date asociate.

Indiferent de mediul de descriere a proiectului, se impune realizarea unei interfețe prietenoase cu utilizatorul pentru toate programele. Acestea vor fi programele pe care proiectantul le va folosi cel mai frecvent.

7.1.1. Editoare de masti.

Editoarele de masti sunt programe folosite pentru a descrie, modifica si intretine geometriile desenelor.

Cele mai multe programe de desenare sunt capabile sa deseneze forme simple, care sunt in mod curent utilizate in proiectarea circuitelor integrate. De asemenea, cele mai multe editoare de masti contin elemente specifice incorporate, pentru proiectarea acestora. Este foarte importanta interacțiunea cu verificatoarele de reguli de proiectare, cat si cu alte unelte de proiectare.

Caracteristicile dorite pentru un editor de masti sunt urmatoarele:

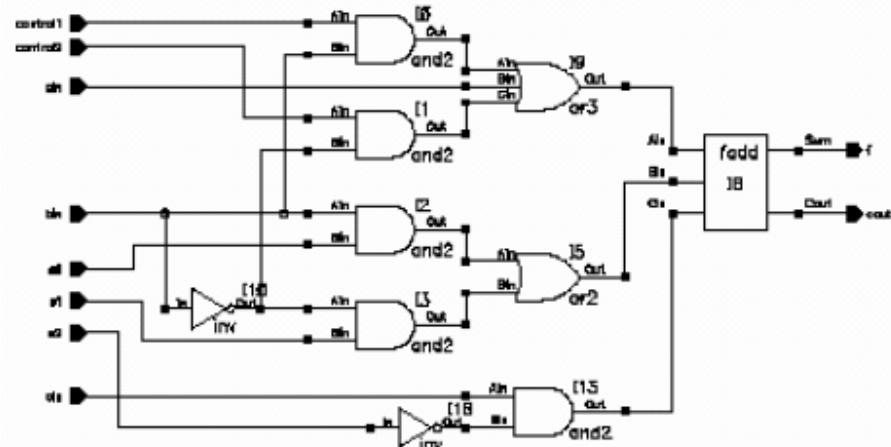
- Programul trebuie sa suporte geometriile des intalnite in desenele de masti. Tipic, cele mai multe proiecte de circuite integrate au de-a face cu poligoane simple avand unghiuri multiple de 45 grade.
- Programul trebuie sa suporte proiectarea ierarhica. Proiectarea ierarhica permite managementul unor proiecte mari, cat si incurajarea reutilizarii unor module proiectate anterior.
- In vederea proiectarii blocurilor, trebuie sa existe suport pentru diferite niveluri de abstractizare. Exista numeroase moduri pentru exprimarea informatiei privind proiectarea. Pentru geometria mastilor se folosesc programe de vizualizare a mastilor/desenelor, iar pentru informatia de conectivitate se utilizeaza listele de componente/conexiuni (net list) si schemele. Gestiunea diferitelor module de vizualizare este esentiala.
- Trebuie intretinuta informatia referitoare la conectivitate, cat si la formele geometrice. Este mult mai convenabil a avea de-a face cu dispozitive de tipul tranzistoarelor si a structurilor, cum ar fi contactele, ca unitati recunoscute de catre program, decat cu reprezentari ale dispozitivelor sub forma de poligoane.
- Comenzile pentru editarea desenelor/mastilor trebuie sa fie capabile sa deplaseze obiecte, mentinand in acelasi timp conectivitatea. De exemplu, atunci cand este deplasat un tranzistor, toate conexiunile sale trebuie sa se deplaseze odata cu el.
- Programul trebuie sa ofere flexibilitate in ceea ce priveste interfata cu alte unelte de proiectare. Un editor de desene/masti, ca atare, nu prezinta o importanta prea mare. Verificatoarele de reguli de proiectare, extractoarele de circuit, simulatoarele si alte asemenea unelte, folosite in procesul de proiectare, trebuie sa fie cat mai strans legate de programul pentru desenarea mastilor.
- Programele trebuie sa fie independente de tehnologie. Progresele rapide in fabricarea circuitelor integrate vor face ca un program legat de un proces dat sa fie depasit inainte de a fi terminat. De asemenea, cei mai multi proiectanti au la dispozitie mai multe procese, iar invatarea unui nou program pentru fiecare ar fi nerealista.
- Vor fi permise celule parametrizate. Trebuie sa fie posibila crearea unor celule ale caror proprietati sa poata fi ajustate prin modificarea parametrilor externi. De exemplu, crearea unui inversor cu latimi ale tranzistoarelor ce pot fi specificate extern.
- Trebuie sa fie permise ambele metodologii de proiectare: "top-down" si "bottom-up". Uneltele nu trebuie sa-l forteze pe proiectant sa foloseasca o anume metodologie; ele trebuie sa fie destul de flexible pentru a permite utilizarea unui numar mare de tehnologii.

In multe cazuri un proiectant de ASIC nu va folosi editorul de masti. El va fi preocupat de mai mult de descrierea schemei, de un limbaj de descriere hardware sau de lista componentelor/conexiunilor. Uneltele pentru desenare, pe de alta parte, sunt esentiale daca se urmareste realizarea unor produse de varf, cum ar fi memorile si procesoarele. De asemenea, daca proiectul are in vedere viteza de lucru ridicata sau daca este vorba de circuite analogice, proiectantul va fi preocupat de detaliile desenelor mastilor. Planurile mastilor afecteaza atat suprafata ocupata, cat si performantele. In cazul circuitelor de foarte mare viteza, diafonia si cuplajele reprezinta probleme legate de desenele mastilor.

7.1.2. Descrierea schemei.

In cazul existentei unor unele sofisticate pentru generarea mastilor, proiectantul nu va mai fi preocupat de geometria acestora.

Descrierea la nivel de schema poate fi utilizata pentru a introduce informatia de conectivitate din cadrul proiectului. Proiectantul realizeaza descrierea circuitului prin plasarea simbolurilor in proiect si prin conectarea corespunzatoare cu fire a porturilor simbolurilor date. In mod curent simbolurile utilizate in cadrul schemelor sunt macrocelulele de mari dimensiuni, porti din biblioteca de celule standard sau tranzistoare disponibile in tehnologia data. In figura care urmeaza se prezinta proiectul schemei unui tronson de Unitate Aritmetica Logica (UAL).



Un proiect ierarhizat poate fi descris prin crearea simbolurilor de reprezentare a blocurilor de nivel mai coborat si conectarea lor ca si in cazul portilor.

Dupa introducerea schemei, informatia de conectivitate intre componente proiectului este extrasă prin urmarirea firelor la porturile simbolurilor. In aceasta etapa pot fi detectate erorile simple de proiectare, cum ar fi retelele neconectate. Descrierea la nivel de schema permite proiectantului sa extraga liste de componente/conexiuni fara sa creeze mastile. Aceasta ofera posibilitatea simularii proiectului inca in faza de proiectare. Proiectul poate fi extins pentru a include intarzieri generice, iar dupa plasare si rutare, parametrii privind intarzierea pot fi din nou introdusi in schema, pentru o simulare mai detaliata.

7.1.3. Limbaje de Descriere a Hardware-lui (LDH - HDL).

In locul descrierii informatiei privitoare la proiect folosind sisteme grafice interactive asociate cu editoare de masti si de scheme, adesea este mult mai convenabil sa se introduca informatia sub forma de text. LDH sunt utilizate pentru a descrie informatia referitoare la proiect in cazul sistemelor mari, pe baza unor constructii mult mai abstracte decat cele permise in cadrul descrierilor la nivel de scheme. LDH vor fi descrise mai tarziu in cadrul acestui capitol.

7.1.4. Bazele de date.

Oricare dintre programele folosite pentru introducerea proiectelor utilizeaza o baza de date pentru stocarea acestora. In mod ideal aceasta baza de date trebuie sa fie accesibila tuturor uneltelor de proiectare. Este mult mai convenabil sa se dispuna de o singura baza de date partajata de catre toate programele, decat sa existe o baza de date pentru fiecare program. In cazul de fata se recomanda ca proprietati:

- Baza de date trebuie sa utilizeze cat mai mult posibil sistemul de fisiere al sistemului de operare. In loc de a stoca toate proiectele intr-un singur fisier mare, care reprezinta biblioteca, fiecare proiect trebuie sa fie salvat intr-un fisier separat, biblioteca fiind, directorul care contine aceste

fisiere. Aceasta permite un management mai facil si mai consistent al bibliotecii de proiectare. Utilitarele sistemului de operare pot fi folosite pentru a gestiona baza de date.

- Baza de date poate stoca informatie in format binar sau text. Formatul binar este mai compact, iar cel de tip text este mai flexibil.

7.2. Unelte de desenare.

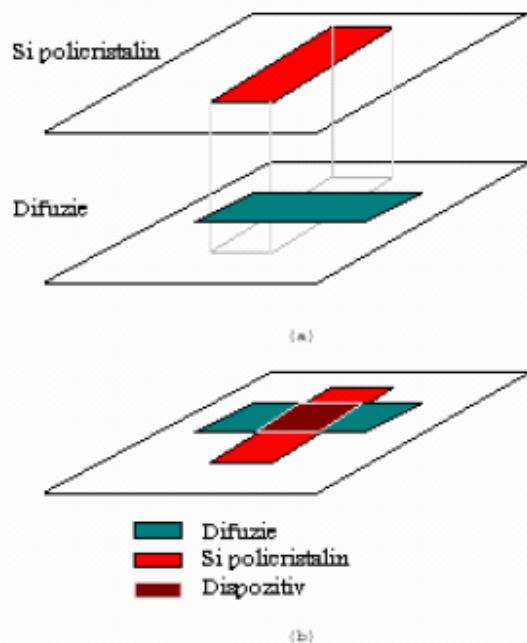
Programele pentru editarea desenelor mastilor sunt de cea mai mare importanta in proiectarea circuitelor integrate. Chiar daca se folosesc pentru descriere si introducere scheme sau LDH-uri, un editor de masti ramane inca necesar. In continuare se vor prezenta unele tehnici folosite pentru gestiunea geometriilor desenelor. De asemenea, se vor prezenta unele dintre programele existente pentru proiectarea circuitelor integrate. Exista, in acest sens, o multime de unelte disponibile, produse de catre Mentor Graphics si de catre Cadence. Capabilitatile unor asemenea unelte vor fi prezentate pe scurt.

7.2.1. Reprezentarile planurilor mastilor.

7.2.1.1. Acoperirea.

Una dintre cele mai simple metode pentru reprezentarea geometriei planurilor se bazeaza pe acoperire.

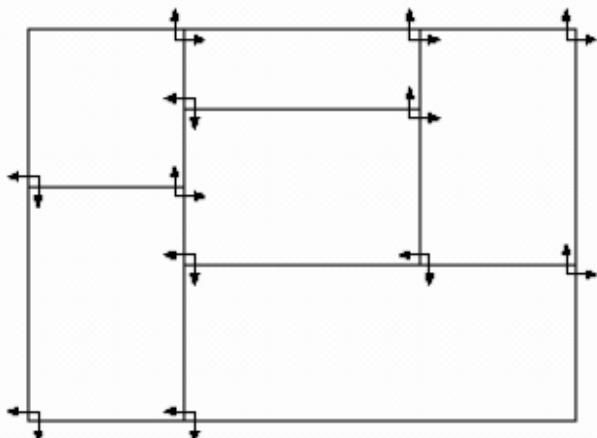
Mastile unui circuit integrat sunt reprezentate ca un set de planuri, fiecare fiind acoperit cu forme care reprezinta geometria mastilor. In loc de a stoca masca prezenta in fiecare punct al proiectului, va fi pastrata o lista de forme care pot acoperi portiuni mari ale structurii.



O baza de date va stoca toate aceste acoperiri permitand executia unor programe de verificare a regulilor de proiectare si de generare a informatiilor privind mastile.

7.2.1.2 Alaturarea colturilor.

Lipirea colturilor reprezinta un algoritm de acoperire, care permite existenta intr-un plan numai a formelor care nu se suprapun. Spatiile libere sunt acoperite cu forme care nu reprezinta masti. Fiecare forma indica alte patru forme, dupa cum se vede din figura.



Cu toate ca aceasta metoda necesita mai mult spatiu de memorie decat alte tehnici, exista o serie de algoritmi eficienti pentru gasirea formelor vecine. Trebuie sa fie posibila, de exemplu, operatia de tip "rasturnare" (plow), ca in sistemul Magic.

Din punctul de vedere al proiectantului, reprezentarea interna a geometriei desenului nu este importanta. Este important sa se realizeze ca, acesta informatie trebuie stocata intr-o maniera care sa permita algoritmilor o eficiente manipulare a datelor.

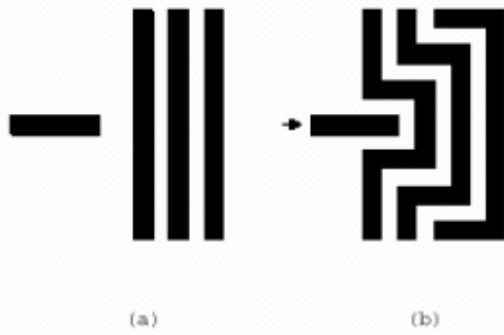
Electric este un program pentru proiectarea circuitelor integrate, disponibil pentru universitatii, si comercializat sub numele “*Electric*”, de catre Electric Editors. In cadrul acestui pachet software, circuitele sunt reprezentate sub forma unei colectii de arce si noduri. Nodurile pot fi noduri simple predefinite, pentru elemente comune de circuit cum ar fi tranzistoarele, contactele, capacitorii, sau noduri mai complexe, care reprezinta celule create de catre utilizator. Fiecare nod poseda porturi care definesc directiile de acces extern, cat si straturile. Arcele sunt folosite pentru conectarea impreuna a nodurilor.

Una din caracteristicile deosebite ale pachetului *Electric* a constat in existenta facilitatii de verificare automata a regulilor de proiectare, pe masura elaborarii proiectului, ceea ce permite corectarea rapida a eventualelor erori.

7.2.2. Magic.

Magic este un program de desenare a mastilor de circuite integrate dezvoltat la Universitatea Berkeley, California. El are in cea mai mare parte caracteristici similare cu *Electric*, dar prezinta si unele particularitati. In continuare se prezinta cateva din caracteristicile pachetului *Magic*.

- *Magic* poseda un Verificator de Reguli de Proiectare (VRP) incremental.
- *Magic* dispune de un “rutor” de canal. Prin introducerea informatiei de conectivitate in fisier, *Magic* poate conecta proiectul dupa amplasarea celulelor.
- *Magic* permite operatia de “rasturnare”, exemplificata mai jos, care asigura permite compactarea si expandarea proiectelor.



(a)

(b)

7.2.4. Cadence.

Cadence reprezinta un pachet industrial puternic, pentru proiectarea circuitelor integrate. *Cadence* ofera o gama larga de unelte de proiectare, bazate pe o platforma comună. Aceste unelte acopera spectrul necesitatilor de proiectare cum ar fi: *amplasarea si rutarea, simularea, modelarea, sinteza si verificarea regulilor de proiectare*.

Spre deosebire de *Electric* si *Magic*, *Cadence* nu furnizeaza codul sursa pentru sistem. In schimb, platforma *Cadence* poseda un limbaj de programare denumit *SKILL*, care este similar cu *C* si *LISP*. Cu ajutorul acestui limbaj, capabilitatile sistemului pot fi extinse fara a mai fi necesara modificarea codului sursa pentru unelte. In mod similar modulul, care genereaza lista componentelor/conexiunilor este suficient de flexibil pentru a permite utilizatorului sa-si adauge interfetele proprii.

- *Cadence* nu este comercializat ca un singur program, ci ca o colectie de unelte, care pot fi adaugate la un sistem de proiectare de baza, la un nucleu.
- *Cadence* dispune de capabilitatile de *Amplasare si Rutare* pentru sinteza automata a planurilor mastilor, plecand de la scheme.
- Spre deosebire de *Electric* si *Magic*, *Cadence* nu permite operarea simultana a diferitelor module/unelte. De exemplu, nu este disponibil un VRP incremental.
- Ca optiuni, *Cadence* include: *descrierea simbolica a mastilor, compactarea, VRP, extragerea circuitului din masti, unelte pentru circuite analogice, generatoare de module*.

7.3. Unelte pentru generarea desenelor mastilor.

Pentru proiecte mici, proiectantul va dori sa elaboreze manual desenele mastilor. Pentru proiectele mari, cea mai mare parte a activitatilor poate fi automatizata, folosind diverse unelte. Pe langa reducerea timpului de proiectare, Uneltele de Proiectarea Automata (UPA) sunt mai putin afectate de de proiectare. Uneltele de generare nu sunt tot atat de bune ca si expertii umani. In orice caz timpul de proiectare poate fi redus pe seama dimensiunilor si a vitezei de lucru. In aceasta sectiune vor fi pe scurt, cativa algoritmi pentru proiectarea automata a mastilor.

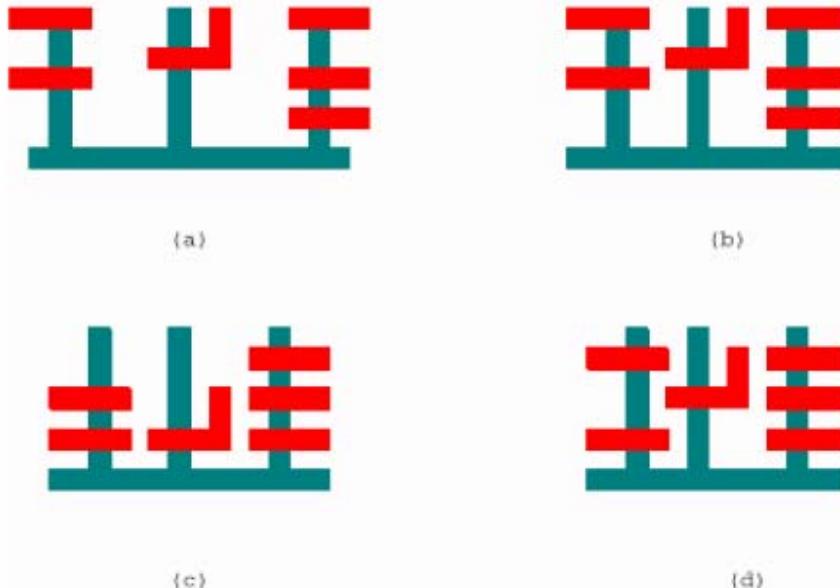
7.3.1. Unelte de compactare.

Compactarea desenelor mastilor este importanta nu numai pentru cresterea densitatii circuitelor, ci si pentru reducerea capacitatilor parazite, asociate cu firele lungi. Desi compactarea se poate realiza pe intregul circuit, complexitatea problemei limiteaza compactarea numai la celule de dimensiuni reduse.

Uneltele de compactare sunt intens utilizate in conjunctie cu programele de generare a celulelor, care efectueaza sinteza desenelor mastilor acestora. Aceste programe au tendinta de a crea desene de masti functionale, dar nu in mod necesar compacte.

Compactarea unidimensiunala este cea mai simpla forma de algoritm de compactare. Mai intai desenul este compactat in directia x prin deplasarea orizontala a obiectelor pana la obtinerea separatiei minime.

Apoi desenul este compactat in directia y . Compactarea in directile x si y este repetata pana cand nu mai este posibila optimizarea. In figura de mai jos este dat un exemplu de compactare unidimensională.



Desenul original este prezentat in figura (a). Dupa prima compactare pe directia x , se obtine rezultatul din figura (b). Rezultatul compactarii dupa directia y este dat in figura (c). In figura (c) se prezinta rezultatul compactarii optime, care nu se poate obtine ca rezultat al compactarii dupa o singura dimensiune.

Compactarea bidimensională incearca sa efectueze compactarea simultan in directiile x si y . Este evident ca aceasta reprezinta o sarcina mult mai complexa necesitand, in general, euristici si algoritmi de Inteligenta Artificiala. (IA). De asemenea, au fost utilizati algoritmi nedeterministi, cum ar fi *calirea simulata* sau *algoritmi genetici*.

*Calirea simulata** efectueaza optimizarea prin deplasarea aleatoare a formelor geometrice. Daca o deplasare imbunatatestea proiectul ea va fi pastrata. Daca o deplasare degradeaza proiectul, modificarea este pastrata cu o probabilitate dependenta de degradare si de temperatura. Aceasta permite algoritmului sa se elibereze de minimele locale in cadrul optimizarii. Temperatura este scazuta pe masura desfasurarii algoritmului.

*Algoritmi genetici*** pornesc de la un ansamblu de solutii posibile, ansamblul genelor. Cele mai bune gene sunt copiate pentru ansamblul reuniune. Sunt selectate aleator perechi de gene, care isi schimba parti ale solutiilor lor. In acest moment se poate introduce o mica mutatie. Aceste gene reprezinta acum generatia curenta a ansamblului de gene. Copierea probabilista, reunirea si mutatia sunt continuante pana la atingerea unei solutii rezonabile. Ca si in cazul *calirii simulate* algoritmul este simplu desi scrierea programului, cat si obtinerea unei solutii acceptabile pot fi dificile.

In continuare se prezinta o serie de aspecte referitoare la compactare.

- Compactarea este dependenta de planul initial. Algoritmi de compactare nu vor pleca de la un plan oarecare pentru a produce planuri bune pentru masti. Este necesar un plan initial rezonabil.
- Insertia unor deviatii in firele de conectare poate imbunatati compactare. Aceasta creste numarul formelor geometrice in planul mastilor.

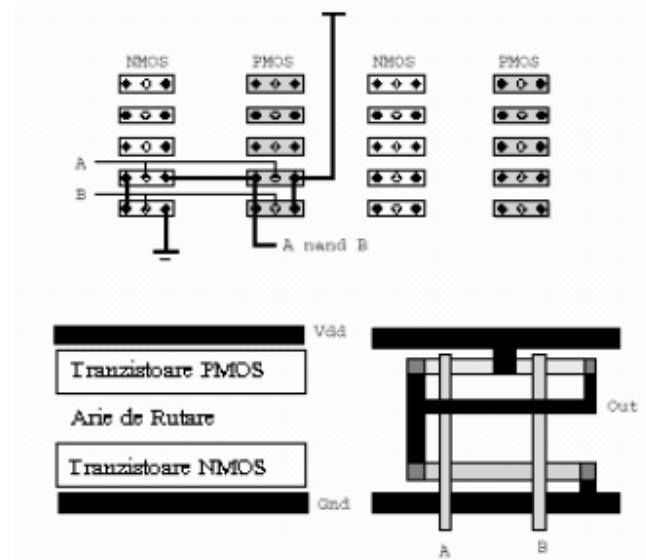
- Interschimbul de componente in locul deplasarii lor poate conduce la imbunatatirea compactarii.
- Trebuie sa se aibe grija sa nu se inlature spatiile introduse pentru motive bine determinate. De exemplu, pentru a imbunatati fiabilitatea, un proiect doreste sa pastreze anumite dispozitive separate printr-o distanta mai mare decat distanta minima.
- In proiectarea ierarhica, deplasarea unui obiect poate afecta obiectele de la diferitele niveluri ale ierarhiei.
- Compactarea *non-mahhattan* introduce mai multe posibilitati de deplasare, care trebuie avute in vedere.
- Compactarea nu trebuie sa modifice separarile, care sunt esentiale pentru proiect. De exemplu, separarea traseelor/sinelor, in celulele standard, si a locatiilor porturilor, in celulele, care urmeaza sa fie suprapuse, trebuie sa fie conectate la locatii specifice.

7.3.2. Generarea celulelor.

Adesea este important sa se dispuna de mijloace automate pentru sinteza planurilor celulelor.

In continuare se vor prezenta diferite tehnici pentru generarea celulelor.

Marea de porti utilizeaza oarie de tranzistoare la care se adauga conexiuni pentru a realiza functiile logice dorite. Aceasta este o idee simpla dar nu foarte eficienta in ceea ce priveste aria ocupata. Pentru crearea planurilor mastilor portilor poate fi utilizata o tehnica de sinteza de tipul *amplaseaza si ruteaza*. Aceasta este asemanatoare cu tehnica amplaseaza si ruteaza ce se va prezenta in sectiunea urmatoare, cu diferența ca, in loc de porti, aici sunt amplasate tranzistoare. Tipic portile vor avea o structura similara. De exemplu, cele mai multe generatoare de celule CMOS vor folosi o structura de tipul celei de mai jos.



Tranzistoarele PMOS sunt plasate pe o singura linie sub traseul/sina de alimentare, iar tranzistoarele NMOS se gasesc pe o singura linie deasupra traseului de masa. Aria cuprinsa intre tranzistoarele PMOS si NMOS este utilizata pentru rutarea conexiunilor la porti ale tranzistoarelor complementare. Pentru structurile regulate de tipul registratorilor, tampoanelor/buffer-elor si numaratoarelor se folosesc generatoare de porti speciale. Aceasta tehnica este cel mai des utilizata pentru a realiza componente de inalta performanta cu structura repetitiva. Pentru fiecare poarta este necesar un generator unic. Generatorul poate dispune de posibilitati de scalare a tranzistoarelor, cat si de posibilitati privind optimizarea parametrilor.

In scopul reducerii ariei ocupate de planul mastilor, dupa sinteza acestora, se poate efectua operatia de compactare.

7.3.3. Minimizarea logica.

Tehnicile de minimizare logica au aparut inainte de proiectarea circuitelor integrate. Pentru gasirea implicantilor primi ai expresiei logice ce descrie un circuit exista numerosi algoritmi, cel mai notabil fiind algoritmul Quine-McCluskey. In cazul proiectantului de circuite integrate este important sa se cunoasca restrictiile privind minimizarea logica.,

- Algoritmul Quine-Mc Cluskey optimizeaza numai functiile cu o singura iesire.
- Un circuit combinational reprezentat ca o implementarea a unei functii cu numar minim de implicanti primi nu furnizeaza in mod necesar si cel mai rapid circuit. Cand se are in vedere viteza, trebuie sa se considere si posibilitatile de comanda ale circuitului.
- Functiile minizmate nu conduc intotdeauna la cea mai mica arie pentru structura ca in cazul PLA-urilor.
- Circuitele minizmate pot fi mai dificil de testat sau mai putin tolerante la defecte. Pentru a creste fiabilitatea se poate introduce in mod intentionat redundanta.

7.3.4. Compilarea Ariilor de Porti.

Uneltele de proiectarea folosite la Ariile de Porti sunt similare cu cele utilizate pentru proiectarea circuitelor la cerere (ASIC). Compilarea ariilor de porti este in principal o operatie de rutat, care va fi discutata in sectiunea urmatoare.

7.4. Amplasarea si rutarea.

Cele mai frecvent utilizate unelte de sinteza sunt *amplasarea si rutarea*. Acestea sunt folosite in special in cadrul metodelor de proiectare cu celule standard, unde specificatiile de proiectare sunt exprimate ca un set de porti conectate impreuna. Fiecare poarta corespunde, de regula, unei operatii logice combinationale sau unui sistem secvential de tipul unui bistabil. Planurile mastilor portilor sunt disponibile in cadrul unei biblioteci.

Uneltele de amplasare si rutare vor amplasa mai intai celulele in cadrul proiectului si apoi vor efectua conexiunile de rutare. Uneltele de amplasare si rutare trebuie sa functioneze impreuna, iar selectarea algoritmilor de amplasare si rutare nu se efectueaza independent, astfel, aceste unelte vor fi examinate simultan.

7.4.1. Algoritmi de amplasare.

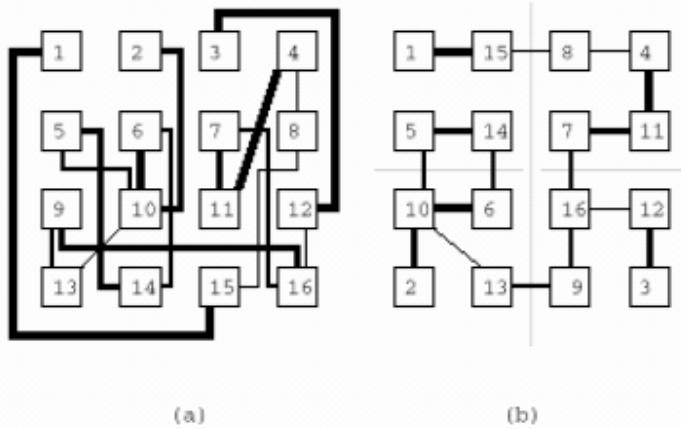
Algoritmii de amplasare sunt cel mai dificil de realizat intr-o maniera eficienta. Scopul uneltelor de amplasare, denumite si “floor planners”, este acela de a amplasa blocurile proiectului astfel incat uneltele de rutare sa permita conectarea corespunzatoare a porturilor acestora. Aceasta problema este NP-completa (complexitatea creste aici exponential cu dimensiunea problemei). Ca rezultat nici unul dintre algoritmii utilizati in mod comun nu sunt optimali. Proiectantul, in nici un caz, nu este interesat de un proiect optimal, ci de un proiect care va satisface specificatiile de proiectare. In cazul unui plan de masti restrictiile se pot referi la o arie ocupata data si la satisfacerea anumitor limitari de timp.

7.4.1.1. Algoritmul bazat pe tajetura minima (min-cut).

Unul dintre cei mai simpli algoritmi folositi pentru amplasare este cel bazat pe *tajetura minima*. Se determina conectivitatea proiectului iar circuitul este partitionat recursiv in jumatati egale. Fiecare partitie este astfel realizata incat un numar minim de conexiuni sa traverseze partitia. Aceasta tehnica de “divizare si cucerire” poate genera rezultate rezonabile. In figura de mai jos este prezentat un exemplu de algoritm bazat pe *tajetura minima*. In figura (a) se prezinta blocurile proiectului. Numarul de conexiuni intre

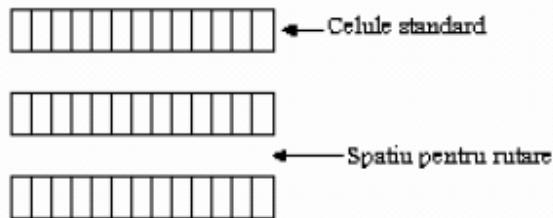
blocuri este reprezentat de grosimea liniei de conectare. Figura (b) arata plasamentul final cu liniile de taiere. Mai intai este taiata linia orizontala, iar apoi cele doua linii verticale.

Un astfel de algoritm permite ca amplasarea proiectului sa puna accent pe reducerea lungimii totale a firelor. Aceasta conduce la un proiect de performanta ridicata, care ocupa o arie redusa.



4.2.1.2. Amplasarea liniilor in cazul celulelor standard.

In cazul utilizarii metodologiei celulelor standard, amplasarea celulelor poate fi simplificata de cunoasterea structurii celulei. In mod tipic celulele standard sunt create cu trasee/sine de alimentare si masa la partile superioara si inferioara ale celulei. Celulele pot fi adiacente fara a conduce la violarea regulilor de proiectare (figura de mai jos). Amplasarea celulelor standard poate fi realizata prin amplasarea celulelor in linii, separate prin canale utilizate pentru rutare. Pentru a determina pozitia celulei in cadrul liniei trebuie folositi algoritmi bazati pe tietura minima sau pe alte tehnici. Liniile de celule mai contin si celule speciale, care permit conectivitatea intre liniile de celule.



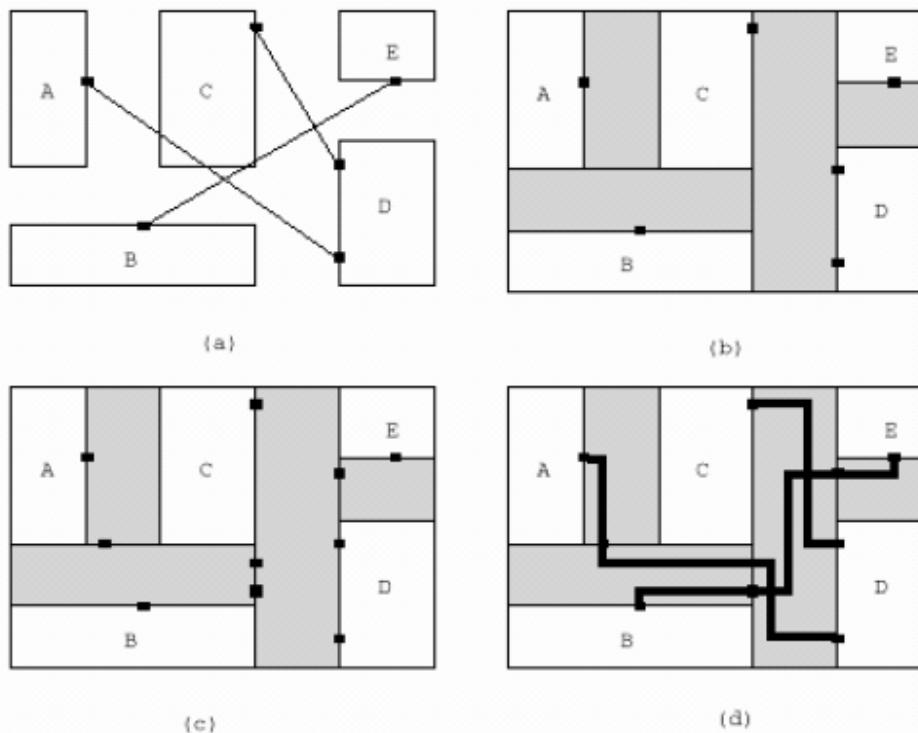
7.4.1.3. Calirea simulata.

Complexitatea algoritmilor de amplasare face ca algoritmul de calire simulata sa devina extrem de atractiv. Acesta va solicita insa mult timp de unitate centrala. Calirea simulata este algoritmul de baza care modeleaza calirea/cresterea rezistentei unui lichid sau solid. Calirea simulata este foarte atractiva si pentru faptul ca este intuitiva si usor de implementat.

7.4.2. Algoritmi de rutare.

Algoritmii de rutare sunt responsabili pentru conectarea porturilor in cadrul unui proiect in care a avut loc deja amplasarea.

Procesul de rutare este prezentat in figura de mai jos. Cele trei etape ale procesului de rutare sunt urmatoarele.



1. *Generarea canalului.* Se identifica regiunile prin care vor fi ruteate/directionate conexiunile. În mod tipic canalele vor fi regiuni rectangulare. Regiunile nerectangulare vor fi partitionate în canale rectangulare (fig. b).

2. *Rutarea globală.* Dacă trebuie rutată o rețea între două canale, la frontieră între cele două canale se va crea un port. În această etapă se determină canalele prin care se rutează o rețea, dar nu se efectuează conexiunile (fig.c).

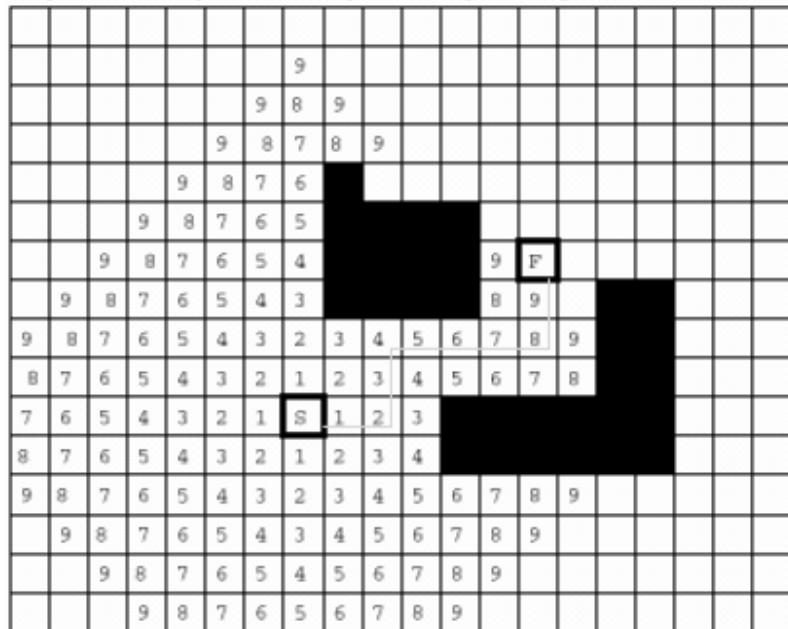
3. *Rutarea canalului.* Sunt ruteate canalele reale, adică sunt amplasate firele care efectuează conexiunile (fig.d).

Rutarea reprezintă o procedură comună în toate tipurile de ASIC-uri, pornind de la cele solicitate de către utilizator, la celulele standard, la ariile de porti și la dispozitivele programabile de către utilizator.

Că și amplasarea, rutarea reprezintă o problema dificilă. Intrucât cele mai multe rutoare nu oferă soluții optimale este important ca retelele să fie tratate pe baza de priorități. Astfel, retelele cu caracter critic vor fi ruteate mai eficient. În continuare vor fi prezentati pe scurt unii algoritmi de rutare.

7.4.2.1. Algoritmul de rutare Lee-Moore.

Algoritmul Lee-Moore rutează câte un fir la un moment dat, fiecare conexiune efectuându-se pe drumul cel mai scurt. Se realizează un tablou de celule ca în figura de mai jos.



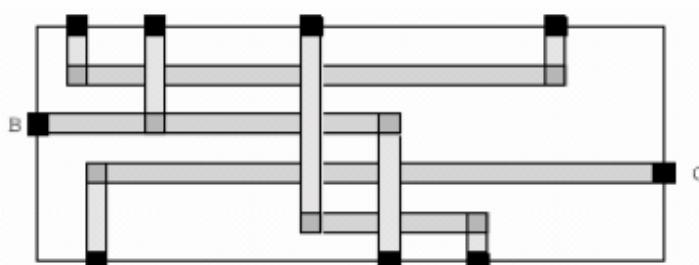
Punctele de plecare si oprire sunt notate cu S si F. Celulei de plecare din tablou i se atribuie valoarea zero. Celulelor adiacente acesteia li se atribuie valoarea unu. De fiecare data cand unei celule i se atribuie o valoare, valoarea incrementata se transfera tuturor celulelor vecine carora nu li s-a atribuit in ca o valoare.

Algoritmul se termina atunci cand se atribuie o valoare celulei finale. Oricare drum de la punctul final la punctul de start, care urmeaza o cale de numere in secventa reprezinta drumul cel mai scurt intre porturi. Dupa asignarea unui drum, seincearca conexiunea urmatoare.

7.4.2.2. Rutarea Canalelor.

Adesea este mai convenabil si eficient sa se ruteze simultan mai multe linii. Acesta este principiul pe care se bazeaza rutorul de canal. Un rutor de canal plaseaza fire paralele in directia principala a canalului.

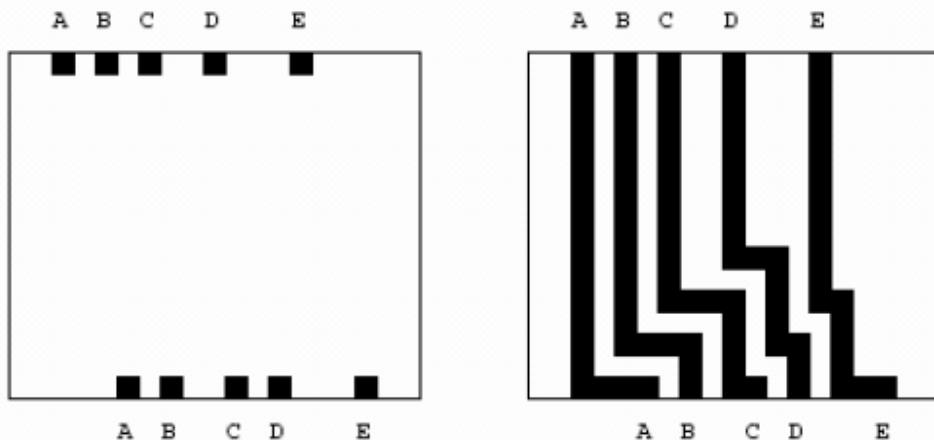
In canal, liniile orizontale sunt rute orizontal, iar liniile verticale sunt rute vertical. In figura de mai jos se prezinta un canal rutat orizontal.



Fiecare retea are o linie orizontala, care conecteaza toate porturile in lungul canalului. Unul din dezavantajele rutarii canalelor consta in aceea ca un semnal fiind rutat prin canale multiple va trebui sa schimbe straturile frecvent. Astfel, procesul de fabricatie va trebui sa asigure mai multe straturi de metal.

7.4.2.3. Rutarea de tip rau (river).

In cazurile in care se stie ca firele nu se vor intersecta, se va putea folosi un algoritm de rutare de tip rau eficient. In figura de mai jos, porturile sunt conectate prin selectarea portului din stanga si rutarea in jos, iar apoi direct la dreapta, catre alt port.



Urmatorul fir este inserat intr-o maniera similara, cu exceptia faptului ca va fi plasat in apropierea firului precedent la distanta minima de separare. Procedand in acelasi mod cu toate firele rutarea poate fi obtinuta la lungime minima . Aceasta tehnica nu se poate aplica in cazul in care firele se intersecteaza.

7.5. Tehnici de verificare.

In aceasta sectiune se vor discuta uneltele mai importante in procesul de verificare a proiectelor. Se vor avea in vedere *Verificatoarele de Reguli de Proiectare* (VRP), *Uneltele de Extractie* si *Verificatoarele de Reguli Electrice*.

7.5.1. Uneltele VRP.

Verificarea Regulilor de Proiectare reprezinta un proces prin care se examineaza daca planul mastilor corespunde restrictiilor impuse de catre procesul de fabricatie. Regulile de proiectare sunt specificate ca un set de restrictii ce guverneaza separarile si conectivitatea in cadrul proiectului. O violare a regulilor de proiectare reprezinta o geometrie, care nu satisface regulile date.

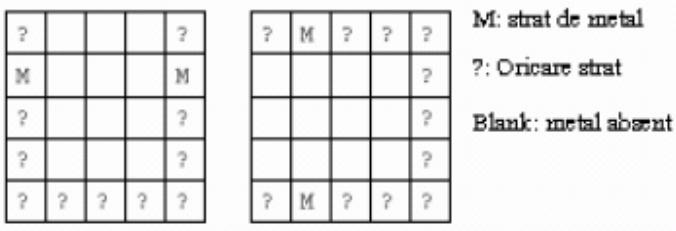
In proiectele abordate ierarhic, VRP ierarhica joaca un rol important. Implementarea ierarhica a unui program VRP poate conduce la o importanta economie de timp. Tipic, aceasta implica verificarea completa a celulelor de pe cel mai coborat nivel si apoi efectuarea verificarii nivelului superior numai in ceea ce priveste periferia celulelor. Doua celule plasate alaturat trebuie sa fie verificate numai in ceea ce priveste cea mai mare distanta de la marginea celulei.

Pentru ca un program VRP sa identifice corect violarea regulilor, el trebuie sa poata determina informatia de conectivitate. In plus, fata de regulile de proiectare, trebuie sa se mai dea un set de reguli de conectivitate. Astfel, un exemplu tipic este acela al regulilor privind straturile *metal1* si *metal2*, care sunt conectate prin *via*. Cand se detecteaza *via* intre doua staturi se presupune ca ele trebuie sa fie conectate.

7.5.1.1. Tehnici de tip rastru.

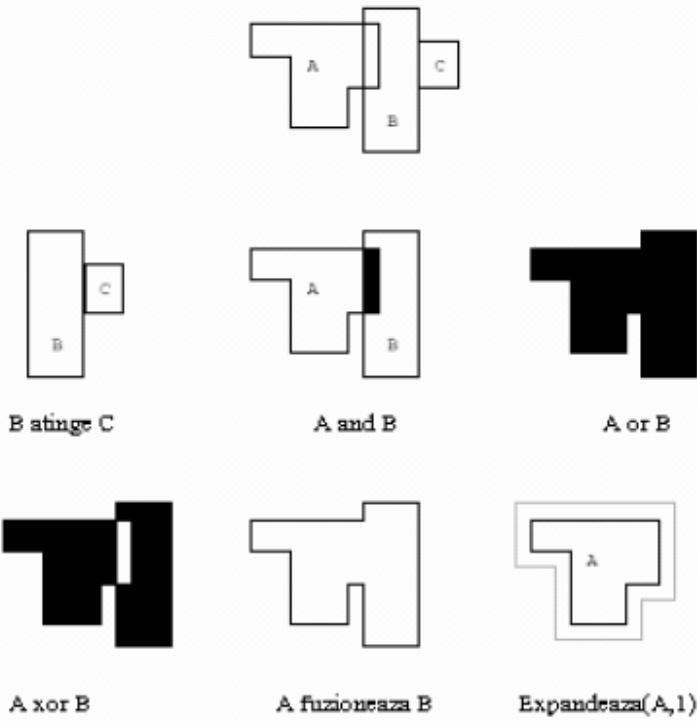
Cea mai simpla tehnica, pentru determinarea violarilor de reguli simple de proiectare, se bazeaza pe utilizarea unei ferestre rastru, care trece peste planul mastilor pentru a stabili eventualele violari ale regulilor simple de proiectare. In figura de mai jos se prezinta ferestrele care trebuie trecute peste planul mastilor pentru a determina separarile straturilor de metal mai mici de 3 μm , cu toate

geometriile pe limite de 1 μm . Daca fereastra se potriveste cu forma de sub ea, s-a gasit o violare a regulilor de proiectare. Tehnica are avantajul ca este relativ simpla, dar ea nu poate sa foloseasca informatia de conectivitate globala si poate raporta violari inexistente. Daca doua linii de metal conectate sunt separate la mai putin de 3 μm se raporteaza o violarea a regulilor de proiectare. Geometriile non-mahattan nu pot fi verificate corect cu aceasta tehnica.



7.5.1.2. Tehnici bazate pe algebra poligoanelor.

Tehnicile bazate pe algebra poligoanelor folosesc sase operatii: *and*, *or*, *xor*, *atinge*, *fuzioneaza*, *expandeaza*. Cele sase operatii sunt prezентate în figura următoare.



O regula de proiectarea, pentru o separare intre traseele de metal de 3 μm , se poate exprima astfel:

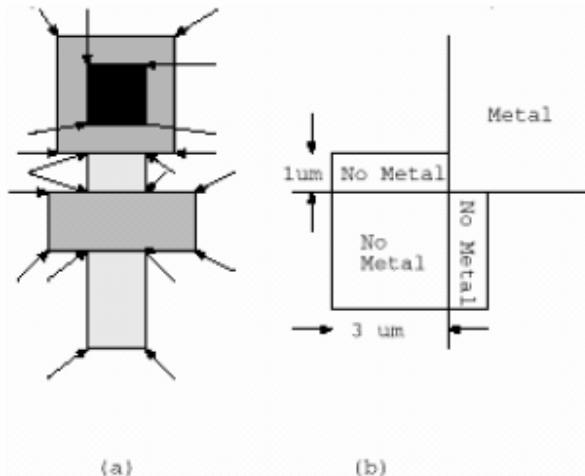
$$EROARE = \text{expandeaza}(\text{poligon}, 1.5\mu\text{m}) \text{ and } \text{expandeaza}(\text{poligon2}, 1.5\mu\text{m})$$

Acste tehnici sunt computational intensive.

7.5.2. Tehnici bazate pe varfurile.

Ultima tehnica VRP utilizeaza varfurile din proiect. Varful este definit a fi locul in care se intalnesc doua muchii. Varfurile unui plan pentru masti sunt arataate in figura de mai jos (a). Fiecare regula de

VRP are un sablon asociat pe care il verifica la fiecare varf. Sablonul pentru spatierea de metal de 3 in (b).



Aceasta tehnica este in general aplicabila geometriilor Manhattan, dar poate fi extinsa si la geometriile non-Manhattan.

7.5.3. Unelte de Extractie.

Dupa generarea planului mastilor, este important sa se extraga din acesta parametri electrici. In mod special intereseaza identificarea urmatoarelor elemente ale planului: tranzistoare, contacte, capacitorsi si rezistente. Algoritmi utilizati pentru identificarea elementelor sunt foarte asemanatoari cu cei dintr-un program VRP. De fapt cele mai multe unelte de extractie implementeaza si functiile de VRP. Elementele de circuite CMOS sunt identificate dupa cum urmeaza:

- Tranzistoarele apar acolo unde un traseu de siliciu policristalin intersecteaza un traseu de difuzie. Trebuie sa se calculeze aria tranzistorului, latimea si lungimea trebuie calculate. Conexiunea la substrat trebuie inregistrata.
- Capacitatatile apar intre toate straturile. Atunci cand un strat intersecteaza alt strat, pentru calcularea capacitatiei se va folosi aria de intersectie. Aceasta implica calcularea arilor si a perimetrelor pentru straturile poligonale.
- Rezistorii apar pe toate liniile. Rezistenta depinde de lungimea si grosimea liniei. Trebuie avute in vedere si trsele curbate. Rezistenta este exprimata in unitati: Ω/patrat .

7.5.4. Verificatoare de Reguli de Natura Electrica (VRNE).

VRNE sunt programe care examineaza informatia din lista componentelor/conexiunilor extrasă din planul mastilor si verifica incalcarea diverselor reguli de natura electrica. In continuare se prezinta cateva dintre regulile verificate:

- *Scurt circuite*: Retelele care sunt conectate la masa sau la tensiunea de alimentare sunt semnalate ca fiind erori posibile.
- *Circuite izolate*: Retelele care nu sunt comandate de nici un semnal se considera a fi eronate.
- *Puterea consumata*: este posibila estimarea puterii consumate si verificarea curentilor in trasee.
- *Rapoartele tranzistoarelor*: Exista posibilitatea de a verifica daca rapoartele tranzistoarelor din circuit sunt rezonabile. Ca erori pot fi semnalate nepotrivirile intre dimensiunile tranzistoarelor NMOS si PMOS, in circuitele CMOS, tranzistoarele de dimensiuni mici, care comanda sarcini mari sau porti OR cablate.
- *Compararea retelelor*: Lista conexiunilor poate fi comparata cu schema originala a circuitului pentru a verifica daca a fost generat acelasi circuit. Acest lucru este dificil daca specificarea

circuitului original a fost structurala si daca pe parcursul sintezei planului mastilor au fost efectuate optimizari. De exemplu, daca au fost inlaturate portile redundante.

7.5.5. Verificarea.

Verificarea la nivel temporal. Circuitul este analizat pentru a gasi cazul cel mai defavorabil privind intarzierile. Daca acestea nu satisfac specificatiile circuitului se semnaleaza o eroare.

Verificarea functionala. Circuitul este analizat pentru a gasi portile logice de baza. Comportarea circuitului este dedusa din acestea. Se poate efectua, in continuare, compararea cu specificatiile de comportare ale circuitului original..

Amplasarea si rutarea bazate pe criterii de performanta se utilizeaza in cadrul uneltelor de proiectare a circuitelor integrate moderne. Daca se detecteaza o violare a criteriului de timp, in cadrul procesului de amplasare si rutare, se poate folosi o alta strategie de implementare. De exemplu, daca un inmultitor nu satisface criteriul de performanta se poate utiliza intr-o masura mai mare tehnica bazata pe banda de asamblare. Aceasta va conduce la oarie mai mare dar si la performante potențiale mai mari.

7.6. Unelte de simulare.

Simularea proiectelor reprezinta o etapa importanta in ciclul de proiectare a circuitelor integrate. Scopul simularii este acela de a determina comportarea proiectului inaintea fabricarii acestuia. In aceasta sectiune se vor examina simulatoarele folosite in proiectarea circuitelor integrate.

7.6.1. Simulatoare la nivelul circuitului.

Simulatoarele la nivelul circuitului sunt utilizate pentru a determina caracteristicile circuitului ca forme de unda. Uneltele de extractie sunt folosite pentru a produce lista conexiunilor ce contine componentele electrice ale circuitului, inclusiv capacitatatile parazite si rezistente. Simulatoarele preiau aceasta informatie si genereaza un sistem de ecuatii diferențiale neliniare. Solutia acestui sistem ofera analiza comportarii in regim tranzitoriu a circuitului.

7.6.1.1. Spice.

Cel mai raspandit program de simulare la nivel de circuit este SPICE. Initial acest program a fost dezvoltat in anii 70, la Berkeley, si a fost scris in FORTRAN si C. Codul sursa pentru SPICE este disponibil si a fost portat pe aproape toate tipurile de calculatoare. Sunt disponibile, de asemenea, si versiuni comerciale: HSPICE, PSPICE. In plus cei mai multi furnizori de circuite integrate ofera componente si lista lor de conexiuni pentru proiectele lor.

SPICE efectueaza analiza in CC prin tratarea capacitorilor ca circuite deschise, iar a inductorilor ca surcurcircuite.

Pentru noul circuit este creata o matrice, care este solutonata. Intrucat ecuatii sunt neliniare, se foloseste o tehnica iterativa cum ar fi metoda Newton-Raphson. Elementele de circuit sunt reprezentate in matrice sub forma:

$$Y_b V_b + Z_b I_b = W_b$$

unde:

- V_b este matricea tensiunilor pe ramuri;
- I_b este matricea curentilor pe ramuri;
- W_b este matricea conditiilor initiale.

De exemplu: un rezistor are $Y_b = 1$, $Z_b = -R_b$ si $W_b = 0$. Aceasta va furniza $V = IR$.

Un capacitor are $Y_b = sC_b$, $Z_b = -1$ si $W_b = C_bV_0$. Aceasta va genera ecuatiei capacitorului ca

$$sC_bV_b - I_b = C_bV_0.$$

Se stie din prima teorema a lui Kirchoff:

$$\begin{aligned} AI_b &= 0 \\ V_b - A'V_n &= 0 \end{aligned}$$

unde:

- A este matricea de incidenta $a_{ij} \in (-1, 0, 1)$

Matricea utilizata pentru a reprezenta elementele de circuit, adesea denumita tablou, este:

$$\begin{bmatrix} 1 & 0 & -A^T \\ Y_b & Z_b & 0 \\ 0 & A & 0 \end{bmatrix} \cdot \begin{bmatrix} V_b \\ I_b \\ V_n \end{bmatrix} = \begin{bmatrix} 0 \\ W_b \\ 0 \end{bmatrix}$$

Acet tablou, cand este solutionat, va determin tensiunile la toate nodurile, cat si tensiunile si curentii dea lungul tuturor ramurilor.

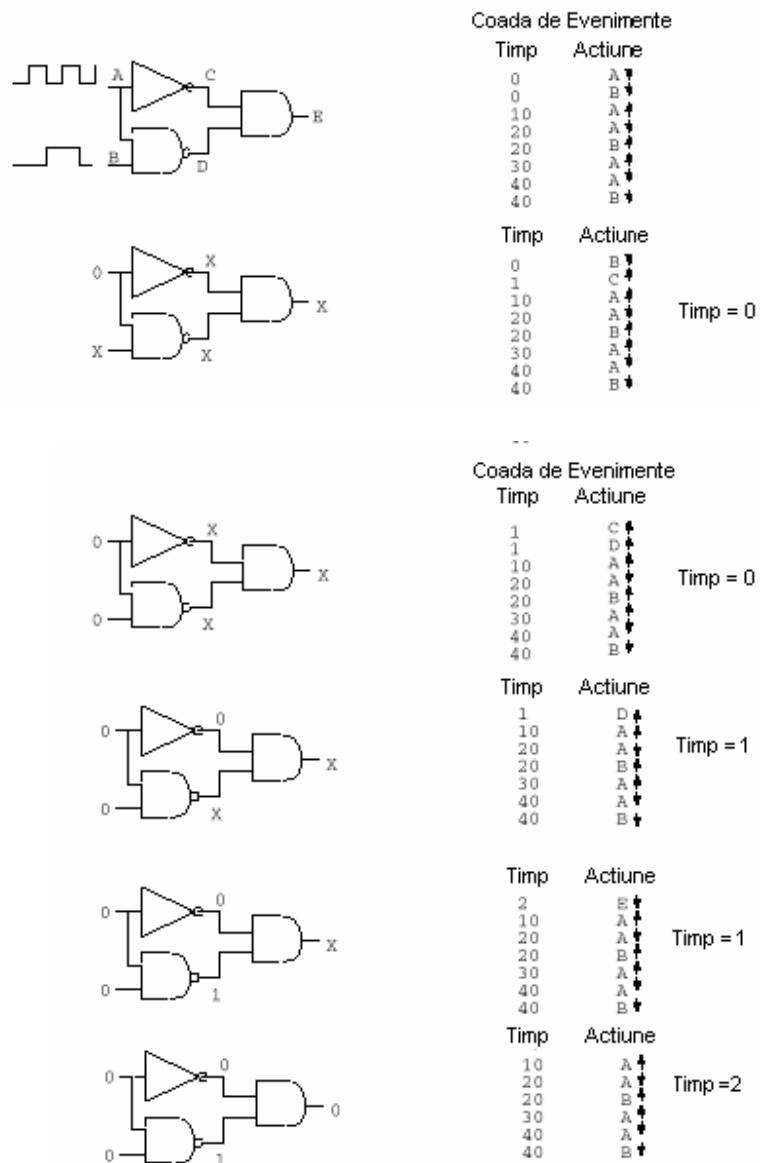
Analiza procesului tranzitoriu este efectuata dupa cum urmeaza:

- Daca nu sunt specificate conditiile initiale se efectueaza analiza in CC.
 - Se sparge intervalul de simulare in pasi de timp discret.
 - Se integreaza numeric ecuatii diferențiale in ecuatii algebrice echivalente. Pentru aceasta se poate folosi Metoda Euler inversa.
 - Se insereaza aceste noi ecuatii in matrice, care se soluzioneaza prin tehnici de analiza de CC.
 - Se repeat pana cand simularea se termina.
- Alte caracteristici ale pachetului nSPICE sunt:
- SPICE suporta analiza de CC. Analiza in regim tranzitoriu si analiza in CA, pentru semnale mici.
 - SPICE suporta tehnologii diferite: MOS, Bipolar etc.
 - Pentru circuite mari complexitatea analizei o face nerealista. Tipic sunt simulate portiuni ale proiectului.
 - Uneori SPICE nu poate ajunge la o solutie convergenta, iar pentru simulare trebuie efectuate ajustari ale parametrilor.
 - SPICE suporta modele diferite pentru tranzistoarele MOS.

7.6.2. Simulatoare Comandate de Evenimente.

In cazul in care intereseaza numai comportarea numerica a circuitului, nu mai este eficienta rezolvarea intregului circuit pentru a afisa starile, de fiecare data cand un nod isi schimba tensiunea. O alternativa se bazeaza pe simularea comandata de evenimente, in care numai partea de circuit afectata de schimbarea de stare este simulata.

Fie circuitul din figura de mai jos. Sunt prezentate starile initiale. Se urmareste simularea circuitului avand in vedere impulsurile aplicate la intrarea portii. Fiecare poarta are o intarziere de 1 ns, iar semnalele de intrare se modifica la intervale de 10 ns. Coada de evenimente pastreaza o lista de evenimente, care urmeaza sa apara la momente specificate de timp. Simularea circuitului, plecand de la continutul cozii de evenimente, este prezentata pentru primele 2 ns. Sunt necesare numai cinci evaluari ale portii (trei care au provocat evenimente ce au fost planificate si doua care nu au fost planificate). Pentru ca circuitul intreg sa fie recalculat pentru fiecare nod, sunt necesare noua evaluari ale portii.



7.6.3. Simulare la nivel de comutator.

In timp ce simularea circuitului ofera rezultate corecte la nivelul formelor de unda, adesea trebuie testate numai functionalitatea circuitului. Simulatoarele la nivel de comutator modeleaza tranzistoarele ca simple comutatoare, care transmit sau nu o valoare. Nivelurile de tensiune sunt modelate prin una dintre cele trei stari: 0-logic, 1-logic sau x-nedeterminat. Cu fiecare stare se asociaza puterea semnalului. Starile si puterile pentru un simulator tipic sunt aratate in laticea din figura (a) de mai jos. Putele au urmatoarele semnificatii:

D Comandat. Aceasta este puterea atribuita lui V_{DD} , V_{SS} si semnalelor de intrare.

Comandat presupune ca nodul este comandat de catre sursa de curent.

W Comandat slab. Aceasta este ca si cu ar fi comandat, dar comandat relativ slab.

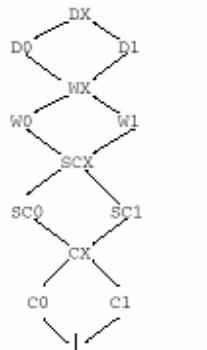
Un tranzistor trage-sus NMOS va comanda un nod catre o stare comanda slab.

SC Supraincarcat. Aceasta este puterea unui nod a carui valoare este stocata pe un capacitor, iar starea a fost asignata de puterea de comanda.

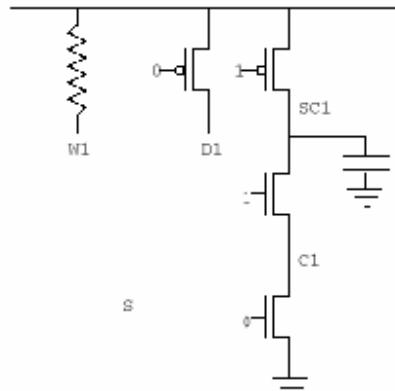
C Incarcat. Aceasta este puterea unui nod a carui valoare este stocata pe un capacitor, dar a carui putere nu a fost stabilita de catre un semnal de comanda.

\perp Nivelul cel mai coborat.

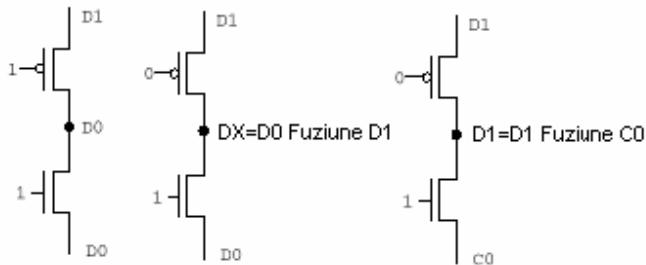
Figura (b) prezinta cateva circuite care ilustreaza cazuri in care pot fi intalnite diferite puteri.



(a)



(b)



In functie de precizia ceruta, simulatoarele pot folosi aceste stari si puteri sau un superset de subseturi ale acestora.

Starea si puterea ale unui nod sunt determinate prin gasirea celui mai coborat punct din latice, care are o cale in jos catre toate starile pe care le comanda. Fie circuitele din figura de mai sus. Primul circuit arata un inversor. Numai unul dintre tranzistoare conduce (NMOS) astfel iesirea capata starea D0. Urmatorul circuit are in conductie ambele tranzistoare NMOS si PMOS, astfel ca iesirea tinde spre D1 Fuziune D0 = DX, o valoare nedeterminata. Cel de-al treilea circuit are iesirea fortata la D1, intrucat D1 este mai puternic decat nodul incarcat C0.

Iata cateva dintre simulatoarele la nivel de comutator:

- SILOS reprezinta un simulator commercial care suporta 12 niveluri logice (Alimentare, Comanda, Rezistiva si Impedanta ridicata, pentru fiecare 0, 1 si X). SILOS permite simularea la nivel de tranzistoare si la nivel de porti. Pot fi utilizate, de asemenea, modele comportamentale.
- VHDL descrie circuitele intr-o maniera convenabila pentru simularea bazata pe evenimente discrete. Tipul construit in program BIT defineste numai nivelurile logice 1 si 0, dar pot fi utilizate niveluri diferite. Functia Fuziune poate fi descrisa de functiile de rezolutie privind magistrala.

7.6.4. Simulator la nivel de porti.

In locul modelarilor circuitelor la nivelul tranzistoarelor, simularea poate fi efectuata la nivel de poarta.

Simularea se poate realiza plecand de la schema cu componente la nivel de poarta si de la un simulator cu modele pentru fiecare tip de poarta. Aceasta permite captarea si simularea proiectelor independent de tehnologia de implementare. Functionalitatea portilor trebuie sa fie aceeasi independenta de procesul de fabricatie. Dupa terminarea proiectului, se poate selecta tehnologia si efectua simularea la nivel de comutator sau de element de circuit.

- SILOS asigura modele la nivel de porti pentru functiile logice commune, latch-uri si bistabile.
- VHDL este potrivit pentru modelarea circuitelor la nivel de porti.

7.7. VHDL.

VHDL sau VHSIC HDL (Very High Speed Integrated Circuits Hardware Description Language) a fost dezvoltat in cadrul Programului VHSIC al Departamentului Apararii al SUA. Limbajul a fost ulterior modificat si acceptat ca standard IEEE, in 1987. In aceasta sectiune se va prezenta pe scurt VHDL. In mod curent se pot intalni si alte limbaje de descriere a hardware-lui (HDL), VHDL fiind selectat aici deoarece este des intalnit in proiectarea circuitelor integrate. Iata cateva dintre caracteristicile limbajului.

- VHDL reprezinta un limbaj HDL de nivel inalt, care asigura descrieri de la nivel de poarta la nivel de sistem. Nu sunt suportate modele de tranzistoare sau de masuri.
- VHDL este asemănător limbajului de programare ADA. Sunt incluse declaratii de tip (scalar, masiv, inregistrare), cheamări de subroutines, cicluri *if* si *do*, pointeri etc.
- VHDL opereaza si cu tipul fizic pentru a asigura informatia de sincronizare/timing.
- VHDL face distinctie intre semnale si variabile. Semnalele sunt utilizate pentru a reprezenta retelele de circuite, in timp ce variabilele sunt folosite pentru a pastra informatia de stare a programului.
- VHDL suporta descrieri generice parametrizate. De exemplu: sumatorul pe N biti.
- VHDL suporta descrieri Structurale. Acestea sunt descrieri de proiect, care prezinta proiectul sub forma de componente si conectivitate.
- VHDL suporta descrieri de tip Flux de Date (Dataflow), Descrierea de tip Flux de date prezinta fluxul datelor prin sistemul proiectat, similar cu Limbajul Transferurilor intre Registru (RTL – Register Transfer Language).
- VHDL suporta descrieri Comportamentale. Acestea sunt utilizate pentru a surprinde comportarea dinamica a circuitelor.

Descrierile de proiecte in VHDL constau in urmatoarele unitati de proiectare.

- *Entitatea Interfata* (Entity Interface) descrie conexiunile externe ale proiectului.
- *Entitatea Corp* (Entity Body) descrie implementarea sau comportarea unei entitati. Fiecare interfata entitate are una sau mai multe entitati de tip corp, asociate cu ea.
- *Configuratie* (Configuration). Aceasta selecteaza care entitate corp va fi folosita in constructia unei ierarhii a proiectului.
- *Pachet* (Package) contine declaratii commune si specificatii, care sunt partajate cu alte unitati de proiectare.
- *Corp Pachet*(Package Body) contine subprogramele de clarare in Pachete.

7.7.1. Exemplu de Descriere Structurala.

```

--          example VHDL program for simple
--          arithmetic unit
--
package misc is
    type control_struct is record
        con0  : bit;
        con1  : bit;
    end record;
end misc;
use misc;
entity testx6 is
    generic
    (
        N : natural := 3
    );
    port
    (
        A : bit_vector(0 To N);
        B : bit_vector(0 To N);
        Output : bit_vector(0 to N);
        Con : control_struct;
        Cin : bit;
        Cout : bit
    );
end testx6;
use misc;
architecture atestx6 of testx6 is
    signal B_not : bit_vector(0 To N);
    signal X1,X2,X3 : bit_vector(0 To N);
    signal Carry : bit_vector(0 To N-1);
begin
    component inv port (
        input : inout Bit;
        output : out Bit);
    component nand2 port(
        Ain      : in Bit;
        Bin      : in Bit;
        Output   : out Bit);

component fadd port(
    Air,Bin,Cin : in Bit
    Sum,Cout    : out Bit);
for i in 0 To N generate
    inv port map(B(i),B_not(i));
    nand2 port map(B(i),Con.con0,X1(i));
    nand2 port map(B_not(i),Con.con1,X2(i));
    nand2 port map(X1(i),X2(i),X3(i));

    if i = 0 generate
        fadd port map(a(0),X3(0),
Cin,Output(0),Carry(0));
    end generate;
    if (i > 0) and (i < N) generate
        fadd port map(a(i),X3(i),Carry(i-1),
Output(i),Carry(i));
    end generate;
    if i = N generate
        fadd port map(a(N),X3(N),
Carry(N-1),Output(N),Cout);
    end generate;
end generate;
end atestx6;

```

7.2.2.Exemplu de Descriere Comportamentală.

```
--      Full Adder behavioral description
--
entity fadd1 is
    port(Ain : bit := '0';
         Bin : bit := '0';
         Cin : bit := '0';
         Sum : bit;
         Cout : bit);
end fadd1;
architecture fadd of fadd1 is
begin
    Sum <= Ain xor Bin xor Cin after 5ns;
    Cout <= (Ain and Bin) or (Ain and Cin)
           or (Bin and Cin) after 6ns;
end fadd;
```

7.8. Formate pentru Interschimb de Proiecte.

HDL-urile de tipul VHDL, discutat mai sus, sunt utilizate de catre proiectanti pentru a descrie circuitul, si astfel, pentru “captarea” proiectului. Formatele pentru Interschimb de proiecte sunt, de asemenea, limbaje pentru descrierea informatiei de proiectare, permitand interschimbul acestora intre diferite programe. In mod curent aceste formate nu sunt proiectate pentru a fi citite sau scrise usor, ci pentru a descrie informatia intr-o maniera usor de analizat si generat. In industria circuitelor integrate se intalnesc formate de interschimb. In continuare vor fi trecute in revista aceste trei formate, cu unele exemplificari.

7.8.1. CIF.

- Formatul Intermediar Caltech (Caltech Intermediate Format).
- Comanda Layer specifica stratul current.
- Comanda BOX deseneaza dreptunghiuri. Comanda WIRE deseneaza fire. Comanda POLYGON deseneaza poligoane si comanda Roundflash deseneaza cercuri.
- Comanda CALL deseneaza o subrutina. DS indica inceputul unei subroutines, iar DD specifica sfarsitul unei subroutines. O subroutine este utilizata pentru a defini o celula (colectie de forme) intr-o ierarhie.
- Toate dimensiunile sunt date in sutimi de microni (10^{-8} m).
- Informatia specificata de catre utilizator poate fi stocata in subroutinesle 0 – 9. Semantica acestora nu a fost standardizata.

Mai jos se prezinta o descriere CIF.

```

DS 1;
9 PROBE_PAD;
L CV;
P 32300,-14800 32800,-14800 32800,-14300 32300,-14300;
P -33700,-14800 -33200,-14800 -33200,-14300 -33700,-14300;
L CM2;
P 8500,3500 -9000,3500 -9000,-14000 -500,
-22500 -500,-34500 0,-34500 0,-22500
8500,-14000;
P -34000,-15100 -16500,-15100 -16500,3500 -34000,3500;
L CM;
P 33500,-14000 16000,-14000 8500,-21500 -9000,-21500
-16500,-14000 -34000,-14000 -34000,-31500 -9000,-31500
-9000,-34500 8500,-34500 8500,-31500 33500,-31500;
DF;
DS 2;
9 TLINE3;
C 1 R 0,1 T -136750,38900;
C 1 MY R U,1 T 400250,38900;
L CM2;
P -102250,38400 365750,38400 365750,38900 -102250,38900;
L CM;
P -102250,29900 365750,29900 365750,47400 -102250,47400;
DF;
DS 29;
9 GREG_CHIP;
C 2 R 0,1 T -45150,70750;
DF;
C 29;
E

```

7.8.2. GDS II.

- GDS II a fost primul format pentru interschimb si este unul dintre cele mai răspândite în PAC a circuitelor integrate. El a fost dezvoltat de către Calma Co. (General Electric)
- GDS II suportă text, masive, care nu sunt întâlnite în CIF.
- Împreună cu CIF, GDS II este un format binar.

7.8.3. EDIF.

- EDIF – Electronic Design Interchange Format este utilizat în mod frecvent ca format pentru descrierea listei de componente/conexiuni necesare diferențelor sistemelor de proiectare.
- EDIF este, din punct de vedere sintactic, similar cu LISP. Ambele sunt bazate pe liste.
- EDIF are trei niveluri, nivelul cel mai coborât nu este programabil și necesită constante în toate expresiile. Nivelul 1 suportă variabile și expresii, ca și celule parametrizate. Nivelul 2 este programabil și reprezintă o extensie a limbajului LISP.
- EDIF este deja acceptat ca standard.
- EDIF suportă diferite abordări ale proiectării. Componentele și lista lor de conexiuni, Scheme, Comportament, Planul mastilor s.a.

7.8.3.1. Exemplu EDIF.

```
{  
EDIF VHDL_design {EDIFversion 2 0 0)  
  (EDIFlevel 0) (keywordmap (keywordLevel 0))  
  (status { written (timeStamp 1990 24 7 19 30 27) })  
  ( EXTERNAL SC_LIB {EDIFlevel 0) ( technology{  
    (numberDefinition { scale 1 (e 1 -9) (unit Distance)}))  
    ( cell and3 { cellType GENERIC )  
      ( view abstract (viewType NETLIST)  
        ( interface  
          ( port { Rename GND "gnd!" } (direction input) )  
          ( port { Rename VDD "vdd!" } (direction input) )  
          ( port { Rename OUTPUT "Out" } (direction output) )  
          ( port { Rename CIN "Cin" } (direction input) )  
          ( port { Rename BIN "Bin" } (direction input) )  
          ( port { Rename AIN "Ain" } (direction input) )  
        )  
      )  
      (COMMENT .... other cells delete from this description .... )  
    ))  
  (library {rename DEFAULT_LIB ".") (EDIFlevel 0)  
  (technology (numberDefinition {scale 1 (e 1 -9) (unit Distance)}))  
    (cell NBIT_ALU1 { cellType GENERIC )  
      (view autoLayout (viewType NETLIST)  
        (interface)  
        (contents { instance AND377 {  
          viewref abstract  
            (cellref and3 { libraryRef SC_LIB ) ) ) )  
            (instance AND3 { viewref abstract (cellref and3  
              (libraryRef SC_LIB ) ))})  
          (COMMENT ..... other cell instances deleted ..... )  
        (net { Rename VDD "vdd!" )  
          ( Joined {  
            ( portref VDD { instanceref INPAD85 ) )  
            ( portref VDD { instanceref INPAD84 ) )  
            ( COMMENT ... rest of VDD connects deleted .. )  
          ( net B_2_  
            ( Joined {  
              ( portref INPUT { instanceref INPAD66 ) )  
              ( portref AIN { instanceref AND248 ) )  
              (Property sigType (string "signal") (owner "Cadence" ))  
            )  
            (COMMENT rest of nets deleted ... )  
          ) ) ) ) ) )
```

7.9. Sinteză comportamentală.

Scopul uneltelor de sinteză comportamentală constă în preluarea unei specificări a comportării proiectului pentru a genera implementarea structurală. Este ca și cum “tu spune ce trebuie să facă circuitul iar uneltele realizează circuitul”. În această secțiune se vor discuta elementele de bază pentru transformarea descrierilor comportamentale în descrieri structurale. Sinteză descrierilor structurale a fost deja examinată.

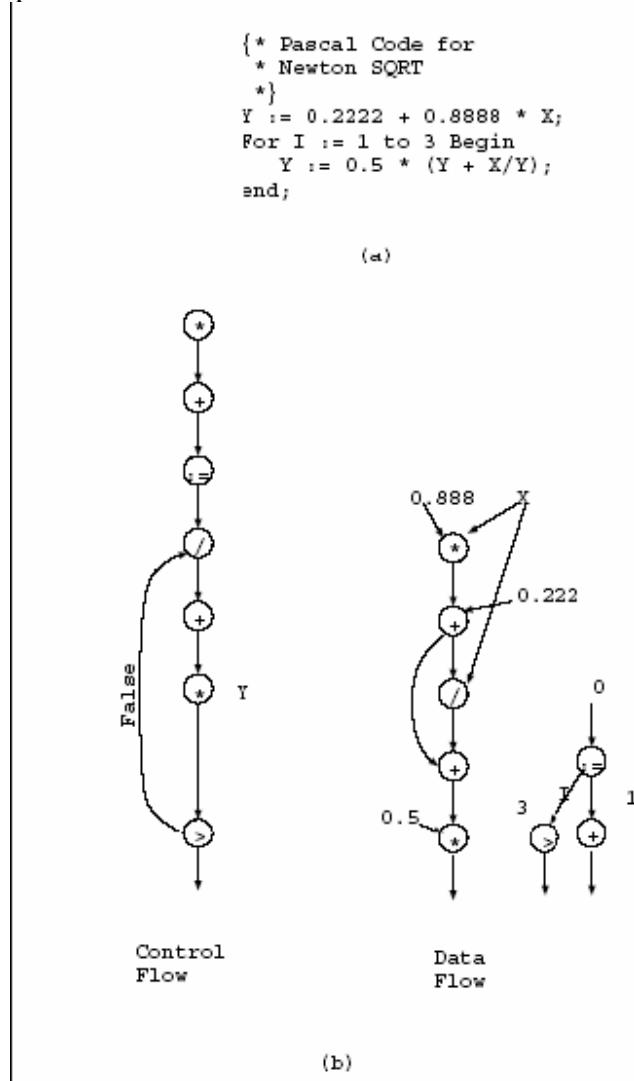
Prima motivare pentru sinteză comportamentală este aceea de a reduce timpul de proiectare și a erorilor.

Sinteză permite, de asemenea, proiectantului să exploreze rapid numeroase soluții pentru proiect, cât și metodologii diferite. Documentația poate fi incorporată ușor în descrierile de nivel înalt. Procesul de sinteză comportamentală poate fi structurat în trei etape.

7.9.1. Definirea Task-ului.

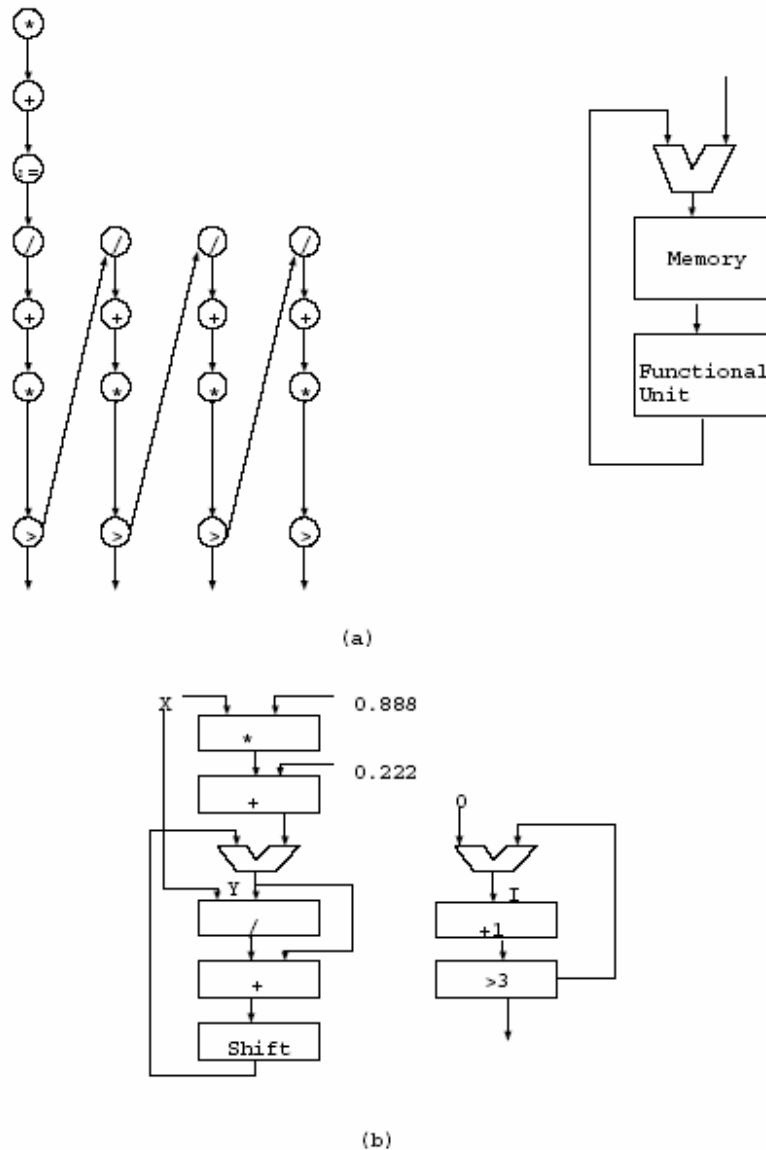
Etapa de definire a task-ului constă în citirea informației de intrare referitoare la descrierea comportamentală pentru a genera o reprezentare internă a proiectului. Sintaxa de intrare pentru descrierea comportamentală poate fi un limbaj de programare, de exemplu C, un HDL cum ar fi VHDL, Verilog sau un limbaj funcțional de programare ca Prolog sau Lisp. Descrierea comportamentală este apoi convertită

in reprezentari grafice, dintre care una contine informatia privind fluxul de control, iar cealalta informatia legata de fluxul de date. Un exemplu este dat in figura urmatoare, care se refera la calculul radacinii patrate a unui numar folosind metoda lui Newton. Algoritmul este descris in limbajul Pascal. Pe acest graf pot fi efectuate optimizari pentru eficientizare. Una dintre optimizari consta in substituirea inmultirii cu 0,5 printr-o deplasare. Algoritmii de optimizare sunt similari cu cei folositi de compilatoare optimizatoare.



7.9.2. Planificarea.

In etapa de planificare se ia decizia privitoare la momentul in care trebuie sa aibe loc operatiile din graful de comanda. Planificarea pasilor in timp (pasii de comanda) in care trebuie efectuate operatiile. In figura de mai jos se prezinta o alocare posibila a operatiilor.



- Unelte pentru proiectarea de sisteme analogice. Uneltele descrise mai sus au avut in vedere proiectarea sistemelor numerice. In prezent se dezvolta pachete de proiectare pentru sisteme analogice plecand de la HDL-uri corespunzatoare.
- Incorporarea unor tehnici de testare in cadrul sintezei circuitelor. Deocamdata este la latitudinea proiectantului specificarea structurilor de testare.

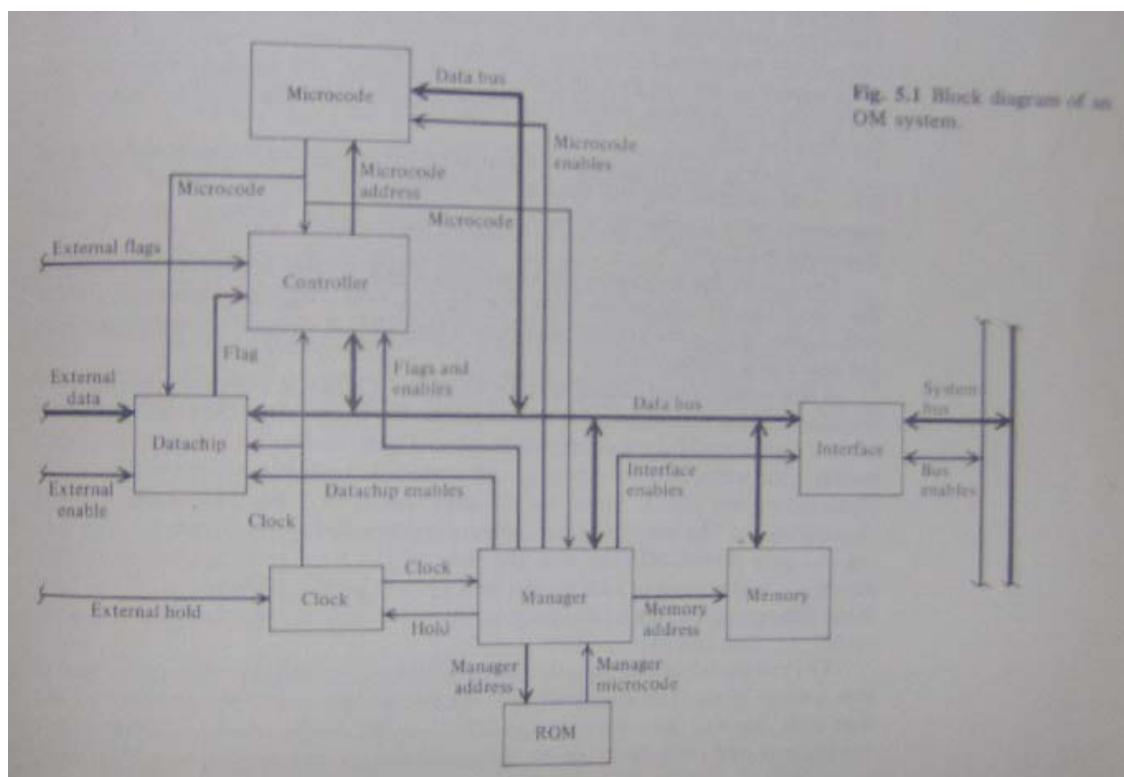


Fig. 5.1 Block diagram of an OM system.

Fig.1 Schema bloc a microprocesorului

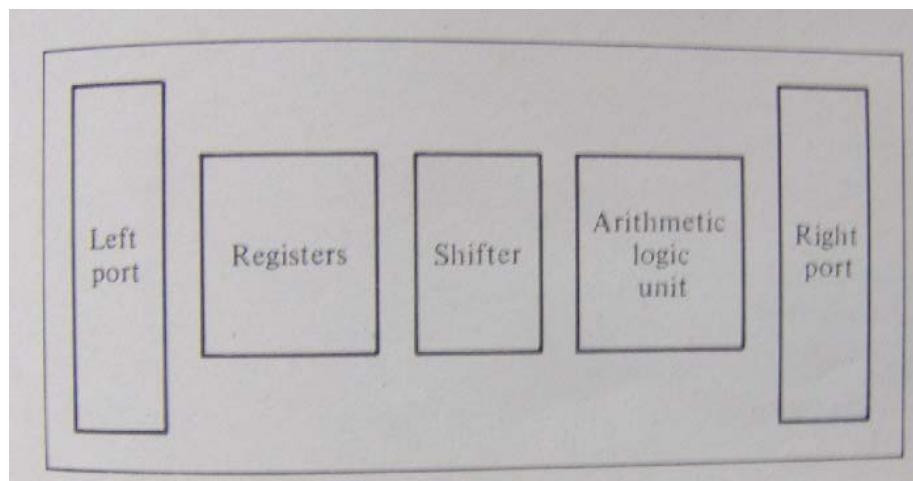


Fig. 2. Planul general al unitatii de executie

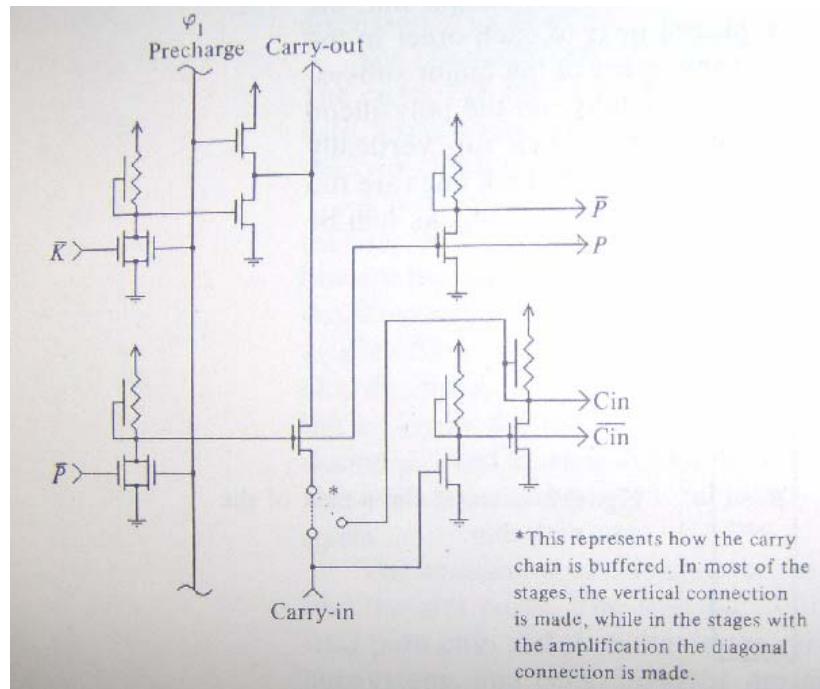


Fig.3. Lantul de transport de tip Manchester, al unitatii de executie

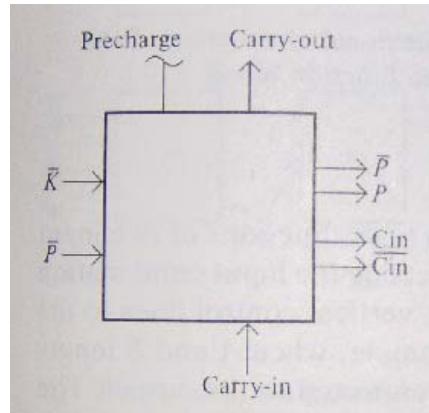


Fig.4. Reprezentarea abstracta a lantului de transport.

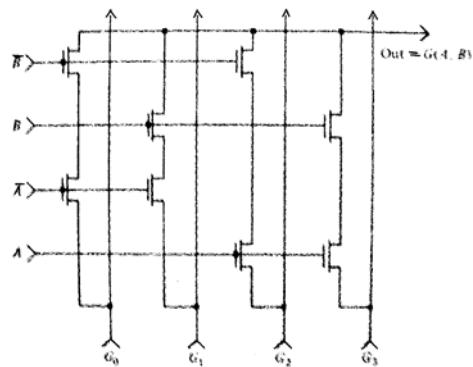


Fig. 5 Diagrama retelei logice programabile, la nivelul tranzistoarelor de trecere,

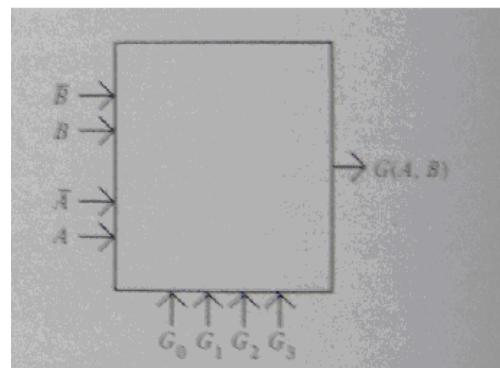


Fig.6. Reprezentarea funcțională abstractă a retelei logice programabile.

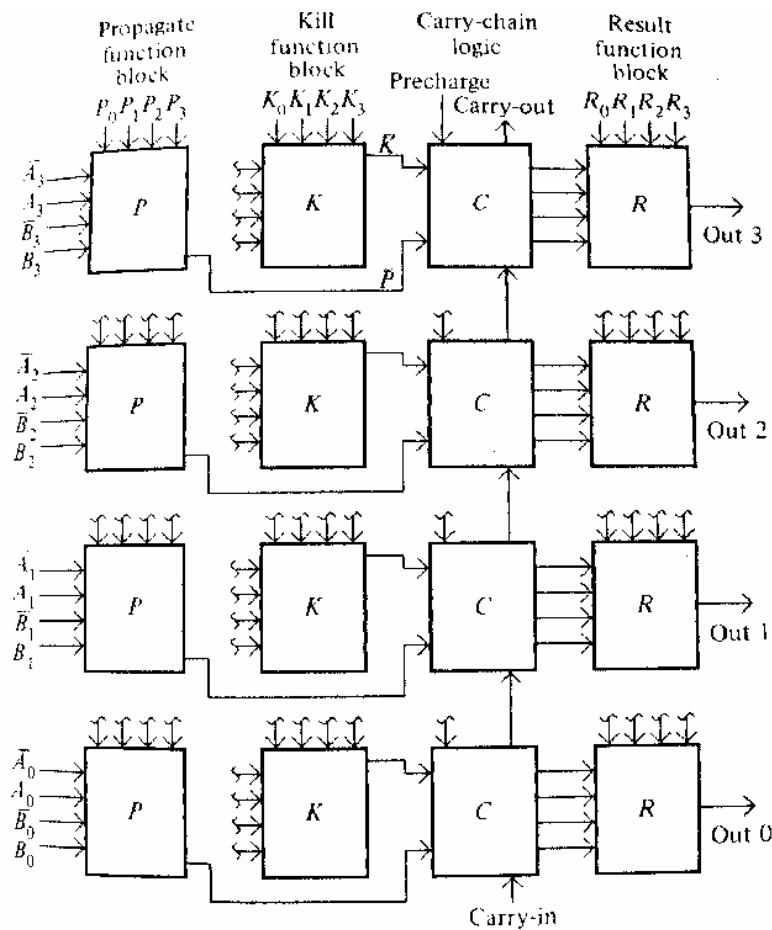


Fig.7. Schema bloc a unei UAL pe 4 biti

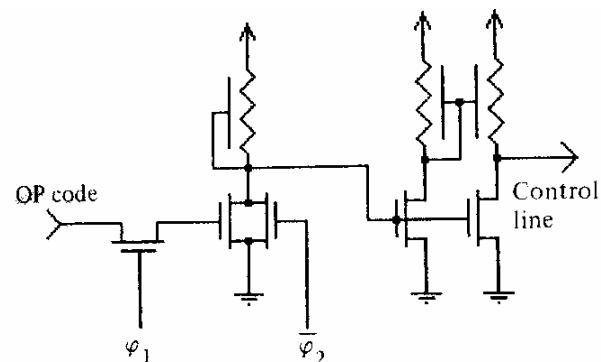


Fig.8. Circuit de comanda pentru UAL. Toate iesirile sunt la nivel ridicat pe durata lui φ_2 negat; termenii selectati sunt pe nivel coborat pe durata lui φ_2 ; codul de operatie este valid pe durata lui φ_1 .

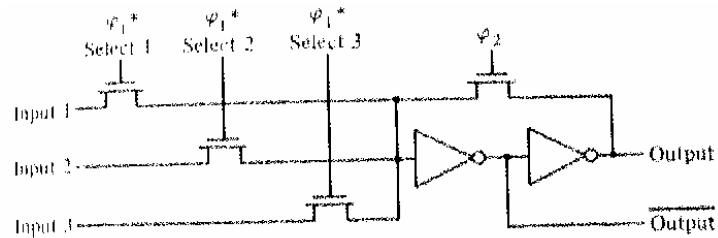


Fig.9. Registrul de intrare UAL si multiplexor.

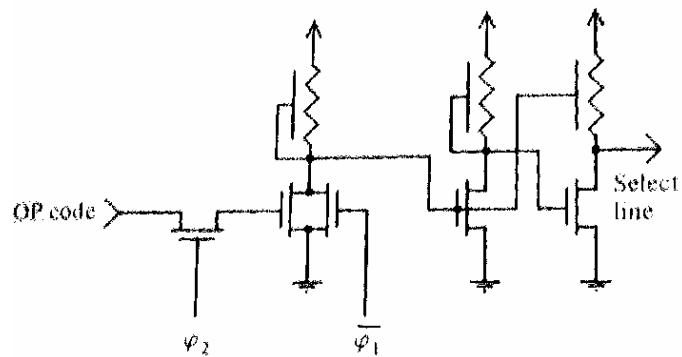


Fig.10. Circuit de comanda pentru selectie . Toate iesirile sunt la nivel coborat pe durata lui φ_1 negat; termenii selectati sunt pe nivel ridicat pe durata lui φ_1 ; codul de operatie este valid pe durata lui φ_2 .

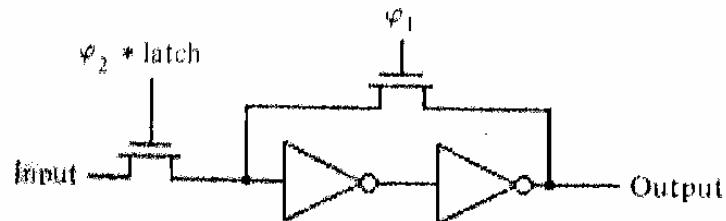


Fig.11. Registrul de iesire.

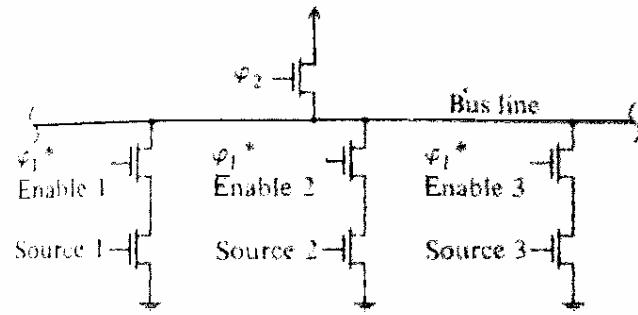


Fig.12. Circuit de magistrala preincarcat

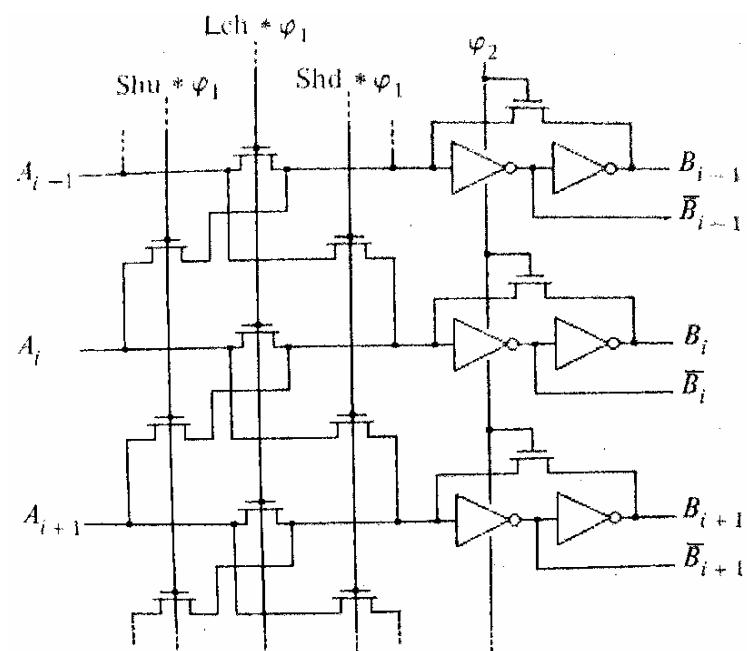


Fig. 13. Circuit simplu de deplasare cu un bit spre stanga.

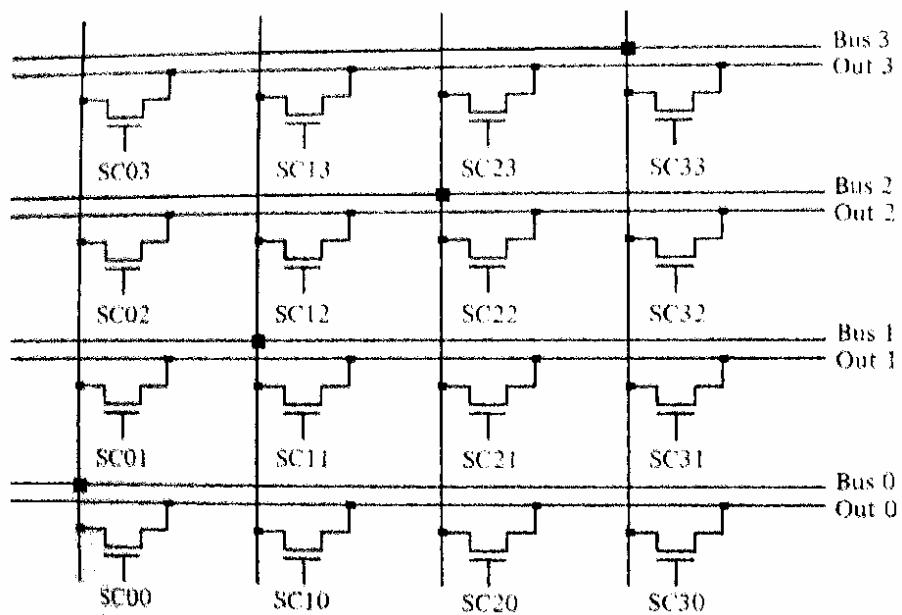


Fig. 14. Comutator crossbar 4 x 4.

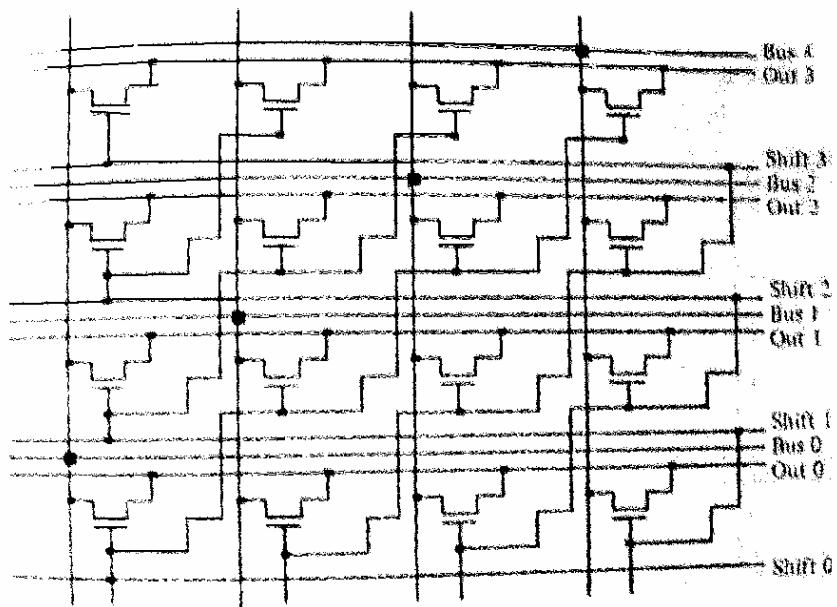


Fig. 15. Circuit de deplasare circulara (barrel shifter) 4 x 4.

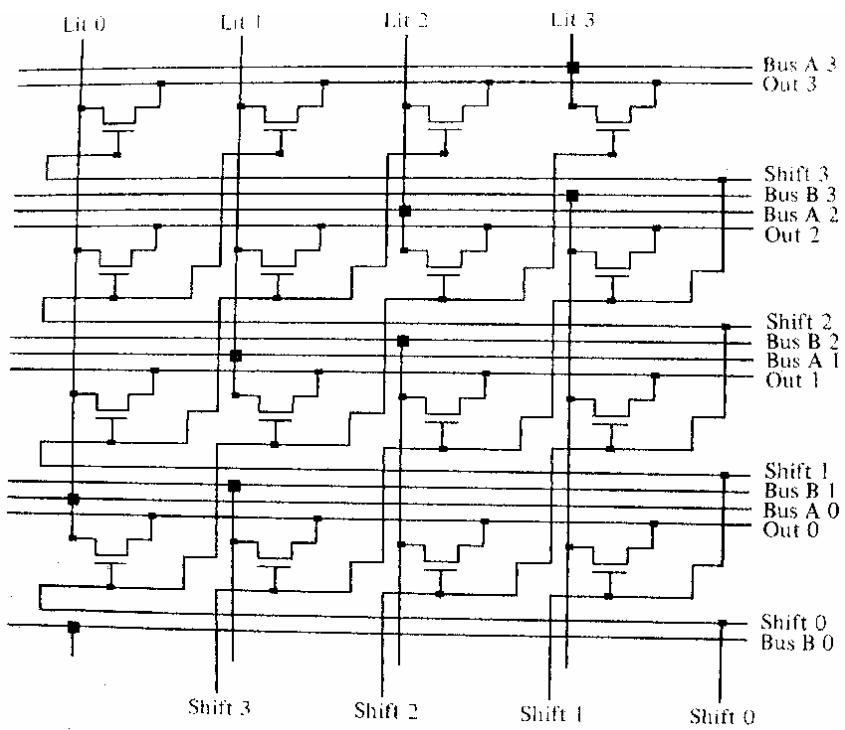


Fig. 16. Circuit de deplasare 4×4 , cu trasee verticale sectionate si doua magistrale de date.

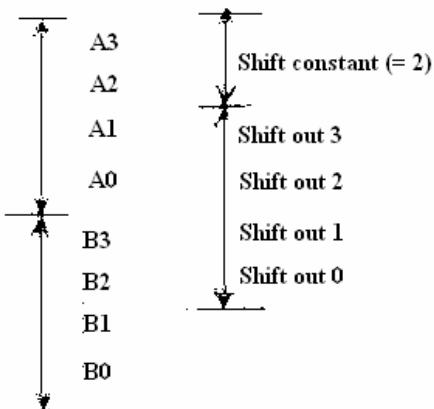


Fig . 17. Reprezentarea conceptuala a operarii circuitului de deplasare.

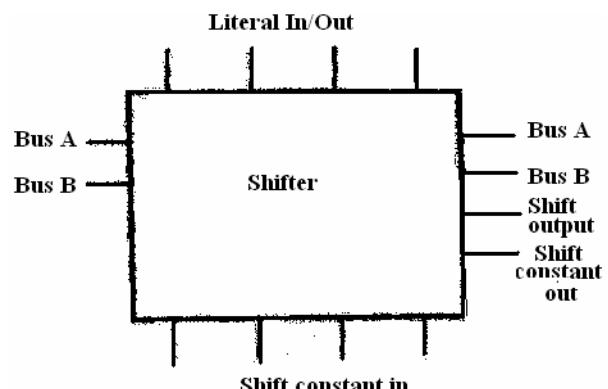


Fig. 18. Diagrama bloc a circuitului de deplasare.

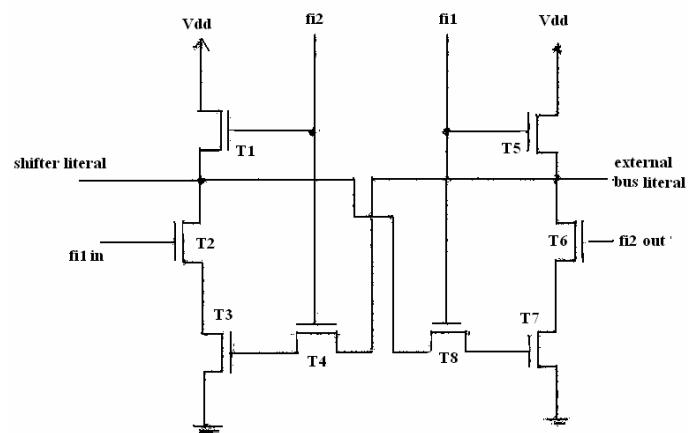


Fig. 19. Interfata Literal.

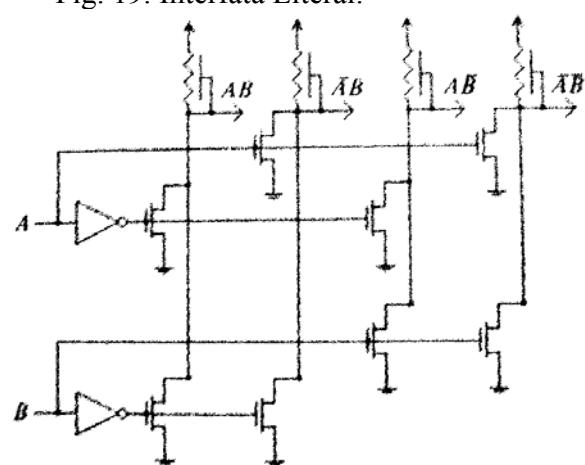


Fig. 20. Decodificator bazat pe NOR

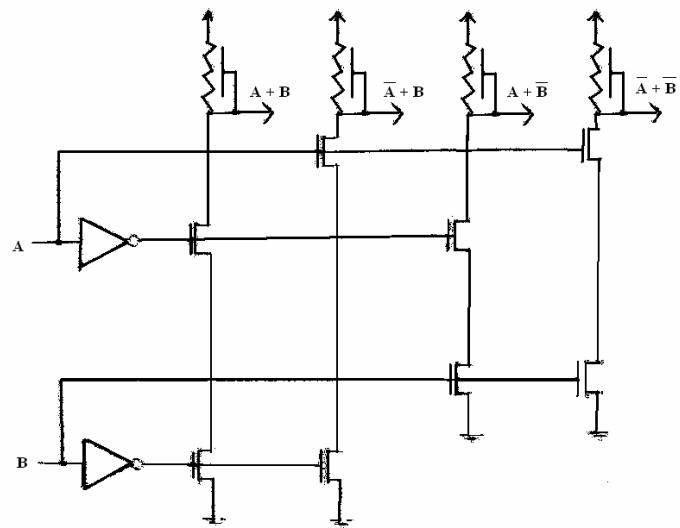


Fig. 21. Decodificator bazat pe NAND

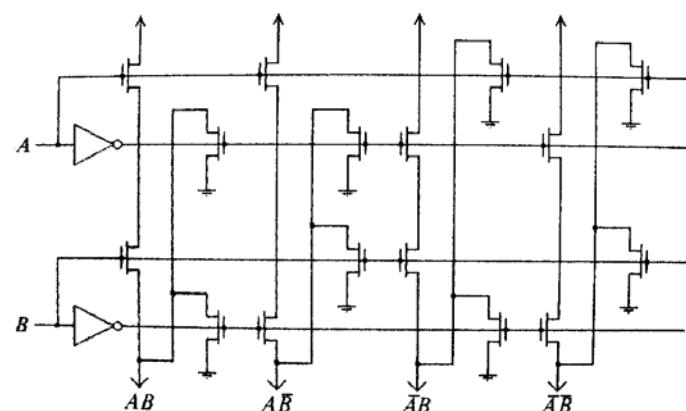


Fig. 22 Decodificator complementar.

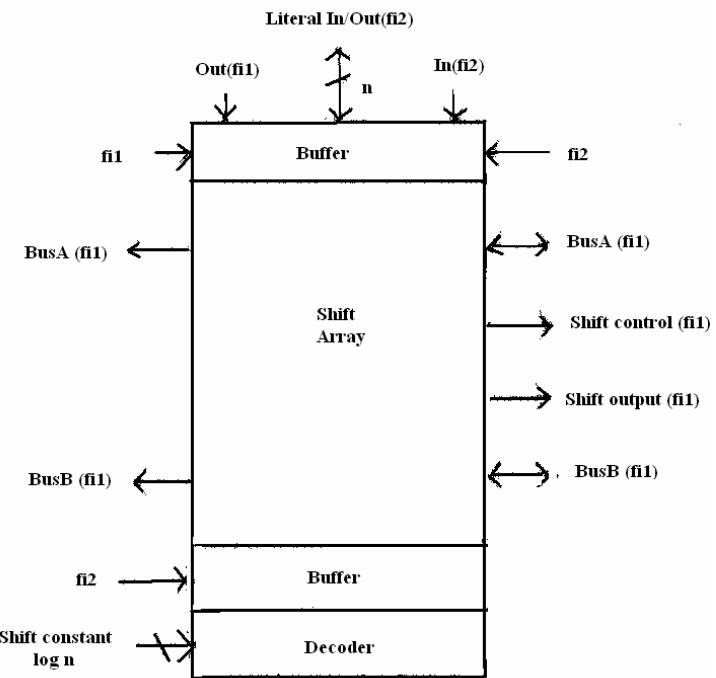


Fig. 23. Circuit de deplasare complet sincronizat.

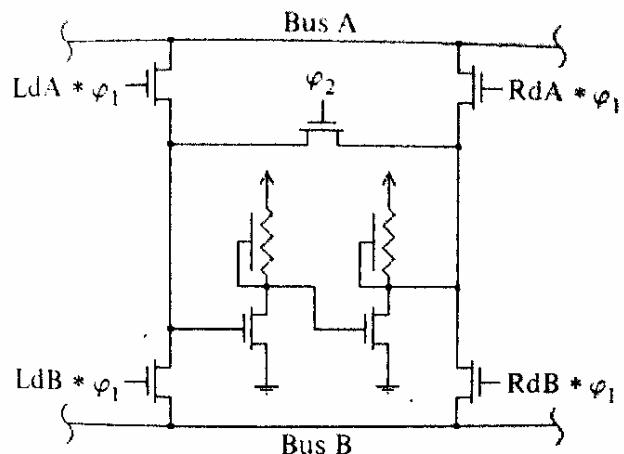


Fig. 24 Celula de regisztr biport.

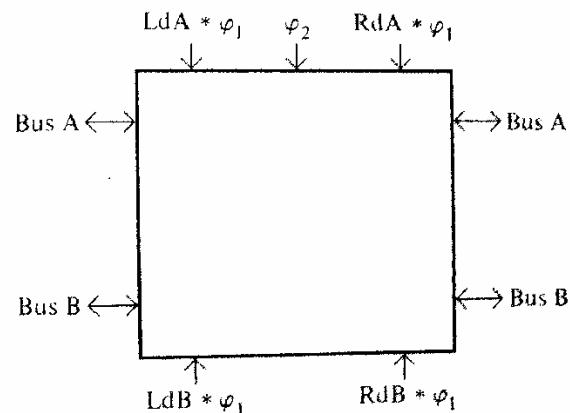


Fig. 25. Diagrama bloc a unei celule de regisztr biport.

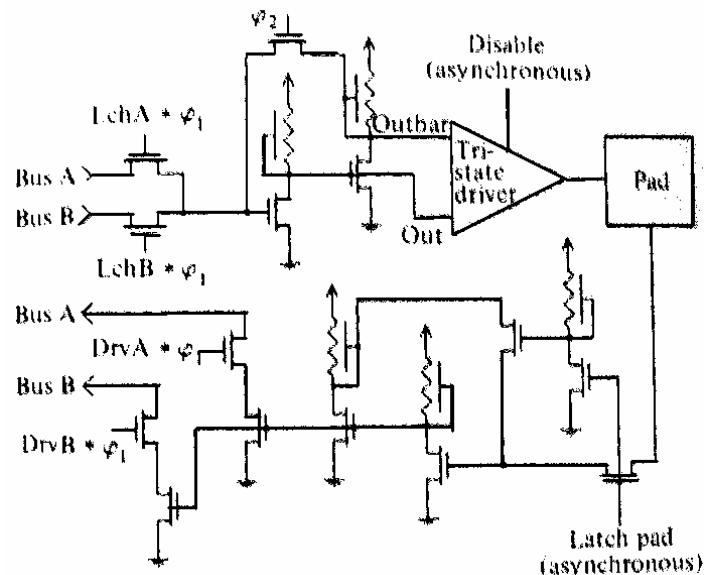


Fig. 26. Circuitul de I/E conectat la un plot bidirectional TS.

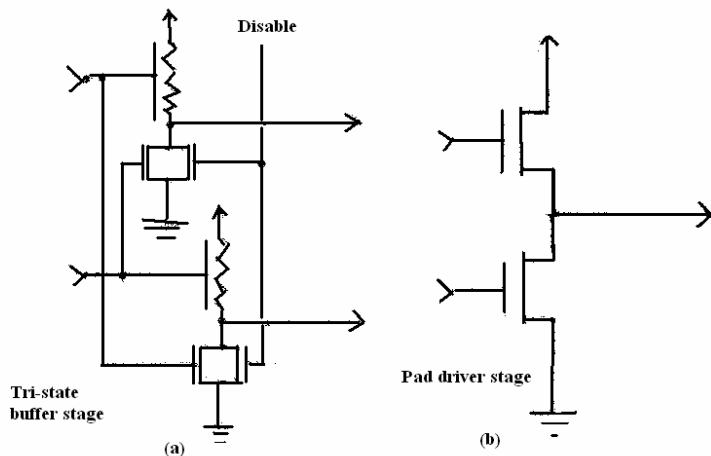


Fig. 28 Etaj tampon TS (a) si Circuit de comanda a plotului (b).

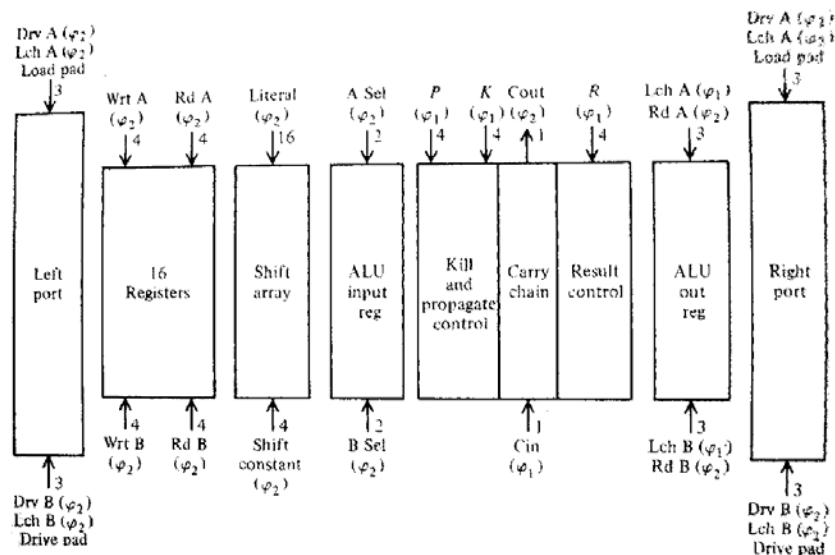


Fig. 29. Diagrama bloc a UAL, cu mentionarea traseelor de comanda.

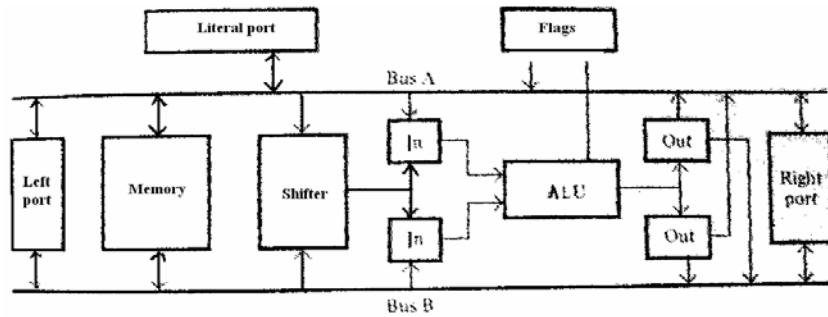


Fig. 30. Diagrama bloc a UAL.

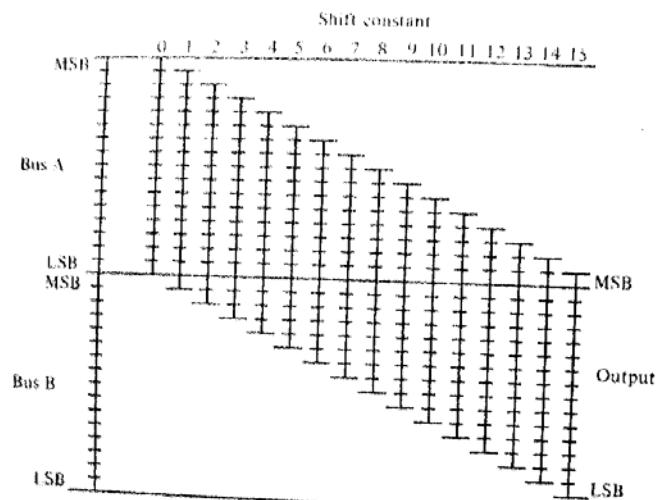


Fig. 31. Operarea circuitului de deplasare.

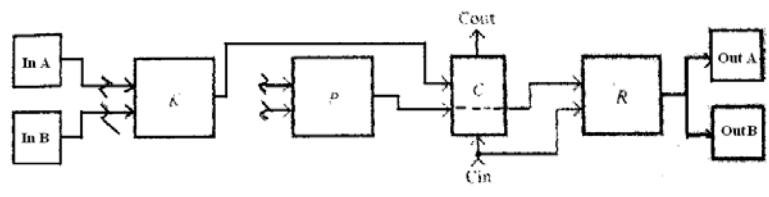


Fig. 32. Diagrama bloc a unui bit al UAL.

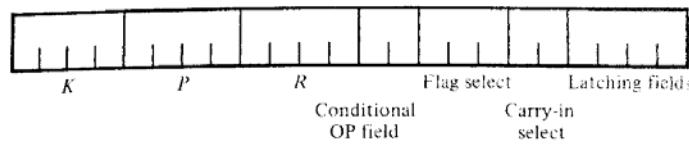


Fig. 33 (a). Faza 2 COP (intra in φ_1)

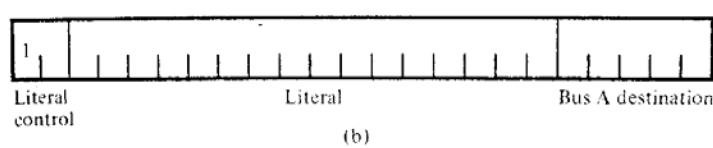


Fig. 33 (b). Faza 1 transfer COP de la Literal (intra in φ_2)

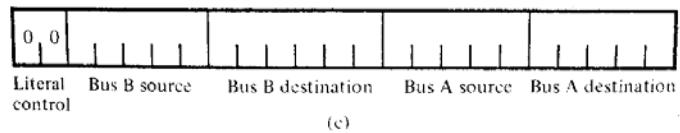


Fig. 33 (c). Faza 1 COP normal (intra in φ_2)

Tab. 1 Transferuri pe magistrale

Bus A Source	Bus A Destination
0nnnn Register n	0nnnn Register n
10000 Right port pins	10000 Left port, drive now
10001 Right port latch	10001 Left port, drive now
10010 Left port pins	1001x Left port, no drive
10011 Left port latch	10100 Right port, drive now
10100 ALU output latch A	10101 Right port, drive now
10101 ALU output latch B	1011x Right port, no drive
10110 Flag register	11000 ALU input latch A
	11001 ALU input latch A gets shift out
	11010 ALU input latch A gets shift control
	11011 Flag register

Tab. 1 Transferuri pe magistrale

Bus B Source		Bus B Destination	
0nnnn	Register n	00nnnn	Register n
10000	Right port pins	010000	Left port, drive now
10001	Right port latch	010001	Left port, drive φ_2
10010	Left port pins	01001x	Left port, no drive
10011	Left port latch	010100	Right port, drive now
10100	ALU output latch A	010101	Right port, drive φ_2
10101	ALU output latch B	01011x	Right port, no drive
		0110xx	ALU input latch B
		10nnnn	ALU input latch B gets shift output, shift constant = n
		11nnnn	ALU input latch B gets shift control, shift constant = n

Tab2. Codificarea operatiilor in UAL, la nivelul microinstructiunilor.

	K	P	R	Cin	Cond
$A + B$	1	6	6	0	Add
$A + B + \text{Cin}$	1	6	6	1	Add with carry
$A - B$	2	9	6	2	Subtract
$B - A$	4	9	6	2	Subtract reverse
$A - B - \text{Cin}$	2	9	6	1	Subtract with borrow
$B - A - \text{Cin}$	4	9	6	1	Subtract reverse with borrow
$-A$	12	3	6	2	Negative A
$-B$	10	5	6	2	Negative B
$A + 1$	3	12	6	2	Increment A
$B + 1$	5	10	6	2	Increment B
$A - 1$	12	3	9	2	Decrement A
$B - 1$	10	5	9	2	Decrement B
$A \wedge B$	0	8	12	0	Logical AND
$A \vee B$	0	14	12	0	Logical OR
$A \oplus B$	0	6	12	0	Logical EXOR
$\neg A$	0	3	12	0	Not A
$\neg B$	0	5	12	0	Not B
A	0	12	12	0	A
B	0	10	12	0	B
Mul	1	14	14	0	Multiply step
Div	3	15	15	0	Divide step
A/O	0	14	12	0	Conditional AND/OR
Mask	10	5	8	2	Generate mask
SHL A	3	0	10	0	Shift A left
Zero	0	0	0	0	Zero

Tabela 3. Selectarea lui Cin

00 0

01 Flag bit

10 1

11 Flag bit complemented

Tab.4. Selectie OP Conditionalala

Select	Flag bit	K	P	R	
0	x	----	----	----	Unconditional
1	0	---0	--0-	--0-	Multipy step
	1	----	0---	0---	
2	0	0--0	-00-	-00-	Divide step
	1	-00-	0--0	0--0	
3	0	----	----	----	AND/OR
	1	----	-00-	----	

Tab. 5. Selectarea noului bit indicator

Select	New flag bit
0	Old flag big
1	Carry-out
2	MSB
3	Zero
4	Less than
5	Less than or equal
6	Higher (in absolute value)
7	Overflow

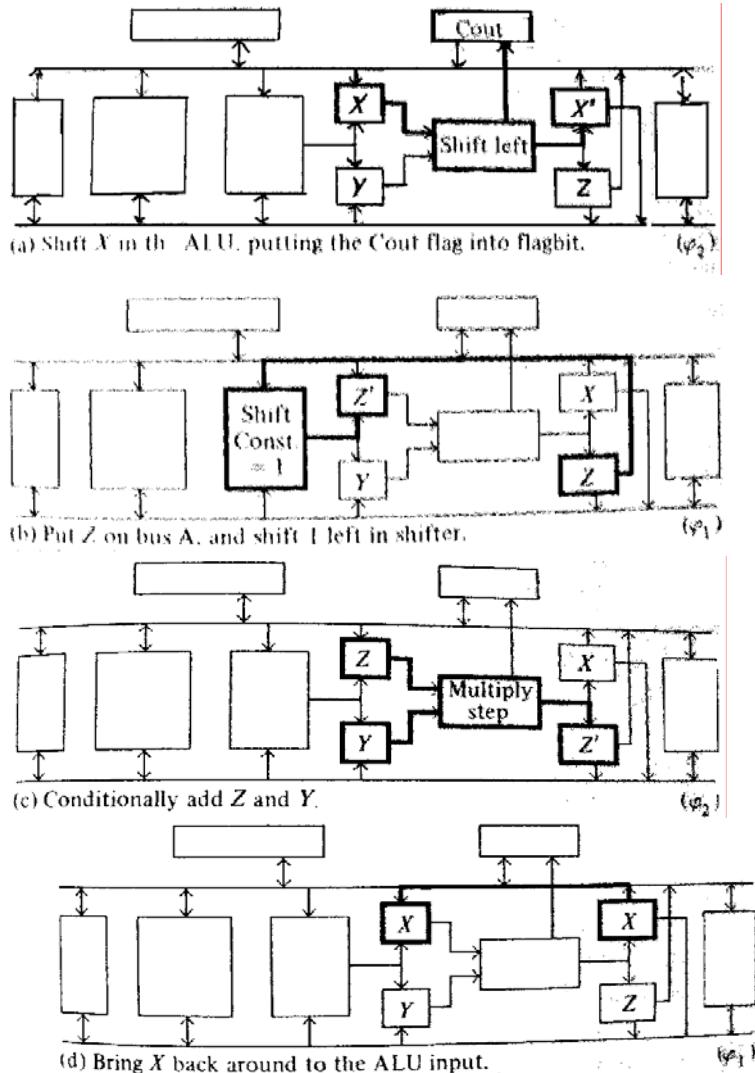
Tab. 6. Indicatorii de conditii.

Bit	Flag
0	Not changed
1	Not changed
2	Not changed
3	Not changed
4	Not changed
5	Previous value of flag bit
6	→ Carry into MSB stage
7	Less than or equal
8	→ Higher (in absolute value)
9	→ Less than
10	LSB
11	→ Zero
12	MSB
13	Overflow
14	→ Carry-out
15	Current flag bit

Tab. 7. Campul de memorare(latching)

Latching field	Register loaded
1xxx	Flag register loaded with current ALU flags
x1xx	ALU output latch A loaded with the ALU output
xx1x	ALU output latch B loaded with the ALU output
xxx1	The literal field during the next φ_2 is loaded with the contents of bus A during the last φ_2
0000	None of these registers are affected

Exemplu de microprogramare a inmultirii $Z \leftarrow X \times Y$, pe 16 biti.



- φ_1 : ALU.Out.A←ALU(Shift A left)←ALU.In.A;
Latch Flags;
- φ_1 : ALU.In.A←Shift.out, Bus.A←ALU.Out.B;
 $R[1] \leftarrow Bus.B \leftarrow R[0]$; This gives a shift constant of 1.
- φ_2 : ALU.Out.B←ALU(Multiply Step); conditionally add.
Flag←Cout;
- φ_2 : ALU.In.A←Bus.A←ALU.Out.A

5.3 SYSTEM OVERVIEW

An informal block diagram of one OM system is shown in Fig. 5.1. Such a system is a complete stored-program, general-purpose computer. Input/output devices are usually interfaced via the external data bus and control lines, located to the left in Fig. 5.1. Several such systems may be interconnected via the system bus to augment one user's overall system. Tasks may then be distributed among the OM systems, for example, using different ones to independently control different input/output devices, thereby improving overall system performance. Groups of different user systems may also share the system bus.

Each OM system is composed of five LSI chips, along with some standard memory chips and a few MSI chips. A brief description follows of the five LSI chips being designed as part of this project.

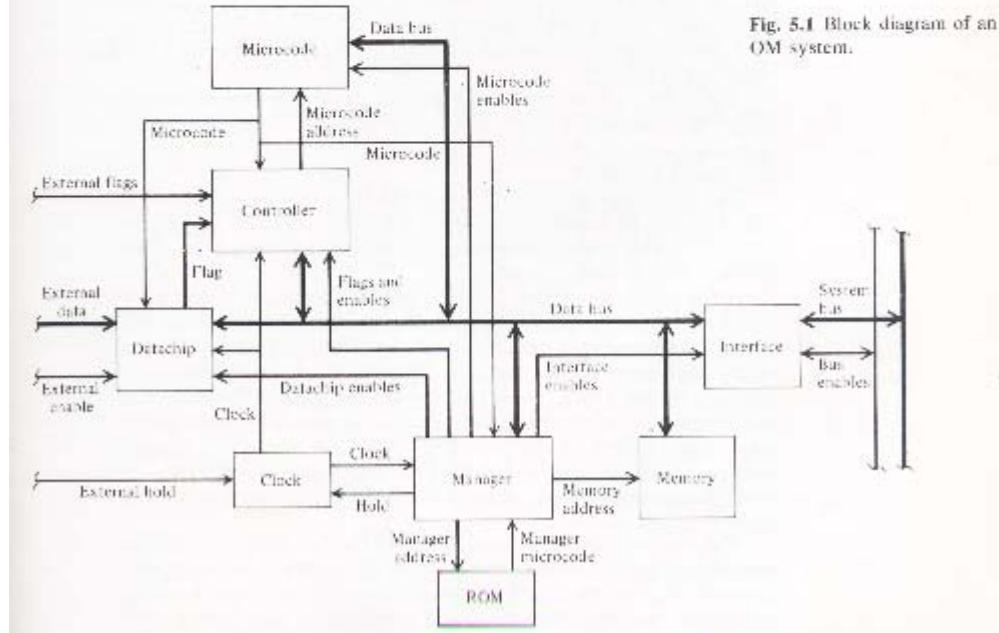


Fig. 5.1 Block diagram of an OM system.

The *data path chip* performs most of the data manipulation functions for the system. The operations are performed as directed by sequences of control micro-instructions, which are fetched from a microcode memory using addresses generated by the controller chip. The main subsystems of the data path chip are a register array, a shifter, and an arithmetic logic unit (ALU). Two buses connect these subsystems. This chip's internal structure is described in detail later in this chapter.

The *controller chip* contains the microprogram counter (μ PC) that stores the microcode memory address, and a counter for the control of microprogram loops. The chip also contains stacks for both the microprogram counter and loop control counter values. The concepts of controller structure and function are fundamental in computer architecture. (Chapter 6 provides an introduction to these ideas and then describes the organization and layout of the controller chip.)

The *memory manager chip* provides addresses for the data memory and directs the communication between chips on the data bus. It also implements some simple data structures in the data memory. The manager can divide the memory into separate partitions and implement a different data structure in each partition. Four basic data structures are implemented: stacks, queues, linked lists, and arrays. When accessing a stack partition, for example, the microcode need only ask the manager to push or pop data on or off the stack. The manager maintains the stack pointers, performs bounds checking to see if the stack is full or empty, and transfers the data.

The *system bus interface chip* provides asynchronous communication with other OM systems via the system bus. There are a whole host of subtleties associated with interfacing asynchronous buses. These issues are among those discussed in Chapter 7.

The *clock chip* generates the two-phase clock signals needed by the system. The clock can be stopped to allow for the synchronization of asynchronous signals. Some chips in the system have only a single clock input; those chips generate the two clock phases on-chip.

A few words about timing may be helpful: In general, during φ_1 data bits are transferred from one subsystem to another on the same chip, while during φ_2 data bits are transferred from one chip to another. The data chip's ALU, and other data modification units, operate during φ_2 . Microcode is available on both phases and is pipelined by one phase. Thus, the opcodes that control the ALU enter the data chip during φ_1 . The microprogram address is generated by the controller chip during φ_2 , gets driven off-chip into the data chip's microcode memory's latches during φ_1 , and is used to look up the next opcode on the following φ_2 . Because of these timing requirements, all jumps in the microcode are pipelined by one clock cycle.

The remainder of this chapter describes the data path chip and is presented in two distinct parts. The first part outlines the architectural requirements for the data path chip and then illustrates, via the detailed design and layout of the chip's subsystems and cells, how the design methodology was applied to satisfy these

requirements. The second part is an external functional description of the data path chip, intended as a user manual for those who microprogram the computer system, and for reference while studying the OM2 controller chip in Chapter 6.

5.4 THE OVERALL STRUCTURE OF THE DATA PATH

The basic requirements initially established for the data path chip were (1) that it be gracefully interconnectable into multiprocessor configurations, (2) that it effectively support a microprogrammed control structure, thus enabling machine instruction sets to be configured to the application at hand, (3) that it be able to do variable field operations for emulation instruction-decoding, assembly of bit-maps for graphics, etc., and (4) that its performance be as fast as possible.

In order to satisfy the first requirement, the data path chip was designed with two ports: one port to be used for a system interconnection, and the other for connection to local memory, input/output devices, etc. The requirement for gracefully handling variable-length words required a shifter at least sixteen bits long. The performance requirement dictated an arithmetic logic unit having considerable flexibility without sacrificing speed. In many systems time is lost in assembling the two operands required for most operations. Therefore, the data path has two internal buses, and all registers on the chip are two-port registers. In order to avoid extensive random wiring for connecting the major subsystems on the chip, the following strategy was adopted at the outset: two internal buses would run through the entire processing array, from one end of the chip to the other. One port was to be located at the left end of the chip, and the other port at the right end.

The three main functional blocks on the chip are the arithmetic logic unit, the shifter, and the register array. These blocks are placed next to each other in the center of the chip, between the two ports. The arrangement of the major subsystems is shown in Fig. 5.2. The system buses run horizontally, on the polysilicon level, through these functional blocks. The major control lines run vertically across these blocks, on the metal level. The power, ground, and clock lines are run parallel to the control signal lines. The details of these functional blocks will be



Fig. 5.2 General floor plan of the data path chip.

described in subsequent sections of this chapter. Included are descriptions of peripheral circuits needed to interface subsystems with each other and to the outside world. Detailed layouts of certain cells in the system are also included. The overall layout of the data chip is shown in the frontispiece.

5.5 THE ARITHMETIC LOGIC UNIT

The carry chain of the ALU, and its associated logic, was the first functional block to be designed in detail, since it was believed that the carry chain would limit the performance of the system. Simulations of several look-ahead carry circuits indicated that they would add a great deal of complexity to the system without much gain in performance. For this reason a decision was made early in the project to implement the fastest possible Manchester-type carry chain, (Chap. 1, §.8) having a carry propagation circuit similar to that shown in Fig. 1.28. The carry chain and its associated logic were allowed to dictate the repeat distance of the cells in the vertical direction. In *n*MOS technology, a Manchester carry chain is particularly limited in its ability to propagate a *high* carry signal. However, it can quite rapidly propagate a *low* carry signal.

In the arithmetic logic unit there will be a null period when the OP code for the next operation is being brought in. Advantage can be taken of this null period to precharge the carry chain and other sections of the data path where timing is particularly crucial. In this way, it is not necessary to propagate high signals through pass transistors where the rise transient would be particularly slow. This strategy was applied in OM's ALU; the resulting carry chain is shown in Fig. 5.3.

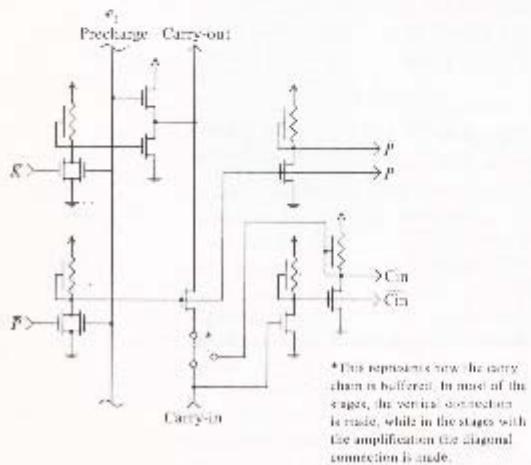


Fig. 5.3 Carry-chain circuit for the arithmetic logic unit.

*This represents how the carry chain is buffered. In most of the stages, the vertical connection is made, while in the stages with the amplification the diagonal connection is made.

The main carry chain runs through a pass transistor from carry-in to carry-out. The carry-in signal is detected by the gate of an inverter that feeds the signal into the subsequent logic of the ALU. Three transistors are used to control the state of the carry-out of each stage. The first one merely precharges the node associated with carry-out during the null period of the ALU. The second is the carry-kill signal that is derived from the inputs to the ALU, and it simply grounds the carry-out through a single transistor. The third is the pass transistor that causes carry-out to be equal to carry-in. These last two signals associated with the carry chain in each stage, carry-kill and carry-propagate, are generated by two NOR gates that have kill-bar and propagate-bar as one input and precharge as the second input. Hence, it is assured that the kill signal and propagate signal are disabled during the null period when the precharging takes place.

After some analysis, we found that nearly all interesting combinations of carry-in and the input signals could be generated using propagate and carry-in from each stage. Thus, as in Fig. 5.4, the carry chain may be seen as a logic block with two inputs, carry-kill and carry-propagate; the outputs, propagate and carry-in; vertical signals, carry-in and carry-out; and one control wire, precharge.

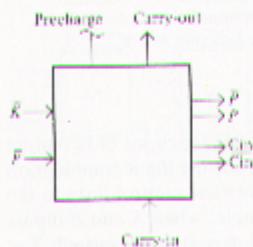


Fig. 5.4 Abstraction of the carry-chain circuit.

The task of designing the balance of the ALU is now reduced to that of designing functional blocks to (a) combine the two input variables to form a propagate-bar and kill-bar, and (b) combine carry-in and propagate to form the output signal, and then designing drivers for controlling the logic function blocks and deriving a timing for precharge.

A number of random logic implementations of function blocks for deriving kill, propagate, and the output were attempted. All seemed to be at variance with the horizontally microprogrammed architecture of the data path and required a large amount of area and power. For this reason it was decided to use the general logic function block illustrated in color Plate 7(a). Recall that the depletion mode transistors, i.e., those covered by ion implanted regions represented by yellow, are always on. Such logic function blocks are used to generate kill-bar and propagate-bar, and for combining carry-in and propagate to form the output.

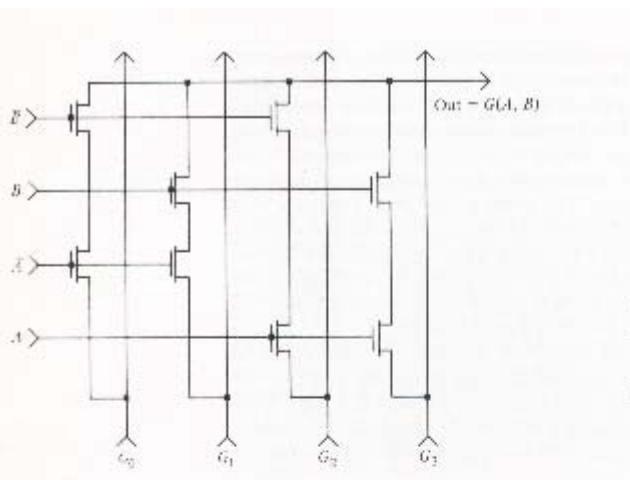


Fig. 5.5 General logic function block diagram for a transistor.

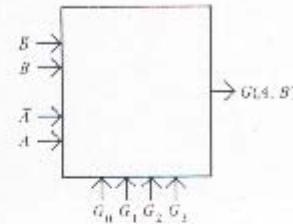


Fig. 5.6 Functional abstraction of the general logic function block.

The circuit, shown in Fig. 5.5, implements the sixteen logic functions of two input variables. It consists of a set of transistors that fully decode the input combination of A and B . The set connects one and only one of the vertical control lines to the output, depending on this input combination. For example, when A and B inputs are both low, the vertical control wire labeled G_0 is connected to the output. The truth table entries for the desired logic function are placed on the G vertical control wires, and the output is then the desired logic function of the two input variables. For example, if the Exclusive-OR of A and B is desired, a logic-0 will be applied to the control wires G_0 and G_3 , and logic-1 will be applied to control wires G_1 and G_2 . Since it is desired to implement the same logic function on all bits of the word, the control variables G_0 through G_3 need not be generated in every bit slice, but may be generated once at either the top or bottom of the array. The functional abstraction of the circuit of Fig. 5.5 is shown in Fig. 5.6. For a color-coded stick diagram of the function block, see Plate 10(a). Plate 10(b) shows a color-coded actual layout of the function block.

The block diagram for a complete arithmetic logic unit is shown in Fig. 5.7. The functional dependence of the output on the two inputs and the state of the carry is determined by a 12-bit number: P_0 through P_3 , K_0 through K_3 , and R_0 through R_3 , together with the carry-in to the least significant bit of the ALU. The ALU is quite general, and its detailed operation set may be left unbound until the control structure of the computer system is designed.

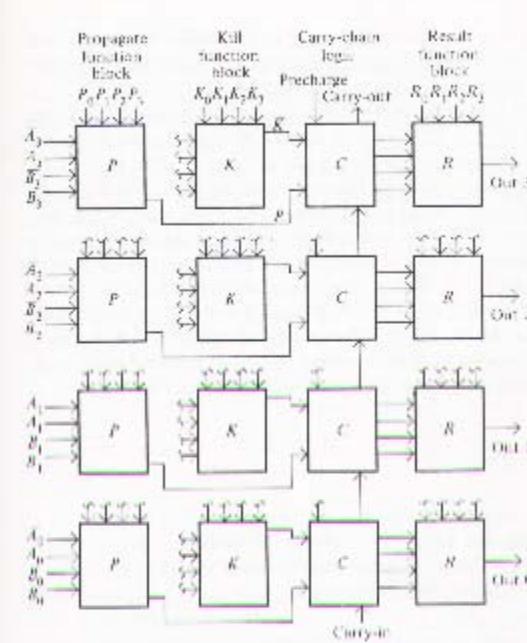


Fig. 5.7 Block diagram of a 4-bit ALU.

There are two general principles illustrated by this design. (1) It is often less expensive in area, time, and power to implement a general function than to implement a specific one. (2) If a general function can be implemented, the details of its operation can be left unbound until later, and hence provide a much cleaner interface to the next level of design. The detailed selection of which functional entities to leave unbound and which to bind early requires a considerable amount of judgment.

Two details must be dealt with before the arithmetic logic unit subsystem is complete. Drivers are needed for the $P_0 \dots P_3$, $K_0 \dots K_3$, and $R_0 \dots R_3$ control lines that will generate signals with the appropriate timing. In addition, inverters must be interposed in the carry chain occasionally to minimize the propagation delay through the entire carry chain. The way we have chosen to implement the interposition of inverters is to recognize that each carry chain function block contains two inverters that produce at their output the carry-in, having been twice inverted from the actual carry-in signal. If we merely substitute this signal for the carry-in signal to the pass transistor, we have doubly inverted our carry-in and buffered it to minimize the propagation delay. This approach avoids putting spaces

for inverters between the carry function blocks. It is illustrated by the dotted connection lines in Fig. 5.3. In the actual implementation, the connection through the inverters was made in every fourth stage (see Section 1.11).

Drivers for the P , K , and R control lines have the following function: At some time during the null period of the ALU (which occurs during φ_1), an OP code specifying the state of each control line arrives at the drivers. It must be latched while the ALU itself is being precharged, and then it must be applied to the P , K , and R control lines as soon as the ALU is activated. The P , K , and R function blocks are themselves composed of pass transistors, and their outputs are more effectively driven low than high. For this reason, we will precharge the outputs of the P , K , and R function blocks as well as the carry chain itself. This is most conveniently done by requiring that all of the P , K , and R control signals be high during the null period of the ALU. Then, independent of the states of A and B inputs, the outputs will be charged high by the time the ALU active period commences. The control driver implementing this function is shown in Fig. 5.8.

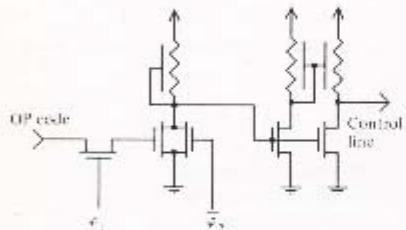


Fig. 5.8 ALU control driver. All outputs high during φ_2 ; selected terms low during φ_1 ; OP code valid during φ_1 .

The OP code is latched through a pass transistor whose gate is connected to φ_1 , and the OP code runs into a NOR gate, the other input of which is also φ_1 . Thus, the output of the NOR gate is guaranteed to be low during the φ_1 period. The NOR gate output is then run through an inverting super buffer, so that during φ_1 the output is guaranteed to be high. At the end of φ_1 , the OP code is driven onto the P , K , and R control lines. The only interface specification for the ALU that must be passed to the next level of system design is that the P , K , and R control signals be valid before the end of φ_1 , and that the A and B inputs likewise be valid by the end of φ_1 and be stable throughout φ_2 , the active period of the ALU. We are then guaranteed that after enough time has passed to allow the carry to propagate, the output of the R function block will accurately reflect the specified function of the ALU and may be latched at the end of φ_2 .

A color-coded plot of the layout of a 1-bit slice through this ALU is shown in color Plate 11. This ALU went through another design iteration before inclusion in the OM2. A plot of the ALU control driver is shown in color Plate 12.

5.6 ALU REGISTERS

In order for the arithmetic logic unit described in the last section to be useful, it must be equipped with a set of registers both for its input variables and for its output. Let us consider the input registers first. Inputs to the ALU may be derived from either the shifter, the buses, or other sources. They may be latched and left unchanged during any $\omega_1\omega_2$ machine cycle or set of machine cycles. This is one of the situations in which combining the multiplexing function with the latching function simplifies the design and achieves better performance. A register operating in this manner is shown in Fig. 5.9.

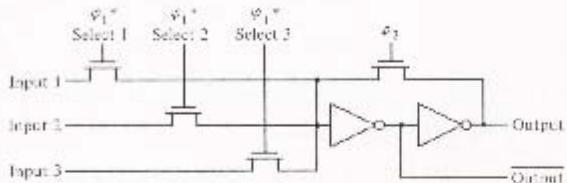


Fig. 5.9 ALU input register and multiplexer.

The input to the first inverter can be derived from four sources: three external sources such as shifter output, bus, etc., and a fourth, the output of the second inverter. When it is desired to latch a new signal into the register, one of the source pass transistors is driven on during φ_1 . The feedback transistor around the two inverters is always activated during φ_2 . Thus, with three vertical control wires plus the φ_2 timing signal, it is possible to select one of three sources into the register, or none of the three sources, thereby leaving the previous value of the register stored on the gate of the first inverter during the φ_1 period. Since it is necessary to have two inverters to form the stable pair when the feedback transistor is on, both the input and its complement are available as required by the P and K function blocks of the arithmetic logic unit. The OP code signal that selects the source that will be applied to the ALU input register during φ_1 must enter the chip during the previous φ_2 . Each of the select signals must be low during φ_2 , and at most one of them may come high during the following φ_1 . A driver appropriate for these control signals is shown in Fig. 5.10, with the corresponding layout shown in color Plate 12. The control OP code is latched during φ_2 , during which time the NOR gate shown disables the output driver. Since the output driver in this case is noninverting, the output select line is held low during all of φ_2 . At the end of φ_2 , the OP code signal is latched and at the beginning of φ_1 the particular select line to be enabled that cycle is allowed to go high.

Note that this timing allows two incoming OP code bits per external wire per machine cycle. In particular, if it were desirable to share a microcode bit between the ALU function and the ALU selector inputs, this could be done by bringing the ALU OP code in during φ_1 and the ALU input selection code in during φ_2 . This technique was suggested by Ivan Sutherland.

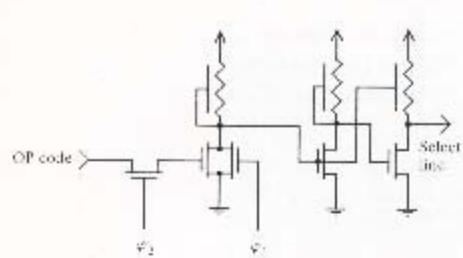


Fig. 5.10 Select control driver. All outputs low during φ_2 ; selected terms high during φ_1 ; OP code valid during φ_2 .

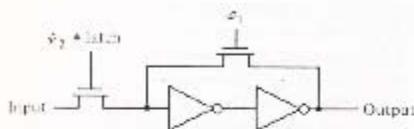


Fig. 5.11 Output register.

The ALU output register is similar to the ALU input register, except that the timing is reversed. The result of the ALU operation is available at the end of φ_2 . An OP code bit will, if desired, enable the latch signal to go high during φ_2 . The feedback transistor is always enabled during φ_1 , and thus the latch is effectively static even though in the absence of a latching signal the data is stored dynamically on the gate of the first inverter through the φ_2 period. Once again, both the output and its complement are available if desired.

5.7 BUSES

An early design decision was made to have data flow through the data path chip on two buses that communicate with all of the major blocks of the system. We have already seen that the ALU performs its operation during the φ_2 period and does not have valid data to place into its output register until the end of φ_2 . If data are to be transferred from the output register of the ALU to its input register, this must be done during the φ_1 period. If we adopt a standard timing scheme in which all transfers on the buses occur during φ_1 , we can make use of the φ_2 period when the ALU is performing its operation to precharge the buses in the same manner that the carry chain was precharged during the φ_1 period. In this way we solve one of the knotty problems associated with a technology designed for ratio logic. If we had insisted that the tri-state drivers associated with various sources of data for a bus be able to drive up as well as down, we would have required both a sourcing and sinking transistor, together with a method for disabling both transistors. While it is perfectly possible to build such a driver (we shall undertake the exercise as part of the design of the output ports), it is a space-consuming matter to use such a driver at every point where we wish to source data onto an internal bus.

By using the bus precharge scheme, our "tri-state drivers" become simply two series transistors as shown in Fig. 5.12. Here the data from one source, for example, the ALU output register, is placed on the gate of one of the series transistors. An enable signal, which may come high during φ_1 , is placed on the other series transistor. If one and only one of the enable signals is allowed to come high during any one φ_1 period, the bus can be driven from as many sources as necessary. The performance of such a bus is limited by the pull down capability of the two series transistors. We attach such a driver to each of the output registers for the ALU.

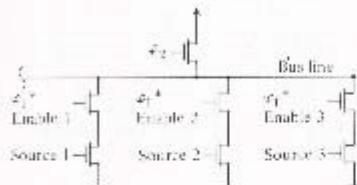


Fig. 5.12 Precharged bus circuit.

5.8 BARREL SHIFTER

Since shifting is basically a simple multiplexing function, one might think that a shifter could be combined with the input multiplexer to the ALU. A simple 1-bit, right-left shifter implemented in this manner is shown in Fig. 5.13. It is identical with the three-input ALU register, with the three inputs used to select among the bus, the bus shifted left by one, and the bus shifted right by one. To support the

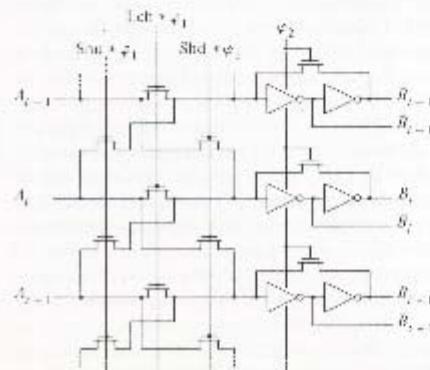


Fig. 5.13 A simple 1-bit, right-left shifter.

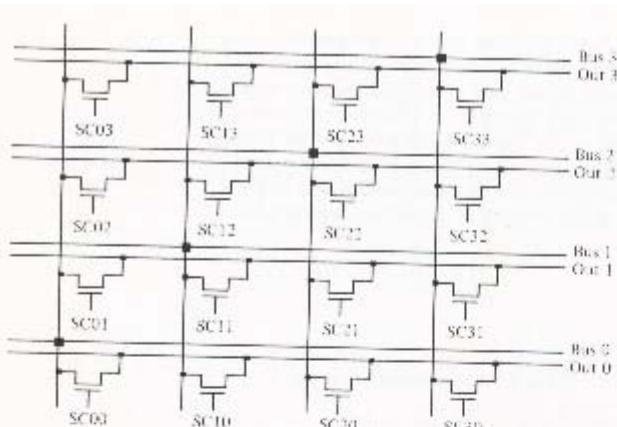


Fig. 5.14 A 4 by 4 crossbar switch.

multibit shifts necessary for field extraction and building up odd bit arrays, something more is required. One is tempted initially to build up a multibit shift out of a number of single shifts. However, for word lengths of practical interest, the n^2 delay problem (see Section 1.11) makes such an approach unworkable.

The basic topology of a multibit shift dictates that any bus bit be available at any output position. Therefore, data paths must run vertically at right angles to the normal bus data flow. Once this simple fact is squarely faced, a multibit shifter is seen as no more difficult than a single bit shifter. A circuit enabling any bit to be connected to any output position is shown in Fig. 5.14. It is basically a crossbar switch with individual MOS transistors acting as the crossbar points, the basic idea being that each switch SC_{ij} connects bus i to output j . In principle this structure can be set to interchange bits as well as to shift them, and it is completely general in the way in which it can scramble output bits from any input position. In order to maintain this generality, the control of the crossbar switch requires n^2 control bits. In some applications, the n^2 bits may not be excessive, but for most applications a simple shift would be adequate. The gate connections necessary to perform a simple barrel shift are shown in Fig. 5.15. The shift constant is presented on n wires, one and only one of which is high during the period the shift is occurring. If the shifter's output lines are precharged in the same manner as the bus, the pass transistors forming the shift array are required to pull down the shifter's outputs only when the appropriate bus is pulled low. Thus, the delay through the entire shift network is minimized and effective use is made of the technology.

A second topological observation is that in every computing machine, it is necessary to introduce literals from the control path into the data path. However,

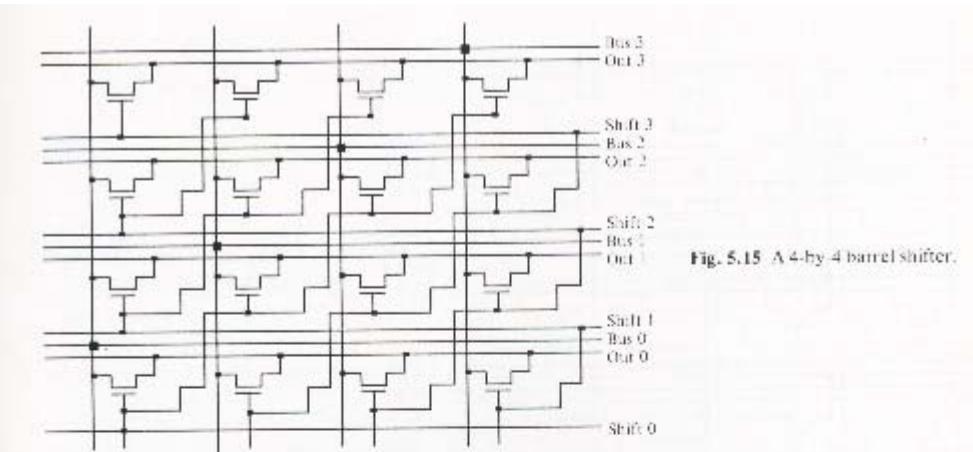


Fig. 5.15 A 4-by-4 barrel shifter.

our data path has been designed in such a way that the data bits flow horizontally while the control bits from the program store flow vertically. In order to introduce literals, some connection between the horizontal and vertical flow must occur. It is immediately obvious in Fig. 5.15 that the bus is available running vertically through the shift array. That is then the obvious place to introduce literals into the data path or to return values from the data path to the controller.

At the next higher level of system architecture, the shift array bit slice may be viewed as a system element with horizontal paths consisting of the bus, the shifter output, and if necessary, the shift constant since it appears at both edges of the array. The literal port is available into or out of the top edge of the bit slice, and the shift constant is available at the bottom of the bit slice. These slices, of course, are stacked to form a shift array as wide as the word of the machine being built.

One more observation concerning the multibit shifter is in order. We stated earlier that our data path was to have two buses. Therefore, in our data path, any bit slice of a shifter such as the one shown in Fig. 5.15 will of necessity have two buses running through it rather than one. We chose to show only one for the sake of simplicity. There remains the question of how the two buses are to be integrated with the shifter. Since we are constructing a two-bus data path, we have two full words available, and a good field extraction shifter would allow us to gracefully extract a word that crosses the boundary between two data path words. The arrangement shown in Fig. 5.15 performs a barrel shift on the word formed by one bus. Using the same number of control lines and pass transistors, and adding only the bus lines that are required for the balance of the data path anyway, we may construct a shifter that places the words formed by the two buses end to end and extracts a full-width word that is continuous across the word boundary between

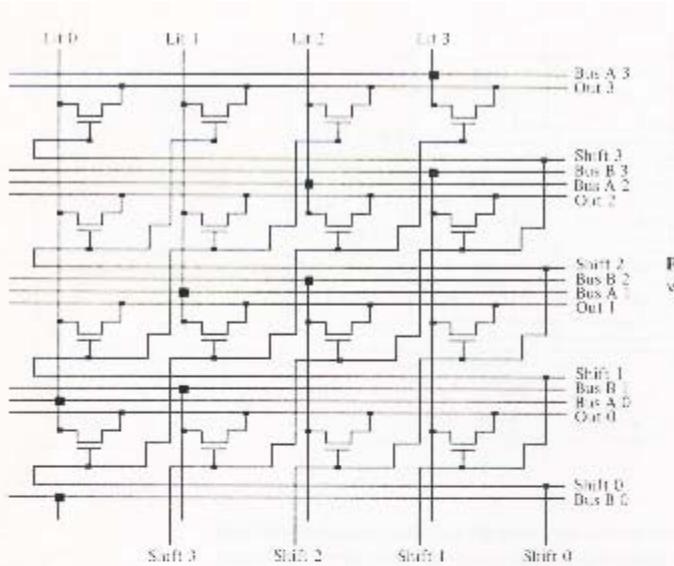


Fig. 5.16 A 4-by-4 shifter with split vertical wires and two data buses.

the A and B buses. This function is accomplished, in as compact a form as just described, with a circuit shown in Fig. 5.16. Notice that the vertical wires have a split in them. The portion of the wire above the corresponding shift output is connected to bus A, and that below the corresponding shift output to bus B. The layout of the barrel shifter is shown in color Plate 13.

It can be seen by inspection that this circuit performs the function shown in Fig. 5.17, which is just what is required for doing field extractions and variable word length manipulations. The literal port is connected directly to bus A and may be run backward in order to discharge the bus when a literal is brought in from the control port. A block diagram representing the shifter at the next level of abstraction is shown in Fig. 5.18.

In order to complete the shifter functional block, it is necessary to define the drivers on the top and bottom that interface with the system at the next higher level. Let us assume that the literal bus from outside the chip will contain data valid on the opposite phase of the clock from that of the internal buses. For that case, a very simple interface between the two buses that will operate in either direction is shown in Fig. 5.19.

The internal shifter output is precharged during φ_2 , and active during φ_1 . It may be sourced from the shifted combination of either the A and B buses or the literal bus and B bus, as shown in Fig. 5.17. The external literal bus itself may be

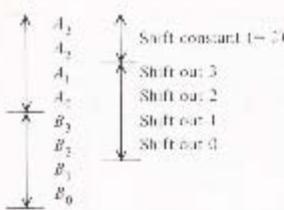


Fig. 5.17 Conceptual picture of the shifter's operation.

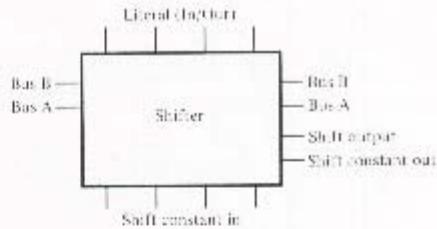


Fig. 5.18 Block diagram of the shifter.

sourced either from the opposite end (the external paths from the program source) or from the end attached to bus A in the shift array shown.

The bus to the external literal path is precharged during φ_1 , and data bits from the literal port of the shifter are enabled onto it by a signal active during φ_2 , as shown in Fig. 5.19. The two signals, $\varphi_1 \cdot \text{In}$. and $\varphi_2 \cdot \text{Out}$, are derived from buffers identical to those shown earlier. The shift constant itself is represented by one line out of n , which is high, the others remaining low. Buffers for these lines are identical to those shown in Fig. 5.10.

There is one more observation concerning the n -bit shift constant. It is represented most compactly by a log n bit binary number. However, in order to generate from such a form a signal that can be used in the actual data path, a decoder is required for converting the binary number into a 1-of- n signal suitable for feeding the buffers. There are a number of ways of making decoders in n MOS technology. The most common form is the NOR form, which is the fully decoded equivalent of the AND plane in the programmable logic array (Chapter 3). It is shown in Fig. 5.20. Notice that the output is a high-going 1-of- n pattern.

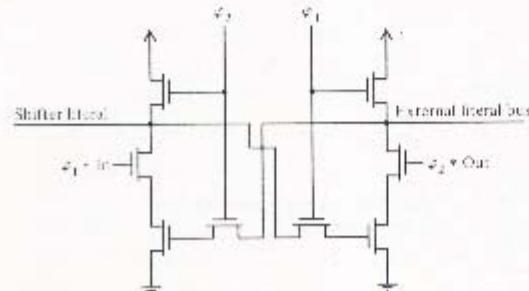


Fig. 5.19 Literal interface

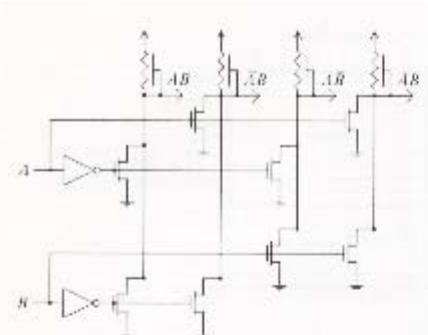


Fig. 5.20 A NOR form 1-of- n decoder.

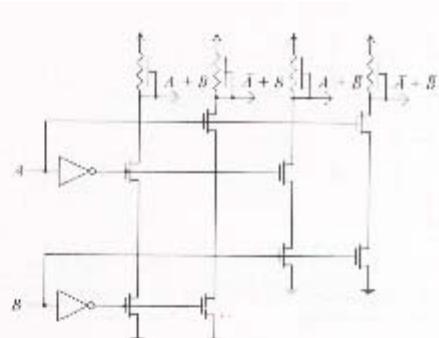


Fig. 5.21 A NAND form 1-of- n decoder.

Decoders can also be made in other forms. For small values of n , the NAND form shown in Fig. 5.21 is often convenient. We used a variant of this form for the ALU function block described earlier. Notice that the output of this form, when used as a decoder, is a low-going 1-of- n pattern. There is also a complementary form of decoder that can be built with this technology, as suggested by Ivan Sutherland. It takes advantage of the fact that in any decoder both the input term and its complement must be present. In this case, the input term can be used to activate pull-up transistors in series, while the complement can be used to activate pull-down transistors in parallel. This logic form is similar in principle to that used with complementary technologies and has similar benefits. It can generate either a high-going or a low-going 1-of- n number and dissipates no static power. A decoder of this sort is shown in Fig. 5.22. Once we have added the appropriate buffers and decoders to our shift array, we have a fully synchronized subsystem ready to be integrated with the system at the next level up. The properties of this subsystem are shown in Fig. 5.23.

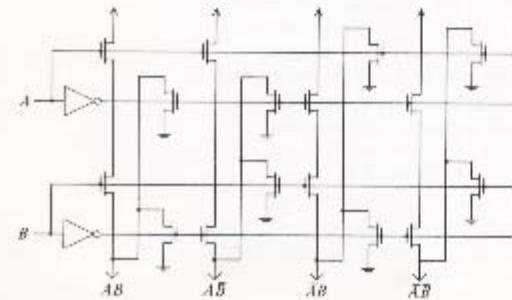


Fig. 5.22 A complementary form 1-of- n decoder.

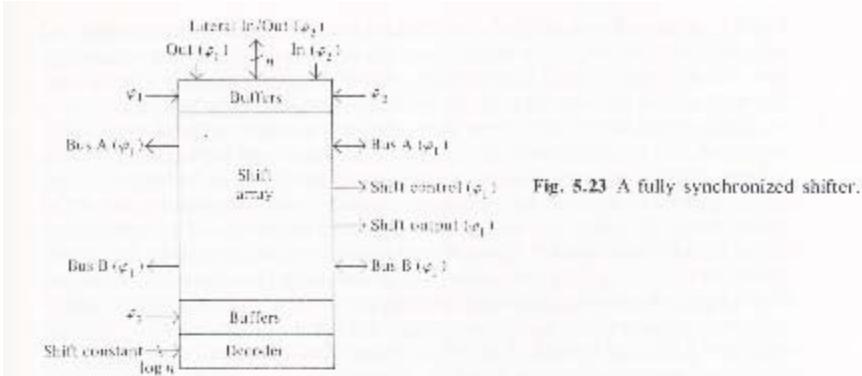


Fig. 5.23 A fully synchronized shifter.

5.9 REGISTER ARRAY

In any microprogrammed processor designed for emulating a higher level instruction set, it is convenient to have a number of miscellaneous registers available, both for working storage during computations and for storing pointers (stack pointers, base registers, program counters, etc.) of specific significance in the machine being emulated. Since the data path has two buses and the ALU is a two-operand subsystem, it is convenient if the registers in the data path are two-port registers. The circuit design of a typical two-port register cell is shown in Fig. 5.24. The layout of a pair of these cells is shown in color Plate 14. This register is a simple combination of the input multiplexer described earlier, the φ_2 feedback transistor, and two output drivers, one for each bus. The registers can be combined into an array m bits long and n bits wide. Each cell of the array can be viewed at the next level up as shown in Fig. 5.25. Drivers for the load inputs and the read outputs are

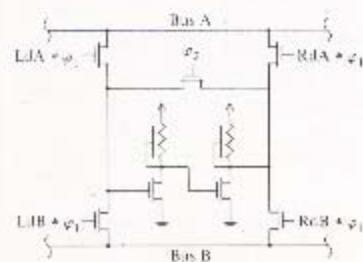


Fig. 5.24 A two-port register cell.

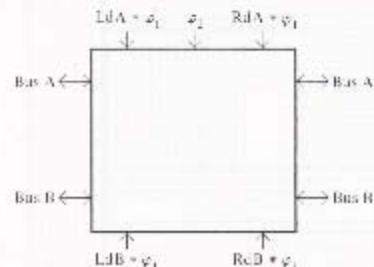


Fig. 5.25 Block diagram definition of the two-port register cell.

identical to those shown in Fig. 5.10. While we could immediately encode the load and read inputs to the registers into $\log n$ bits, we shall delay doing so until the next level of system design. There are a number of sources for each bus besides the registers, and we will conserve microcode bits by encoding them together.

Before we proceed, there is one important matter that must be taken care of in the overall topological strategy. Routing of VDD and ground paths must generally be done in metal, except for the very last runs within the cells themselves. Often the metal must be quite wide, since metal migration tends to shorten the life of conductors if they operate at current densities much in excess of 1 milliamperc per square micron cross section. Thus, it is important to have a strategy for routing ground and VDD to all the cells in the chip before doing the detailed layout of any of the major subsystems. Otherwise, one is apt to be faced with topological impossibilities because certain conductors placed for other reasons interfere with the routing of VDD and ground. A possible strategy for the overall routing of VDD and ground paths is shown in Fig. 5.26. Notice that the VDD and ground paths form a set of interdigitated combs, so that both conductors can be run to any cell in the chip. Any such strategy will do, but it must be consistent, thoroughly thought through at the beginning, and rigidly adhered to during the execution of the design.

5.10 COMMUNICATION WITH THE OUTSIDE WORLD

Although in particular applications the interface from a port of the data path to the outside world may be a point-to-point communication, the ports will often connect

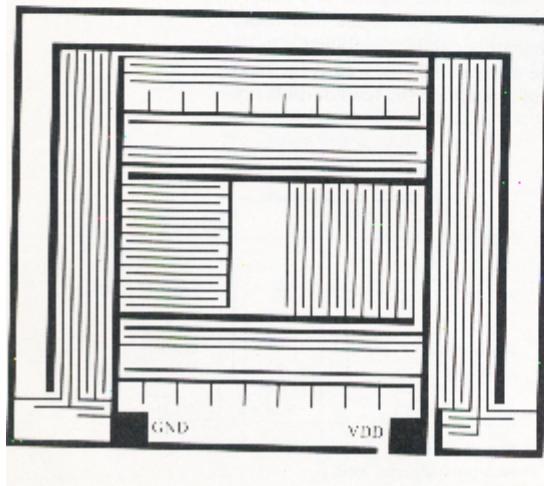


Fig. 5.26 VDD and GND net for the data path chip.

to a bus. Thus it is desirable to use port drivers that may be set in a high impedance state. Drivers that can either drive the output high, drive the output low, or appear as a high impedance to the output are known as *tri-state* drivers. Such drivers allow as many potential senders on the bus as necessary. Figure 5.27 shows the circuit for a tri-state interface to a bonding pad. Here, either bus A or bus B can be latched into the input of a tri-state driver during φ_1 . Likewise the pad may be latched into an incoming register at any time independent of the clocking of the chip. Standard bus drivers are enabled on bus A and bus B. The only remaining chore is the design of the tri-state driver that drives the pad directly. Details of the tri-state driver are shown in Fig. 5.28. The layout of an output pad and its associated driver circuitry is shown in color Plate 15.

The terms *out* and *outbar* are fed to a series of buffer stages that provide both true and complement signals as their outputs and are disabled by a *disable* signal. Note that this Disable signal does not cause all current to cease flowing in the drivers, since the pull-up transistors are depletion type. In general there will be a number of super buffer stages of this sort. The very last stage of the driver is shown in Fig. 5.28(b). It is not a super buffer but employs enhancement mode transistors for both pull-up and pull-down. These transistors are very large in order to drive the large external capacitance associated with the wiring attached to the pad. They are disabled in the same manner as the super buffers, except that when the gates of both transistors are low, the output pad is truly tri-stated. The

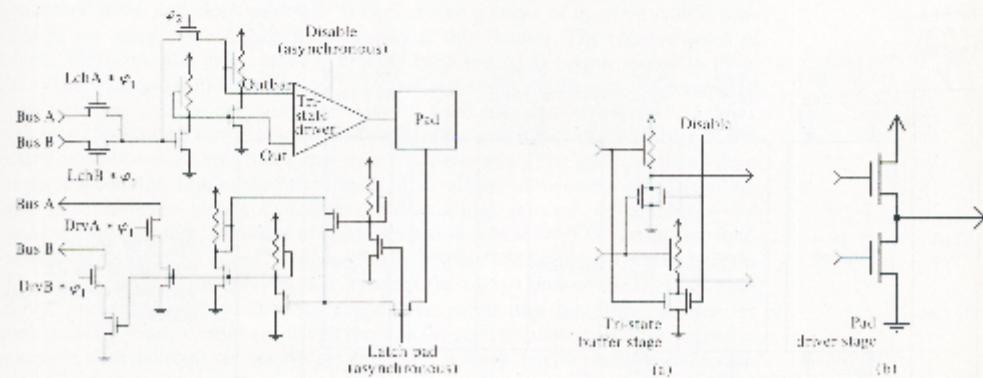


Fig. 5.27 Data port tri-state pad circuit.

Fig. 5.28 The tri-state driver consists of any number of tri-state buffer stages followed by a pad driver stage. The current design used two tri-state buffer stages.

two output transistors are a factor of approximately e larger than the last super buffer in the buffer string.

As we have seen, a rather large inverter string is required to transform the impedance from that of the internal circuits on chip to that sufficient for driving a pad attached to wiring in the outside world; the large size imposes a delay, of some factor times a logarithm of this impedance ratio, upon communications between the chip and the outside world. Any help that can be obtained in making this transformation is of great value. For example, the latch and buffers associated with the input bus circuit to the pad drivers can themselves be graded in impedance level, so that by the time the out and outbar signals are derived, they are at a considerably higher current drive capability than minimum nodes on the chip, and thus the initial latch buffers can be larger than typical inverters on the chip. All such tricks help to minimize the number of stages between the bus and the outside pad and consequently the total delay in going off-chip.

5.11 ENCODING THE CONTROL OPERATION OF THE DATA PATH

By now we have defined a complete functional data path with ports on each end and functional blocks through the center, as shown in Fig. 5.29. The data path operation code bits required to control the data path are shown, as is the phase of the clock on which they are latched. There are forty-nine such bits together with

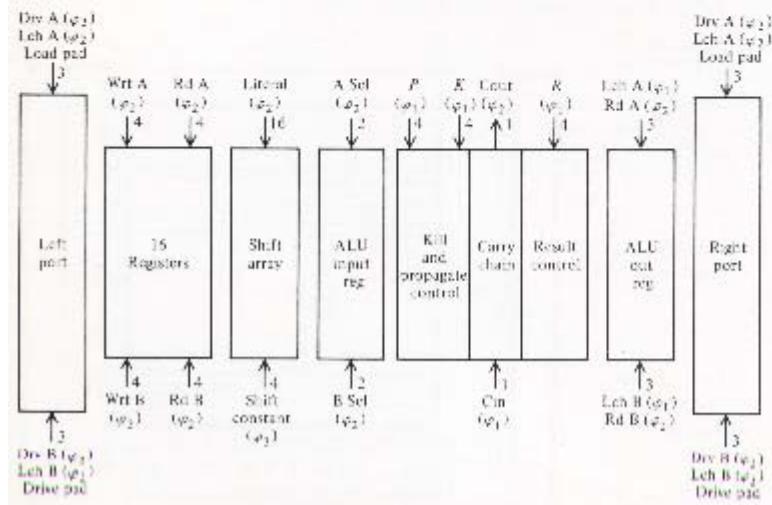


Fig. 5.29 Block diagram of the data path with control wires added.

the four asynchronous bits for latching and driving the pad to the external world. In addition, there are the carry-out wire and the sixteen literal wires. These sixty-six wires together with the thirty-two from the left and right ports must go to and come from somewhere. (Schemes for encoding internal data path operations into microinstructions of various lengths are discussed in Chapter 6.) At one extreme all the data path control wires can be brought out to a microcode memory driven by a microprogram counter and controller, in which case all operations implementable by the data path may be done in parallel. The opposite extreme is to tightly encode the operations of the data path into a predefined microinstruction set. In the present system, this encoding would be most conveniently done by placing a programmable logic array or set of programmable logic arrays along the top and the bottom of the data path. A condensed microinstruction could then be fed to the programmable logic arrays that would then decode the compact microinstruction into the data path operation code bits.

An important point of the design strategy used here is that we can orthogonalize the design of the data path and the design of the microinstruction set in such a way that the interface between the two designs is not only very well defined and very clean, but it can be described precisely, in a way that system designers at the next higher level can understand and work with comfortably. The data path can then be viewed as a component in the next level system design.

Using the approximate capacitance values given in Table 2.1, we can estimate the minimum clock period for sequencing the data path. We would expect a φ_1 time for the data path of $\approx 50\tau$ (same as the general estimate given in Section 1.13 on transit times and clock periods). However, the φ_2 time of the data path is limited by the carry chain, as discussed earlier in this chapter. The relative areas of metal, diffusion, and gate can be estimated from the ALU layout shown in Plate 11. The metal and diffusion occupy ≈ 15 and ≈ 8 times the area of the propagate pass transistor gate, respectively. Metal is ≈ 0.1 and diffusion is typically 0.25 times the gate capacitance per unit area. Thus the total capacitance of each stage of the carry chain is ≈ 4.5 times that of the pass transistor gate. The effective delay time is correspondingly longer than the transit time τ of the transistor itself. The effective delay through n stages of such pass transistor logic is $\approx n^2\tau$. In the OM2, $n = 4$ and the effective delay for 4 bits of carry chain is $\approx 4.5 \cdot 16\tau = 72\tau$. To this must be added the delay of the doubly inverting buffers at the end of every 4 bits of straight Manchester logic. This delay is $(1 + k)$ times the transit time of the inverter pull-down, properly corrected for stray capacitance in the inverter. Here the inverter ratio k is ≈ 8 , since its input is driven through the pass transistors. Conservatively, strays in such a circuit are always several times greater than the basic gate capacitance, and we may estimate the inverter delays at $\approx 30\tau$. Our estimate for the total carry time is thus ≈ 100 times the transit time for each block of 4 ALU stages. The total φ_2 time should then be $\approx 400\tau$. In 1978, the fastest commercial n MOS processes yield a transit time τ of ≈ 0.3 ns, and we would expect a minimum total clock period of $\approx 450\tau$, or ≈ 135 ns.

5.12 FUNCTIONAL SPECIFICATION OF THE OM2 DATA PATH CHIP

5.12.1 Introduction

This specification describes a 16-bit data path chip referred to as OM2 [#986]. The OM2 contains 16 registers, an ALU, and a 16-bit shifter, and is designed as part of a microprogrammed writable-control-store digital computer. The companion chip is the controller chip, which contains the program counter, stacks, and so on. The controller is described in Chapter 6. The entire system is designed to run on a single 5-volt supply.

The OM2 data chip has two data ports for communication with the external system and a communication path to the controller chip. The data ports are tri-state with either internal or external control. Communication with the controller consists of a 16-bit literal port and a single flag bit. Seven control bits come directly from the microcode memory.

The system runs on a single clock, generating φ_1 and φ_2 internally. When the clock is high, the internal buses transfer data; when the clock is low, the ALU is performing its operation. Microcode bits enter the data chip the phase before that code is to be executed. Therefore, the bus transfer code enters the data chip when the clock is low, and the ALU code enters when the clock is high. Figure 5.30 sketches a possible OM system.

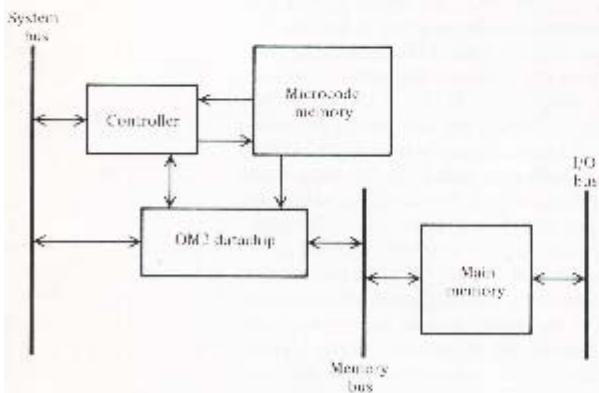


Fig. 5.30 One possible OM2 system configuration.

⁴Section 5.12 contains a functional specification of the OM2 data path chip, contributed by Dave Johannsen of California Institute of Technology. This specification was originally documented in Display File #1111, by Dave Johannsen and Carver Mead of the Caltech Computer Science Department, and copyrighted by Caltech. The specification is reprinted here with the permission of the California Institute of Technology. See also the later document, "Our Machine: A Microcoded LSI Processor," Display File #1826, by Dave Johannsen, Caltech Computer Science Department, for a general description of the OM System.

Throughout this section a positive logic convention is used. A "1" refers to a high voltage level, while a "0" refers to a low voltage level.

5.12.2 Data paths

A block diagram of OM2 is shown in Fig. 5.31. There are two buses that connect the various elements of the chip. The buses transfer data while the clock is high, the period referred to as φ_1 . During φ_2 , when the clock is low, the buses are pre-charged. Each bus can get data from only one source and give data to only one destination during any one cycle.

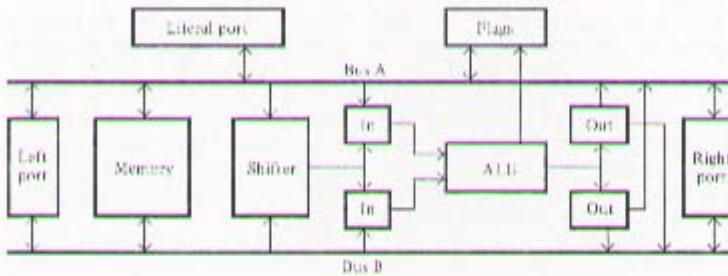


Fig. 5.31 Block diagram of the OM2.

The left and right ports communicate between the data chip and the outside world. The right port has been traditionally known as the memory bus port while the left port has been the system bus port, but since the two ports are identical, this is an arbitrary convention. Each port has both an input latch and an output latch to provide facilities for synchronizing the data chip to the outside buses. Under program control either of the two buses can load the output latch during φ_1 . There are three modes of driving data from the output latch to the pins, two of which are under program control and one of which is under hardware control. The first method is to output the data as soon as it comes from the bus, during the same φ_1 . The second method is to latch the data from the bus during φ_1 and drive it out during the following φ_2 . The final method is to latch the data from the bus during φ_1 , but output the data when an enable pin is pulled low. The enable pin would be controlled by a bus manager, and can be asynchronous with respect to the data chip. Inputting from the port is similar. By pulling down on another enable pin, data bits from the external bus are loaded into the input latch, which can be read later under program control. Alternatively, the microcode can force the data currently on the external bus into the internal bus during the current φ_1 . With this scheme, many types of synchronous and asynchronous buses may be interfaced to OM2's. For internal control only, the external enable pins can be left floating.

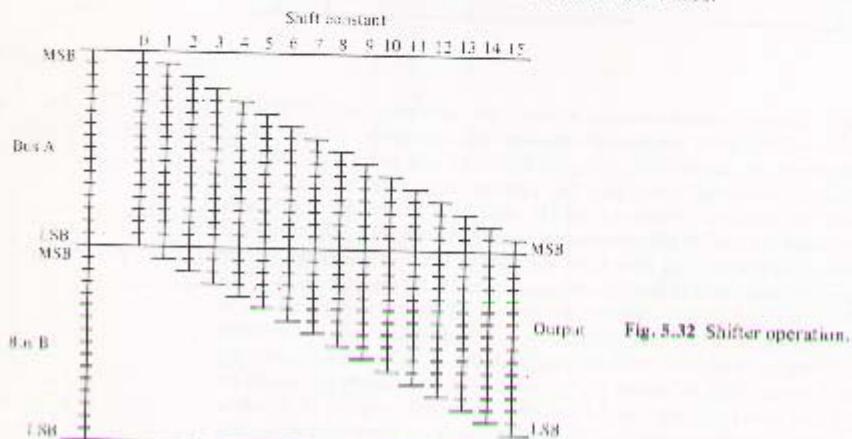
5.12.3 Registers

The registers are static and dual port. Any one of the 16 registers may source either or both of the buses, while any one of the 16 may be the destination for either bus, but not both. There are only two restrictions on the use of the registers;

1. One register may not be the destination for both buses on the same cycle.
2. One register may not be both the source for one bus and the destination for the other bus on the same cycle.

5.12.4 Shifter

The shifter concatenates the two buses, resulting in a 32-bit word, with bus A being the more significant half. The shift constant then selects the bit position where the 16-bit output window starts. The shift constant specifies the number of bits from bus B present in the output; i.e., a shift constant of 0 returns bus A, while a shift constant of 15 returns the LSB of bus B in the MSB of the output, followed by all but the LSB of bus B in the rest of the word. A conceptual picture of the shifter is shown in Fig. 5.32. The ALU can select as inputs either the bus, the shift output, or shift control. If shift control is selected, the entire word is 0 except where the LSB of bus A appears in the shift output. The shifter operates on φ_1 ; it may be viewed as an extension of the buses.



5.12.5 ALU

A block diagram of a single bit of the ALU is shown in Fig. 5.33. The ALU operates on the data that is contained in its two input latches. Input latch A may be

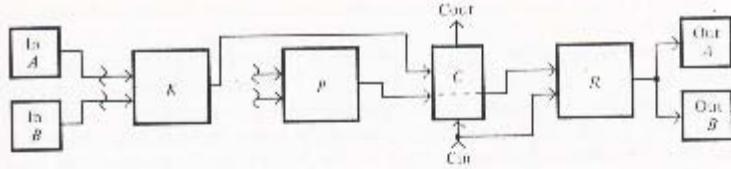


Fig. 5.33 Block diagram of one bit of the ALU.

loaded from bus A, the shifter output, or the shift control, while the input latch *B* may be loaded from bus B, the shifter output, or the shift control.

The outputs of the two latches become the inputs to two function blocks that determine what will happen on the carry chain. Function block *P* determines whether the carry chain propagates, while *K* decides if it is to kill the carry. If neither is true, the carry chain generates a carry. Each function block has four control inputs, which, for the propagate function block, are referred to as *PFF*, *PFT*, *PTF*, and *PTI*. If *PFF* is enabled, the *P* block output is high if both input latches are false (contain 0). Enabling *PFT* activates the output if input *A* is false and input *B* is true, and so on. If, for example, both *PFF* and *PFT* are enabled, the output is active if input *A* is false, regardless of the state of input *B*. To further illustrate the operation of the function blocks, consider addition. If both inputs contain a 1, the carry is to be generated, while if both inputs are 0, the carry is killed. If the two inputs are different, the carry is to be propagated (carry-out=carry-in). To do this operation, the kill output should be active if both inputs are false, so *KFF* is enabled. Both *PTI* and *PTF* should be enabled to propagate properly. Therefore, *K* = (*KFF*, *KFT*, *KTF*, *KTT*) = (1,0,0,0), and *P* = (*PFF*, *PFT*, *PTF*, *PTI*) = (0,1,1,0).

The result of the ALU is produced by the *R* function block, which has as inputs *P*-block out and carry-in. For the addition example above, the output should be the exclusive-OR of *P* and *Cin*, so *R* = (0,1,1,0). *P*, *K*, and *R* values for common ALU operations are listed in Table 5.2.

Two ALU output latches (*A* and *B*) can be loaded from the *R* block output; either one may later be used to source either bus.

5.12.6 Flags

The carry input to the LSB of the ALU is a logical combination of a flag bit and two control inputs. The two control inputs can force the carry-in to be either 1 or 0, or they can select either flag or flag bar as the input.

There is also a method for doing conditional ALU operations under the control of a 2-bit conditional OP field. A conditional operation performed by the ALU is not only a function of the control inputs but also of the flag bit. The conditional operation control forces some of the control inputs low, regardless of what the *P*, *K*, and *R* microcode says. The coding for conditional operations allows the use of operations like multiply step and divide step without the necessity for branching in the microcode.

There is a 16-bit flag register that can also be a source or destination of bus A. This register can also be loaded with the ALU flags during φ_2 . The ALU flags include *carry-out*, *overflow*, *carry-in to the MSB*, *zero*, *MSB*, *LSB*, *less than*, *less than or equal to*, and *higher* (in unsigned value). The last three flags are comparison flags used after a subtraction. For example, after subtracting ALU input latch B from latch A, the "less than" flag is true if the value in ALU input latch B is larger than the value in ALU input latch A. The MSB of the flag register is called the flag bit, and this bit may be modified every φ_1 by loading it with the value of one of the other bits of the flag register. The flag bit is used in the calculation of carry-in and modification of conditional ALU OPs. This bit is also sent to the controller chip to be used for conditional branching, etc.

5.12.7 Literal

The one remaining data path is the literal port. It is used to send data from the data chip to the controller, and vice versa. It is a source or destination for bus A. When the literal port is being used, standard bus operations are suspended for that cycle.

5.12.8 Programming

The data chip requires 23 bits of microcode on each phase of the clock. This section of the memo specifies the encoding of the fields within that microcode. Figure 5.34 shows the arrangement of the microcode word.

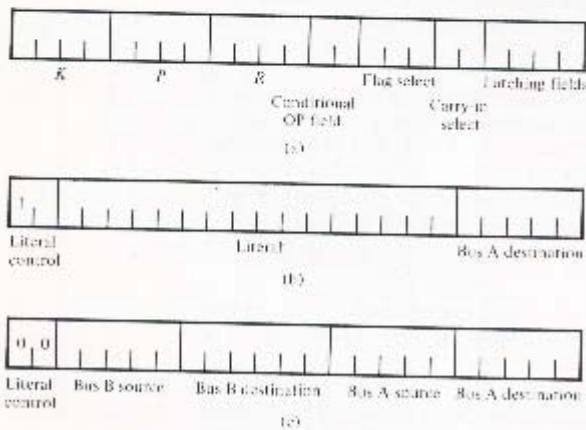


Figure 5.34
(a) Phase 2 OP code (in on φ_1).

(b) Phase 1 literal transfer OP code (in on φ_2).

(c) Phase 1 normal OP code (in on φ_2).

Bus Transfer

The bus transfer control bits enter the data chip during φ_2 and are used during the following φ_1 . There are two buses, bus A and bus B, that interconnect the modules

Table 5.1

	Bus A Source	Bus A Destination
<i>0annn</i>	Register n	0nnnn Register n
10000	Right port pins	10000 Left port, drive now
10001	Right port latch	10001 Left port, drive φ_2
10010	Left port pins	10011 Left port, no drive
10011	Left port latch	10100 Right port, drive now
10100	ALU output latch A	10101 Right port, drive φ_2
10101	ALU output latch B	1011x Right port, no drive
10110	Flag register	11000 ALU input latch A 11001 ALU input latch A gets shift out 11010 ALU input latch A gets shift control 11011 Flag register
	Bus B Source	Bus B Destination
<i>0annn</i>	Register n	0nnnn Register n
10000	Right port pins	010000 Left port, drive now
10001	Right port latch	010001 Left port, drive φ_2
10010	Left port pins	01001x Left port, no drive
10011	Left port latch	010100 Right port, drive now
10100	ALU output latch A	010101 Right port, drive φ_2
10101	ALU output latch B	01011x Right port, no drive 0110xx ALU input latch B 1Dannn ALU input latch B gets shift output, shift constant = n 1Dannn ALU input latch B gets shift control, shift constant = n

of the data chip. These two buses are similar in many respects; however, there are a few asymmetries as to sources and destinations. Also, when a literal is being transferred, the only bus transfer field that is active is the bus A destination, which stores the literal entered on bus A. A listing of bus sources and destinations appears in Table 5.1.

ALU Input Selection

The two ALU input latches are destinations for the two buses, as shown above. In addition to being loaded directly from the buses, these two latches can be loaded from the outputs of the shift array. The shift constant always comes from the four least significant bits of the bus B destination field, even though the destination of bus B is not the ALU input latch B. For example, bus B may be transferring the contents of register 3 into register 5 while bus A is transferring the contents of register 4 to the ALU input latch A through the shifter. In this case, the shift constant would be "5" because the four least significant bits of the bus B destination field contain "0101".

Table 5.2

	<i>K</i>	<i>P</i>	<i>R</i>	Cin	Cond
$A + B$	1	6	6	0	Add
$A + B + \text{Cin}$	1	6	6	1	Add with carry
$A - B$	2	9	6	2	Subtract
$B - A$	4	9	6	2	Subtract reverse
$A - B - \text{Cin}$	2	9	6	1	Subtract with borrow
$B - A - \text{Cin}$	4	9	6	1	Subtract reverse with borrow
$-A$	12	3	6	2	Negative <i>A</i>
$-B$	10	5	6	2	Negative <i>B</i>
$A + 1$	3	12	6	2	Increment <i>A</i>
$B + 1$	5	10	6	2	Increment <i>B</i>
$A - 1$	12	3	9	2	Decrement <i>A</i>
$B - 1$	10	5	9	2	Decrement <i>B</i>
$A \wedge B$	0	8	12	0	Logical AND
$A \vee B$	0	14	12	0	Logical OR
$A \oplus B$	0	6	12	0	Logical EXOR
$\neg A$	0	3	12	0	Not <i>A</i>
$\neg B$	0	5	12	0	Not <i>B</i>
<i>A</i>	0	12	12	0	<i>A</i>
<i>B</i>	0	10	12	0	<i>B</i>
Mul	1	14	14	0	Multiply step
Div	3	15	15	0	Divide step
<i>A</i> : <i>O</i>	0	14	12	0	Conditional AND:OR
Mask	10	5	8	2	Generate mask
Sll, <i>A</i>	3	0	10	0	Shift <i>A</i> left
Zero	0	0	0	0	Zero

ALU Operations

Table 5.2 shows coding for ALU operations that are commonly found useful. The user is encouraged to encode other operations if these are not suitable. The numbers given are the decimal representation of the 4-bit control word. For *P* and *K*, $A'B' = 1$, $A'B = 2$, $AB' = 4$, $AB = 8$. For *R*, $P'C' = 1$, $P'C = 2$, $PC' = 4$, $PC = 8$. Cin is the carry-in select, and Cond is the conditional OP select.

Carry-In Select

The carry-in select field determines what the carry into the 1.SB of the ALU will be, according to the following table:

00	0
01	Flag bit
10	1
11	Flag bit complemented

Conditional OP Select

The conditional OP select field is used to generate three basic conditional type operations: multiply, divide, AND/OR step. In a great many cases, the conditional OP allows functions dependent on a flag to be performed in one cycle, rather than sending the flag to the controller and branching to two separate instructions depending upon that flag. When a conditional OP is selected, certain ALU control bits are forced to zero. Which bits are zeroed depends on the conditional OP select and the flag bit, as follows:

Select	Flag bit	K	P	R	
0	x	----	----	----	Unconditional
1	0	...0	-0-	-0-	Multiply step
	1	----	0...	0...	
2	0	0-0	-00-	-00-	Divide step
	1	-00-	0--0	0--0	
3	0	----	----	----	AND/OR
	1	----	-00-	----	

For example, consider multiplication: If the flag bit is high, P_3 and R_3 are grounded, so the ALU OP (1,14,14) becomes (1,6,6), which is the code for "ADD". If the flag bit is low, K_0 , P_1 , and R_1 are pulled low, transforming (1,14,14) into (0,12,12), i.e., the code for "input A".

Flags

The flag select field determines which of the ALU flags becomes the new flag bit. The following table lists the selection options.

Select	New flag bit
0	Old flag bit
1	Carry-out
2	MSB
3	Zero
4	Less than
5	Less than or equal
6	Higher (in absolute value)
7	Overflow

The ALU flags are loaded into the flag register under the control of the latching field, bit 3. They are loaded into the following positions.

Bit	Flag
0	Not changed
1	Not changed
2	Not changed
3	Not changed
4	Not changed
5	Previous value of flag bit
6	→ Carry into MSB stage
7	Less than or equal
8	→ Higher (in absolute value)
9	→ Less than
10	LSB
11	→ Zero
12	MSB
13	Overflow
14	→ Carry out
15	Current flag bit

Latching Field

The latching field specifies which of four registers should be loaded, as shown in the following table:

Latching field	Register loaded
1xx	Flag register loaded with current ALU flags
x1xx	ALU output latch A loaded with the ALU output
xx1x	ALU output latch B loaded with the ALU output
xx.x1	The literal field during the next ω_1 is loaded with the contents of bus A during the last ω_1
0000	None of these registers are affected

Literals

The 2-bit literal field specifies when a literal is to be used and which direction it goes. If both bits are 0, no literal transaction will occur. If the first bit is 1, a literal will be transferred. If the second bit is 1, the literal goes off-chip, while if the bit is 0, the literal comes on-chip.

5.12.9 Programming Examples

Here we present three programming examples that should provide a better understanding of the various data paths within OM2.

The first example is 16-bit integer multiplication. The two inputs, X and Y , are multiplied to produce the result, Z . In the multiply loop, the number X is shifted left and the MSB is stripped off. Z is shifted left, then Y is added to the new Z if the MSB of X was a 1. The sequence of instructions is repeated 16 times, using the counter in the controller to signal when the 16 iterations have been performed.

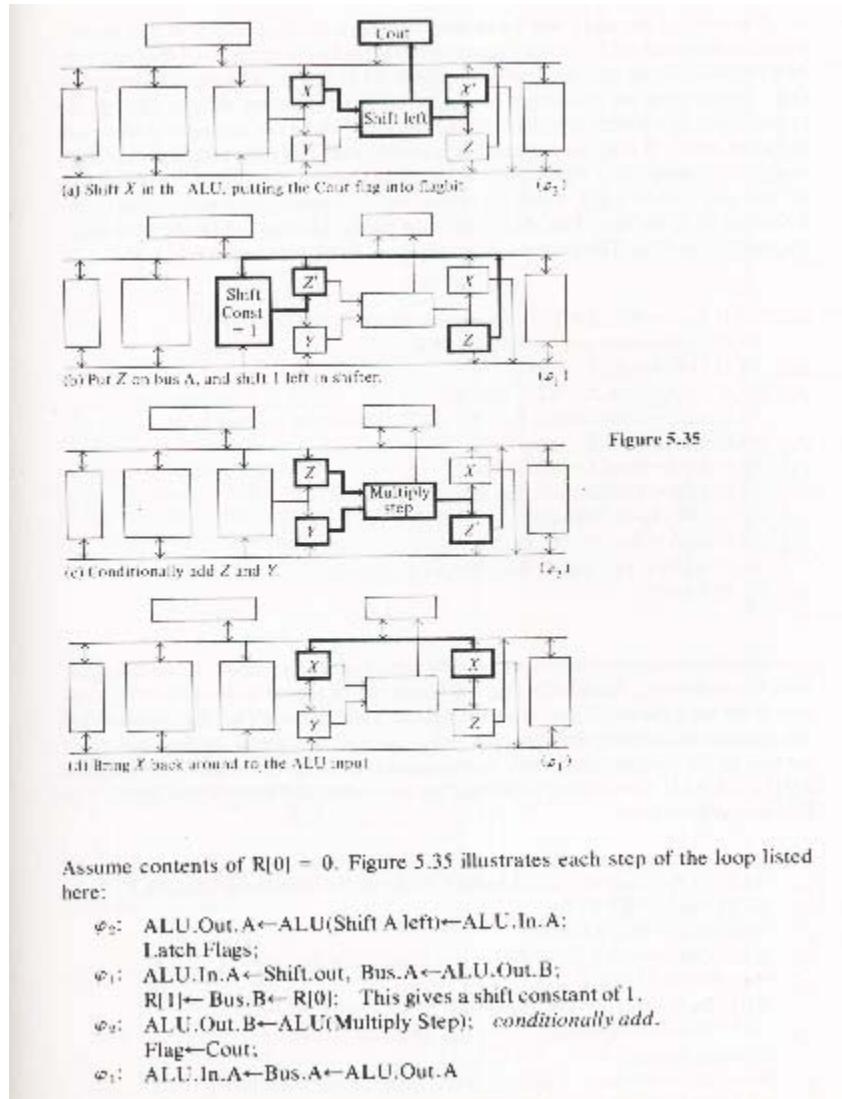


Figure 5.35

The second example will be to generate a parity flag, which is not directly available from the ALU. Parity is generated by exclusive-oring all of the bits of the data together. If the data are loaded into both ALU inputs, with input B rotated by one, performing an exclusive-or operation will give an output that is the exclusive-or of adjacent bits; bit i of the output will be bit i of the input \oplus bit $i-1$ of the same input. If this same operation is performed, this time rotating input B by two, bit i becomes $i-1 \oplus i-2 \oplus i-3$. By doing this two more times, rotating B first by four and then by eight, every bit of the output is equal to the parity, that is, the EXOR of all of the bits. The MSB flag is the parity odd flag, while the zero flag is the parity even flag. The program is listed below and illustrated in Fig. 5.36.

```

 $\varphi_1$ : ALU.In.A←Bus.A←R[0]; generate the parity of register 0.
        ALU.In.B←Shift.out(1); Bus.B←R[0];
 $\varphi_2$ : ALU.Out.A←ALU(Exor);
 $\varphi_3$ : ALU.In.A←Bus.A←ALU.Out.A;
        ALU.In.B←Shift.out(2); Bus.B←ALU.Out.A;
 $\varphi_4$ : ALU.Out.A←ALU(Exor);
 $\varphi_5$ : ALU.In.A←Bus.A←ALU.Out.A;
        ALU.In.B←Shift.out(4); Bus.B←ALU.Out.A;
 $\varphi_6$ : ALU.Out.A←ALU(Exor);
 $\varphi_7$ : ALU.In.A←Bus.A←ALU.Out.A;
        ALU.In.B←Shift.out(8); Bus.B←ALU.Out.A;
 $\varphi_8$ : ALU(Exor);

```

The third example illustrates how the data path can compute its own instruction. When driving a literal off-chip, the literal values appear in 16 of the microcode bits. If we have the literal port drive the data off chip, but don't set the disable bits in the instruction decoder, the data path will "execute" the literal. In the code below we sum all the registers (as further illustrated in Fig. 5.37). The basic literal transfers R[1] to the ALU to be added to R[0]; if we increment and execute the literal, then R[2] is transferred; etc.

```

 $\varphi_1$ : ALU.In.A←Literal "Bus.A←R[1]; ALU.In.B←Bus.B←ALU.Out.B";
 $\varphi_2$ : ALU.Out.B←ALU(A);
 $\varphi_3$ : ALU.In.A←Bus.A←R[0];
 $\varphi_4$ : ALU.Out.A←ALU(0); This is just setup, now the loop!
 $\varphi_5$ : Bus.A←ALU.Out.B;
        ALU.In.B←Bus.B←ALU.Out.A;
 $\varphi_6$ : ALU.Out.A←ALU(add);
        Execute Literal;
 $\varphi_7$ : ALU.In.A←A.Bus; The rest of this instruction is the literal!
 $\varphi_8$ : ALU.Out.B←ALU(increment B)←ALU.In.B; point to next register.

```

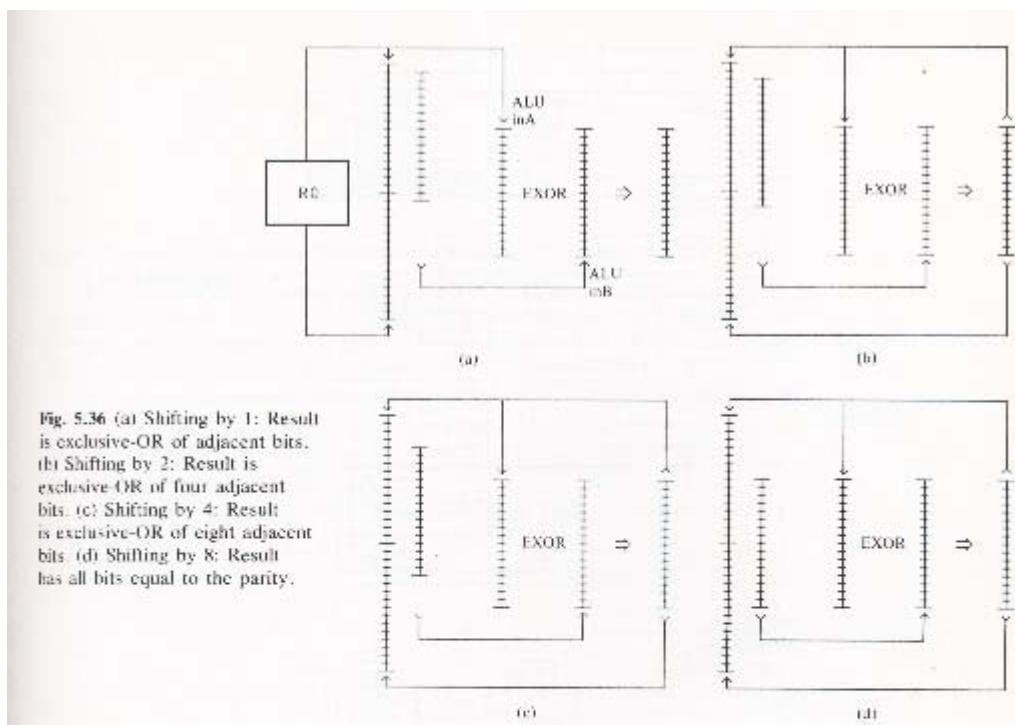


Fig. 5.36 (a) Shifting by 1: Result is exclusive-OR of adjacent bits.
 (b) Shifting by 2: Result is exclusive-OR of four adjacent bits.
 (c) Shifting by 4: Result is exclusive-OR of eight adjacent bits.
 (d) Shifting by 8: Result has all bits equal to the parity.

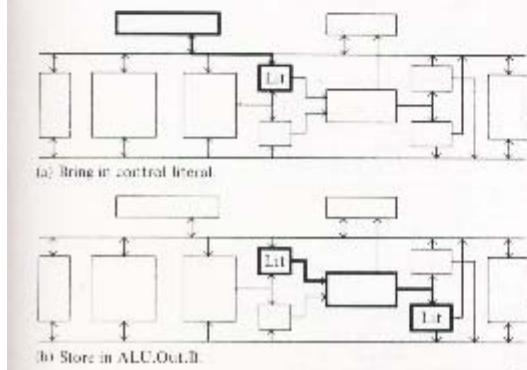


Figure 5.37

(Continued)

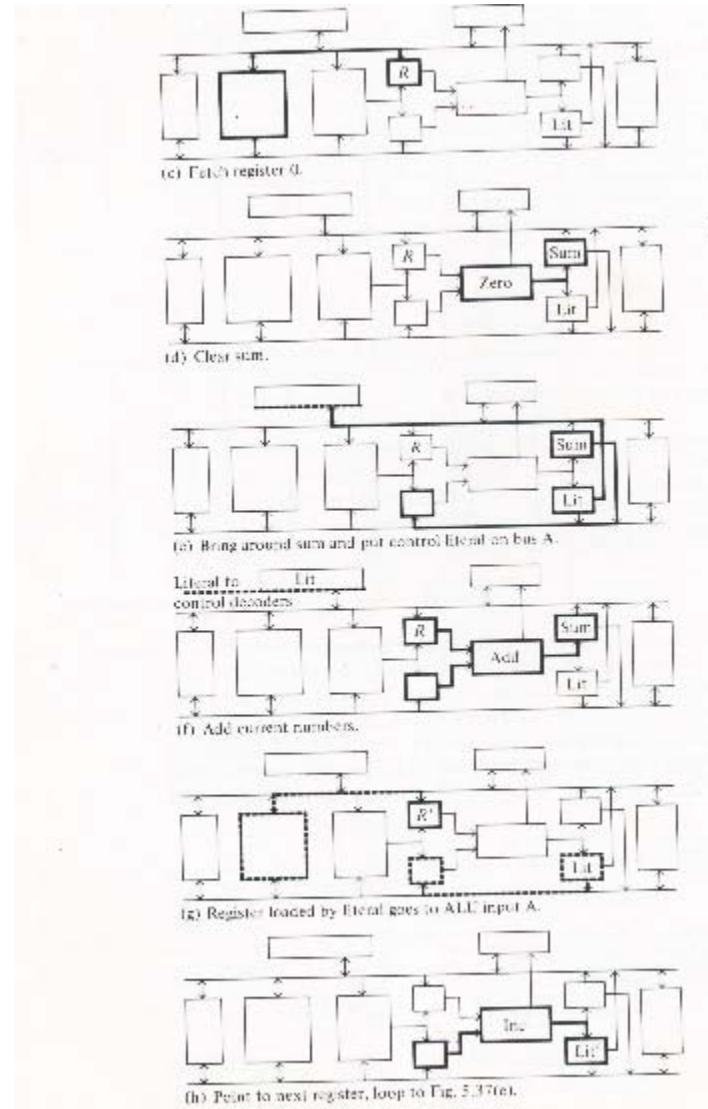


Figure 5.37 (cont.)

Proiectarea subsistemelor VLSI

97-1

Probleme speciale:

- generează unei celule de bază care poate implementa funcțiile prezentate ale sistemului.

Celule de bază doctoră:

- un set de caracteristici/mecanisme de sincronizare
- Geometria celulei și caracteristicile de sincronizare doctoră proiectării circuitului de comandă care încorporează unele de protecție constătoare din celule de bază.

După implementarea circuitelor de comandă rezultă un sub-sistem cu modul funcțional ca o rotație în direcția poziției:

- specificații funcționale;
- - " - geometrice
- - " - de I/E - date
- - " - de comandă și sincronizare.

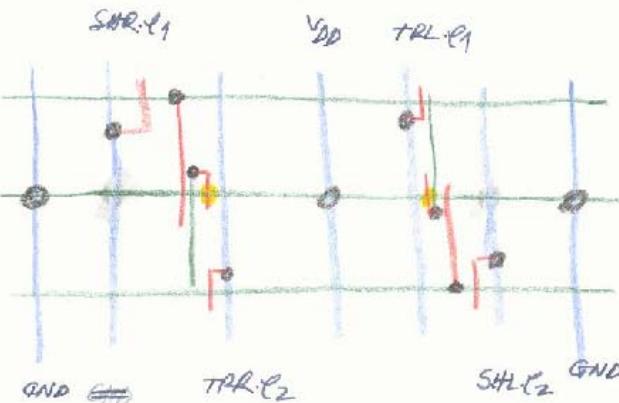
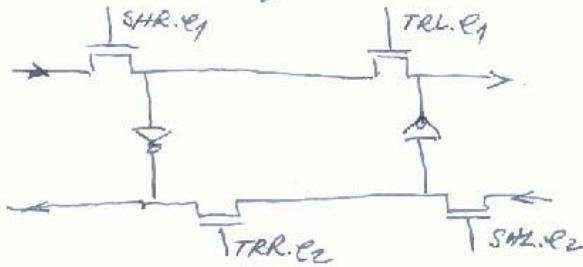
Sistemul de arhive de sincronizare are următoarele:

- reținere de date,
- iesire de date,
- reținere de comandă,

Exemplu: Proiectarea unei memorii de tip SRAM - LIFO

⁹⁷⁻²
Celula de bază este capabilă să realizeze operațiile: Push - Pop - Nap, și conditile unui sau lărjor și al unui rod de operare.

Celule de bază:



Rândul de sus se aplică pe durata lui E_1 , iar rândul de jos - pe durata lui E_2 .

Fluxul de comenzi este vertical, pe linii de metalizare, iar fluxul de date este orizontal pe linii de difuzie.

- 3 -

5703

Deplașen dreapta : POSH

$C_1 : SHR = 1$ și $C_2 : TRR = 1$

Deplașen stanga : POSP

$C_2 : SHL = 1$ și $C_1 : TRL = 1$

Memorare : NOP

$C_1 : TRR = 1$ și $C_2 : TRR = 1$.

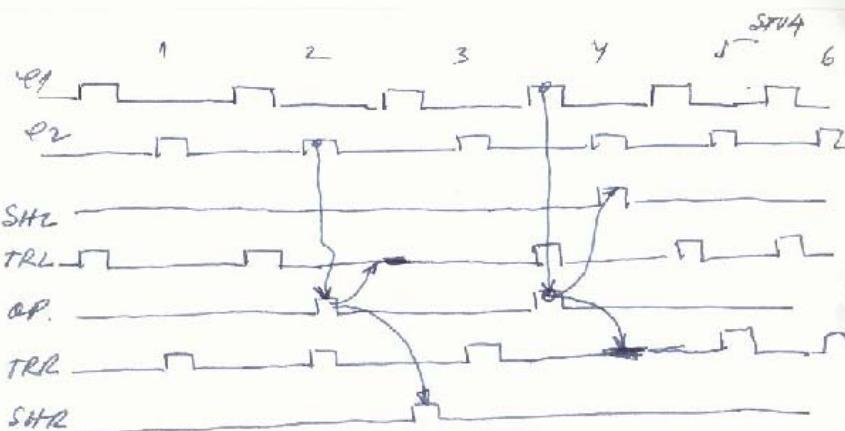
Generarea semnalelor de comandă :

Schimba este reletor orasla și poate fi generata printr-o repetie.

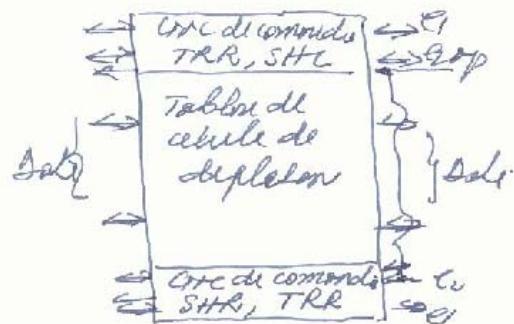
Se foloseste un singur cod de operatie de un bit, apelat faza C_1 ($op=1$), sau C_2 ($op=2$).



POSH : $op = 1$, nu C_2
 $op = 0$, nu C_1



Forme subciclului



Proiectarea logicii combinatoriale

Sunt trei categorii de probleme

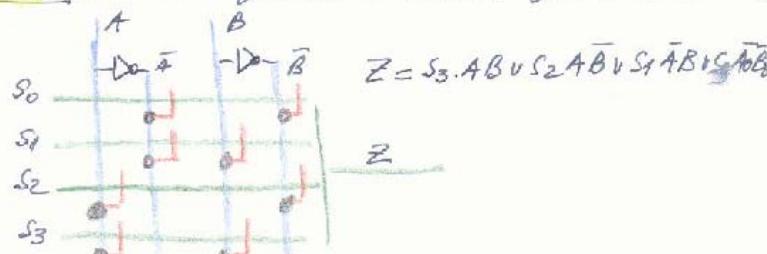
- Se cere o circuitul logic

În cazul comenzii unei circuite formate dintr-o singură celulă, care poate fi multiplicată

Se folosesc procedeele proiectării logice tradiționale cu circuite NAND, NOR sau după repetiție, foarte și mai folosesc procedeele formale de minimizare.
Utilizarea portilor statici nu este recomandată deoarece

- nu conduce la forme regulațe
- nu asigură aria minimă
- putem consuma mult energie
- nu realizează maximizarea numărului de funcții logice pe circuitele care sunt

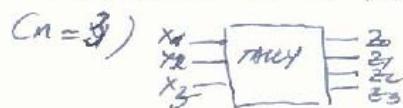
Soluție: Rețele formate din trecătoare de tip



- Adună dăse de problema

Suplementarea unei funcții logice complexe poate fi făcută să se crească metate de structuri topologice

Fie circuitul TALLY cu n variabile reale



$$z_0 = \bar{x}_3 \bar{x}_2 \bar{x}_1$$

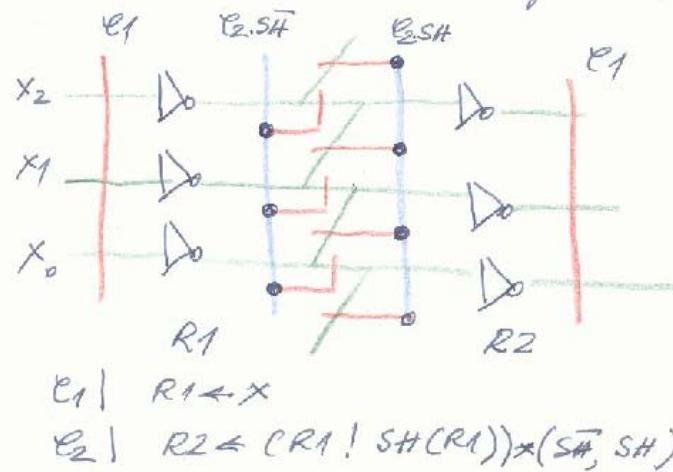
$$z_1 = \bar{x}_3 \bar{x}_2 x_1 \vee \bar{x}_3 x_2 \bar{x}_1 \vee x_3 \bar{x}_2 \bar{x}_1$$

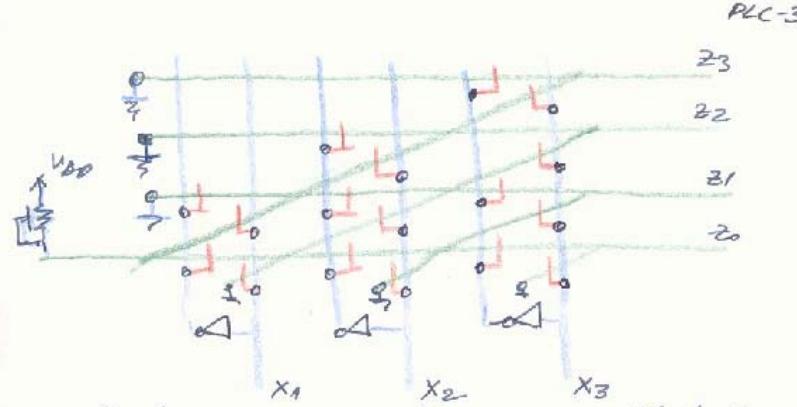
$$z_2 = \bar{x}_3 x_2 x_1 \vee x_3 \bar{x}_2 x_1 \vee x_3 x_2 \bar{x}_1$$

$$z_3 = x_3 x_2 x_1$$

Suplementarea conduce la structuri regulațe sub aspect topologic

Realizarea unei circuite de deplasare





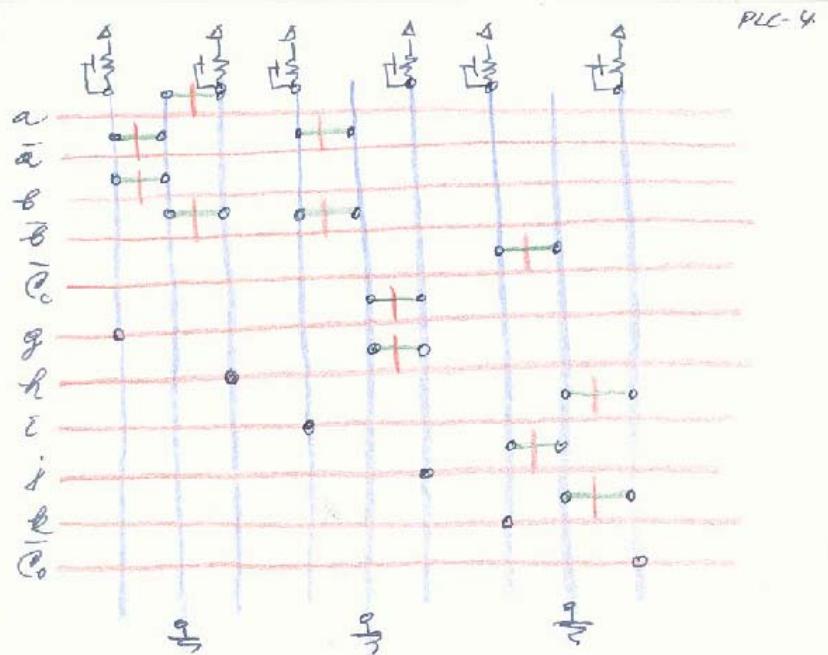
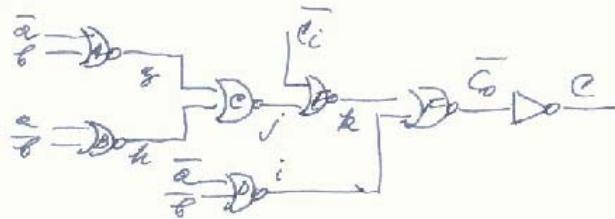
Pentru a manipula mai mult de 3 rețele se vor introduce bufferi suplimentari. Schema este similară doar că logicele sunt conectate în paralel.

Retele Weinberger

Fie funcția

$$C_0 = ab \vee C_i (\bar{a}b \vee \bar{a}\bar{b})$$

reprezentată
ca structură NOR



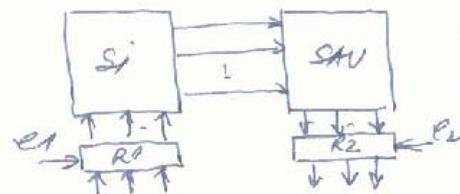
Retele logice programabile

O față de logice complexe trebuie să fie suplimentată fără a avea nevoie de o placă logică și obiectiv să fie structura reglabilă. Se specifică logice rețele negatice.

Retelele programabile permit oprire și activare funcțiilor logice negatice și structuri reglabilă. F. p. pot fi modificate selecțional fără a fi necesari modificările mecanice și mărirea spațiului.

Utilizarea memorilor pe baza stocare PLC-5
tabelilor de adrese me este considerabilă
deosebit ocupă un volum mare.

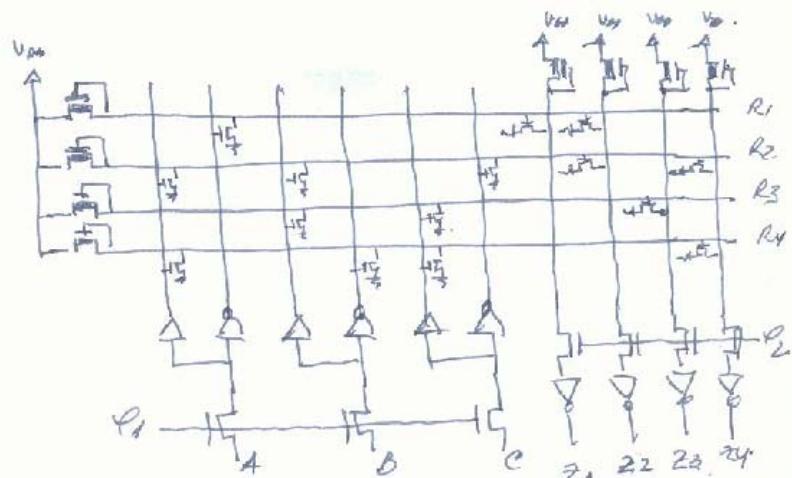
Scheare bloc și unei I/O



Analiza Si-SAV
sunt realizate cu
corante NOR.

$$\begin{aligned}Z_1 &= A \\Z_2 &= A \vee \bar{A}\bar{B}\bar{C} \\Z_3 &= \bar{B}\bar{C} \\Z_4 &= \bar{A}\bar{B}\bar{C} \vee \bar{A}\bar{B}\bar{C}\end{aligned}$$

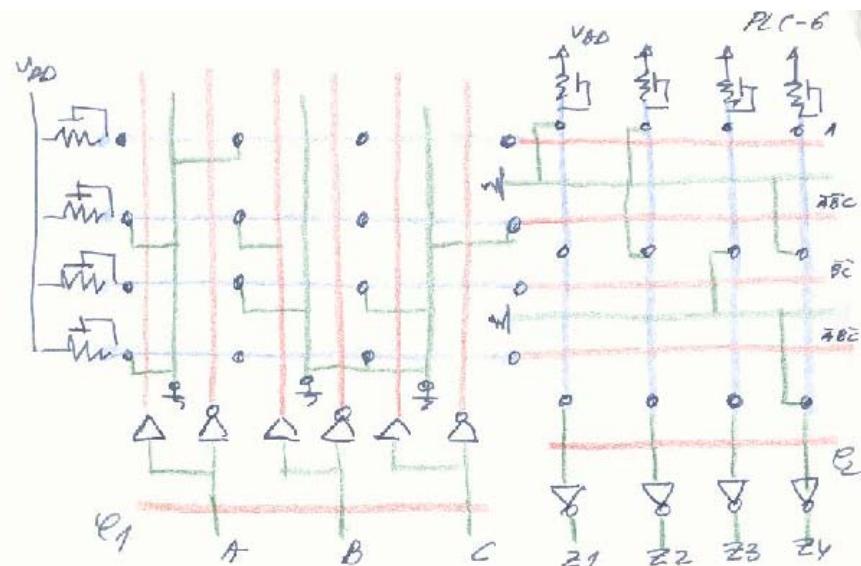
reprazentări fizice:
 $A, \bar{A}\bar{B}\bar{C}, \bar{B}\bar{C}, \bar{A}\bar{B}\bar{C}$



Se obține că

$$\begin{aligned}R_1 &= A = \bar{A} \\R_2 &= \bar{A}\bar{B}\bar{C} = \overline{A \vee B \vee C} \\R_3 &= \bar{B}\bar{C} = \overline{B \vee C}\end{aligned}$$

$$\begin{aligned}R_4 &= \bar{A}\bar{B}\bar{C} = \overline{A \vee B \vee C} \\R_5 &= \bar{A}\bar{B}\bar{C} = \overline{A \vee B \vee C}\end{aligned}$$



Prin urmare programările rezultă constând
în anumite locuri de elimentare de circuit numai pentru
o anumită producție care să fie în formă cunoaște
fauncției

Anumite variabile și forme generale depind
de aranjamentele parametrii

- număr de relași;
- numărul elementelor de tip produs;
- numărul de reză;
- următoare de lungime λ .

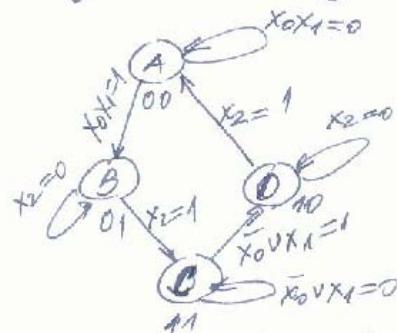
Sunt suficiente de scantei reportat ^{PLC-8} în perioada cvasilui

Exemplul: Gătitozii cuiai se vor avea cu:

- bei rutiori: x_0, x_1, x_2
- craci usori: $x_3, -x_4$
- patru stoni A, B, C, D codificați prin:

Y ₀	Y ₁	stare
0	0	A
0	1	B
1	1	C
1	0	D

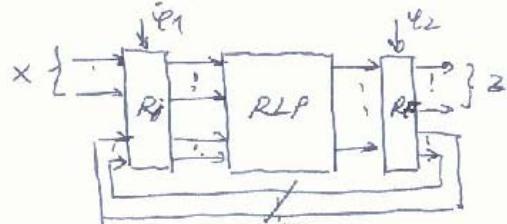
Acționatul efectuat după cvasilante organizare și diagramă de lucru:



Realizarea tabelelor de lucru pentru stoni și a tabelui pentru resori

AUTOMATE FINITE (AF)

AT se realizează folosind o rețea logică programabilă într-un registrator și codificând o rețea de lemnare sau întrare



Săvârșirile de reacție sunt raportate ca săvârșiri de stonă Y și sunt stocate în registrul de reacție R_i, pe deasupra lui ē_i. Săvârșirile se programă prin RPL până la reținere leii R_E, care îl sătorește pe deasupra lui ē_i.

Tramelele care se rezolvă, la valoarea perioadei cvasilui rezolvăresc rețeaua RPL și lăsă să dea rezultatul de reacție determinat în stare. Acționarea este săcru.

$$\begin{aligned} Y(k+1)_j &= f[Y(k_j), X(k_j)] \\ Z(k+1)_j &= g[Y(k_j), X(k_j)] \end{aligned}$$

Deși tramele de reacție ar fi permanente astfel, automatul să fie compus din atâtva. Mascaile structurale efectuante pe care le dețin intervalele pot corelați de reacție

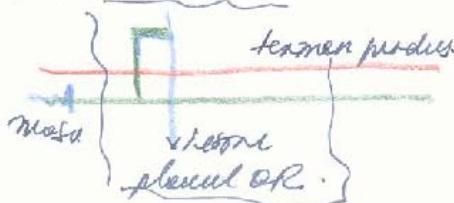
Seasale monogram
in Rio de Janeiro

Sewards memorials PC-5
rule to C2

netconi	Horn projecte	Stone acetone	Recom
X0X ₁ X ₂	X ₀ Y ₁	Y ₀ Y ₁	Z ₀ Z ₁ Z ₂ Z ₃ Z ₄
0***	0 0	0 0	0 0 0 1 0 R ₁
0	0 0	0 0	0 0 0 1 0 R ₂
11*	0 0	0 1	1 0 0 1 0 R ₃
**0	0 1	0 1	0 0 1 1 0 R ₄
**1	0 1	1 1	1 0 1 1 0 R ₅
10*	1 1	1 1	0 1 0 0 0 R ₆
0**	1 1	1 0	1 1 0 0 0 R ₇
1	1 1	1 0	1 1 0 0 0 R ₈
**0	1 0	1 0	0 1 0 0 1 R ₉
**1	1 0	0 0	1 1 0 0 1 R ₁₀

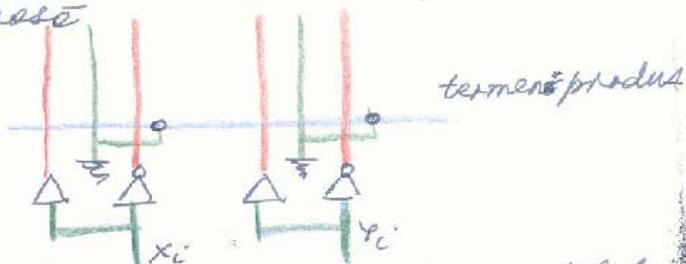
Reguli practice progresive ale tutelii legale

1. Preetura Brescovită și logodnicul său tabără pleau în
starea cernecătoare și pleau în răză se va bate
o căde de difuzie rău băta consecucționari
și resimță (nematicat, metal), peste tessutul de
polișoftron și tessutul oarzoidal de difuzie
conectat la mase

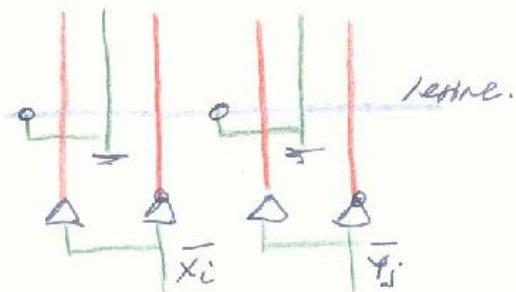


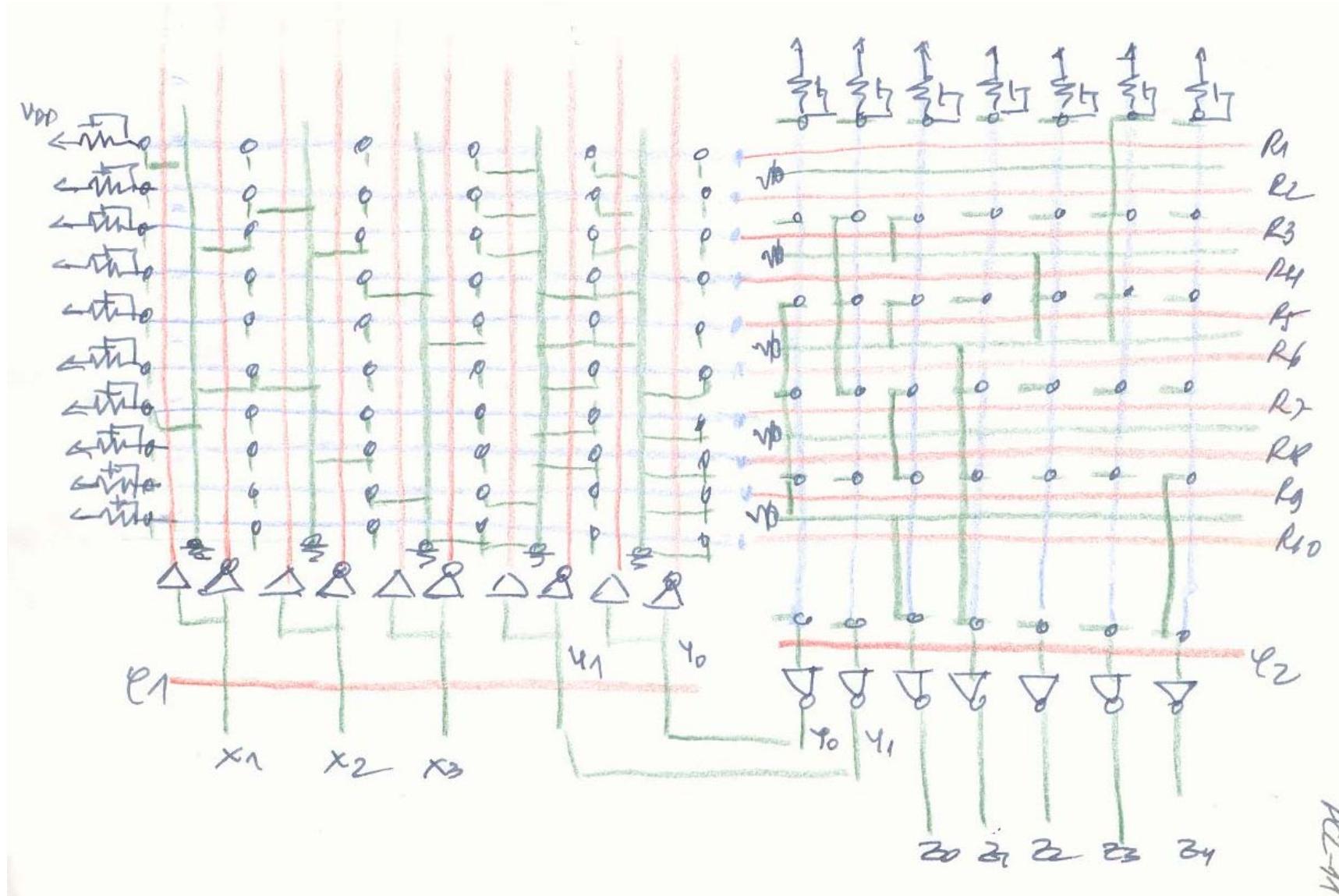
Pentie placut AND sent doce neguli PCL-10

② Pe elini trebuie să fie lărgită din tabele pentru
nășteri și să fie proiectat se bazează o
col de difuzie și la limita compresiunii
a termenului predeces este nășterea răzvrătită
la treseal verticală de difuzie corectă le
moșe.



③ Pectru fizicul termen egal cu o refochele răție-
rilor și e stocii prezente se va lăsa o cantitate
oportunitate de disponere de la termenul produc-
re mesaj retransmisiunea traseul de pe bălăica
corespondator rățorii directe





PC-11

Mesură se selectată se poate reprezenta $PCL-12$
printr-o sumă de $(150\lambda)^2$

Rezultat $\lambda = 3 \mu\text{m}$ (1978) (locuri $\lambda = 0,3 \mu\text{m}$)
suma se face de $(450 \mu\text{m})^2 \approx 0,002 \text{ cm}^2$

Mesura crește peste 150 treptătoare
și ocupă $1/25$ din aria unei etaj
integrale ocupate - $(0,25 \text{ cm}^2)$

Se trage ca urmare că $1/25.000$ din
aria etajului ocupat este integrat