Facultat de Ciències

XARXES NEURONALS VLSI D'ALTA VELOCITAT/CAPACITAT

Memòria presentada per en
Jordi Carrabina i Bordoll
per optar al grau de
Doctor en Informàtica.

Bellaterra, Juliol 1991.

Facultat de Ciències

XARXES NEURONALS VLSI D'ALTA VELOCITAT/CAPACITAT

Memòria presentada per en
Jordi Carrabina i Bordoll
per optar al grau de
Doctor en Informàtica.

Bellaterra, Juliol 1991.

12. 200930

Elena Valderrama Vallés, Catedràtics d'Arquitectura i Tecnologia de Computadors d'aquesta Universitat,

CERTIFICA:

que la present memòria ha estat realitzada sota la meva direcció per en Jordi Carrabina i Bordoll, constituint la tesi doctoral per accedir al grau de Doctor en Informàtica.

Bellaterra, Juliol de 1991.

Elena Valderrama Vallés

Als fusters que van fer la nau,
als companys d'abord,
i amb tot el cor,
a l'Alba radiant, que em sap admirador seu.

ÍNDEX

PRÒLEG 1

1 INTRODUCCIÓ A LES XARXES NEURONALS 7

1.1 Definició 11

1.2 Evolució històrica 12

1.3 Elements d'una xarxa neuronal 15

1.4 Paradigmes d'aprenentatge 17

1.5 Propietats funcionals 18

1.6 Tipus de xarxes neuronals 19

1.7 Funcionalitat associada a la topologia de la xarxa 24

1.7.1 Àlgebra d'hiperplans 25

1.7.2 Síntesi de circuits combinacionals 25

1.7.3 Síntesi d'elements de memòria i oscil.ladors 27

1.7.4 Síntesi de circuits seqüencials 29

1.7.5 Síntesi de funcions complexes 30

2 MICROELECTRÒNICA APLICADA A XARXES NEURONALS 31

2.1 Nivells d'abstracció associats a les xarxes neuronals 33

2.2 Nivells d'abstracció associats a la microelectrònica 34

2.3 Tendències fonamentals de les implementacions VLSI 37

2.4 Xarxes neuronals programables d'alta velocitat 41

2.4.1 Memòria autoassociativa 43

2.4.2 Memòria heteroassociativa 52

2.5 Xarxes neuronals de preprocés 57

2.5.1 Transistor MOS en lògica subllindar 57

2.5.2 Còclea electrònica 59

2.5.3 Retina electrònica 62

2.5.4 Conversor analògico-digital 67

5.6.2 Exemple d'aplicació:
Reconeixedor de caràcters manuscrits 140
5.7 Comparació amb altres estratègies 142
5.7.1 Realitzacions analògiques 142
5.7.2 Realitzacions digitals sistòliques 143
5.7.3 Neurocomputadors 144

6 AMPLIACIONS FUTURES 147
6.1 Aplicacions 149
6.1.1 Processament d'imatges 150
6.1.2 Processament de la veu 153
6.2 Arquitectura 153
6.3 Xips 155
6.4 Algorismes 157

CONCLUSIONS 159

APÈNDIX 1. REGLES D'APRENENTATGE PER A PESOS DISCRETS 167
A1.1 Capacitat d'emmagatzemament en xarxes de tipus Hopfield 169
A1.2 Estudis teòrics sobre pesos discrets 170
A1.3 Regles d'aprenentatge per a l'obtenció de pesos discrets 171
A1.3.1 Regles d'aprenentatge sobre l'espai discret 171
A1.3.2 Transformacions de l'espai continu al discret 173

APÈNDIX 2. CEL.LES VLSI 179
A2.1 Memòria associativa bidireccional 181
A2.1.1 Sinapsi 182
A2.1.2 Neurona 182
A2.2 Conversor analògico-digital 185
A2.2.1 Resistències MOS 185
A2.2.2 Resistències de polisilici 187
A2.3 NDN2 189
A2.3.1 Comprovador de codi Berger de 12 bits 189

PRÒLEG

El cervell és l'element més elegant i perciós dels que posseeix el cos humà. Ho és per la simplicitat amb la que realitza certes funcions, les quals potser pel costum de realitzar-les habitualment no mostren la complexitat del processament que amagen. Ho es també per la complexitat de la seva estructura, de la qual en tenim encara un coneixement força superficial. En coneixem algunes dades com ara el número aproximat de neurones que el composen, aproximadament 10^{10}, amb grau de connectivitat intern molt elevat, fins a 10^4 connexions per neurona. Sabem d'altra banda, localitzar les regions del cervell corresponents al control de la resta de membres de l'organisme, i al mateix temps, som capaços d'identificar certes estructures amb formes determinades de tractar la informació (òrgans sensors, memòria, etc.).

Des del punt de vista del tractament de la informació, per part de les estructures de processat que ha desenvolupat l'home en la construcció d'ordinadors, l'estudi d'aquest immensa estructura és una font inesgotable de nous recursos tant materials com conceptuals enfocats a la resolució de problemes, el grau de dificultat dels quals, des d'un punt de vista bàsicament algorísmic, és molt elevat.

Aquest estudi és paral.lel a l'increment de la potència i la capacitat de processat de les màquines algorísmiques que l'home sap desenvolupar, ajudat sens dubte pel desenvolupament d'una tecnologia que soporta aquest creixement: la tecnologia microelectrònica. Mercès a aquesta eina, el disseny dels sistemes computacionals evoluciona cap a un grau de paral.lelisme cada cop més important. El concepte de sistema complex i fortament interconnectat porta l'aparició d'un nombre important de problemes (sincronització, redundància, rebustesa, etc.) la major part dels quals estan "solucionats" amb èxit pel nostre sistema nerviós.

Però la informàtica, com a ciència dedicada a l'estudi i el processat de la informació i el control de processos, i la electrònica, com a tecnologia de suport, no poden deixar de banda altres conpectes que apareixen de manera natural en les formulacions associades a les xarxes neuronals: la "substitució" de la programació per l'aprenentatge i la utilització de dispositius elementals únics per a la construcció de la xarxa neuronal.

El concepte d'aprenentatge és, sens dubte, identificatiu de les xarxes neuronals. Tot i que la dificultat de trobar una definició adient, podem dir que es basa en l'assimilació de les característiques comunes de certs grups d'informació mitjançant únicament les dades, amb un procés que normalment és repetitiu. La substitució esmentada fa referència a que en el procés d'aprenentatge no és necessari donar al sistema que apren cap algorisme, la recerca del qual, en molts cassos és complexa.

La utilització d'un únic element de procés, d'altra banda ens acosta la concepció de les xarxes neuronals a la microelectrònica, en particular a la que utilitza el procés MOS per al disseny i fabricació de circuits integrats. L'element de procés és igualment únic, el transistor MOS, i tot i que, en detall és prou complexe, pot ser modelat de manera molt simple com un interruptor.

Diverses realitzacions microelectròniques, en alta escala d'integració, han intentat portar més enllà aquesta similitud, mitjançant el disseny de xarxes neuronals artificials. Aquesta vía oberta a mitjans de la década dels 80, ha produït i segueix produint resultats inesperants i noves solucions i nous conceptes en el disseny de sistemes electrònics. Però, l'enllaç entre aquests dos camps, xarxes neuronals artificials i microelectrònica, pot produir-se a altres nivells. Les estratègies de disseny i el grau d'automatització desenvolupats al voltant de la tecnologia microelectrònica, han permes desenvolupar diferents nivells s'abstracció modelant estructures cada cop més complexes per al seu ús com a elements de base per al disseny de sistemes més complexos. La realització VLSI de xarxes neuronals ha utilitzat aquests diferents nivells, aprofitant-ne al màxim les característiques de cadascun d'ells.

En aquest treball presentem la nostra aportació humil al disseny VLSI de xarxes neuronals artificials. L'hem estructurat de la següent manera. Al primer capítol fem una introducció als conceptes bàsics associats a les xarxes neuronals. Expliquem com es modelitzen les neurones, com realitzar xarxes mitjançant diferents models d'interconnexió de neurones, i quines són les característiques computacionals associades. Introduïm els algorismes corresponents a algunes de les regles d'aprenentatge més importants. També comentem la importància de la topologia de la xarxa. Mostrem que a part de les funcionalitats complexes associades a les xarxes neuronals també es possible realitzar la síntesi de circuits combinacionals i seqüencials.

En el segon capítol presentem un seguit de realitzacions microelectròniques de xarxes neuronals. En particular, fem un especial esment d'aquelles que introdueixen metodologies noves al VLSI: les xarxes neuronals de proprocés, i les xarxes neuronals programables. Presentem dos circuits realitzats, un de cada estratègia i l'anàlisi de les seves característiques més importants i del seu rang d'aplicacions.

El tercer capítol està dedicat a l'estudi de la dinàmica de relaxació de les xarxes neuronals des d'un punt de vista teòric. Aques estudi es fonamental per a la realització de xarxes neuronals programables. Estudiem les implicacions de la dinàmica de la xarxa en les característiques qualitatives d'aquesta i fem una comparació de quatre dinàmiques de relaxació diferents: una dinàmica paral.lela i tres dinàmiques seqüencials, que anomenem de criteris aleatori, analític i probabilístic. La dinàmica seqüencial de criteri probabilístic és la que dóna uns millors resultats pel que fa a calitat i velocitat de recuperació.

En el quart capítol, desenvolupem els algorismes per al disseny de circuits digitals que implementin la dinàmica seqüencial amb criteri probabilístic tenint en compte d'una banda, les condicions donades pel disseny VLSI i de l'altra, la restricció de treballar amb pesos discrets per tal de aconseguir un número màxim de neurones i una elevada velocitat de recuperació.

En el capítol cinqué presentem els xips que hem dissenyat, així com les dues estratègies que hem seguit que anomenem, estatègia trasparent i estratègia

especial, i que fan referència a la manera de processar les matrius totalment interconnectades, igual per a totes elles o preprocessant certes matrius característiques (zero, identitat, etc.), respectivament. Es presenten els xips NDN2, NDN3, i NDN3, les seves característiques i les avaluacions de superfície i velocitat de procés. Es mostra també les implicacions a nivell de sistema digital i es mostra la placa de circuit imprés realitzada per la connexió del sistema basat en el xip NDN3 a un PC, sobre el qual estem desenvolupant una aplicació de reconeixement de caràcter òptics. Finalment, es realitza una comparació amb altres estratègies de disseny de sistemes neuronals, realitzacións analògiques, realitzacions digitals sistòliques i neurocomputadors.

Finalment, es proposen al capítol sisé possibles estratègies futures d'ampliació i millora d'aquest sistema a diferents nivells: aplicació, arquitectura, xips i aprenentatge.

Capítol 1

INTRODUCCIÓ A LES XARXES NEURONALS

La intel.ligència humana ha estat al llarg del temps, una eina que ha permès l'home evolucionar cap a situacions globalment més satisfactòries, tant des del punt de vista socio-cultural com tecnològic.

Però, l'hombre no s'ha conformat amb disposar d'una eina capaç de generar noves eines sinó que l'ha intentat reproduir en màquines amb una certa "intel.ligència". Podem dir que aquesta tasca ha estat iniciada amb un cert éxit en diferents camps de treball en els quals l'home ha estat capaç d'ajudar-se d'aquestes eines: per a treballs mecànics pesats, en condicions crítices (química, física nuclear o enginyeria espacial), encara que també per a treballs casolans o en entorns culturals de creació (música, arts plàstiques, disseny) o difusió (medis de comunicació).

El següent repte abordat per l'home creador ha estat l'estudi i reproducció de la mèdula d'aquesta estructura: la seva pròpia intel.ligència capaç de generar aquestes mateixes eines, la qual cosa suposa atacar un nivell d'abstracció superior. Per aquesta raó s'ha investigat i s'investiga el sistema nerviós desde diversos punts de vista i amb diferents enfocs en funció de la "utilitat" per a la qual es busquen solucions.

Donada la dimensió de l'empresa, el cervell conté un número d'elements bàsics (neurones) aproximadament igual a 10^{10}, la resolució d'aquest problema ha hagut de ser atacada mitjançant una partició del sistema. Els criteris per a aquesta partició son bàsicamente dos: criteris funcionals i criteris topològics.

Els criteris funcionals fan referència a l'estudi de parts del sistema nerviós segons la funció que realitzen. Mitjançant aquesta técnica s'han aïllat les tasques transductores dels sentits, que realitzen la percepció de la realitat física exterior, les tasques de control d'elements motors, els sistemes fisiològics interns autònoms

(cor, estómac, etc.), les funcions de càlcul basades en estructures lògiques i en estructures de memòria i reconoceixement i la capacitat d'aprenentatge.

En canvi, els criteris topològics fan referencia al estudi de les estructures que realitzen les esmentades funcionalitats. A nivell genèric, es coneix la localització física a les àrees del cervell de les tasques associades a les diferents parts del cos humà. A un nivell més detallat, s'ha conseguit estudiar la connectivitat entre neurones de grups localitzados per a números de neurones limitats.

L'estudi de la relació entre les propietats funcionals i l'estructura topològica de determinades parts del sistema nerviós que anomenarem xarxes neuronales, constitueix el núcli de coneixement necessari per a la realització de xarxes neuronals artificials. Malgrat tot, a aquest nivell, els esmentats estudis no presenten cap element comú unificador ja que els temes d'estudi són suficientemente dispars.

Per trobar aquest element unificador és necessari baixar al nivell de base del sistema nerviós: la neurona. Des d'aquest punt de vista, podem reformular el problema com la síntesi de xarxes neuronals amb certes propietats funcionals donades per les característiques topològiques de conectivitat entre neurones i pel model de neurona utilitzat.

El principal avantatge d'aquesta aproximació ve donada per l'elevat coneixement de que es disposa sobre els esmentats elements. Aquesta descripció dels models de neurona pot restringir-se per al seu ús en xarxes neuronals artificials en funció de les característiques demanades per l'usuari. Psicòlegs, fisiòlegs, matemàtics, físics, electrònics, informàtics, òptics, bioquímics o biotecnòlegs entre altres, han utilitzat diferents models o dispositius que realitzen la funcionalitat de la neurona amb prestacions diferents de complexitat, velocitat, dimensions, límits de conectivitat, etc.

La restricció al treball amb neurones com a únics elements de procés, suposa una forta restricció per a la construcció de sistemes ja que partim d'un únic element de base, però simultàniamente es simplifica la seva realització tecnològica si podem implementar aquest dispositiu bàsic i utilizar-lo de forma repetitiva.

1.1. Definició

Les xarxes neuronals, models connexionistes o, segons la seva denominació més actual, els sistemes neuromorfs de càlcul (neurocomputadors, ANNs: Artificial Neuronal Networks, etc.), són sistemes concebuts per aprofitar certs principis d'organització estructural, que s'intueix que utilizen els sistemes neuronals naturals, en tasques de percepció i processat de la informació. Són xarxes d'elements de càlcul molt simples, de tipus adaptatiu, altament connectats i amb organització jeràrquica, que pretenen interaccionar amb el món real d'una forma similar a com ho fan els sistemes neuronals naturals.

Les neurones i el cervell són, sens dubte, els exemples més autèntics de xarxes neuronals. D'alguna manera que desconeixem les esmentades xarxes pensen, senten, aprenen, recorden, ... L'esforç humà per comprendre el seu funcionament passa per la construcció de models, el seu contrast amb la realitat, i la seva millora succesiva. Aquests models es divideixen en dos categories: els models biològics i els models tecnològics.

En el modelat biològic es pretén estudiar l'estructura i funcionament real del cervell, amb l'objectiu de descubrir com es reflexen els aspects biològics en les manifestacions de més alt nivell (el comportament per exemple). En el modelat tecnològic, contrariament, eawpretén estudiar el cervell amb l'objectiu d'extreure'n idees que puguin utilizar-se en el desenvolupament de noves tecnologies de cómput. Es trata doncs, d'inspirar-se en el model de funcionament del cervell per desenvolupar noves estructures que realitzin eficientment funcions costoses per al computador actual. En aquest sentit reben el nom més escaient, de "xarxes neuronals artificial" (ANNs), o "neurocomputador".

Els objectius de la investigació en xarxes neuronals són diversos. En primer lloc, es tracta de descubrir com realitza el cervell operacions tals com la percepció, el volcat associatiu de records, el raonament, l'aprenentatge, etc. En segon lloc, es tracta de desenvolupar un conjunt de models de xarxes neuronals que, encara que no siguin biològicament fidels al seu model, enfatitzin aquest tipus de "poder de computació" propi del cervell humà, segons sembla tan diferent del que actualment posseeixen les nostres computadores.

1.2. Evolució històrica

Malgrat que la gènesis de les xarxes neuronals es remunta als años 40, el tema ha reaparegut darrerament amb força renovada degut, entre altres raons, als recents desenvolupaments de nous models teòrics, a les noves tecnologies susceptibles d'implementar les esmentades xarxes, i als nous algorismes d'aprenentatge.

No es pretén tampoc fer una "història de les xarxes neuronals", però sí apuntarem alguns dels esdeveniments clau:

L'any 1943, McCulloch i Pitts [McCul43] proposaren el següent model de neurona formal: Una neurona és un element amb m entrades x_1 , x_2 ,..., x_m (m>>1) i una sortida d, que ve caracterizat per m+1 números; el seu llindar # i els pesos w_1 , w_2 ,..., w_m , on w_i està associat a l'entrada x_i. El mòdul opera en una escala discreta de temps, i l'activació en el moment n+1 queda determinada per l'activació de les seves entrades en l'instant n d'acord amb la següent regla "El mòdul (neurona) envia un impuls per la seva sortida (això) solament si la suma dels pesos de les seves entrades actives en l'instant n supera el valor de llindar # de la neurona".

Formalment,

$$d(n+1) = 1 \Leftrightarrow \sum w_i x_i \geq \# \qquad (1.1)$$

Els pesos positius corresponen a sinapsis excitadores, mentre que els pesos negatius corresponen a sinapsis inhibidores.

Una xarxa neuronal és un conjunt de neurones formals, funcionant totes elles amb la mateixa escala de temps i interconnectades. Les possibilitats d'interconnexió de les xarxes neuronals (arquitectura de les xarxes) són variades, i poc es sap, a priori, respecte de la "bondat" de cadascuna d'elles.

McCulloch i Pitts varen dedicar els seus esforços a demostrar que la seva neurona formal era capaç de realitzar operacions lògiques simples. A més foren els primers en reconèixer que la "classificació de patrons" és un problema central en qualsevol teoria sobre el comportament intel.ligent.

Més endavant, al 1949, D.O. Hebb [Hebb49] postulava que la "conectivitat" en el cervell canvia contínuament, i que aquests canvis són la base de

de l'aprenentatge. Hebb va proposar un model actualment conegut com la "llei d'Hebb de l'aprenentatge": Si una neurona A és activa repetidament quan una altra neurona B, connectada a una de les seves sinapsis (entrades) està activa, la conductivitat de la sinapsis A-B augmenta" (és a dir, el pes de la connexió A-B s'incrementa). Segons aquesta regla, si un grup de neurones dèbilment connectades (amb pesos petits) s'activa repetidament, les neurones tendeixen a organitzar-se formant grups fortament connectats.

Al 1958, F. Rosenblatt [Rosen58] desenvolupa unes xarxes neuronals amb neurones formals del tipus de les de McCulloch-Pitts i amb pesos ajustables que, després d'una fase d'entrenament, han estat capaces de realitzar tasques de classificació de certs patrons. L'esmentada xarxa va rebre el nom de "perceptró". Un preceptró típic constava de tres capes de cel.les: una primera capa d'unitats "sensitives" (entrades), una segona capa de neurones que rebien entrades de les unitats sensitives via conexions de pesos $w_{ij,}$ i una tercera capa de sortida, que anomenà "capa motora". Els pesos s'inicialitzaven a un valor arbitrari i, durant la fase d'entrenament, es modificaven per a que la xarxa respongués de la forma desitjada als estímuls d'entrada.

L'any 1960, Widrow i Hoff [Widro60] presentaven la seva "Adaline" (ADAptative LInear NEuron), en una línia similar a la del perceptró.

El desenvolupament del perceptró i les seves variants provocaren força enrenou en aquells temps. Es discutia (i encara es segueix discutint) si el cervell i els sistemes de càlcul són sistemes basats en la manipulació de símbols o no. Dit d'una altra manera, l'escola que recolça l'afirmació anterior defén que el procés de "resolució de problemes" (problem-solving) és esencialment algorísmic, mentre que l'altra escola ho nega. El perceptró demostrava que la idea que defenia aquesta darrera era, si més no, viable.

En la década següent aquesta postura va rebre un cop dur amb el libre de Minsky i Papert "Perceptrons" [Minski65], que evidenciava els límits del perceptró. En aquest libre, Minsky i Papert provaren que el perceptró era incapaç de realitzar operacions tan simples com la "OR exclusiva" i, en conseqüència, no podia actuar com element universal en el sentit de Turing. Posteriorment es va

veure que, ampliant el número de capes del perceptró, sí que es poden realitzar tals operacions. Un perceptró de tres capes és un element universal.

El fet és que, després del llibre de Minsky i Papert, l'estudi de les xarxes neuronals, que havia adquirit un important renom durant la década dels 50-60; va quedar-se relegat a una sèrie d'"incondicionals" (Grossberg, Kohonen, Anderson, Amit,...), que varen continuar treballant sense desànim.

El "renaixement" de les xarxes neuronals s'associa a una publicació de J.J. Hopfield [Hopfi82] de 1982, en la que es desvetllaven certes analogies entre els comportaments col.lectius de grups de neurones i els comportaments col.lectius de grups d'àtoms. D'una manera senzilla, una xarxa neuronal pot veure's com un conjunt d'elements (neurones) sotmeses a unes forçes d'inhibició i excitació (pesos) provocats pels elements veïns (la resta de les neurones que formen la xarxa). Una situació similar es dóna en la matèria, on un conjunt d'àtoms es troben sotmesos a unes forçes d'atracció i repulsió, resultat de l'acció dels àtoms veïns. Hopfield va establir una analogia formal entre una xarxa neuronal "restringida" (xarxa de Hopfield), i un tipus de materials magnètics, els "vidres d'espín" (spin-glasses). Si la xarxa és totalment connectada i simètrica ($w_{ij}=w_j$ i $w_{ii}=0$), es pot associar una funció energia E a la xarxa neuronal de tal manera que la xarxa evoluciona lliurement cap a aquelles configuracions que minimitzen la funció energia. Aquestes configuracions estables poden utilizar-se per emmagatzemar informació. La funció de "memòria associativa" apareix així d'una manera natural.

A partir de Hopfield, el número de publicacions sobre el tema es dispara escandalosament. Deixarem, malgrat tot, la breu ressenya històrica en aquest punt, per intentar sintetitzar l'estat actual del tema.

1.3. Elements d'una xarxa neuronal

Una xarxa neuronal és un sistema format per:

1) Un conjunt d'unitats de procés (neurones)
2) L'estat d'activació
3) Una funció de sortida per a cada unitat
4) Un patró de conexionat
5) Una regla de propagació
6) Una regla d'activació
7) Una regla d'aprenentatge

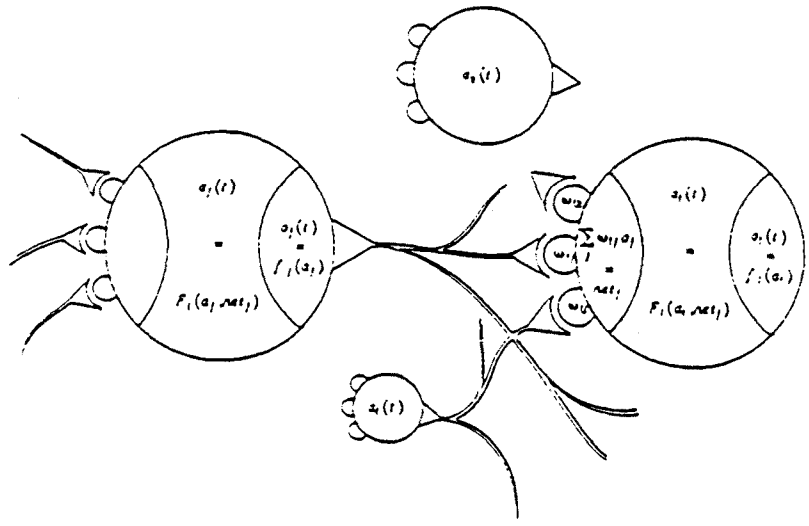


Figura 1.1. Components d'una xarxa neuronal.

1) *Conjunt de neurones* ($\{u_i\}$)

Cada unitat pot representar diverses entitats : lletres, paraules, característiques,... o les seves representacions abstractes. Anomenem u_i a les diverses unitats.

Cal distinguir, per qüestió de nomenclatura, les unitats d'entrada (reben els impulsos exteriors), les unitats de sortida (observables des de l'exterior), i les unitats "ocultes".

2) *Estat d'activació (a(t))*

Cada unitat u_i, en un instant t, es troba en l'estat $a_i(t)$. L'estat d'activació de la xarxa neuronal és un vector **a**(t) de tantes components com unitats té la xarxa. Cada component $a_i(t)$ pot pendre, en principi, qualsevol valor.

3) *Funció sortida (o(t))*

Associada a cada unitat u_i existeix una funció $F_i(a_i(t))$ que genera la sortida $o_i(t)$. Anàlogament el cas anterior, el vector **o**(t) resumeix la sortida de la xarxa.

4) *Patró de conexionat*

Especifica el conexionat entre les unitats. Cada conexió porta associat un pes T_{ij} (pes de la conexió entre la neurona i i la j) que pot pendre valors positius i negatius. Si $T_{ij}>>0$, es diu que la unitat u_j excita a la unitat u_i; si $T_{ij}<<0$, es diu que la unitat u_j inhibeix a la unitat u_i. Si $T_{ij} = 0$, no existeix conexió entre la unitat u_i i u_j. Els pesos de totes les connexions formen la matriu de pesos $T = \{T_{ij}\}$.

5) *Regla de propagació*

La regla de propagació és una funció que resumeix l'efecte sobre cada unitat de l'activació de les unitats connectades a ella. Les contribucions de la resta de les neurones poden ser diferents si pertanyen a diferentes capes. Per a la unitat u_i, la contribució de totes les entrades provinents de neurones de tipus i ve donada per $h_i(t)$, el camp local associat a la neurona i.

$$h_i = \sum_j TijOj \qquad (1.2)$$

Si tenim un sol tipus d'unitats o la mateixa regla de propagació per a totes:

$$h=T.o \qquad (1.3)$$

6) *Regla d'activació (f_i)*

Dóna l'estat d'activació en funció l'estat anterior i la regla de propagació:

$$a_i(t+1) = f_i(a_i(t),h_i(t)) \qquad (1.4)$$

En el cas de tenir un sol tipus de neurones:

$$h = To \tag{1.5}$$

La funció f_i difereix en cada model. Algunes vegades, f_i és simplement la identitat, altres vegades f és una funció llindar, i en el cas de que sigui una funció contínua, és comú definir-la com una funció "sigmoidal". La figura 2 mostra diverses definicions de la funció f.

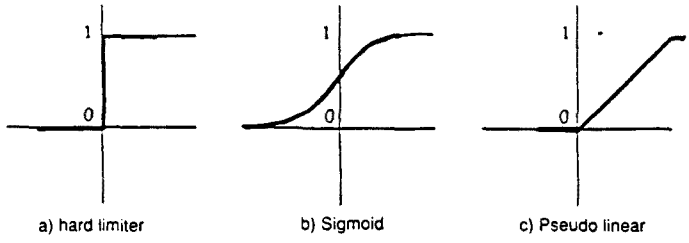


Figura 1.2. Diverses funcions llindar.

7) *Regla d'aprenentatge*

Diverses regles d'aprenentatge s'utilitzen en diferents models (regla d'Hebb, regla de Widrow-Hoff, backpropagation, etc). En l'apartat dedicat als tipus més comuns de xarxes neuronals es comenten algunes d'aquestes regles.

1.4. Paradigmes d'aprenentatge

A pesar del gran nombrer de regles d'aprenentatge que és possible definir, les xarxes neuronals són capaces de realitzar, bàsicament 5 funcions diferents:

1) *Auto-associació.*

Un conjunt de patrons es presenten repetidament al sistema i aquest els "emmagatzema" de forma que posteriorment, si se li presenten informacions parcials dels patrons, o patrons "similars", el sistema és capaç de reconstruir l'original.

2) *Associació de patrons.*

És una variant de l'anterior. Al sistema se li presenten pars de patrons, de forma que aprenguin a identificar-los. Posteriorment, quan se li presenta un dels patrons d'una certa parella, el sistema produeix l'altre.

3) *Classificació.*

Agrupa els patrons en classes. Durant la sessió d'aprenentatge, al sistema se li presenten els patrons (estímuls) i la seva classe. L'objetiu és que aprengui, de manera que si se li presenta una versió lleugerament distorsionada d'un dels patrons, o un patró "similar", el classifiqui correctament. El perceptró, i el seu teorema de convergència, funcionen en aquest entorn.

4) *Detecció de regularitats.*

Al sistema se li presenta una "població" de patrons, de forma que cada patró S_k apareix amb una probabilitat p_k. El sistema suposadament "descubreix" característiques estadístiques de la població. Aquest paradigma és similar al de la classificació; però en aquest cas el sistema defineix les seves pròpies classes (aprenentatge no supervisat).

5) *Detecció/Generació de seqüències temporals*

El sistema és capaç d'apendre a generar o reconèixer les relacions internes que lliguen les sequències temporals.

1.5. Propietats funcionals

El cómput basat en xarxes neuronals presenta certes propietats anàloges a les del cervell humà:

- *Capacitat d'associació* entre patrons d'entrada i patrons de sortida, com és el cas de la memòria associativa, en la qual l'accés a la informació es realitza per contingut en comptes de per posició com en les memòries convencionals.

- *Capacitat de generalització*, en virtut de la qual un patró que es presenta per primera vegada a la xarxa és correctament tractat. Aquest punt és important de cara a les memòries associatives ja que permet respondre correctament a informacions degradades.

- *Capacitat de búsqueda* en paral.lel, de forma que la velocitat de resposta no depén bàsicament del número de patrons amagatzemats.

- *Capacitat d'aprenentatge*. La xarxa és capaç de reordenar-se en funció de la informació rebuda i de la regla d'aprenentatge implícita.

- *Flexibilidad*, en el sentit que una xarxa neuronal amb capacitat d'aprenentatge pot adaptar-se a situacion diverses.

1.6. Tipus de xarxes neuronals.

Segons un estudi de Hecht-Nielsen [Niels88] realitzat l'any 1988, hi ha més de 50 tipus de xarxes neuronals actualment en ús, de les quals al menys 13 s'utilitzen freqüèntment. La figura 3 mostra un resum amb les més destacades.

Neural Network Model	Network Topology *PE layers*	range of input values	Recall/Learning Phase *Propagation Rule*	*Threshold Function*	*Weight Updating*	*Error Calculation*
Hopfield/ Kohonen	single-layer with feedback	binary	$net = \sum S.W$	hard limiter	$\Delta w_{ij} = s_i.s_j$	
Perceptron	single-layer feed-forward	binary or continuous	$net = \sum S.W$	hard limiter	$\Delta w_{ij} = \eta.s_i.e_j$	$e_j = t_j - s_j$
Perceptron (Delta Rule)	single-layer feed forward	continuous	$net = \sum S.W$	linear	$\Delta w_{ij} = \eta.s_i.e_j$	$e_j = t_j - s_j$
Back Propagation	multi-layer bi-directional links	continuous	$net = \sum S.W$	sigmoid	$\Delta w_{ij} = \eta.s_i.e_j$	$e_{jo} = T'(net).(t_j - s_j)$ $e_{jh} = T'(net).\sum E.W$
Boltzmann Machine	multi-layer or randomly connected	binary	$net = \sum S.W$	sigmoid	$\Delta w_{ij} = \eta.e_j$	$e_j = \eta.(<p_{ij}> - <p'_{ij}>)$
Counter Propagation	multi-layer feed forward	binary	$net = \sum S.W$	hard limiter	$\Delta w_{ij1} = -\eta.e_{j1}$ $\Delta w_{ij2} = \eta.e_{j2}$	$e_{j1} = s_i - w_{ij1}$ $e_{j2} = \eta_1.t_j - \eta_2.w_{ij2}$
Self-Organising Map	two-dimensional grid of outputs PEs	continuous	$net = \sum S.W$	sigmoid	$\Delta w_{ij} = \eta.e_j$	$e_j = s_i - w_{ij}$
Neocognitron	hierarchical multi-layer feed forward	continuous	$net = \frac{(1+S_e.W_e)}{(1+S_h.W_h)} - 1$	linear	$\Delta w_{ij} = \eta.s_i$	

Figura 1.3. Models de xarxes neuronals.

1) *Xarxes de Hopfield/Kohonen* ([Hopfi82],[Kohon85])

És un model de memòria associativa unicapa, amb todes les neurones interconnectades, amb pesos simètrics en el cas particular de la xarxa de Hopfield. Les entrades són originalment binàries, i la funció llindar és de tipus graó.

Como ja s'ha anomenat, a la xarxa neuronal se li associa una funció energia E definida utilitzant una terminologia eléctrica com:

$$E = -\frac{1}{2}\sum_i \sum_j G_{ij} V_i V_j - \sum_i I_i V_i \qquad (1.6)$$

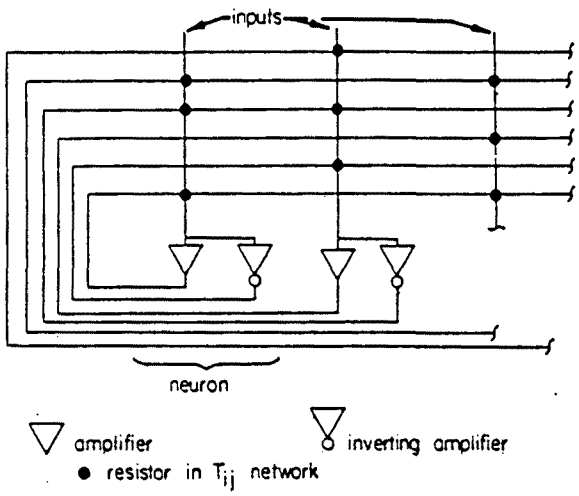


Figura 1.4. Xarxa de Hopfield.

A la xarxa se li entra una configuració i es deixa evolucionar lliurement. Es demostra que la xarxa evoluciona cap a mínims de la funció energia. El número de patrons que poden emmagatzemar (és una esperança de recuperació alta, de l'ordre de 90%) és de 0.15*N, essent N el número de neurones de la xarxa, si els patrons estan correlacionats.

2) *Perceptró* ([Rosen59])

Es una xarxa neuronal uni-capa, amb entrades que poden ser binàries o contínues, funció llindar del tipus escaló, i per a la qual existeix un teorema de convergència per a la regla d'aprenentatge associada. El problema, com ja s'ha esmentat, es basa en que les entrades han de ser separables per que el procés convergeixi (figura 5).

Box 4. The Perceptron Convergence Procedure

Step 1. Initialize Weights and Threshold

Set $w_i(0)$ $(0 \leq i \leq N-1)$ and θ to small random values. Here $w_i(t)$ is the weight from input i at time t and θ is the threshold in the output node.

Step 2. Present New Input and Desired Output

Present new continuous valued input $x_0, x_1 \ldots x_{N-1}$ along with the desired output $d(t)$.

Step 3. Calculate Actual Output

$$y(t) = f_h\left(\sum_{i=0}^{N-1} w_i(t)x_i(t) - \theta\right)$$

Step 4. Adapt Weights

$$w_i(t+1) = w_i(t) + \eta[d(t) - y(t)]x_i(t),$$
$$0 \leq i \leq N-1$$

$$d(t) = \begin{cases} +1 \text{ if input from class A} \\ -1 \text{ if input from class B} \end{cases}$$

In these equations η is a positive gain fraction less than 1 and $d(t)$ is the desired correct output for the current input. Note that weights are unchanged if the correct decision is made by the net.

Step 5. Repeat by Going to Step 2

Figura 1.5. Procediment de convergència del perceptró.

3) *Regla Delta* ([Rume86])

És una generalització del perceptró desenvolupada per Widrow-Hoff amb objecte de salvar les limitacions d'aquest. La "Delta rule" utilitza el métode dels mínims quadrats per a minimitzar l'error entre la sortida desitjada de la xarxa i la suministrada per aquesta. Com a major diferència respecte al perceptró, la regla Delta treballa amb una funció llindar de tipus lineal.

La regla d'aprenentatge intenta associar parelles d'entrada-sortida (aprenentatge supervisat). En primer lloc, s'introdueix el patró d'entrada, i es deixa a la xarxa que produeixi lliurement la seva sortida. Aquesta sortida es compara amb el patró desitjat, i els pesos es modifiquen d'acord amb la regla:

$$\Delta_p w_{ji} = \rho\,(t_{pj} - o_{pj}) i_{pj} = \rho\,\delta_{pj} i_{pj} \qquad (1.5)$$

T_{pj} és la j-èssima component del patró de sortida p, i O_{pj} la j-èssima component de la sortida produida per la xarxa. I_{pi} és la i-èssima component del patró d'entrada. Δw_{ji} és la modificació a aplicar al pes de la connexió entre les neurones j,i.

4) *Backpropagation* ([Rumel86])

És una xarxa multi-capa, d'aprenentatge supervisat basat en una generalització a varies capes de la regla Delta. La idea bàsica del backpropagation és propagar enrera els errors a les unitats ocultes que no reben influència directa de les entrades. Així, les unitats de sortida calculen l'error entre la sortida esperada i l'obtinguda, l'error es propaga a les unitats ocultes (neurones de les capes intermitjes) i aquestes avaluen el seu error. La modificació dels pesos es realitza en funció dels errors obtinguts per a totes les unitats.

Box 6. The Back-Propagation Training Algorithm

The back-propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multilayer feed-forward perceptron and the desired output. It requires continuous differentiable non-linearities. The following assumes a sigmoid logistic non-linearity is used where the function $f(\alpha)$ in Fig. 1 is

$$f(\alpha) = \frac{1}{1 + e^{-(\alpha-\theta)}}$$

Step 1. Initialize Weights and Offsets

Set all weights and node offsets to small random values.

Step 2. Present Input and Desired Outputs

Present a continuous valued input vector $x_0, x_1, \ldots x_{N-1}$ *and specify the desired outputs* $d_0, d_1, \ldots d_{M-1}$. If the net is used as a classifier then all desired outputs are typically set to zero except for that corresponding to the class the input is from. That desired output is 1. The input could be new on each trial or samples from a training set could be presented cyclically until weights stabilize.

Step 3. Calculate Actual Outputs

Use the sigmoid nonlinearity from above and formulas as in Fig. 15 to calculate outputs $y_0, y_1 \ldots y_{M-1}$.

Step 4. Adapt Weights

Use a recursive algorithm starting at the output nodes and working back to the first hidden layer. Adjust weights by

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x_i'$$

In this equation $w_{ij}(t)$ is the weight from hidden node i or from an input to node j at time t, x_i' is either the output of node i or is an input, η is a gain term, and δ_j is an error term for node j. If node j is an output node, then

$$\delta_j = y_j(1 - y_j)(d_j - y_j),$$

where d_j is the desired output of node j and y_j is the actual output.

If node j is an internal hidden node, then

$$\delta_j = x_j'(1 - x_j')\sum_k \delta_k w_{jk},$$

where k is over all nodes in the layers above node j. Internal node thresholds are adapted in a similar manner by assuming they are connection weights on links from auxiliary constant-valued inputs. Convergence is sometimes faster if a momentum term is added and weight changes are smoothed by

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x_i' + \alpha(w_{ij}(t) - w_{ij}(t-1)),$$

where $0 < \alpha < 1$.

Step 5. Repeat by Going to Step 2

Figura 1.6. Algorisme d'aprenentatge del backpropagation.

El backpropagation treballa amb una funció llindar de tipus sigmoidal, i els valors d'entrada són continus.

5) *Boltzmann machine* ([Rumel86],[Ackle85])

Es tracta novament d'una xarxa neuronal multi-capa, que utiliza les técniques del "simulated annealing" per a modificar els pesos. La màquina de Boltzmann accepta valors d'entrada binaris, i aplica la distribució de Boltzmann per ajustar els pesos.

La regla d'aprenentatge és una modificació de la de Hopfield per a permetre el "simulated annealing". El gap d'energia entre els estats 1 i 0 de la neurona k és E_k, l'estat següent de l'anomenada neurona S_k pendrà el valor 1, independentment de l'estat anterior, amb una probabilitat:

$$p_k = \frac{1}{1+e^{-\frac{E_k}{T}}} \tag{1.8}$$

on T és un paràmetre de "temperatura" del sistema. Aquesta regla de decisió local assegura que en l'equilibri térmic la probabilitat relativa de dos estats globals (P_a i P_b) depén únicament de la seva energia (E_a i E_b), i sigueixen la distribució de Boltzmann,

$$\frac{P_a}{P_b} = e^{-\frac{(E_a - E_b)}{T}} \tag{1.9}$$

A temperatures baixes, s'aconsegueixen amb major probabilitat els valors de baixa energía, però el procés és lent. A temperatures altes és més probable quedar-se en "estats cíclics", però el procés és ràpid. El "simulated annealing" proposa fer evolucionar la xarxa primer a temperatures altes, i després baixar gradualment la temperatura, com un métode d'evitar els mínims locals d'energia.

6) *Counter propagation*

És una xarxa de tres capes, i un algoritme d'aprenentatge de dues fases. Durant la primera fase, els pesos de les connexions que van de la capa d'entrada a la capa d'unitats ocultes s'actualitzen en funció de l'algoritme no supervisat de Kohonen (només una neurona de la capa oculta està activa durant la fase

d'aprenentatge). En la fase dos s'actualitzen els pesos de les conexions que van de la capa oculta a la capa de sortida, en funció de l'algoritme supervisat de Grossberg. És una xarxa que aprén ràpidament encara que, degut a que sol una neurona de la capa oculta està activa cada vegada, és incapaç d'apendre configuracions d'una certa complexitat.

7) *Mapes auto-organitzats*

Tenen un vector de neurones d'entrada connectat a una matriu bidimensional de neurones de sortida. Els nodes de sortida estan molt interconnectats (encara que no totalment), amb connexions fonamentalment de caràcter local. L'algorisme d'aprenentatge és no-supervisat.

8) *Neocognitró* ([Fukus88])

És una xarxa neuronal jeràrquica, multi-capa, unidireccional (feed-forward), construida alternant capes de dos tipus: Capes de cel.les S, o cel.les d'extracció de característiques, i capes de cel.les C, que ajuden a corregir qualsevol error posicional en els patrons a apendre.

L'existència de les celdes C proveeix el sistema d'una certa capacitat de reconèixer patrons degradats. S'usa principalment com classificador.

1.7. Funcionalitat associada la topologia de la xarxa

Els elements que descriuen les xarxes neurals que hem enumerat a l'apartat 1.3 són "uniformes" per a una xarxa donada de manera que les propietats de les xarxes neuronals vindran donades bàsicament per la seva organització espaial, que és una propietat "col.lectiva" i que es reflexa en el que anomenem la topologia de les interconnexions.

Aquest concepte és realment important ja que un cop d'ull al tipus de topologia que proposa una determinada xarxa ens dirà a grans trets quin tipus de funcionalitat realitza.

El número de capes o "layers" neurones, la direcció de les connexions entre dos layers, bé sigui unidireccional o bidireccional, la existència de realimentació dins d'un mateix layer i altres són alguns dels criteris que, de forma similar a les

estructures sintetitzades mitjançant disseny lògic o analògic, ens permetran de predir les propietats d'una determinada xarxa.

La regla d'aprenentatge, en canvi, donarà el procés i les constants associades que s'encarregaran d'especificar amb un nivell de precisió superior les característiques de la xarxa, donant els valors de les sinapsis.

1.7.1 Algebra d'hiperplans

El concepte d'hiperplà fa referència a la superfície que "selecciona" l'estat de la xarxa per a cada neurona. Si veiem el conjunt de pesos sinàptics associats a una neurona com un vector director d'un hiperplà, i l'estat de la xarxa com un vector en aquest hiperespai, el camp local de la neurona, donat pel producte escalar entre els dos vectors indica el grau d'orientació d'aquest estat. Si la funció de transferència utilitzada és del tipus signe, llavors la sortida de la neurona indica si l'efecte global del pattern d'entrada ha estat excitador o inhibidor. En ambdós cassos, la sortida de la neurona mostra una característica activa.

Aquest característica diferencia aquesta àlgebra de l'algebra booleana, que podriem obtenir amb neurones {0,1}, en la qual només s'activarien els hiperplans actius (camp local positiu), i per tant només no totes les neurones restarien actives per a cada estat d'entrada.

En el cas de neurones lineals o sigmoidals, el concepte d'hiperplà és útil en termes de visualització de les zones de classificació encara que l'operació amb valors continus fa perdre molt del sentit associat a la lògica bivaluada.

Per això, tot i que és una eina usual en la visualització de les xarxes neuronals, no existeix una formulació del que anomenem l'algebra d'hiperplans.

Aquests conceptes, però, ens seran utils en els apartats següents en els que mostrarem la possibilitat d'utilització de les xarxes neurals com a alternativa al disseny lògic.

1.7.2 Sintesi de circuits combinacionals

La sintesi de funcions combinacionals és un dels camps en els quals el disseny lògic és més potent ja que és capaç de construir qualsevol funció lògica amb només dos nivells de portes lògiques del tipus nand o nor.

Aquest resultat és senzill de reproduir amb una xarxa neuronal amb característiques determinades.

Si definim la neurona de forma similar a la definició McCulloch-Pitts (figura 1.7), es possible construir emb el mateix esquema qualsevol tipus porta lògica simple (Inversor, nand, nor, etc.), tal i com es mostra a la taula I.

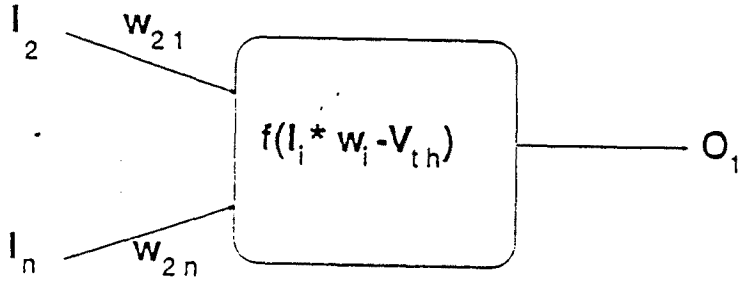


Figura 1.7. Neurona utilitzada per a la sintesi de funcions combinacionals.

Funció	W_{1n}	V_{th1}
Inversor	-1	0
AND n	1	n-1
OR n	1	1-n
NAND n	-1	n-1
NOR n	-1	1-n

Taula I. Valors de sinapsis i llindars per a les diferents funcions lògiques sintetitzades.

Aquesta forma senzilla de sintetitzar funcions universals (nand, nor) permet, a l'igual que la sístesi lògica, el disseny de qualsevol funció combinacional en dos nivells de neurones, a les quals s'afegeix un nivell de neurones d'entrada per donar l'estructura multilayer perceptron de tres capes, que és una de les més utilitzades en xarxes neuronals.

Els algorismes d'aprenentatge associates a les xarxes però, no utilitzen normalment la síntesi basada en neurones amb threshold, sinò que únicament es determinen els valors de les sinapsis i es considera el threshold constant i igual per a totes les neurones de la xarxa.

Amb aquesta restricció, es poden constuir funcions combinacionals encara que la seva síntesi no és tan directa, donada la poca flexibilitat que tenim sobre l'algebra d'hiperplans.

Les regles d'aprenentatge s'encarreguen d'obtenir, normalment amb processos iteratius i partint únicament de les dades, els valors de les sinapsis a implementar.

En el cas de funcions combinacionals les regles d'aprenentatge (tipus backpropagation) poden adaptar-se per d'obtenir valors discrets per als pesos sinaptics (apèndix 1).

1.7.3. Sintesi d'elements de memoria i oscil.ladors

El disseny de màquines algorísmiques requereix, a més dels circuits combinacionals, els circuits seqüencials. La base d'aquests circuits són els elements de memòria que permeten emmagatzemar dades que són carregades sota el control d'un senyal particular (senyal de càrrega o de rellotge). Aquest esquema es necessari quan es realitza una ordenació seqüencial, o bé quan les dades tractades s'obtenen amb uns certs criteris temporals (p.e. dades provinents de sensors).

Llavors seria d'esperar que mitjançant xarxes neuronals es poguessin obtenir certes característiques de memòria.

Per fer aquesta síntesi, analitzarem l'esquema d'un punt de memòria: un latch RS (figura 1.3.a). La característica bàsica d'aquest dispositiu és l'existència d'una realimentació interna que permet la biestabilitat del dispositiu.

Una de les característiques més remarcables de certes xarxes neuronals és l'existència d'una realimentació entre totes les neurones de la xarxa. Per tant, assignant els valors dels pesos convenientment, podem recuperar l'estructura del dispositiu de memòria (figura 1.3.b). En aquest cas, utilitzem les mateixes neurones definides anteriorment.

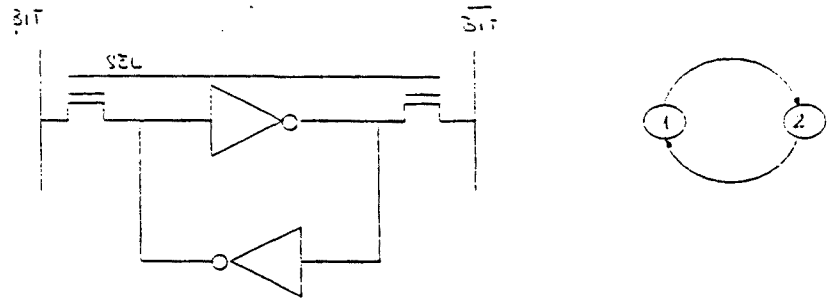


Figura 1.8. Esquema d'un latch . (a) i la xarxa neuronal equivalent (b).

La introducció d'un senyal de rellotge pot realitzar-se de manera equivalent encara que en general la sincronització de les xarxes neuronals. Aquesta alternativa és però, poc utilitzada a nivell d'implementació de les xarxes i es segueix la filosofia de les xarxes neuronals naturals d'utilitzar elements asíncrons.

Tot i això, els senyals de rellotge u oscil.ladors també poden ser sintetitzats mitjançant xarxes neuronals, de forma similar a la síntesi dels elements de memòria. Aquests, es caracteritzen per que el número d'inversions necessàries per tal d'aconseguir la biestabilitat és parell, mentre que en la síntesi d'oscil.ladors és senar. Llavors, les característiques de l'oscil.lació venen donades per les característiques temporals de cada implementació. La figura 1.4 mostra un esquema simple d'anell oscil.lador sintetitzat a nivell de protes lògiques, i la xarxa neuronal equivalent.

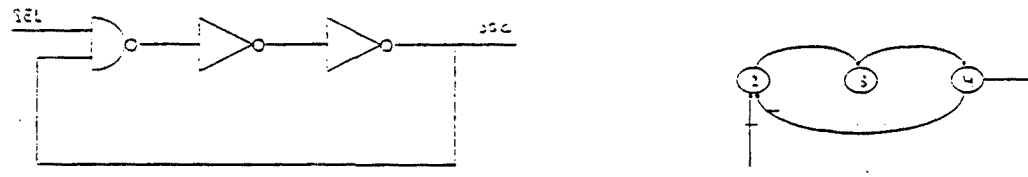


Figura 1.9. Esquema oscil.lador (a) i xarxa neuronal equivalent (b).

5.4 Sintesi de circuits seqüencials

La síntesi de circuits sequencials es pot fer utilitzant funcions combinacionals i elements de memòria. En aquest cas la regla d'aprenentatge s'encarregarà de fer aprendré al sistema no només cominacions entrada sortida sinó les relacions temporals que es dedueixen de l'ordre en que es presentes.

Els diversos algorismes d'aprenentatge que manipulen seqüencies temporals (Elmann, Jordan, Weibel, etc.), utilitzen una formulació d'aquest estil sense dedicar explícitament interconnexions per a funció de memòria. Tanmateix, utilitzan aquests recursos implicitament quan retarden senyals.

Les topologies d'aquest tipus de xarxes (a la figura 1.5 es poden veure les corresponents a xarxes d'Elmann i Jordan) són molt similars als esquemes clàssics corresponents a les màquines de Moore y Mealy.

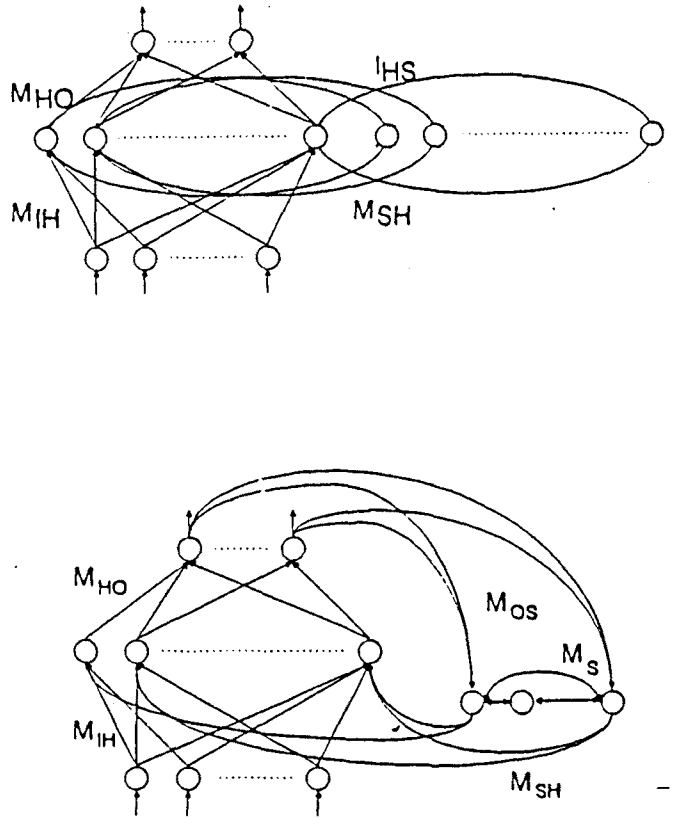


Figura 1.10. Topologia per a les xarxes d'Elman (a) i Jordan (b).

5.5 Síntesi de funcions complexes

En els dos apartats anteriors hem vist que minjançant xarxes neuronals podem sintetitzar tant funcions combinacionals com seqüencials. Tot i això la potència de les xarxes neuronals artificials no ve donada per la possible competència que puguin fer a les técniques de disseny lògic ja que aquestes estan consolidades i disposen de tecnologies que les implementen de forma òptima.

L'aventatge principal de les xarxes neuronals ve donat per la simplicitat de la síntesi, mitjançant les regles d'aprenentatge, del circuit per a funcionalitats complexes, manipulació de dades amb soroll, etc. En aquest cas la xarxa realitza no només la síntesi sinó que també pren certes decisións, com per exemple, si un vector d'entrada pertany a un conjunt o un altre, etc.

D'entre les funcionalitats complexes que tenen una relació directa amb la topologia de la xarxa mostrarem aquelles que presenten una estructura més diferent d'estructures clàssiques.

Típicament la memòria associativa, i la sintesi i reconeixement de seqüencies temporals requeriran de estructures amb retroalimentació, mentre que el reconeixement de patrons i la classificació podran realitzar-se únicament amb xarxes feed-forward.

Capítol 2

MICROELECTRÒNICA APLICADA A XARXES NEURONALS

La realització microelectrònica de xarxes neuronals ha estat un dels puntals del resorgiment de les xarxes neuronals. Disposar d'una tecnologia amb unes prestacions elevades significa fer possible la resolució de problemes amb un alt grau de paral.lelisme i complexitat elevada. Les velocitats de procés associades a la tecnologia CMOS (100 MHz) i la densitat d'integració 10^6 transistors/xip, estan molt per sobre dels disponibles en la primera etapa de les xarxes neuronals (anys 60s), en la qual es disposava d'operacionals de baixes prestacions (ample de banda per sota del KHz) i 100 transistors/xip.

2.1. Nivells d'abstracció associats a les xarxes neuronals.

Els criteris d'implementació microelectrònica no han partit necessàriament de la descripció de la xarxa a alt nivell, segons criteris d'utilització funcional, sinó que han pres com a base altres estructures associades a les xarxes.

Les estructures de que disposem en el disseny de xarxes neuronals estan lligades a la "jerarquia" intrínseca d'aquestes, ordenada segons el nivell d'abstracció.

SISTEMA

XARXA

LAYER

NEURONA

TECNOLOGIA

En cadascun d'aquests nivells, existeix un conjunt de variants que ens permeten obtenir diferents característiques.

A nivell de neurona podem definir diferents funcions d'activació i de sortida, però també característiques temporals, restriccions sobre els pesos, etc.

A nivell de layer, es pot definir el grau de connectivitat (topologia) entre les neurones d'un layer (local, global, inexistent).

Al nivell de xarxa estan normalment associades les regles i processos d'aprenentatge així com les característiques funcionals.

Mentre que a nivell de sistema, entés com a conjunt de xarxes, es defineixen funcionalitats amb un grau de complexitat més elevat.

Aquestes definicions no són però restrictives, de manera que existeixen xarxes composades d'un únic layer, o sistemes que utilitzen la mateixa regla d'aprenentatge per a totes les xarxes de les quals es composa, i que fins i tot defineixen una nova xarxa neuronal.

Tots aquest nivells al seu torn són independents de la tecnologia utilitzada. En el cas de xarxes neuronal naturals, la tecnologia utilitzada és la tecnologia bioquímica. Les xarxes neuronals artificials utilitzen altres suports com ara la microelectrònica, l'òptica o la biotecnologia.

La microelectrònica és, de totes les tecnologies que permeten fer xarxes neuronals artificials, la més estabilitzada en termes d'àmplia disponibilitat del procès tecnològic estandard, elevat grau d'automatització del procés de disseny, criteris de disseny funcional d'alt nivell (lògics i analògics), etc., tot i que no sigui segurament la més òptima.

2.2. Nivells d'abstracció associats a la microelectrònica.

L'estandardització i l'ús industrial de la tecnologia microelectrònica han permés de generar una estructutra molt més rica que s'associada a altres tecnologies, en termes de la generació de nivells d'abstracció intermedis entre les funcionalitats i el procés tecnològic, fins al punt que els processos d'alt nivell es defineixen independentment de la tecnologia. Aquesta independència és similar

a de les xarxes neuronals artificials respecte de la bioquímica entesa com a procés tecnològic de creació de les xarxes neuronals naturals.

El disseny de sistemes basats en tecnologia microelectrònica es pot estructurar de forma jeràrquica en funció del nivell d'abstracció, segons el següent esquema

SOFTWARE

HARDWARE

XIP

DISSENY LOGIC DISSENY ANALOGIC

PORTES LOGIQUES PORTES ANALOGIQUES

LAYOUT

MODELS

DISPOSITIUS

PROCES TECNOLOGIC

Sobre aquest esquema, la implementació de xarxes neuronals artificials pot partir de qualsevol nivell de la jerarquia malgrat que, amb o sense la intervenció directa de l'usuari, sempre serà necessari d'arrivar al nivell de procés.

El nivell del qual es parteixi determinarà algunes de les característiques de les xarxes degut a l'existència de passos intermitjos sobre els quals s'aplicaran

processos d'automatització. Per exemple, si es treballa a nivell de software amb llenguatges de descripció d'alt nivell, l'automatització s'estendrà des del compilador al sistema informàtic, els xips interns, etc. mentre que si treballem a nivell de layout, en full-custom, l'automatització serà referent bàsicament al procés de fabricació i caracterització paramètrica.

Els trets distintius de les dues estratègies extremes serien:

- Nivells d'abstracció alts:
  - . Desenvolupar sistemes més complexos i diversos.
  - . Alt nivell de programabilitat i reconfiguració.
  - . L'ús d'eines de desenvolupament més potents.
  - . Un alt grau d'automatització.
  - . Poc control sobre els processos automatitzats.
  - . Models poc precisos.

- Nivells d'abstracció baixos:
  - . Funcionalitats específiques.
  - . Llibertat a l'hora de dissenyar dispositius.
  - . Optimització de certes característiques (temps, superfície,..).
  - . Major fiabilitat dels paràmetres utilitzats.
  - . Temps de desenvolupament més elevats.
  - . Menor disponibilitat d'eines d'ajuda al disseny.

Dins el rang que abasta el disseny de sistemes amb tecnologia microelectrònica, el disseny de circuits integrats VLSI lligaria els nivells de descripció que van des del xip fins al procés, mentre que els nivells més alts de descripció estarien lligats a altres branques de la concepció electrònico-informàtica.

Des d'aquest punt de vista global, la majoria de realitzacions computacionals de xarxes neuronals han estat realitzades a nivell de software. En aquestes implementacions, els elements neuronals de baix nivell: neurones i

sinapsis, estan descrits normalment de forma molt simplificada, típicament com a funcions simples els primers i com a elements de matriu els darrers.

Les realitzacions a nivell de sistema hardware utilitzen estructures específiques per tal d'accelerar alguns dels càlculs involucrats, com per exemple els productes vector-matriu.

Les dues aproximacions més potents són: la utilització de **processadors digitals de senyal,** a nivell d'acceleració dels càlculs elementals (realització ràpida dels productes vector-matriu amb l'ajuda de dispositius especialitzats) o de **sistemes basats en transputers,** a nivell d'arquitectures paral.leles, donada la seva elevada velocitat de transmissió de les dades entre processadors (amb càlcul i memòria interns).

Les aplicacions amb aquests sistemes són, però, poc òptimes ja que les eines per treballar-hi són poc desenvolupades (software per l'assincronisme de les comunicacions, compiladors, etc.) des del punt de vista de resoldre problemes de xarxes.

Altres aproximacions amb processadors menys potents però amb soft més desenvolupats (68030+68880, 80386+80387,...) són poc òptimes per a la realització dels tipus de càlculs que realitzen les xarxes neuronals.

Algunes d'aquestes xarxes estan desenvolupades a nivell comercial i sobre elles s'hi realitzen aplicacions industrials.

3. Tendències fonamentals d'implementacions VLSI.

Les implementacions microelectròniques de xarxes neurals intenten materialitzar les xarxes fins al nivell de neurona: sistema, xarxa, layer, neurona, en algun dels nivells del procés de disseny abans esmentats.

L'avantatge principal d'aquest tipus d'implementacions és que, tant el procés tecnològic com les diferents estratègies de disseny microelectrònic, han desenvolupat **eines y metedologies potents d'ajuda a la concepció de sistemes:** jerarquia del disseny, simulació (de dispositius, elèctrica, temporal, lògica, estructural, funcional, de sistema, etc.), testabilitat, verificació, caracterització,

alternatives d'implementació (full-custom, gate arrays, standard cells, generadors, etc.), disseny de sistemes electrònics, etc.

En gairebé tots aquests estadis intermitjos hi ha hagut realitzacions de xarxes neurals de característiques i complexitats diferents.

Potser l'únic camp relacionat amb les xarxes neurals artificials que no ha estat objecte d'un esforç important des del VLSI ha estat **la implementació d'algorismes d'aprenentatge.** Aquest abandó es produeix per una qüestió de **generalitat.** Mentre que sobre una xarxa podem utilitzar diferents algorismes d'aprenentatge, les regles d'aprenentatge implementen diferents operacions, fins i tot per un tipus determinat de xarxa neuronal.

Una causa adicional ve donada pel fet que, una part important de les xarxes neuronals, principalment les realitzades a baix nivell, implementen **funcionalitats específiques no programables,** per la qual cosa l'aprenentatge, si existeix, és previ al procés de concepció dels xips.

Les estructura física de la xarxa (sintesi de neurones i sinapsis) de les diferents realitzacions existents són molt variades, i en cada cas les seves característiques estàn restringides en funció de diferents paràmetres: velocitat de procés, número de neurones, complexitat de les neurones, número de bits necessaris per representar els pesos, grau de connectivitat de la xarxa, grau de programabilitat, etc.

A tall d'enumeració, les realitzacions VLSI de xarxes neuronals o d'estructures de base per aquestes són:

- Nivell de sistema. Transputers.
  Xips/Sistemes de processat digital de senyal.
  Coprocessadors.
  Processadors específics.
  Integració a nivell d'oblea (WSI).

- Nivell d'estructura funcional. Sistòlics.
  ALUs simplificades (per a WSI).

- Nivell de porta lògica. Lògica de pulsos.
  Lògica aleatòria.

- Nivell de porta analògica. Xarxes no programables (Hopfield, Kohonen,..)
  Xarxes analògiques programables.
  Xarxes neuronals sensores amb models MOS subllindar.

- Nivell de layout. Xarxes programables a nivell de transistor.
  Implementació de resistències.

- Nivell de model. Model MOS per a digital com a resistència.
  Model MOS subllindar per a xarxes analògiques.

- Nivell de procés. Modificacions al procés MOS (fusibles, EEPROM, EPROM, polisilici resistiu).
  Dispositius amb acoblament de càrregues (CCD).
  Dispositius bipolars.
  Dispositius optoelectrònics.

Aquests diferents punts de vista reflexen tant la complexitat del desenvolupament adquirit per la tecnologia microelectrònica, com l'elevat nivell d'especialització necessari per a comprendre i avaluar cadascuna de les implementacions esmentades.

Al mateix temps, l'avaluació des d'un punt de vista neuronal, tampoc no és senzilla ja que s'inclouen el mateix camp de recerca, des de nous dispositius que poden realitzar de manera més òptima alguna de les funcionalitats requerides, fins a xarxes de característiques programables (topologia, número de neurones, etc.).

El criteri de classificació que hem escollit, **el grau de programabilitat de les xarxes**, intenta establir un compromís entre les característiques associades a la utilització de les xarxes neuronals i les lligades a la realització microelectrònica.

Des del punt vista de les xarxes neuronals podriem diferenciar les xarxes segons **l'algorisme d'aprenentatge** utilitzat, de forma que tenim xarxes amb aprenentatge supervisat i xarxes amb aprenentatge no supervisat.

Aquest criteri no involucra necessàriament la topologia de la xarxa, ja que sobre una xarxa amb una topologia determinada podem utilitzar diferents tipus d'aprenentatge, encara que permet una diferenciació basada directament en el procés d'obtenció dels pesos.

Les **regles d'aprenentatge supervisades**, realitzen un procés de comparació de les sortides de la xarxa sense entrenar amb les sortides esperades per tal de generar els pesos, normalment mitjançant processos iteratius. Diferents condicions sobre els paràmetres de la iteració (condicions inicials, velocitat de l'aprenentatge, etc.), poden portar a solucions diferents.

En canvi, les **regles d'aprenentatge no supervisades** no realitzen l'aprenentatge en funció d'una comparació sinó que, o bé, donada una funcionalitat concreta sobre uns valors específics determinen de manera directa tots els valors associats a la xarxa, o utilitzen algorismes que, mitjançant processos iteratius sobre les entrades, determinen els coeficients de la xarxa (sistemes autoorganitzats). El determinisme podem trobar-lo igualment en la síntesi de sistemes neuronals basats en òrgans sensors o de preprocés, encara que en aquest cas la programabilitat no és necessària donat que la funcionalitat és específica.

Des del punt de vista de la microelectrònica, i per a sistemes de processat de la informació, la classificació bàsica es realitza en funció del mètode de tractament i representació de les dades. Existeixen dues estratègies bàsiques que també es reflexen en l'estructuració dels mètodes de disseny microelectrònics: **el disseny de circuits integrats digitals i el disseny de CIs analògics.**

Aquestes dues estratègies estan suportades per dos mètodes de disseny diferents: el disseny lògic, basat en **l'algebra de Boole**, i el disseny analògic, basat en **l'àlgebra de les transformades de senyal (s, z,...).**

Les **xarxes neuronals digitals** realitzen els productes sinàptics i les sumes de productes amb estructures de càlcul complexes i costoses (ALUs, multiplicadors, etc.), que no permeten un grau de paral.lelisme tant elevat com les analògiques, però que permeten l'expansió de la dimensió de la xarxa de

manera més simple. Aquestes realitzacions, però, utilitzen **elements comuns** en el disseny de circuits digitals.

Les **xarxes neuronals analògiques** basen els càlculs de sumes de productes sinàptics en **la suma de corrents**, d'aquesta manera poden realitzar un gran número de càlculs en paral.lel.

Aquestes realitzacions presenten elements nous en el camp del disseny de circuits integrats analògics: el treball amb **models subllindar per al transistor MOS, i l'augment de la connectivitat interna dels circuits.**

En els següents paràgrafs en presentem les innovacions més importants en aquest camp: el disseny de xarxes programables d'alta velocitat i el disseny de xarxes neuronals de preprocés.

4. Xarxes neuronals programables d'alta velocitat

La idea de realitzar xarxes neuronals programables paral.leles d'alta velocitat, parteix directament de les propostes d'en J. Hopfield [Hopfi84] de realitzar elèctricament les sinapsis com a resistències $R_{ij} = 1/T_{ij}$, de manera que el camp local $\Sigma T_{ij}O_j$, es calcula com a suma de corrents $\Sigma_j V_j/R_{ij}$. Aquest valor entra a un amplificador que realitza la funció d'activació, la sortida del qual és l'estat de la neurona V_j. El circuit equivalent es mostra a la figura 2.1.

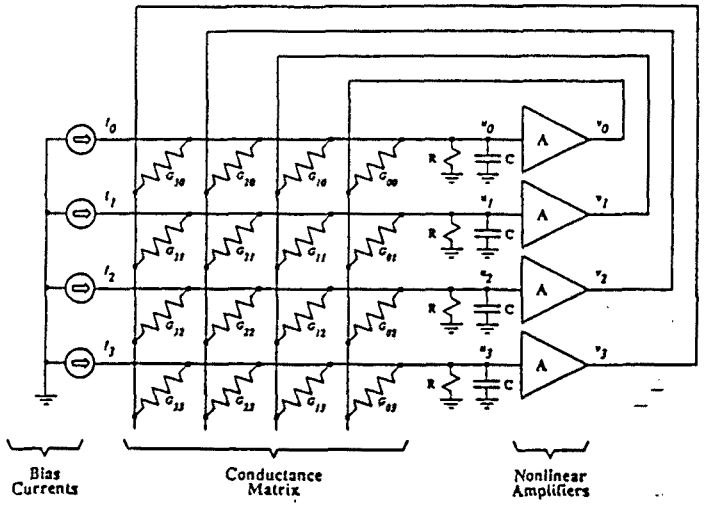


Figura 2.1. Implementació de xarxa neuronal a nivell elèctric.

La xarxa més general possible, i per tant la que ha estat implementada més sovint, és la xarxa d'Hopfield, amb les neurones totalment interconnectades.

Per a xarxes totalment interconnectades en les quals totes les neurones i sinapsis estan implementades físicament, la dimensió del xip neuronal ve donada per la dimensió de la sinapsi bàsica, de les quals n'hi ha N^2, respecte de N neurones. Si a més han de ser programables, les sinapsis estaran composades del dispositiu que realitza la funció de resistència, les cel.les de memòria corresponents al pes que dóna el valor de la resistència i d'una lògica d'operació. Per aquests motius resulta crítica, en la implementació de xarxes neuronals paral.leles i programables, la **reducció de les dimensions de les sinapsis.**

Aquest procés ha estat paral.lel a la reducció de dimensions dels dispositius en les tecnologies MOS, la qual cosa ha permés d'augmentar en un ordre de mangnitud el número de sinapsis per xip en quatre anys.

En aquests moments, ambdues reduccions estan més o menys estabilitzades i per tant sembla que s'ha arrivat al límit d'aquestes implementacions.

A nivell de realitzacions, aquesta evolució s'ha fet també en el tipus de dispositiu utilitzat en els circuits. S'han utilitzat circuits analògics, dispositius basats en les propietats resistives dels transistors i altres dispositius basats en modificacions del procés tecnològic.

Les xarxes a nivell de transistor, han estat les més utilitzades per a la realització de xarxes neuronals programables amb procés paral.lel, ja que permeten un màxim número de neurones totalment interconnectades per xip.

La tecnologia CMOS no està optimitzada per a la realització de resistències (per a processos estàtics), ja que normalment es manipulen fonts de corrent o resistències no linials carregant i descarregant capacitats (procés dinàmic). Amb un procés estandard, la realització d'una resistència amb una superfície mínima es fa amb un transistor en zona de no saturació.

En aquest cas el transistor MOS es comporta com una resistència no lineal, amb una relació corrent-tensió donada per l'expressió següent:

$$I_{DS} = \beta((V_{GS}-V_T)V_{DS}-\frac{V_{DS}}{2}) \qquad (2.1)$$

Aquest comportament no és problemàtic si treballem amb funcions d'activació abruptes. En aquest cas, la linealitat només és important al voltant de la tensió de llindar.

Aquesta simplificació es pot realitzar ja que, per tal de tenir un número màxim d'interconnexions per xip, el rang de pesos permesos és $\{\pm 1,0\}$, i per tant s'utilitza un únic tipus de resistència per a les sinapsis excitadores o inhibidores. La forma d'implementar aquestes resistències depén de l'estratègia de disseny i de la tecnologia utilitzada.

En qualsevol cas, es busca que es produeixi una transició quan el signe de la diferència entre el número de sinapsis excitadores i el d'inhibidores associat a una neurona, no coincideix amb el signe de l'estat de la neurona (per neurones bipolars).

La precisió de la tensió de comparació, que ha de ser inferior a la mínima diferència entre aquests valors, determina el dispositiu que podem utilitzar com a neurona. Podem, per exemple, utilitzar un inversor mínim de precisió 0.3V per treballar amb menys de 30 neurones però, per a un número de neurones més elevat, hem d'utilitzar amplificadors millors com per exemple amplificadors diferencials.

Si el rang dels pesos augmenta, els efectes de segon ordre lligats al procés tecnològic (bec de moixó, etc.) modifiquen l'àlgebra bàsica ($1+1\neq 2$) i afecten la precisió dels pesos, de manera que algunes transicions que compleixen la condició numèrica es produiran i altres no.

Les implementacions que presentem són representatives de l'evolució seguida en implementacions que utilitzen aquesta estratègia, tant pel que fa a la complexitat dels elements utilitzats, com en el número de neurones o millores qualitatives.

4.1. Memòria autoassociativa.

Totes aquestes realitzacions utilitzen el transistor com a resistència i emmagatzemen els pesos en cel.les de memòria "digitals".

4.1.1. Primera Implementació NMOS.

La primera implementació de que tenim constància fou realitzada al Caltech per l'equip d'en Carver Mead [Sivil86], pioner de la utilització del VLSI en la realització de xarxes neuronals.

La idea bàsica d'aquesta realització és implementar les resistències com a transistors. Donat que a l'época la tecnologia més disponible era la NMOS d'empobriment, només es podien controlar els transistors de descàrrega.

Això obligava a una estratègia de doble raïl per a la realització del esquema excitació/inhibició. L'exitació activa un transistor de descàrrega d'una de les línies, mentre que la inhibició n'activa un que descarrega l'altra. El procés NMOS implica l'existència d'una càrrega resistiva fixa (pull-up) i l'activació d'un número major de transistors de descàrrega fa disminuïr el nivell de tensió a la sortida.

L'estat de la neurona es detecta amb un amplificador diferencial amb entrada i sortida doble raïl, dissenyat amb NMOS amb lògica subllindar que presenta un guany superior, connectat a les dues línies d'excitació/inhibició tal i com es mostra a la figura 2.2.

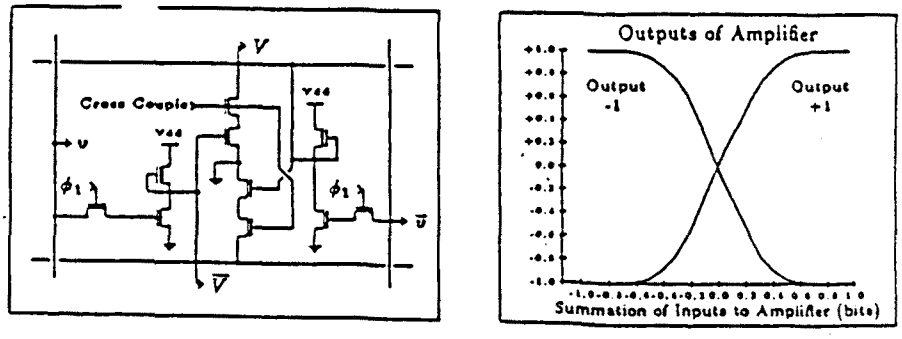


Figura 2.2.- Neurona NMOS en el model de Mead.

La sinapsi de la figura 2.3, permet la connexió entre les línies presinàptiques (v_i,v^{-i}) i postsinàtiques (V_i,V^{-i}) a través de transistors de pas, de manera que si la sinapsi es excitadora la correspondència és directa (v_i.V_i),(v^{-i}.V^{-i}), mentre que si és inhibidora és creuada (v_i.V^{-i}),(v^{-i}.V_i).

La programació es realitza mitjançant un autòmat intern a cada sinapsi, de manera que no és necessari introduir en sèrie (p.e. amb scan-path) els pesos.

El xip realitzat en NMOS 4 micres conté 22 neurones i ocupa 5.7 x 6.7 mm^2 amb 53 I/O pads. Com a memòria associativa pot emmagatzemar 3 patterns (!) amb propietats de recuperació d'errors.

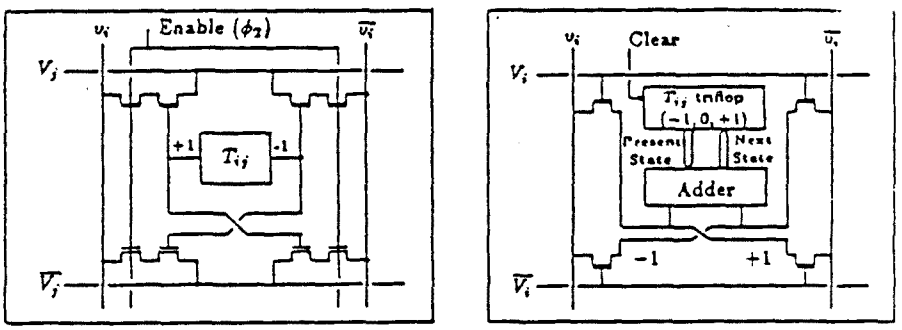


Figura 2.3.- Sinapsi NMOS en el model de Mead.

Un dels problemes principals associats a aquesta representació ve donat per l'acoblament entre les resistències de les sinapsis i la resistència de sortida de les neurones, que fa que el valor de sortida d'una neurona depengui del número de sinapsis actives sobre aquella línia.

Un altre comentari mereix el fet que tot i que el funcionament de les xarxes neuronals totalment interconnectades és asíncron, normalment s'introdueixen fases de rellotge per permetre sincronitzar el sistema. La raó d'aquesta sincronització podem trobar-la en la necessitat de testejar el xip, tasca no gens senzilla quan el grau de paral.lelisme és elevat.

4.1.2. Evolució cap al CMOS

Els treballs d'en Mead van evolucionar cap a la implementació de xarxes neuronals sensores analògiques treballant en la regió subllindar (apartat 5).

La implementació de xarxes neuronals programables va ser seguida, entre altres, per l'equip d'en Graf als Bell Labs [Graf86,7,8].

La pretensió d'en Graf era d'augmentar el número de neurones gràcies a dues millores: la utilització de sinapsis més compactes d'un sol rail presinàptic i la reducció de l'escala d'integració.

L'estratègia seguida per a la sinapsi es mostra a la figura 2.4. En ella es superen els problemes de l'acoblament d'impedàncies fent que la neurona postsinàptica ataqui la porta d'un transistor i el connecti bé a V_{dd} (excitadora) o V_{ss} (inhibidora).

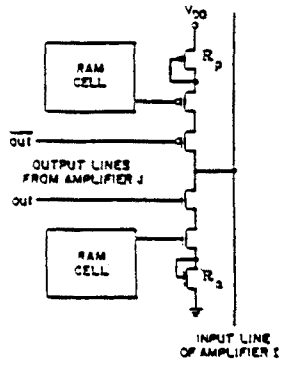


Figura 2.4.- Sinapsi CMOS en el model de Graf.

Com a neurona utilitza, per a poques neurones, un parell d'inversors amb la lògica d'entrada-sortida associada (figura 2.5).

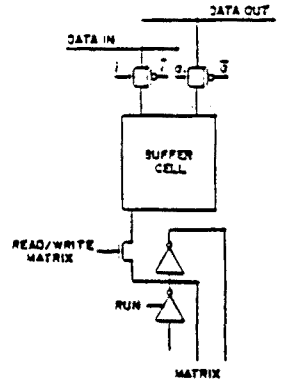


Figura 2.5.- Neurona CMOS en el model de Graf.

Això els ha permés l'any 1989, arrivar a implementacions en CMOS 0.9 micres, 400.000 transistors i una superfície de 1 cm^2, a 256 neurones totalment interconnectades que permeten xarxes amb pesos $\{\pm 1,0\}$.

Per al treball com a memòria associativa, proposa una xarxa amb un esquema de tipus xarxa d'Hamming amb un mecanisme competitiu que fa que després de la fase de relaxació només un pattern sigui actiu. A més, associa una etiqueta a cada pattern de manera que a la sortida pot tenir tnat el pattern recuperat, com el bit que el referencia (a manera de classificador).

El sistema s'utilitza per al reconeixement de caràcters manuscrits i aconsegueix uns resultats propers al 95%. Per aquesta aplicació, la xarxa realitza també els processos gràfics de thinning and thicking associats al preprocés de les imatges.

4.1.3 Tolerància a procés CMOS

El problema de la realització anterior resulta ser la utilització de transistors NMOS i PMOS per a la implementació de connexions excitadores i inhibidores.

Els dos transistors de connexió a alimentació (PMOS) i massa (NMOS), reben les seves característiques de conducció per processos independents (les que depenen del dopatge) i, degut a que el nivell de comparació de la neurona és fixe (es necessita molta superfície per emmagatzemar o introduir valors analògics diferenciats per a cadascuna), la tolerància a procés d'aquestes implementacions és baixa.

Per resoldre aquest problema, la solució és tornar a una estratègia de doble raïl sense transistors PMOS, la qual cosa redueix superfície, atacant a porta per evitar l'acoblament d'impedàncies i utilitzant diferencials analògics per a número de neurones elevat.

Associada a aquesta arquitectura, proposa una regla d'aprenentatge per a xarxes amb pesos discrets en la qual utilitza una determinació dels pesos per columnes mitjançant mètodes d'optimització lineals (simplex). Amb aquesta regla arriva a una capacitat d'emmagatzemament per a memòria autoassociativa del 30% del número de neurones.

Una realització d'aquestes característiques permet a en M. Verleysen [Verle89] realitzar una xarxa amb 14 neurones en 9 mm^2 en CMOS 3 micres.

Les sinapsis i les neurones són elements senzills de realitzar que ocupen poca superfície de silici (figura 2.6).

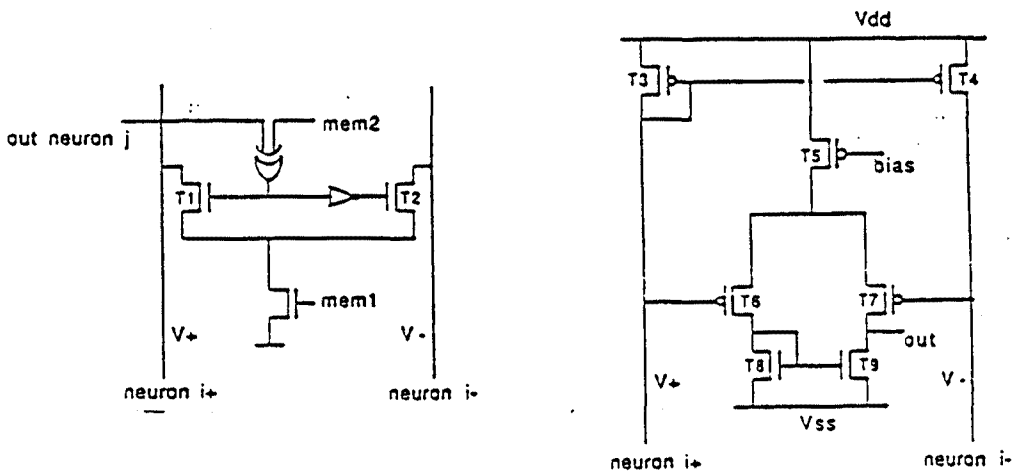


Figura 2.6.- Sinapsi i Neurona CMOS en el model de Verleysen.

4.1.6 Modificació del procés tecnològic

El desig de tenir xarxes neuronals amb millors característiques (major número de neurones, rang més elevat de programabilitat dels pesos, menor superfície de sinapsi,etc.) han portat a modificacions en el procés tecnològic CMOS tendents a realitzar dispositius amb una característica òptima entre funcionalitat i superfície.

Aquestes realitzacions les han pogut fer investigadors associats a centres de fabricació de xips, i per tant han estat reduïdes. Tot seguit en farem una breu referència.

(i) Resistències amb fusibles.

La realització de resistències amb fusibles presenta com a principal avantatge la reducció de la superfície ocupada per les sinapsis al xip, ja que no

es requeriex les cel.les de memòria RAM (2 per sinapsi com a mínim) a l'interior del xip.

Això requereix d'una tecnologia de tipus PROM, com per exemple les que utilitzen silici amorf hidrogenat per a la realització de fusibles que, un cop curtcircuitats, fan de resistències de valor baix (figura 2.7). Això va permetre a un grup del M.I.T. l'any 1987 la realització de 40 neurones totalment interconnectades [Thako87].

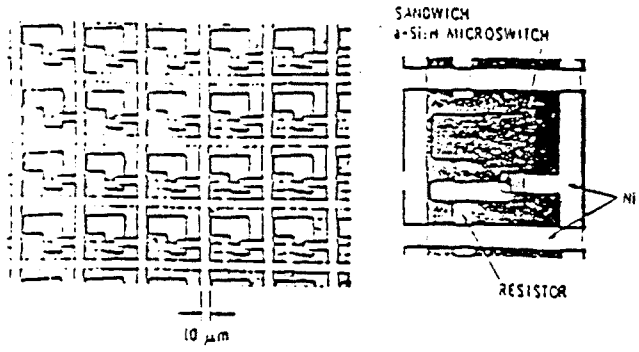


Figura 2.7. Array de sinapsis realitzades amb fusibles, i detall d'una d'elles.

Altres solucions [Jacke87] passen per una programació no elèctrica sinó amb feix d'electrons. La matriu de les sinapsis de la xarxa neuronal, està composada de línies de metall realitzades amb tungsté sobre el qual se situa un layer de poliamida dielèctric. Un cop definits els valors de les sinapsis de la xarxa, aquestes es programen realitzant forats allà on no ha d'anar connexió (figura 2.8).

L'array realitzat contenia 22 x 22 sinapsis, però la complexitat de la programació el fan poc operatiu.

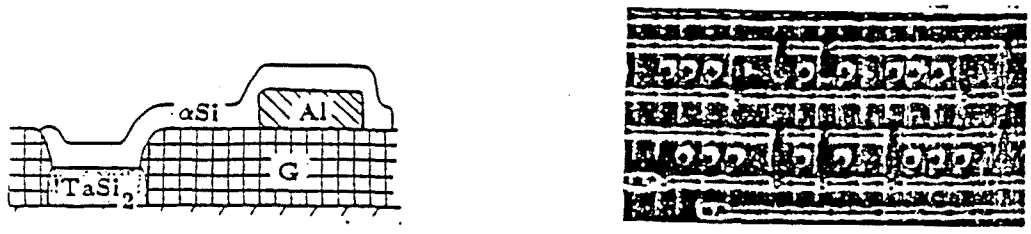


Figura 2.8. Array de sinapsis programades per e-beam, i detall d'una d'elles.

(ii) Dispositius CCD

Una altra proposta d'implementació interessant fou la que van proposar investigadors del MIT [Sage86] i de Dortmund [Rueck87], basada en la suma de càrregues en lloc de en la suma de corrents, tal i com es fa de manera directa utilitzan dissenys amb dispositius de càrrega acoblada (CCD) i tecnologia MNOS de doble porta.

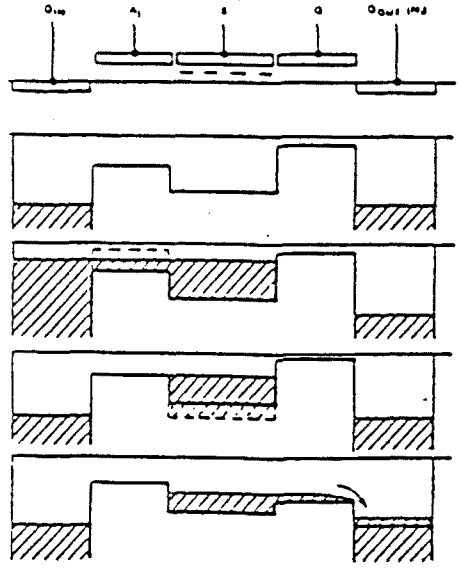


Figura 2.9. Càlcul d'un producte sinàptic en una estructura MNOS CCD.

En aquest cas, les sinapsis es realitzen com a barreres de potencial, programables de manera analògica mitjançant la injecció de càrrega sobre la porta enterrada realitzada amb una capa de nitrur (figura 2.9). D'aquesta manera s'arriva a xips d'alta densitat que també realitzen la suma de productes de manera natural, amb una precisió superior a 4 bits en els pesos.

(iii) Pesos analògics

Una tecnologia de porta flotant pot utilitzar-se també en un procés CMOS per a la implementació de memòries programables no volàtils (figura 2.10).

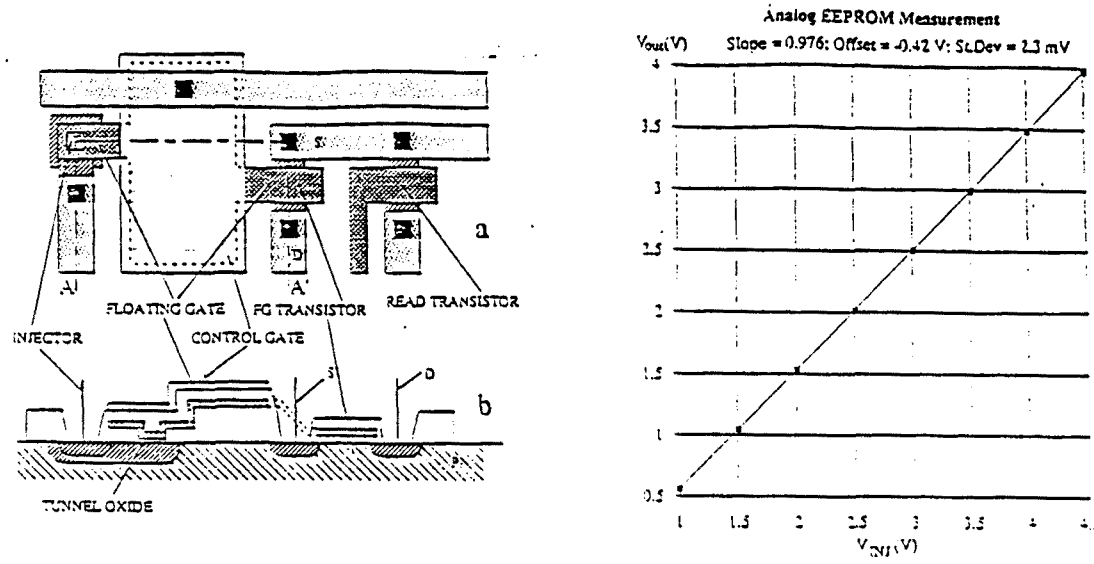


Figura 2.10. Detall d'una sinapsi amb punt de memòria analògic.

Tot i que el mecanisme de programació no és senzill, la linealitat de la tensió electrònica injectada fa possible la utilització d'aquest tipus de dispositius per emmagatzemar pesos analògics [Vitto90].

La superfície utilitzada no és superior a la necessària per implementar els dos bits de memòria RAM més la circuiteria associada a la sinapsi.

Malgrat això, cal un procés específic de doble porta amb capes primes (100 amstrong) per a la programació, per efecte tunel, de la porta enterrada.

4.1.5 Restriccions a nivell de sistema

Aquest conjunt de realitzacions, tot i que han seguit una evolució espectacular, estan arrivant a una saturació.

La causa fonamental d'aquest límit ve donada perqué no és possible realitzar matrius únicament de sinapsis connectables entre elles i amb neurones en un dels extrems.

Això és degut a que la transferència d'un valor analògic de tensió o corrent de l'interior del xip a l'exterior, amb la variació de les condicions de treball que això comporta (nivells de capacitat, corrent, soroll, etc.), no es pot realitzar amb la precisió necessària quan el número de neurones és molt elevat.

4.2. Memòria heteroassociativa

En aquest apartat hi ha també diverses realitzacions basades en el treball de Kosko sobre memòries heteroassociatives [Kosko87], que estableix una xarxa que connecta bidireccionalment dos layers de neurones entre els quals produeix la memòria heteroassociativa, coneguda com a memòria associativa bidireccional (BAM).

D'entre elles mostrem la que hem realitzat nosaltres [Carra89], amb tecnologia CMOS 2 micres dins del programa MPC, que implementa dos layers de 25 neurones cadascun en 3.7x3.7 mm^2 (figura 2.11).

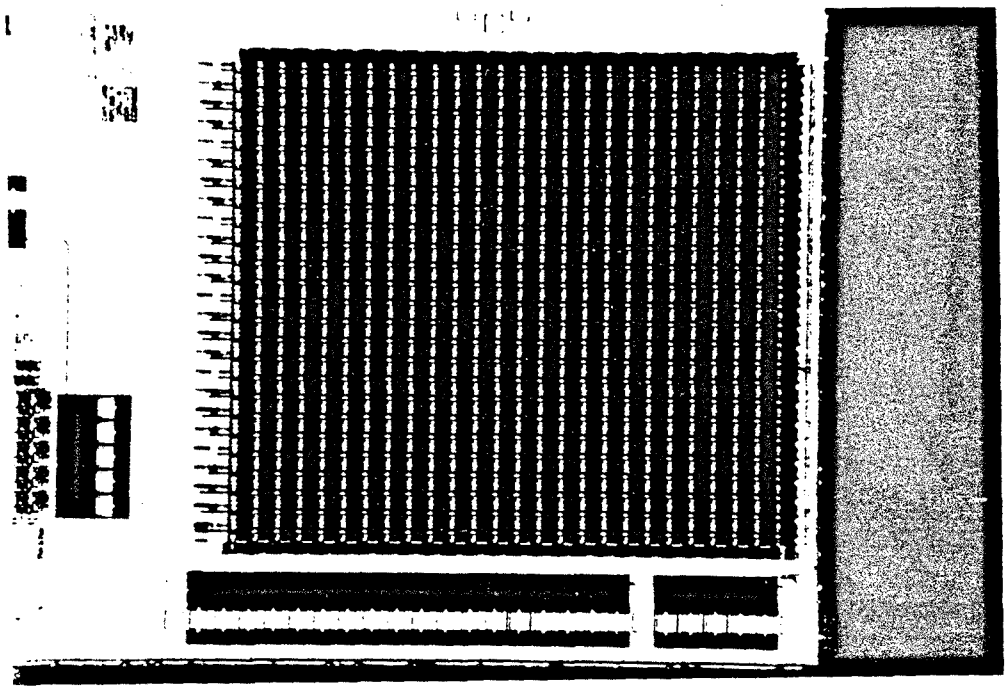


Figura 2.11. Fotografia del xip que implementa una xarxa de tipus BAM.

La cel.la bàsica de la sinapsi (figura 2.12) segueix el model de Graf i el conexionat adicional necessari per aconseguir la bidireccionalitat és mínim, 2 línies de metall i 4 transistors MOS (a part d'un multiplexor a l'entrada de la cel.la), ja que la matriu de pesos és simètrica.

La neurona ha estat realitzada amb un inversor MOS i la introducció de dades en aquesta versió es realitzava mitjançant un registre de desplaçament.

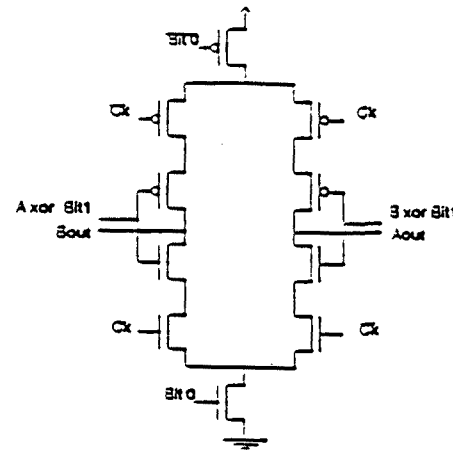


Figura 2.12.- Esquema en transistors de la sinapsi CMOS per a BAM.

En la implementació realitzada, que es mostra a la figura 2.13, les dimensions de la sinapsi són 128*119 μm^2. Aquesta superfície està repartida en parts aproximadament iguals per a l'estrucrutura de la sinapsi, els dos bits de memòria RAM i la lògica necessària, multiplexor i porta or-exclusiva.

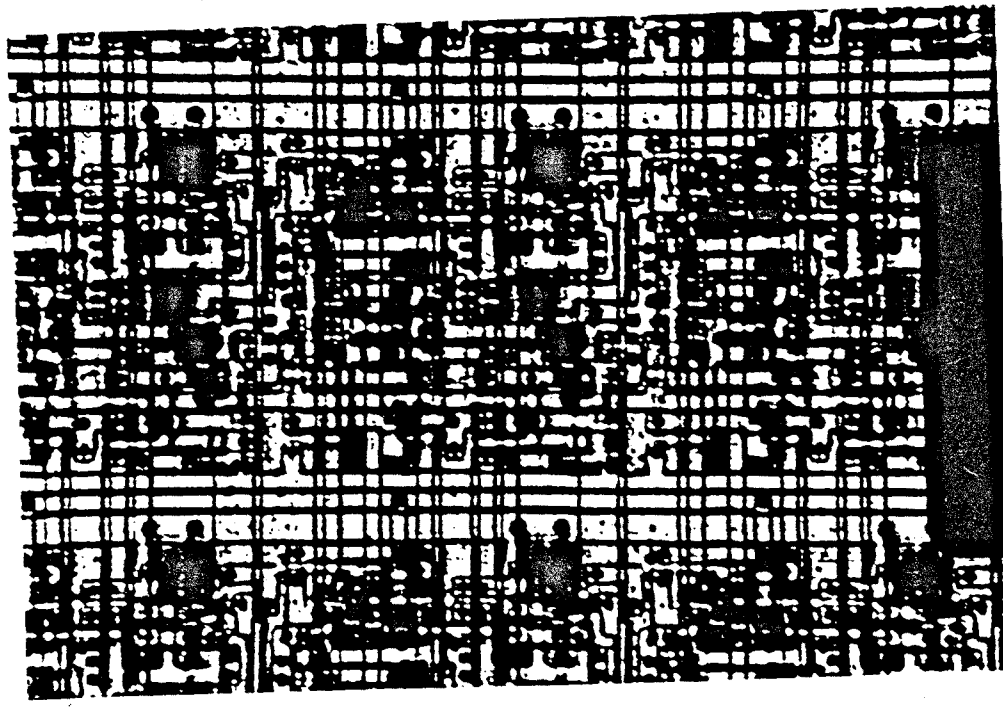


Figura 2.13. Fotografia de la sinapsi CMOS implementada.

Les característiques de la BAM com a memòria heteroassociativa a nivell teòric, són similars a les de la memòria autoassociativa descrites anteriorment per a les mateixes regles d'aprenentatge.

A nivell d'arquitectura de sistema, la figura 2.14 mostra la connexió com recurs de càlcul a un bus per a la funcionalitat de memòria heteroassociativa. El cost en superfície de silici s'incrementa únicament en N neurones i N registres, respecte de les N^2 sinapsis que tenen ambdós.

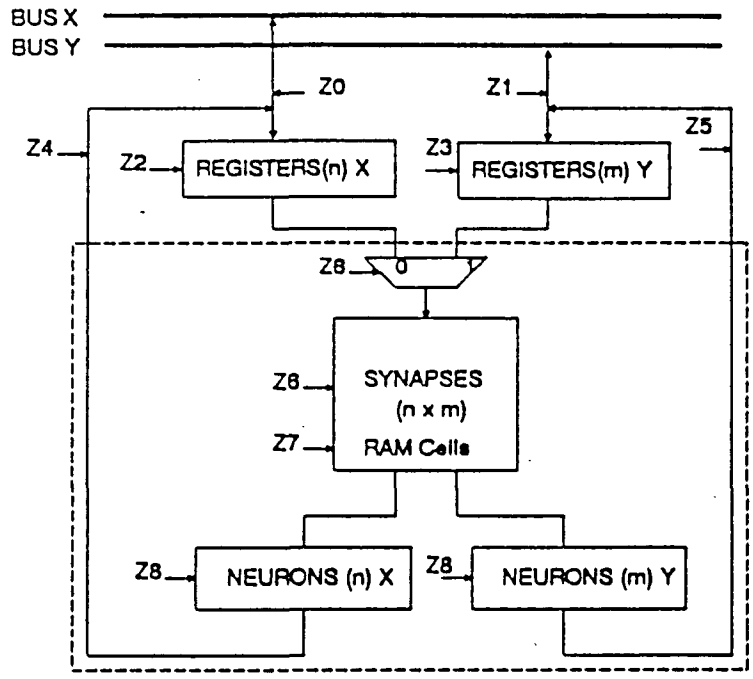


Figura 2.14. Arquitectura per a memòria heteroassociativa.

Aprofitant la simetria de la matriu de pesos, podem utilitzar la BAM com a memòria autoassociativa, incrementant la complexitat temporal, és a dir, realitzant el procés de manera síncrona en el doble d'iteracions que en el cas autoassociatiu (figura 2.15) per recuperar les dues meitats del valor emmagatzemat a partir de les dues inicials. Donada aquesta situació, obtenim un guany en número de neurones però no en número de sinapsis.

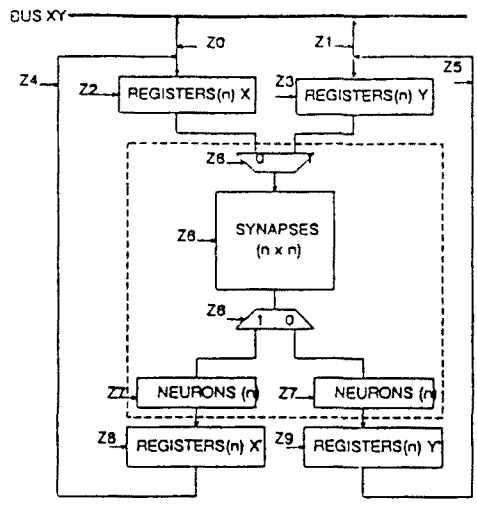


Figura 2.15. Modificació de l'arquitectura per tenir memòria autoassociativa a partir d'una BAM (a) amb la matriu de pesos equivalent (b).

El número de patterns **p** que podem emmagatzemar com a la memòria autoassociativa no augmenta respecte d'un Hopfield equivalent, si considerem que la capacitat disminueix d'una forma aproximadament linial al número de sinapsis que eliminem $\alpha_{BAM} = \alpha_H/2$, (la meitat són zero com mostra la figura 8). Llavors tenim,

$$p = \alpha_{BAM} N_{BAM} = \alpha_H N_H \tag{2.2}$$

En canvi, augmenta el número de patrons emmagatzemats per sinapsi, **p'**, ja que el número de sinapsis físicament implementades es divideix per 4. Si fem una xarxa de Hopfield de les mateixes dimensions, tenim que $S_{BAM}=4S_H$ (la meitat de les sinapsis són zero i de les altres dues matrius, només se n'implementa una ja que són l'una transposta de l'altra) i per tant el rendiment com a memòria autoassociativa respecte de la superfície utilitzada és millor.

$$p = \alpha'_{BAM} S_{BAM} = 2\alpha'_H S_H \tag{2.3}$$

Aquesta estratègia permet la partició en més submatrius si augmentem el número de pasos a realitzar. La figura 2.16 mostra l'arquitectura i la matriu en el cas d'una partició en tres submatrius.

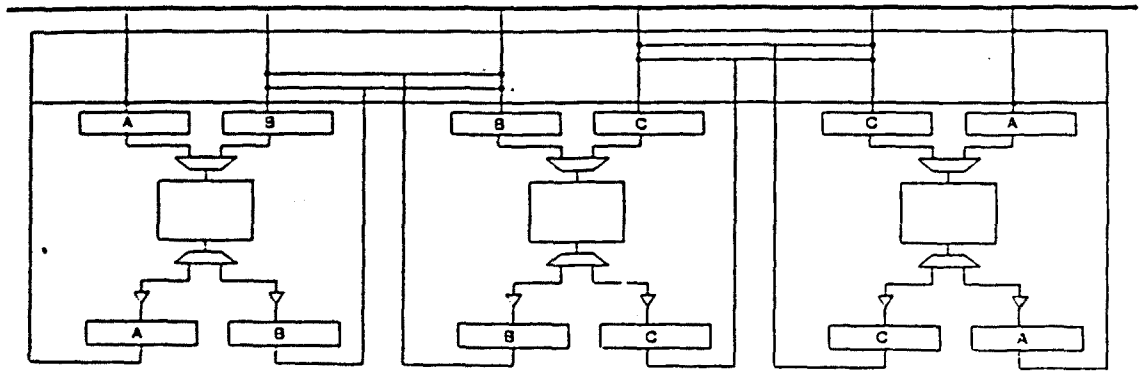


Figura 2.16. Modificació de l'arquitectura per obtenir una memòria autoassociativa a partir de tres xips BAM

A aquestes tècniques d'increment de capacitat, podem afegir tècniques de tolerància a faltes, ja que disposem de tres resultats (A,B,C) per a cadascun dels valors inicials, corresponents a les tres parts diferents del vector presentat a la xarxa, i per tant s'incrementa la qualitat de recuparació del sistema.

Tot i les millores en la capacitat d'emmagatzemament d'aquest sistema, la possibilitat de tenir xarxes operatives amb aquestes quantitats de neurones resulta difícil, a causa de que el procés de divisió no es pot portar al límit de treballar amb cos bits. En aquest cas, les condicions d'operativitat de les regles d'aprenentatge sobre cadascuna de les parts del vector (autocorrelació nul.la) són molt restrictives sobre el rang de vectors als quals podem aplicar la funcionalitat de memòria associativa.

4. Xarxes neuronals de preprocés

Anomenem xarxes neuronals de preprocés el conjunt de xarxes neuronals realitzades en VLSI, la funcionalitat de les quals està enfocada a la preparació de les dades per a un processat posterior més que no pas a una presa de decisions, com seria el cas de xarxes que realitzen classificació, memòria associativa, reconeixement de patrons, etc.

Les funcionalitats típiques d'aquests sistemes estan lligades a processat de senyals provinents de captadors i el seu lligam amb el camp de les xarxes neuronals ve donat per la correspondència amb els processos que realitzen els òrgans sensors biològics (vista, oïda, etc.).

Dins d'aquest plantejament, es poden utilitzar tant xarxes analògiques com digitals, encara que és en el camp dels analògics on s'ha introduït un nombre important de nous conceptes tant a nivell de model com a nivell de porta o de xip.

Una part important d'aquest treball ha estat realitzat, recollit i presentat per en Carver Mead en dos llibres sobre xarxes neuronals analògiques [Mead89a,b]. Aquí a Europa, cal esmentar els diversos treballs realitzats a l'EPFL pel grup de recerca d'Eric Vittoz [Vitto89a,b].

4.1 Transistor MOS en lògica subllindar

L'aportació més revolucionaria en el camp del disseny de xarxes neuronals analògiques, ha estat la utilització del transistor MOS en la regió de funcionament subllindar, fins al moment utilitzada bàsicament per al disseny de circuits integrats de baix consum.

La característica principal d'aquest mode de funcionament és la relació exponencial que lliga les tensions associades a l'estructura MOS (de porta, drenador i font) i el corrent drenador font en la zona d'inversió feble del MOS.

$$I_{DS} = I_0 e^{-\frac{q\kappa V_G}{kT}} \left(e^{\frac{qV_S}{kT}} - e^{\frac{qV_D}{kT}}\right) \qquad (2.4)$$

Aquesta característica és molt important també des del punt de vista dels òrgans sensors humans, ja que es conegut que molts d'ells presenten una sensibilitat logarítmica respecte, per exemple, dels senyals auditius i visuals.

Altres característiques a destacar del treball amb estructures que utilitzen aquest model són:

(i) La restricció de la zona de feble inversió.

Tot i que el comportament logarítmic s'observa en un rang de tres a cinc décades de corrent, els corrents manipulats són molt petits (nanoampers) per la qual cosa cal tenir especial cura respecte d'aïllaments, etc. El rang de tensions és també reduït (per sota de 1 volt).

(ii) L'elevada dependència dels paràmetres respecte del procés tecnològic.

Aquesta tolerància fa que s'hagi de dissenyar amb estructures tolerants a aquestes variacions o que incloguin mecanismes de compensació o sintonia.

(iii) La baixa estandarització dels models utilitzats de simulació elèctrica.

La majoria dels fabricants no faciliten els paràmetres associats a aquest modelat, mentre que els models MOS utilitzats (típicament els tres nivells de SPICE) no caracteritzen aquesta zona de funcionament de manera exponencial.

Malgrat aquests inconvenients, ha estat dissenyada una gran varietat d'estructures de nivell funcional superior, algunes de les quals parteixen de l'adaptació d'elements utilitzats en el disseny d'analògics, mentre que altres estan directament relacionades amb les xarxes neuronals:

. Portes

Miralls de corrent. Amplificadors de transconductància. Amplificadors de rang ample. Seguidors de tensió. Funció valor absolut. Multiplicador de Gilbert. Exponencial i logaritme. Arrel quadrada. Expansió i compresió de funcions. Funció d'activació linial. Integradors i diferenciadors. Estructures diferencials de segon ordre. Axons. etc.

. Estructures

Estructures d'agregació de senyals. Xarxes RG. Detecció de valor màxim. Cadenes de retards. etc.

Amb aquestes estructures de base s'han realitzat chips amb funcionalitats específiques en full-custom, i per tant, amb grau de repetitivitat màxim.

A tall d'exemple n'estudiarem tres realitzacions: la còclea i la retina realitzats per en Carver Mead, com a exemples de modelat de sistemes neuronals naturals amb estructures analògiques, a nivell de portes la primera, i amb propietats distribuïdes la segona, i un conversor A/D de 4 bits proposat per en J. Hopfield i realitzat per nosaltres, com a exemple de funcionalitats associades a sistemes electrònics no naturals.

4.2 Còclea Electrònica

La còclea és un sistema neurofisiològic compost d'un recipient, dins el qual hi ha un líquid incompresible elèctricament conductor, tencat per una membrana per a l'entrada de senyals acústics i amb una sortida de senyals elèctrics sobre cel.lules nervioses. Pot considerar-se com un dels dispositius més elegants dels que relacionen el sistema nerviós intern amb el món exterior.

La figura 2.17 (a) mostra la còclea i les seves parts principals: la membrana basilar (que separa la còclea en dues parts) i les finestres oval (membrana mòbil) i rodona (membrana fixa).

Sota la membrana hi ha l'òrgan de Corti, figura 2.17 (b), que és l'encarregat de la transducció i que està compost de cel.lules cil.liades. N'hi ha de dos tipus, internes (≈3500) i externes (≈10000) les quals realitzen funcionalitats diferents, les primeres detecten la velocitat del fluid, i les segones produeixen un control del guany en funció de canvis en el nivell de só.

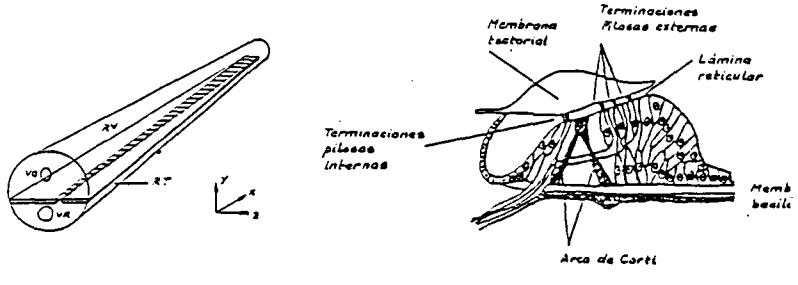


Figura 2.17. Esquema de la coclea (a) i l'òrgan de Corti (b).

La propagació d'ones sonores es transmet, des de l'exterior als ossos de l'oïda mitja, i d'aquests fins a la finestra oval de la coclea. L'impacte sobre aquesta finstra produeix unes diferències de pressió entre les dues cares de la membrana i es genera una la propagació de l'ona sonora sobre la seva superfície.

La funció de transductor la realitza la còclea amb un canvi d'escala logarítmic de freqüències. Aquest procés es realitza directament mercés a les característiques físiques de la membrana basilar, l'elàsticitat i l'amplada de la qual es modifiquen longitudinalment.

La física, en particular la teoria de circuits elèctrics, sap modelar el sistema mecànic (en aquest cas de mecànica de fluids) previ a la funció de transducció mitjançant sistemes elèctrics de segon ordre. Els paràmetres associats als filtres elèctrics es poden obtenir directament dels paràmetres de la còclea.

Els sistemes elèctrics al seu torn, són implementables en VLSI de manera relativament senzilla. Donades la densitat i mobilitat del fluid, elasticitat de la membrana, etc. es posible extreure els paràmetres del sistema per a un model simplificat i obtenir la resposta desitjada.

En el cas de la implementació d'en Carver Mead els paràmetres associats a cada filtre s'assignen segons mostra la gràfica dels pols del sistema al **pla s** de la figura 2.18, en la qual es pot observar l'increment exponencial de la distància en freqüència característica entre dos pols consecutius del sistema quan ens allunyem de l'origen de coordenades.

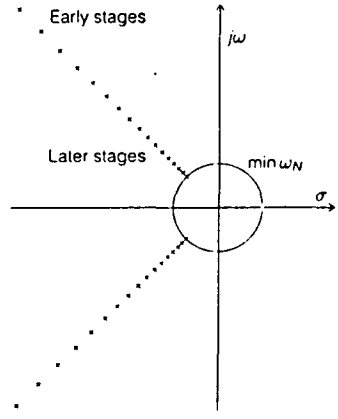


Figura 2.18. Pols al pla s del sistema de filtres que implementa la còclea.

L'efecte d'aquests filtres és ressonar a freqüències lleugerament diferents de manera que per entrades sinusoidals, la longitud a la qual es produeix la amplitud d'oscil.lació màxima indica el valor d'aquesta freqüència. Quan introduïm ones més complexes, aplicant el principi de superposició, obtenim una anàlisi espectral del senyal d'entrada.

En el cas de sons, la principal característica dels quals és temporal més que no pas freqüencial (consonants sordes), el model implementat no ofereix uns resultats tan clars.

El sistema implementat es correspon al pla de base donat per la figura 2.19, i està composada bàsicament per un número elevat de filtres de segon ordre. Les dues línies de τ i Q, que determinen els paràmetres del filtre, són controlables externament mitjançant dos valors situats als extrems d'una xarxa resistiva. Típicament per a τ posarem dos valors diferents que donaran les freqüències màxima i mínima del sistema, ja que són valors de tensió introduïts mitjançant una línia resistiva de polisicili.

La freqüència de sintonia de cada filtre està relacionada amb l'exponencial de la tensió de porta del transistor, el qual realitza la funcionalitat de font de corrent de l'estructura analògica corresponent. Aquesta tensió depén, al seu torn, de la longitud (resistència) a la qual es troba respecte de l'inici de la línia de polisilici, degut a que el transistor de polarització treballa en la regió subllindar. Com a resultat, un gradient lineal en tensió es converteix en un gradient exponencial en les freqüències dels filtres.

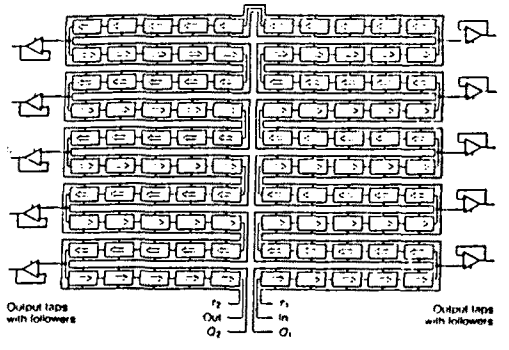


Figura 2.19. Pla de base del xip que implementa la còclea.

Els valors de Q en principi són constants per a tots els mòduls, introduïr al circuit valors diferents equival a treballar amb filtres adaptatius, la qual cosa la realitza el cervell per controlar el nivell d'atenuació necessari per a diferents entorns d'escolta auditiva.

Respecte d'aquestes implementacions, en les quals la possibilitat de variar els paràmetres és reduïda, les implementacions digitals de filtres permeten un grau més elevat de programabilitat, encara que a canvi de ocupar una superficie molt superior.

Aquest és el cas del xip realitzat pel nostre grup de treball [Avell91], que implementa un filtre de tercer ordre en una superfície de silici de 30 μm^2 en CMOS 1.5 micres.

4.2 Retina electrònica

La retina de Mahowald-Mead [Mahow89] modela el sistema nerviós de preprocés que hi ha a la retina amb les cinc cel.les nervioses associades (figura 2.20) que tenen com a funcions bàsiques: representació d'informació de senyals amb variació suau (fotoreceptors, cel.les bipolars i horitzontals), extracció d'events de moviment per propagació de senyals amb ràpida variació temporal (cel.les amacrines) i digitalització (ganglis retinals). Aquesta estructura es repeteix per a cadascuna de les neurones que formen la xarxa bidimensional de la retina.

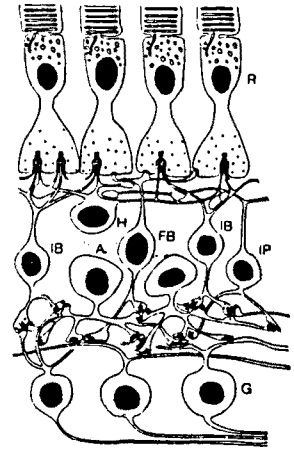


Figura 2.20. Esquema de l'organització de les cel.lules de la retina.

El modelat d'aquestes propietats es fa amb els dispositius següents:

1. Fotoreceptor amb relació logarítmica entre intensitats de llum i corrent.

2. Xarxa resistiva que treu la mitja espacial i temporal de la sortida del fotoreceptor, en una funcionalitat que podriem considerar típica de les xarxes neuronals que prenen en consideració la localitat dels fenòmens.

3. Sortida de cel.les bipolars proporcional a la diferència entre els senyals provinents del fotoreceptor i els provinents de la xarxa resistiva, com a part de la funció de preprocés.

Això configura un xip amb una topologia de connectivitat bidimensional, amb un pla de base com el de la figura 2.21.

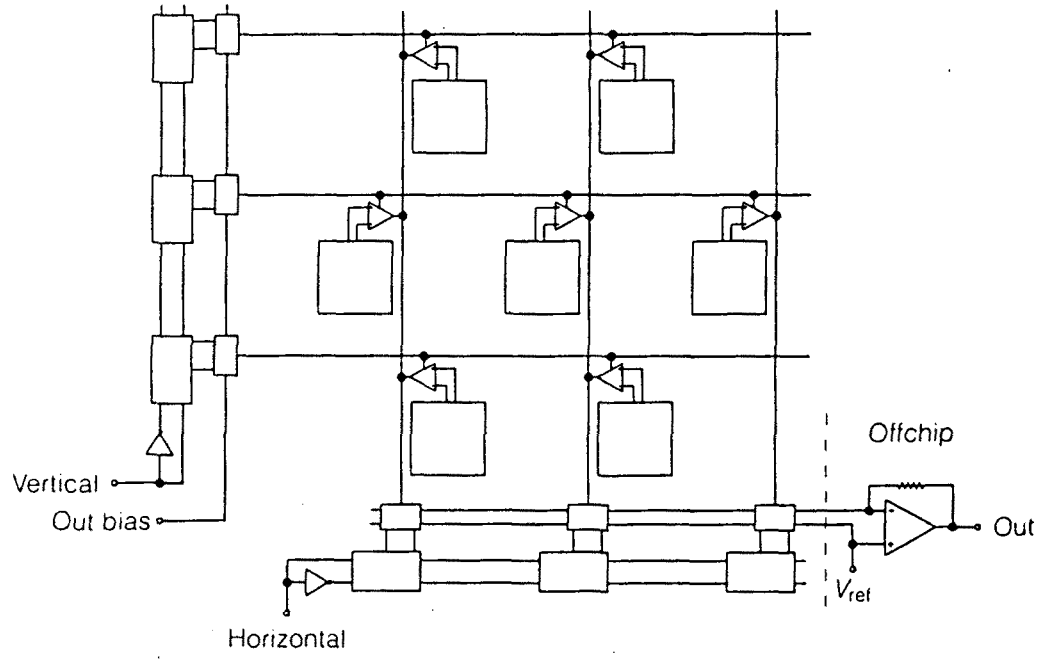


Figura 2.21. Pla de base del xip que implementa la retina artificial.

Com en tot bon disseny full-custom, en la retina predomina la repetitivitat de les cel.les dissenyades, de les quals només n'hi ha un conjunt reduit:

- Fotoreceptor

El treball en subllindar permet treure de manera directa el logarisme de la intensitat d'un bipolar en forma de pad.

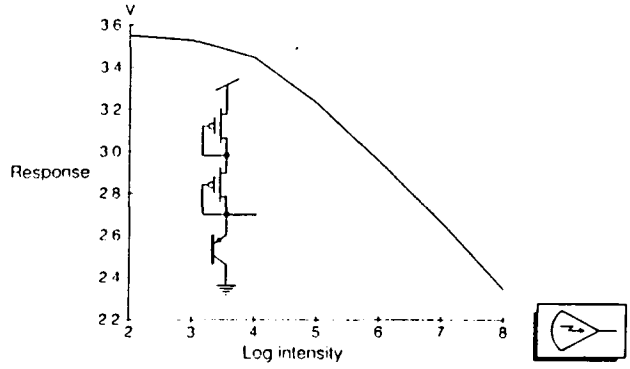


Figura 2.22. Cel.la del fotorreceptor.

- Pixel

El pixel consta del fotoreceptor, un seguidor que el connecta a la xarxa resistiva i el comparador de sortida. La dimensió del pixel és de 109x97 (mesurats en unitats lambda).

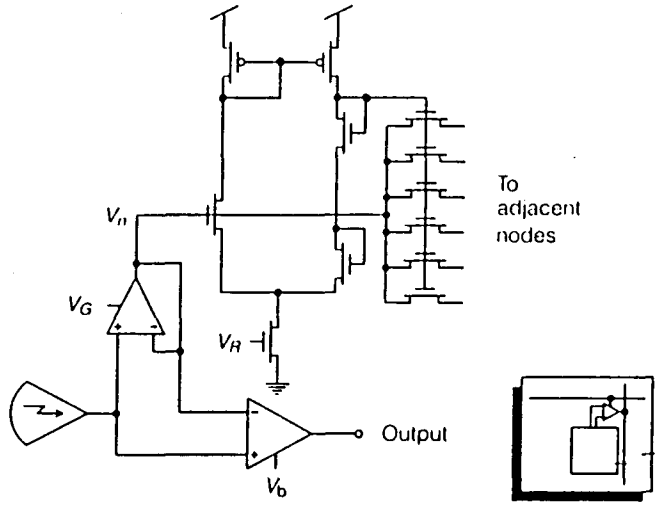


Figura 2.23. Cel.la del pixel.

- Circuiteria de selecció i lectura

La circuiteria de lectura (tipus scanner) està composada de dos registres de desplaçament (horitzontal i vertical), un circuit de polarització del diferencial de sortida de pixel i un de selecció del corrent de sortida produït per aquest.

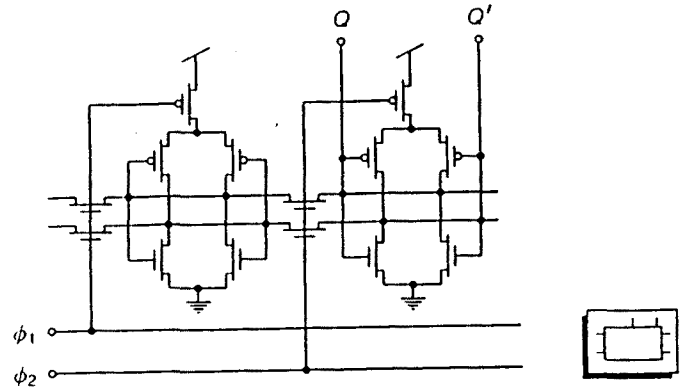


Figura 2.24. Cel.les de la circuiteria de selecció.

Les característiques d'aquests sistema (figura 2.25) venen donades per:

(i) Les corves de sensibilitat, mostren que la resposta del fotoreceptor és independent de la il.luminació de fons (Figura 2.25.a).

(ii) la resposta temporal ve determinada per la magnitud dels corrents de polarització de l'amplificador de rang ample i del circuit resistiu (Figura 2.25.b).

(iii) la constant d'espai, que depèn de la diferència entre els corrents de polarització de l'amplificador de rang ample i del circuit resistiu (Figura 2.25.c).

(iv) la resposta als flancs (canvis d'il.luminació espaial i temporal), que mostren clarament el pretractament de les cel.les nervioses a la retina. Flancs pronunciats produeixen sortides majors que flancs suaus (Figura 2.25.d).

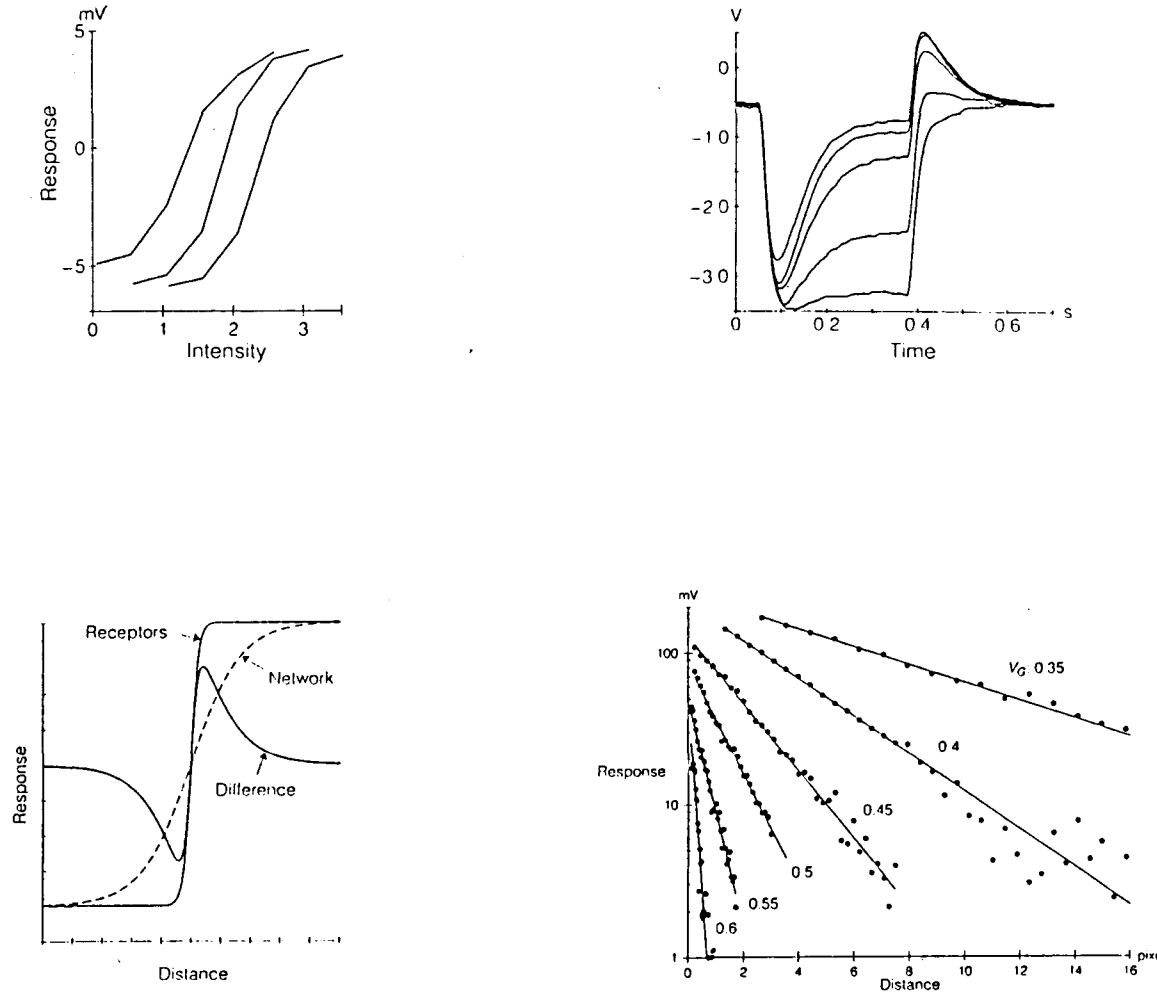


Figura 2.25. Corves associades a les característiques de la retina.

Com a resultat, aquest xip treu els contorns de les imatges captades. Aquests contorns es poden portar a altres estructures de procés per a la realització de funcionalitats específiques, per a les quals, les xarxes neuronals també tenen característiques òptimes, però normalment requereixen de la possibilitat de programació (equivalent al procés d'aprenentatge). Entre aquestes funcionalitats podriem destacar, reconeixement de textures, identificació d'elements (lletres, cares,...), enregistrament digital, etc.

5.1.2 Conversor analògico-digital

En J.J. Hopfield és, sens dubte, un dels pares de la segona generació de xarxes neurals mercés als seus treballs de principis de la dècada dels 80. D'entre les moltes propostes que presenta, en fa alguna a nivell d'aplicació: proposa de fer un conversor A/D de 4 bits basat en una xarxa neuronal [Hopfi86].

La idea és molt senzilla: obtenir els valors de les sinapsis associades a la xarxa per igualació de dues expressions. D'una banda, l'equació de la funció energia de Liapunov associada a una xarxa neuronal totalment interconnectada, simètrica i de diagonal nul.la, com la de la figura 2.1, i de l'altra, la funció que realitza el conversor expressada com a funció a minimitzar més la restricció de valors digitals per als pesos.

$$\sum_{i=0}^{n-1} I_i V_i + \sum_{\substack{j,k=0 \\ j\neq k}}^{n-1} T_{jk} V_j V_k = \frac{1}{2}(X - \sum_{i=0}^{n-1} 2^i V_i)^2 + \sum_{i=0}^{n-1} 2^{2i-1} V_i (1 - V_i) \qquad (2.5)$$

$$T_{ij} = -2^{(i+j)-1} \qquad I_i = 2^{2i-1} + 2^i X \qquad (2.6)$$

A nivell de circuit, una part de les sinapsis estan lligades a tensions (V_i, X) i per tant podem implementar-les com a resistències ($R_{ij}=1/T_{ij}$ i R_{xi}), mentre que altres són independents i poden ser realitzades com a polaritzacions, tensió llindar, etc. Les resistives podem expressar-les de forma matricial com,

$$R_{ij} = -1 \begin{pmatrix} 0 & 1 & \frac{1}{2} & \frac{1}{4} \\ 1 & 0 & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{16} \\ \frac{1}{4} & \frac{1}{8} & \frac{1}{16} & 0 \end{pmatrix} \qquad R_{xi} = -1 \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{1}{4} \\ \frac{1}{8} \end{pmatrix} \qquad (2.7)$$

El mateix Hopfield realitza les simulacions amb amplificadors i resistències, però apareixen mínims locals que no permeten un funcionament correcte. Estudis posteriors mostren que algunes de les realimentacions porten problemes d'histèresi que provoquen la caiguda en aquests mínims locals.

Per a la realització del nostre conversor, hem plantejat les condicions d'operativitat següents:

(i) El rang dels valors de tensió va de V_{ss} = 0 v. a V_{dd} = 5 v, per a l'entrada externa X i els estats de les neurones V_i.

(ii) Les neurones tenen una funció d'activació 'hard-limiter' amb una tensió llindar $V_{th} = V_{dd}/2$ = 2.5 v., per la qual cosa treballem amb variables transformades.

$$V' = V - V_{th} \tag{2.8}$$

(iii) Volem obtenir uns valors de resistència positius, de manera que intentem implementar el signe negatiu amb un inversor (aproximat per un model lineal), obtenint així unes noves magnituds transformades.

$$V'' = (V_{dd} - V) - V_{th} \tag{2.9}$$

En aquestes condicions, reformulem la part de la funció a optimitzar com,

$$\frac{1}{2}[(X-V_{th}) - \frac{1}{2^N-1}\sum_{i=0}^{N-1} 2^i((V_{dd}-V_i)-V_{th})]^2 + \frac{1}{(2^N-1)^2}\sum_{i=0}^{N-1} 2^{2i-1}((V_{dd}-V_{th})-V_i)(V_i-(V_{ss}-V_{th})) \tag{2.10}$$

La resolució de l'equació resultant d'igualar l'expressió anterior amb la funció energia, dóna com a resultats,

$$T_{ij} = \frac{2^{(i+j)-1}}{(2^N-1)^2} \qquad I_i = \frac{2^i}{2^N-1} X - \frac{2^{i+1}}{2^N-1}\frac{5}{2} \tag{2.11}$$

En aquestes expressions podem observar que tots els valors que podem implementar amb resistències són magnituds positives. L'única magnitud negativa és la tensió llindar, que a més depén de la neurona seleccionada.

Aquesta dependència pot evitar-se si tenim en compte la independència del subsistema format per una neurona i les sinapsis associades. Aquesta independència fa referència al fet que, tot i que hi hagi connexions resistives entre els elements de la xarxa, a nivell teòric, se suposa que aquesta influència passa sempre a través de la neurona. En electrònica, equival a dir que la impedància de sortida de la neurona és zero. Llavors, podem aïllar cada neurona i les seves sinapsis, tal i com es mostra a la figura 2.26, i realitzar un procés de normalització sobre cadascuna.

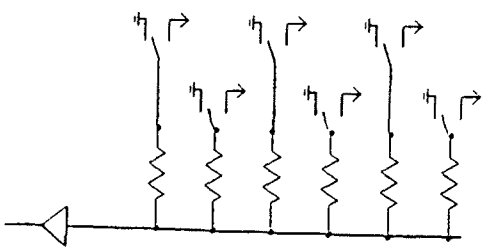


Figura 2.26. Neurona i les seves sinapsis associades.

La normalització escollida ens porta la tensió llindar a 2.5 volt per a cada neurona. Per tant, podem utilitzar el mateix dispositiu de comparació per a totes.

Amb aquesta normalització, obtenim uns valors per a la implementació de les resistències del circuit electrònic.

$$R_{ij} = \frac{2^N - 1}{2^{j-2}} \qquad R_{xi} = 2 \qquad V_{thi} = \frac{5}{2} \tag{2.12}$$

$$R_{ij} = 15 \begin{pmatrix} 0 & 4 & 4 & 4 \\ 2 & 0 & 2 & 2 \\ 1 & 1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} \qquad R_{xi} = \begin{pmatrix} 2 \\ 2 \\ 2 \\ 2 \end{pmatrix} \tag{2.13}$$

El canvi dels valors dels paràmetres resol els problemes d'implementació de la xarxa, però no permet obtenir la funcionalitat del conversor. La figura 2.27 mostra els **problemes d'evolució dinàmica de la xarxa: un cop assolit un mínim local, al sistema li costa evolucionar cap al mínim que dóna la solució correcta.**

Aquest problema d'evolució dinàmica, no apareix en un conversor formulat clàssicament. Per tant, intentem establir un paral.lelisme entre ambdós conversors.

Per al conversor clàssic, les equacions es poden formular com,

$$\begin{aligned} V_3 &= \theta(X-2.5) \\ V_2 &= \theta(X-2.5V_3-1.25) \\ V_1 &= \theta(X-2.5V_3-1.25V_2-0.625) \\ V_0 &= \theta(X-2.5V_3-1.25V_2-0.625V_1-0.3125) \end{aligned} \tag{2.14}$$

Si el reformulem utilitzant la mateixa tècnica que en el cas anterior, amb uns rangs de tensió de 0 a 5 volt, i una inversió en els valors, tenim

$$\begin{aligned} V_3 &= 5\theta(X-2.5) \\ V_2 &= 5\theta((X-2.5)+\frac{1}{2}(2.5-V_3)) \\ V_1 &= 5\theta((X-2.5)+\frac{1}{2}(2.5-V_3)+\frac{1}{4}(2.5-V_2)) \\ V_0 &= 5\theta((X-2.5)+\frac{1}{2}(2.5-V_3)+\frac{1}{4}(2.5-V_2)+\frac{1}{8}(2.5-V_1)) \end{aligned} \tag{2.15}$$

La realització d'un conversor analògico-digital d'aquest estil, amb resistències i inversors, dóna unes gràfiques de funcionalitat correctes encara que la baixa amplificació donada per la característica de transferència dels inversors (0.2 volt) fa que en molts cassos no obtinguem valors digitals a la sortida de les neurones, com en el cas de la figura 2.27 en el qual introduïm a la xarxa 2.54 volt.

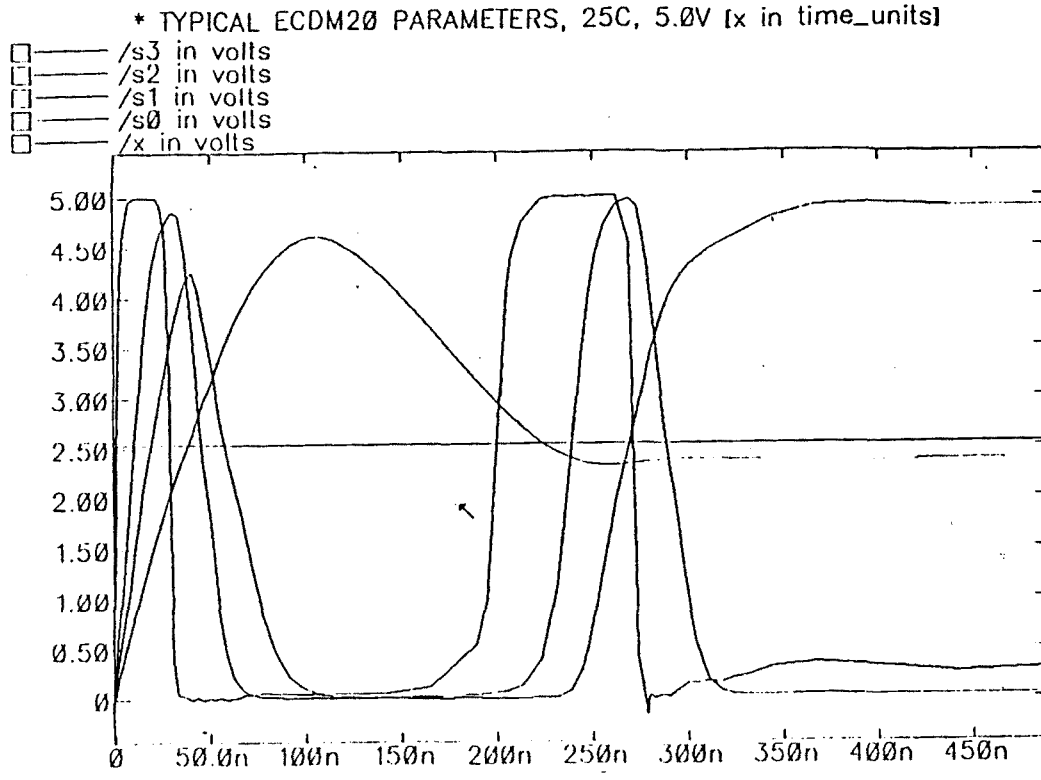


Figura 2.27. Simulació del conversor clàssic fet amb resistències per a 2.54 volt d'entrada.

Els valors de resistències sinàptiques que obtenim per al conversor clàssic es poden expressar utilitzant una expressió matricial com:

$$R_{ij} = 5 \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & \frac{1}{8} \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad (2.16)$$

En la comparació de les matrius corresponents als dos conversors s'observen dues diferències:

(i) Un factor d'escala (1/32).

(ii) La submatriu diagonal inferior és nul.la en el cas clàssic.

El primer punt es refereix a un escalat que podem introduir a la formulació de la funció d'activació a minimitzar. El segon, en canvi, és qualitativament més important ja que té una influència directa sobre la realimentació a la xarxa neuronal.

En el conversor clàssic, el càlcul dels bits del valor digital corresponent a l'analògic d'entrada, segueix una estratègia ordenada del bit més significatiu (el BMS s'obté primer) al bit menys significatiu (el bms s'obté el darrer).

En el conversor neuronal, l'ordre d'obtenció dels bits depén de l'estat inicial de la xarxa. El fet que es tracti d'una estructura amb realimentació entre dues neurones similar a l'element de memòria digital (composada de dos inversors capiculats), fa que l'estat final depengui de l'ordre dels canvis d'estat i que aparegin mínims locals.

Davant d'aquesta problemàtica, és necessari portar al conversor neuronal a un estat inicial més proper a l'estat final desitjat. Aquest procés es pot realitzar mitjançant una estratègia de dues fases:

1ª Fase. Tallar les connexions dels bms als BMS.

2ª Fase. Restablir-les i relaxar la xarxa.

Amb aquesta estratègia, després de la primera fase s'arriva al estat donat pel conversor en la realització clàssica, mentre que amb la relaxació s'assoleix l'estat digital amb la precisió correcta donada pels elements de memòria.

El resultat es pot veure sobre la figura 2.28, en la qual es parteix del mateix valor analògic d'entrada que en el cas de la figura 2.27. Es produeix una evolució dinàmica del sistema i quan aquest s'estabilitza, s'introdueixen les connexions de realimentació que ens donen un valor digital amb una precisió elevada (en la figura de 0.01 volt).

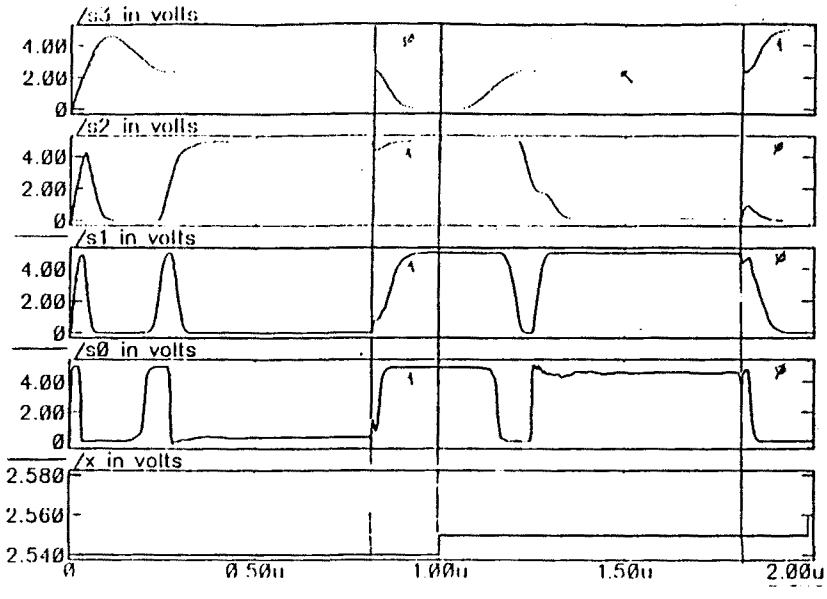


Figura 2.28. Conversor analògico digital neuronal amb dues fases per a dos valors de tensió diferents (V = 2.54 volt, V = 2.55 volt).

El conversor va ser implementat amb una estratègia de disseny 'full-custom' segons l'esquema anteriorment proposat, en dues estratègies que usaven dos dispositius diferents com a resistències:

(i) **línies de polisilici.**

(ii) **transistors MOS en zona lineal.**

Les resistències de polisilici ocupen una superfície més elevada i tenen uns valors de tolerància relativa menors que els transistors MOS, que tenien un valor de resistència no lineal.

En les fotografies de la figura 2.29 s'observen els elements resistius utilitzats per a ambdues implementacions.

En superfície de silici, layout de la figura 2.30, la implementació MOS ocupa 0.16 mm^2, mentre que la de resistències en polisilici ocupa 0.6 mm^2 sense tenir en compte els PADs que comparteixen. El xip fou dissenyat en tecnologia CMOS 2u, amb procés de ES2 dins del projecte MPC.

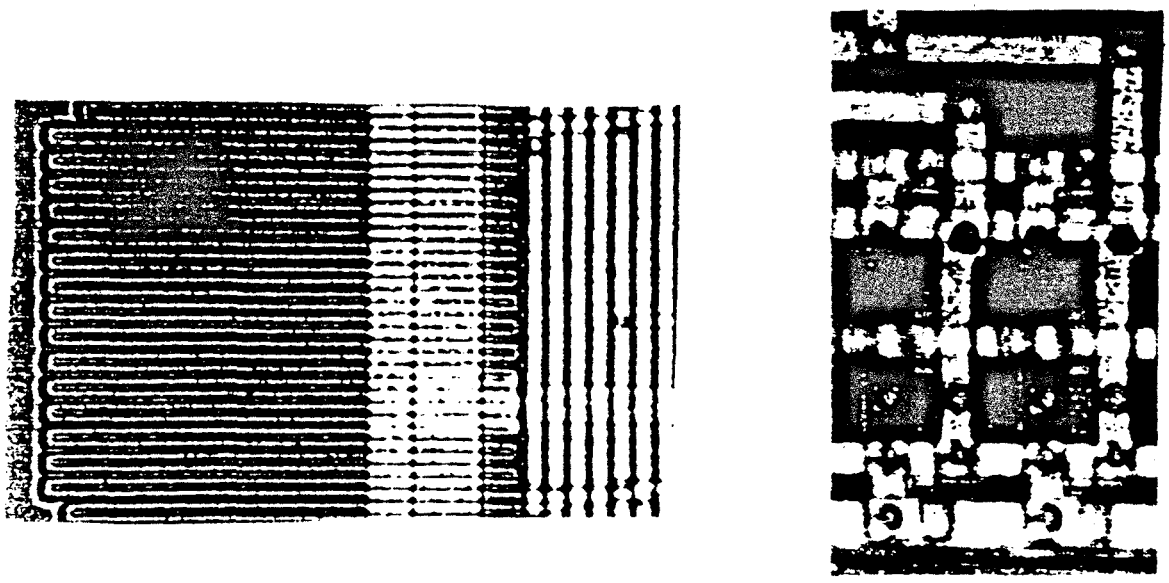


Figura 2.29. Resistències de polisicili (a) i resistències MOS (b).

El conversor basat en resistències ha estat testejat i funciona a una velocitat de 5 MHz, mentre que la implementació MOS no funciona. Els errors que mostra són de tipus unidireccional, la qual cosa ens fa pensar que la variació de procés tecnològic ha produït una modificació, tant en el valor de tensió llindar de les neurones com en els valors de resistència relatius dels transistors NMOS i PMOS.

A les següents figures es mostren les fotografies, realitzades sobre l'oscil.loscop, corresponents a un senyal d'entrada de tipus rampa, i els bits de sortida del conversor, en tres cassos: en el primer s'utilitzen les dues fases (figura 2.31) i n'obtenim la funcionalitat correcta, en el segon tenim només la fase de relaxació (figura 2.32.a), és a dir, sense tallar la realimentació cas, observant-se l'evolució incorrecta de les sortides, i el tercer cas, es correspon a la formulació clàssica amb inversors com a comparadors, sense realimentació (figura 2.29.b). En aquest cas, el funcionament és molt similar a de la realització neuronal, però les transicions d'estat són més suaus. Les sortides digitals corresponen a valors negats respected dels esperades, com a consecuència de la realització amb inversors, de forma coherent amb la formulació de Hopfield.

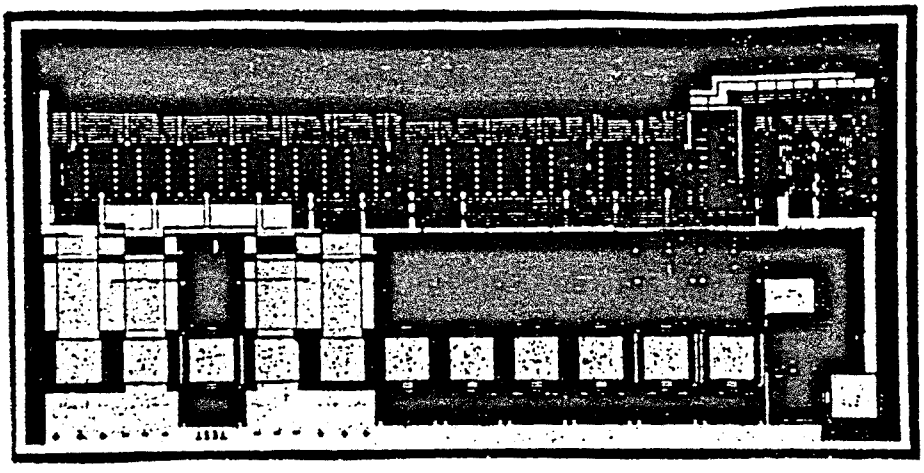


Figura 2.30. Fotografia del conversor neuronal analògico digital de 4 bits.

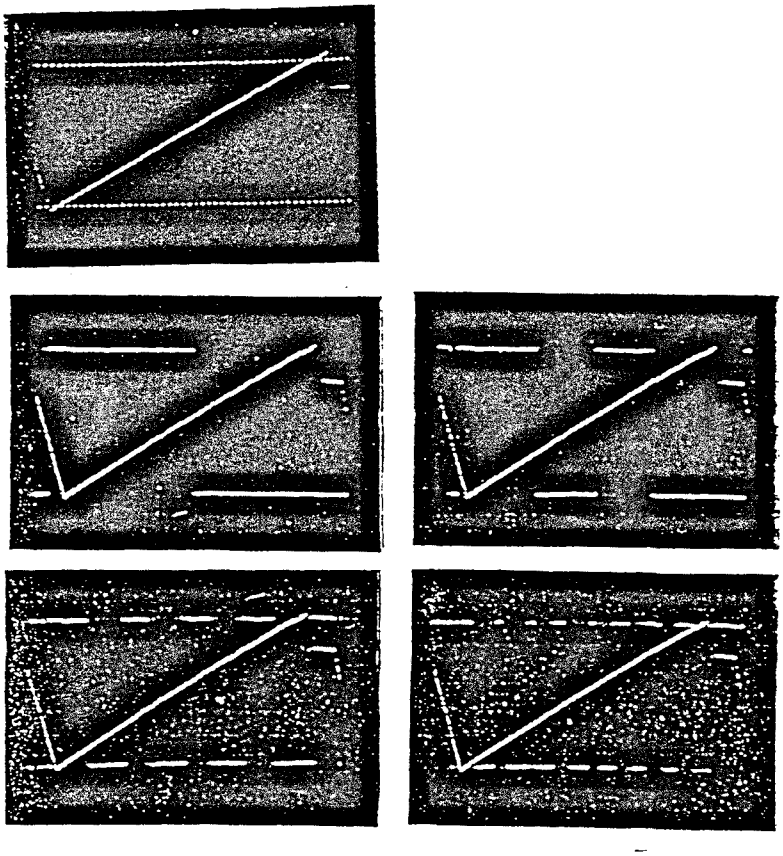


Figura 2.31. Fotografies corresponents a l'entrada rampa analògica i les quatre sortides digitals del conversor A/D.

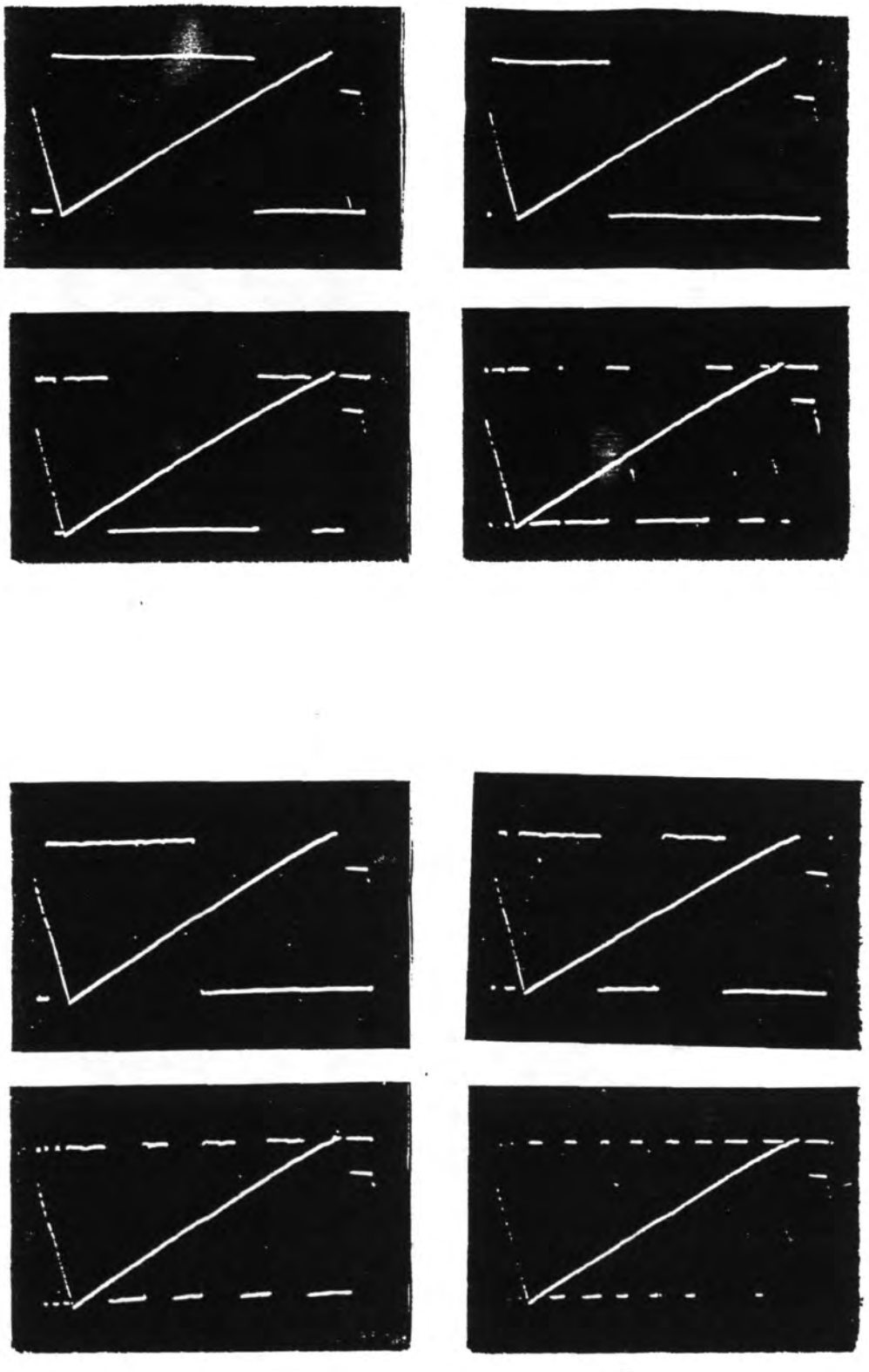


Figura 2.32. Fotografies corresponents a l'entrada rampa analògica i les quatre sortides digitals del conversor A/D en el cassos de funcionament parcials: (a) només amb fase de realimentació i (b) només sense fase de realimentació.

Capítol 3

DINÀMICA EFICIENT PER A LA RELAXACIÓ DE XARXES NEURONALS

L'estudi de les propietats d'una xarxa neural es centra habitualment en l'anàlisi de l'estàtica del sistema. En general es considera que els paràmetres associats a la dinàmica del sistema donen informació sobre el procés d'evolució de la xarxa, però no poden modificar el seu comportament global, el qual vindrà especificat per altres elements tals com la regla d'aprenentatge o la topologia.

Tot i això, l'elecció de la dinàmica és un punt important en el disseny de xarxes neuronals programables. En funció de qui utilitzi la xarxa neural, es poden definir dinàmiques de característiques diferents (seqüencial o paral.lela, analítica o random, síncrona o asíncrona,...), les quals restringeixen el nivell de descripció de les neurones generant models computacionals (sigmoidal, lineal o graó), elèctrics o fisiològics.

En aquest capítol, demostrem que aquestes característiques infueixen en les propietats qualitatives associades a les xarxes neuronals (capacitat de recuperació, velocitat de procés, etc.).

3.1. Representació de la dinàmica d'una xarxa neuronal

Les característiques dinàmiques d'una xarxa neural venen donades pel conjunt d'equacions lligat al sistema. Aquest sistema depén al seu torn de la descripció que es fa dels elements de la xarxa que són bàsicament dos: neurones i sinapsis.

Les sinapsis, què són els elements que es descriuen de manera més simple, donen informació sobre la interacció entre neurones. Per avaluar l'efecte de la neurona **j** (presinaptica) sobre la neurona **i** (postsinaptica) es fa el producte del valor de la sinapsis que les connecta (el seu pes T_{ij}), pel valor de l'estat de sortida de la neurona presinaptica, **Oj**. Els processos d'aprenentatge es cuiden bàsicament de determinar els valors de les sinapsis.

Les neurones en canvi tenen diferents **models de representació.** Els més complexes de tots són els models a **nivell fisiològic.** En aquests models el nombre

de paràmetres i efectes associats és realment elevat: Potencials químics, funcions d'activació, temps de propagació, periode refractari, efecte d'integració, característiques dels pulsos generats,...

La manipulació de models d'aquest tipus en una xarxa neural complexa és computacionalment costós de manera que usualment s'utilitzen models simplificats.

El model més senzill i més utilitzat en computació és el que conté una **funció d'activació, f**, que relaciona el **camp local, h_j**, amb l'**estat** de la neurona, O_j, que ens els estudis següents prendrem igual a la **sortida** d'aquesta.

Existeixen tres funcions d'activació d'ús freqüent: **hard-limiter, lineal** o **sigmoidal.** Als seus models desapareix qualsevol relació temporal explícita, i el què fisiològicament és una dinàmica continua en el temps, passa a ser una **dinàmica discreta.** El principal avantatge d'aquest mètode és que la discretització permet el càlcul de la dinàmica via productes vector-màtriu, fàcils de realitzar en ordinadors.

Les equacions que governen la dinàmica de la xarxa són, habitualment:

$$O_i(n) = f(h_i(n)) \tag{3.1}$$

$$h_i(n+1) = \sum_{j=1}^{N} T_{ij} O_j(n) \tag{3.2}$$

El **model elèctric**, en canvi, es correspon a una **dinàmica contínua** i considera la dependència temporal de la manera més simple possible: mitjançant una equació diferencial de primer ordre. La funció d'activació pot ser qualsevol de les citades, i per tant, la xarxa ve definida per un conjunt d'equacions diferencials del tipus,

$$O_i(t) = f(h_i(t)) \tag{3.3}$$

$$C_i \frac{dh_i(t)}{dt} = -\frac{1}{R_i} h_i(t) + \sum_{j=1}^{N} T_{ij} O_j(t) \tag{3.4}$$

Aquesta expressió és un cas particular del formulisme proposat per Cohen-Grossberg [Cohe83], en el qual C_i i R_i són valors associats a paràmetres elèctrics del sistema. En un principi aquest model és util per a la realització de màquines basades en estructures analògiques per a les quals és fàcil construir, amb dispositius electrónics, tant el producte vector-matriu, com la integració i la funció d'activació.

En termes de dinàmica, aquesta definició de neurona permet una evolució temporal contínua [Hopfi84], similar a la utilitzada pels sistemes biològics, tot i que no està basada en la lògica de cadenes de pulsos com aquests.

El sistema d'equacions resultant imposa de manera determinista el seqüenciament dels estats de sortida.

3.2. Relació entre dinàmica i energia

Una xarxa neural amb realimentació, sota unes certes restriccions té associada una **funció energia**. L'objectiu de les **regles d'aprenentatge** es aconseguir que els mínims d'aquesta funció energía coincideixin amb els vectors que nosaltres volem emmagatzemar (en el cas de que volguem una funcionalitat de memòria associativa).

Aquesta funció energia és la mateixa si estem treballant amb una dinàmica discreta (seqüèncial o paral.lela), però per tal d'assolir els mínims de la funció energia cal fer un tipus de dinàmica tal que, per a cada iteració només permeti el canvi en la sortida d'una neurona. Aquesta dinàmica s'anomena **dinàmica seqüencial o asíncrona.**

Quan es permeten canvis múltiples, **dinàmica paral.lela o síncrona,** en xarxes amb realimentació poden aparèixer **cicles límit.** En aquest cas, els punts fixos de la dinàmica estan composats de diversos estats, la qual cosa provoca una situació poc aconsellable per a la majoria d'aplicacions. La primera xarxa neuronal estudiada és una xarxa amb **dinàmica síncrona.**

En una dinàmica seqüencial, l'elecció de la neurona que canvia el seu estat de sortida, dins del conjunt de neurones que poden fer-ho, pot realitzar-se de diferents formes.

La segona dinàmica que estudiem, **dinàmica seqüencial amb criteri aleatori**, correspon al sistema clàssic d'elecció a l'atzar de la neurona l'estat de la qual canviem, del conjunt de posibles neurones que poden produir transició d'estat. La tercera dinàmica, **dinàmica seqüencial amb criteri analític**, l'obtenim resolent analíticament, per trams, el sistema d'equacions diferencials temporals (obtingudes a partir del model "elèctric" de neurona) que descriuen una xarxa neural monolayer amb realimentació. Per aquest sistema trobem també el seqüenciament dels canvis d'estat del sistema. Per a la quarta dinàmica, **dinàmica seqüencial amb criteri probabilistic**, l'elecció de la neurona que canvia d'estat, l'obtenim de mesures d'inestabilitat sobre l'estat de cada neurona.

Les simulacions realitzades mostren una diferència qualitativa en el comportament de les quatre dinàmiques. La dinàmica seqüencial amb criteri probabilístic es comporta millor pel que fa a la qualitat de recuperació, convergeix més ràpidament, i és implementable algorismicament de manera més òptima.

Els estudis sobre les dinàmiques tenen en comú la utilització d'una funció d'activació per a les neurones de tipus "hard-limiter", de sortida bipolar {+1,-1}.

3.3. Dinàmica paral.lela

En una dinàmica paral.lela, a partir dels estats inicials es calculen els camps locals corresponents a totes les neurones segons les expressions (3.1,3.2). Aquelles neurones per a les quals el signe del camp local és diferent del de l'estat canviaran simultàniament el seu estat a la següent iteració. Amb aquesta nova configuració es recalculen els camps locals.

El procés iteratiu de relaxació de la xarxa continua fins assolir un estat estable, que pot ser un únic estat, o bé un seqüència d'estats. En aquest cas el sistema ha entrat en un cicle límit.

3.4. Dinàmica seqüencial amb criteri aleatori

Les neurones de la xarxa neural, governada per les equacions (3.1,3.2), no contenen cap dependència temporal explicita, de forma que no imposen una

dinàmica determinista per al seqüenciament dels canvis d'estat. La neurona que canvia el seu estat s'elegeix **aleatòriament** d'entre les que tenen el producte del camp local per l'estat de sortida de la neurona negatiu.

Aquesta formulació, anomenada dinàmica de Glauber, parteix del criteri que diu que s'arriva als mínims locals de la funció energia independentment del seqüenciament utilitzat. Donada l'aleatorietat en l'elecció de la neurona l'estat de la qual canvia, no es pot reproduir la trajectoria que va d'un estat inicial a un punt fixe de la dinàmica (mínim local de la funció energia).

3.5. Dinàmica seqüencial amb criteri analític

3.5.1 Funcions d'activació

Les funcions d'activació de les neurones es caracteritzen, en el cas de neurones de tipus elèctric, per ser funcions instantànies, de manera que la resposta temporal a una certa entrada és el valor donat per la funció sense retard.

L'evolució dinàmica d'un sistema com el presentat al capítol anterior (figura 2.1) es deguda a la integració temporal del camp local de cada neurona, i a la influència mútua donada per la connectivitat. Aquesta propietat ens permet considerar dos tipus de xarxes segons si la funció d'activació és continua o no.

En el cas de funcions d'activació contínues, p.e. lineals o sigmoidees, existeix un conjunt de variables únic amb el qual es pot descriure l'estat del sistema (equacions 3,4), ja siguin les funcions d'activació, $\mathbf{h}_j$, o els estats de les neurones, $\mathbf{O}_j$, però no són necessaris tots dos simultàniament.

En canvi, si tenim funcions d'activació discontínues (p.e. del tipus hard-limiter) no existeix una funció inversa que recuperi el valor del camp $\mathbf{h}_j$ a partir de l'estat $\mathbf{O}_j$, i per tant, no n'hi ha prou amb saber l'estat de la neurona per predir l'evolució temporal, sinò que es necessari conèixer el camp local, donat per l'equació (3.4), per calcular-ne la sortida. Aquesta assimetria ens permet discretitzar la dinàmica de la xarxa.

El valor de sortida de la neurona només canvia quan el valor del camp arriva a un cert valor llindar, però no durant tota la seva variació temporal. Si no canvien els valors de sortida, l'estat del sistema és manté tot i que hi hagi evolució

temporal. Si l'estat del sistema és manté, podem calcular analíticament quina és l'evolució dinàmica del sistema entre dos canvis consecutius, ja que podem "desacoblar" el conjunt d'equacions diferencials, per tal d'obtenir neurones aïllades, com la que es mostra a la figura 3.1.a.

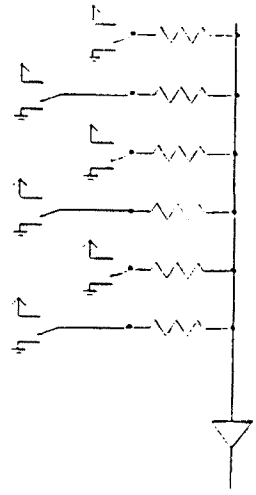


Figura 3.1.- Exemple de neurona desacoblada (a), i el comportament del camp local i l'estat de sortida durant la seva evolució (b).

3.5.2 Solució del sistema d'equacions acoblades

A simple vista la resolució del sistema d'equacions diferencials acoblades que representa una xarxa neural és impracticable quan el número de neurones creix una mica però sota certes condicions el sistema té solució fàcil.

Si reformulem l'expressió (3.4), per a cada neurona tenim:

$$\frac{dh_i(t)}{dt} + \frac{1}{R_i \cdot C_i} h_i(t) = \frac{1}{C_i} V_{fi}(t_0) \qquad (5)$$

$$V_{fi}(t_0) \equiv \sum_{i=0}^{N} T_{ij} O_j(t_0) \qquad (6)$$

En el cas de neurones bipolars {+1,-1} amb funcions d'activació "hard-limiter" podem calcular la contribució de l'estat del sistema sobre cada neurona a partir de l'instant t_0, com:

$$V_{fi}(t_0) = \sum_{j=1}^{N} T_{ij} O_j(t_0) = \sum_{\{J_+\}} |T_{ij}| - \sum_{\{J_-\}} |T_{ij}| \qquad (7)$$

$$j \in \begin{cases} \{J_+\} & si\ sign(T_{ij}) \cdot O_j = +1 \\ \{J_-\} & si\ sign(T_{ij}) \cdot O_j = -1 \end{cases}$$

Aquest terme (equivalent al producte vector-matriu) pot identificar-se com el valor final del camp local, a l'estat estacionari, per a la neurona i.

L'ús d'una funció de transferència 'hard-limiter' fa que només quan el camp local es igual a un valor llindar es pot produir un canvi d'estat en la neurona, de manera que el valor del terme $\mathbf{V_{fi}(t_0)}$ de l'equació (3.6) és constant entre dos instants consecutius pels quals es compleix què el camp $\mathbf{h_i(t)}$ és igual al llindar de la funció per a dues neurones qualsevol i,j (figura 3.1.b).

En el cas de que hi hagi canvis d'estat, aquest terme s'actualitzarà, comportant-se com el 'valor final' associat al instant inicial $\mathbf{t_0}$.

Pel que fa als paràmetres elèctrics, podem considerar que el valor de $\mathbf{C_i}$ és una constant associada a la velocitat de resposta de la neurona, que, en general, és igual per a totes les neurones de la xarxa, i per tant, que es comporta com un factor d'escala que depén de la implementació del dispositiu, i que podem menysprear a efectes de fer una avaluació de l'algorisme independent de l'estratègia d'implementació.

El valor $\mathbf{R_i}$ té en compte totes les sinapsis connectades a la neurona,

$$R_i = \frac{1}{\sum_{j=1}^{N} T_{ij}} \qquad (8)$$

La distribució dels valors de les sinapsis a l'entrada de cada neurona influencien la seva característica dinàmica. Aquesta influència s'estén a nivell de tota la xarxa (per a xarxes amb realimentació), ja que apareixen velocitats de resposta diferents. La determinació dels valors de les sinapsis per a les regles d'aprenentatge, fa que no es tingui en compte la manera en que afectaran la dinàmica de la xarxa.

La solució analítica de la equació diferencial (3.5) és directa,

$$h_i(t) = R_i V_{fi}(t_0) + (h_i(t_0) - R_i V_{fi}(t_0)).\exp(-\frac{t-t_0}{R_i C_i}) \qquad (9)$$

En la fase de relaxació d'una xarxa, partim de l'estat inicial del sistema que no determina, segons la definició de la xarxa donada per les equacions (3.3,3.4), el valor del camp local en l'instant inicial $h_i(t_0)$. Com a resultat, a un mateix estat li correspon una varietat infinita de combinacions dels valors dels camps. Hi ha, per tant, un conjunt infinit de dinàmiques possibles que partint d'un estat inicial poden portar a estats finals diferents.

Per a la nostra dinàmica prenem el valor del camp igual al valor de l'estat,

$$h_i(t_0) = O_i(t_0) \qquad (10)$$

Aquesta elecció és arbitrària i prèvia al procés de relaxació i podem considerar-la una forma d'introduir la inestabilitat a la xarxa.

3.5.3. Solució analítica per trams

El procès de desacoblament de les respostes temporals de cada neurona l'utilitzem per discretitzar la dinàmica, de manera que els canvis d'estat de les neurones ens vinguin donats per l'aplicació de les expressions analítiques sobre les equacions de la dinàmica.

Segons l'espressió (3.8), podem calcular per a cada neurona si es produirà o no transició i en quin temps. La primera transició la produirà la neurona que primer arrivi al valor de canvi d'estat V_{Thi} amb el mínim temps.

El temps que la neurona **i** necessita per arrivar al llindar ve donat per

$$\Delta t_i = R_i \ln \frac{h_i(t_0) - R_i V_{fi}(t_0)}{V_{Thi} - R_i V_{fi}(t_0)} \tag{11}$$

Suposem que la primera neurona que canvia és la neurona **k**, i $\Delta t_k = \min \Delta t_i$. Llavors, la sortida de la neurona canvia segons,

$$O_k(t_0 + \Delta t_k) = -O_k(t_0) \tag{12}$$

A partir d'aquest instant de temps tindrem un altre conjunt d'equacions que definiran la dinàmica del sistema. Ara bé, els paràmetres d'aquest nou sistema es poden obtenir a partir dels anteriors.

El nou valor del camp local inical $\mathbf{h_i(t_1)}$, vindrà donat per l'expressió

$$h_i(t_0 + \Delta t_k) = R_i V_{fi}(t_0) + (h_i(t_0) - R_i V_{fi}(t_0)) \cdot \exp - \frac{\Delta t_k}{R_i} \tag{13}$$

Mentre que per a l'actualització dels valors finals tenim

$$V_{fi}(t_0 + \Delta t_k) = V_{fi}(t_0) - 2 O_k(t_0) T_{ik} \tag{14}$$

El canvi en els valors estacionaris, conseqüència de les propietats d'interacció entre les neurones, és responsable de la seqüència d'estats pels quals passa la xarxa.

Les diferents dinàmiques basades en models continus (p.e. equacions diferencials d'ordre superior), es diferencien per l'ordre dels canvis d'estat de les neurones, que juntament amb les interaccions entre elles poden portar la xarxa a un estat final diferent.

Amb aquest canvis, tornem a tenir totalment definit el sistema d'equacions (5) i per tant, podem trobar el nou valor Δt corresponent a la neurona l'estat de la qual canvia en la següent iteració.

Per aquest mètode aconseguim relaxar la xarxa i podem estudiar analíticament l'evolució en cada instant de temps, calculant només el valor dels camps locals als punts en els quals hi ha un canvi d'estat.

3.6. Dinàmica seqüencial amb criteri probabilista

El model de xarxa basat en un sistema d'equacions diferencials de primer ordre acoblades és extremadament senzill i didàctic, i permet una sèrie d'implementacions elèctriques [Carra90]. La següent qüestió a investigar seria si un model d'aquest estil dóna una dinàmica amb propietats òptimes. De les propietats estacionàries dels models de primer ordre no en podem treure cap conclusió en aquest sentit.

Per tal de generar un model de dinàmica a partir d'una informació més precisa sobre la qualitat del seqüenciament de la xarxa, utilitzem mesures d'estabilitat.

La condició d'estabilitat per a una memòria $\{\xi_i\}$ i per a un conjunt donat de sinapsis $\{T_{ij}\}$ ve donada per la següent expressió [Garne88] que han de complir totes les neurones.

$$\Delta_i^\mu \equiv \xi_i^\mu \cdot \sum_{j-1}^{N} T_{ij} \xi_j^\mu > 0 \qquad \forall i,\mu = 1,...,p \tag{15}$$

Si volem uns dominis d'atracció superiors i per tant una estabilitat millor per al pattern, podem fer-ho amb la restricció

$$\Delta_i^\mu = \xi_i^\mu \cdot \sum_{j-1}^{N} T_{ij} \xi_j^\mu > k \tag{16}$$

Aquesta mesura, que s'utilitza en certes regles d'aprenentatge [Verle89][Rérez90], pot utilitzar-se també per a l'anàlisi de l'estabilitat dels patrons en la fase de relaxació.

En aquest sentit, el producte de l'estat pel camp local de cada neurona $\mathbf{O_i h_i}$, és una mesura de l'estabilitat del bit **i** del vector d'entrada $\{\mathbf{O_j}\}$. Si l'estabilitat és positiva la neurona no canviarà el seu estat, mentre que per a neurones amb estabilitats negatives aquest canvi pot produir-se.

La probabilitat de que es produeixi un canvi en el valor de sortida d'una neurona és proporcional a l'estabilitat donada per l'estat resultant. Una neurona amb una estabilitat alta requereix que els canvis en altres neurones l'afectin totes en el mateix sentit inestabilitzador. Com més gran és l'estabilitat, més improbable

és una seqüència d'aquest tipus i les realimentacions influiran principalment en que no es produeixin algunes de les transicions amb probabilitat baixa.

Aquesta anàlisi ens permet definir un tercer criteri per a la dinàmica seqüencial: **Es realitzen els canvis segons la seva probabilitat. Quan varies neurones poden canviar d'estat, s'escollirà la neurona capaç d'assolir un estat més fortament estable.**

Aquest enunciat reflexa el fet que volem estudiar les característiques de la dinàmica independentment de les regles d'aprenentatge, de les quals n'obtenim els valors de les sinapsis $\{T_{ij}\}$ convenientment normalitzats.

En aquest cas, la mesura d'estabilitat de l'estat de la neurona **i** s'obté per

$$E_i(n) \equiv O_i(n) \cdot \sum_{j-1}^{N} T_{ij} O_j(n) \qquad (17)$$

Com per al criteri analític, després d'un canvi d'estat $\mathbf{O_k(n+1)=-O_k(n)}$, es produeix una actualització del valor de l'estabilitat de totes les neurones segons:

$$E_i(n+1) = E_i(n) - 2O_i(n)O_k(n)T_{ik} \qquad (18)$$

excepte per a la neurona que ha canviat, per a la qual

$$E_k(n+1) = -E_k(n) + 2T_{kk} \qquad (19)$$

Aquest procés continua fins assolir l'estat estacionari quan totes les estabilitats són positives.

3.7. Resultats de simulació

De l'estudi sobre les diferents estrategies n'esperem determinar quina és la millor dinàmica de relaxació de la xarxa, tant pel que fa a les qualitats de recuperació, com per una reducció del número de passos necessaris per arrivar a l'estat estable. L'avaluació temporal haurà de tenir en compte, a més, la complexitat dels càlculs a realitzar pels diferents algorismes.

Aquest punt ens permet de fer una primera diferenciació entre dinàmiques paral.leles i seqüencials.

7.1 Algorisme òptim

L'estratègia típica utilitzada en simulacions i emulacions, consisteix en realitzar càlculs del tipus vector-matriu, amb una complexitat $O(N^2)$, que per a cada iteració realitza un producte vector-matriu. Aquesta complexitat es correspon a la que implementen les dinàmiques paral.leles.

Per a les dinàmiques seqüencials que hem proposat en aquest capítol, és necessari realitzar el producte vector-matriu únicament per a la primera iteració. Posteriorment, per a cada canvi d'estat d'una neurona els càlculs són diferents; per exemple, per al criteri probabilístic, els calculs corresponents a les actualitzacions dels vectors d'estat i de les estabilitats de totes les neurones. D'aquest forma, la complexitat dels calculs per a cada iteració esdevé lineal O(N).

Per als criteris aleatori i analític es realitzen càlculs similars, de la mateixa complexitat O(N). Aquest resultat és important en termes d'implementació en xips ja que, per a números de neurones elevats, l'increment de temps necessari per a realitzar una estratègia paral.lela és excessiu.

7.2. Mesura de característiques de les xarxes

Hem realitzat els estudis quantitatius sobre una xarxa de tipus Hopfield amb funcionalitat de memòria autoassociativa, utilitzant com a regla d'aprenentatge la llei de Hebb. L'elecció ve donada, no tant per les qualitats de recuperació d'aquesta regla, sinò per ser més coneguda d'aprenentatge no supervisat, la qual cosa fa més ràpida l'obtenció de la matriu d'interconnexions per a la relaxació de la xarxa.

Hem executat un conjunt de simulacions per a les quatre dinàmiques esmentades, que presentem en dos grups: el primer correspon a la comparació entre les dinàmiques seqüencials (amb criteris aleatori, analític i probabilístic), que són les que tenen una complexitat dels càlculs menor, i el segon correspon a una comparació entre la dinàmica paral.lela i la dinàmica seqüencial amb criteri probabilistic, en termes de propietats de recuperació i independentment de la complexitat dels càlculs.

Les simulacions han estat realitzades variant el número de neurones de la xarxa (entre 100 i 400) per a diferents valors de la capacitat d'emmagatzemament α (des de 0.02 fins a 0.20). En elles es pot veure el canvi que es produeix al voltant de la capacitat $\alpha=0.15$ (punt de saturació) en les propietats de recuperació del sistema, tal com prediuen els models de mecànica estadística.

Els resultats que presentem són qualitativament similars en a totes les simulacions realitzades per sota del punt de saturació. Han estat obtinguts per a 200 neurones, una capacitat $\alpha=0.10$, per a 100 matrius de sinapsis diferents amb els patrons elegits de manera aleatòria, a partir d'un overlap inicial de 0.70 fins a un overlap de 1 amb un pas de 0.05 i fent per a cadascun d'ells 10 proves diferents.

En la relaxació de la xarxa s'utilitza el mateix estat inicial tant per a les tres dinàmiques seqüencials, com en la comparació entre paral.lela i seqüencial.

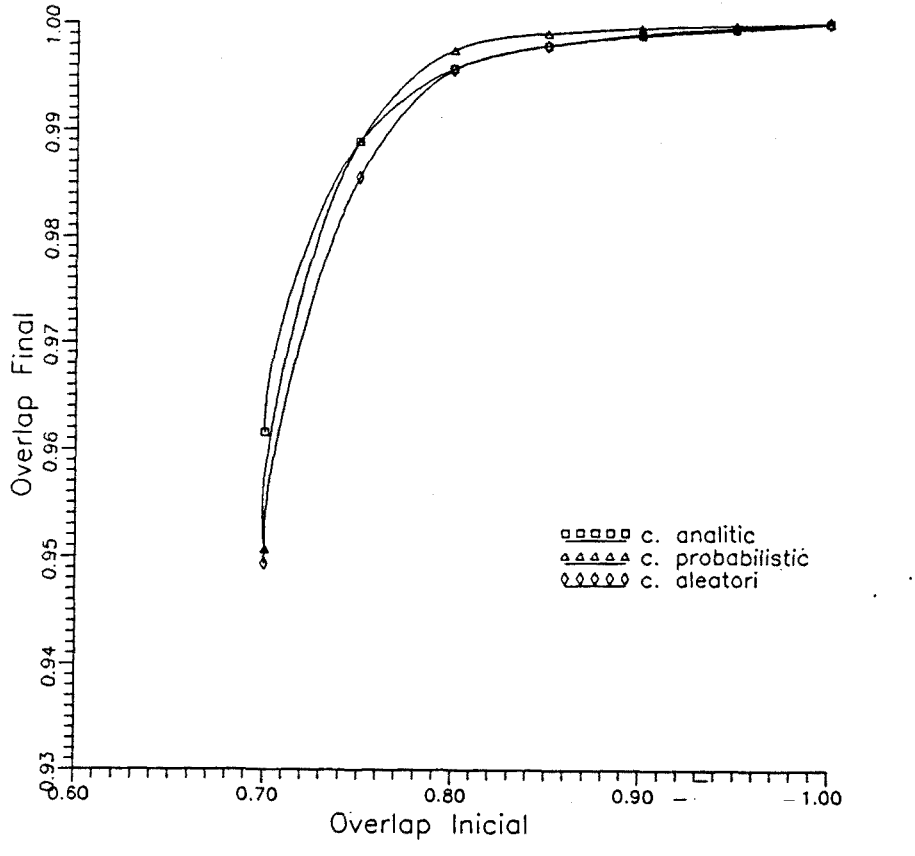


Figura 3.2. Mesures de l'overlap final respecte de l'inicial per a les dinàmiques seqüencials.

La primera gràfica de resultats de simulació (fig.3.2) mostra l'overlap final respecte de l'overlap inicial i és per tant, una mesura de la qualitat de recuperació. En ella es pot veure que la dinàmica seqüencial amb criteri probabilístic, té una recuperació lleugerament millor que l'aleatòria per overlaps inicials per sobre de 0.75, i aquesta és millor que l'analítica.

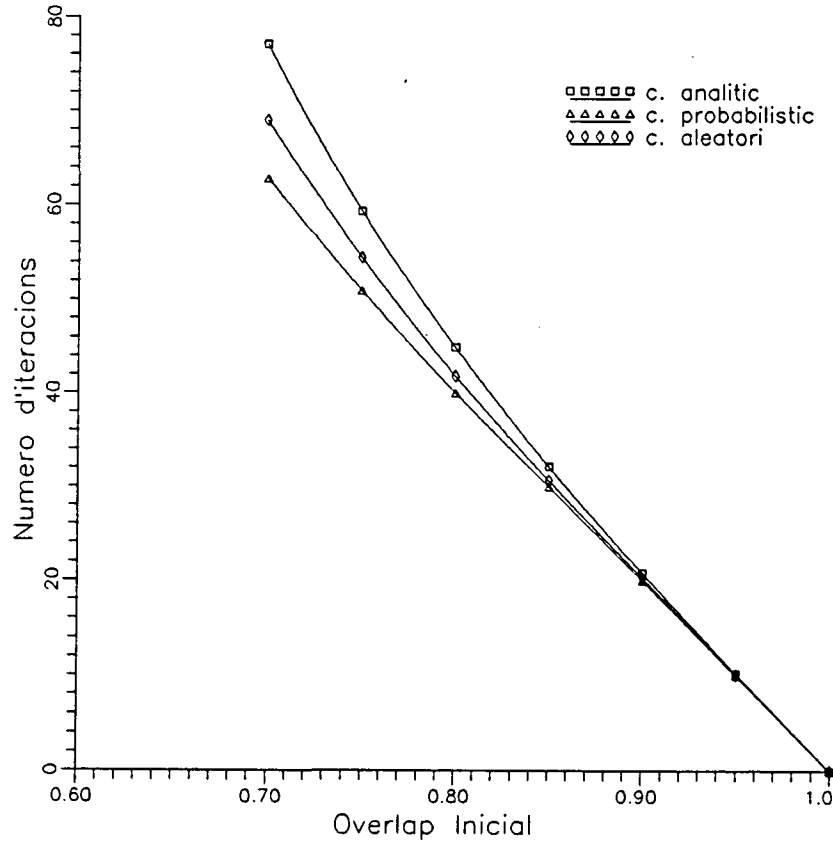


Figura 3.3. Mesures de número d'iteracions respecte de l'overlap inicial per a les tres dinàmiques seqüencials.

El mateix resultat s'obté per al número de passos que triga el sistema en arrivar a l'estat estable, representat a la segona gràfica (fig. 3.3). S'observa per a la dinàmica probabilística, una dependència d'ordre inferior a la corresponent a les altres dues, la convergència de les quals és més lenta per overlaps inicials grans. Aquests valors són especialment significatius quan el número de neurones és elevat, afectant de manera més significativa la velocitat de relaxació de la xarxa.

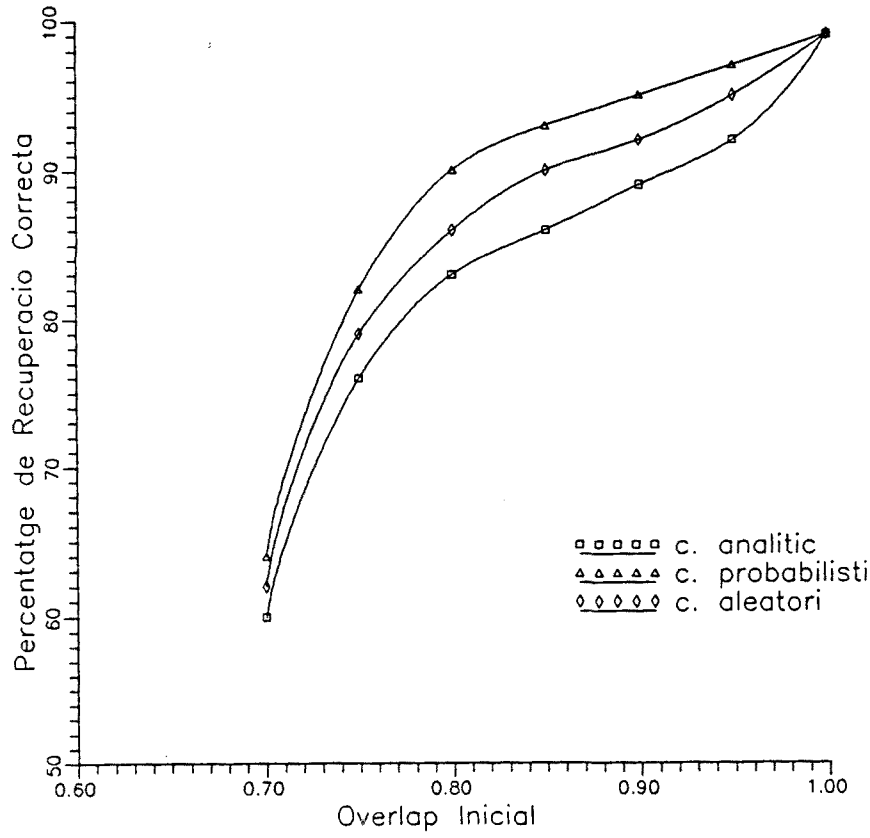
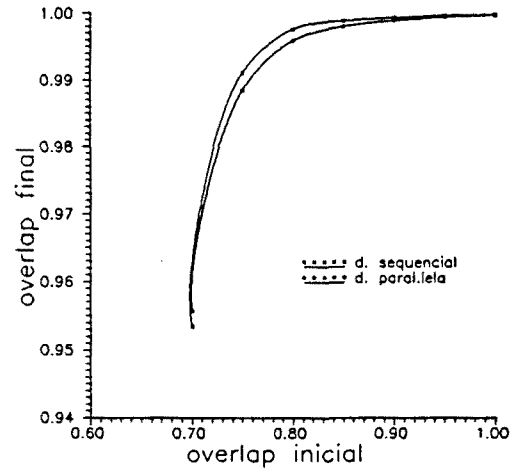
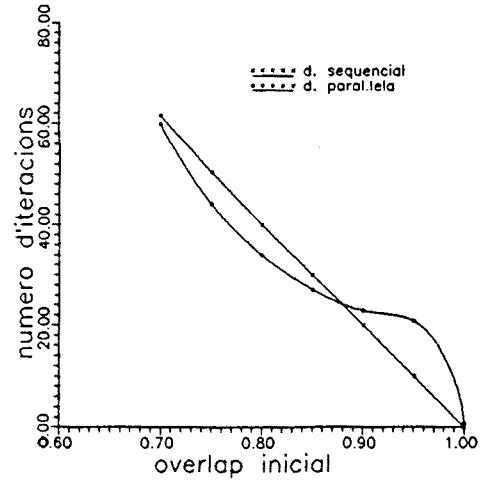


Figura 3.4. Mesures de percentatge de recuperació correcta respecte d'overlap inicial per a les tres dinàmiques seqüencials.

La tercera gràfica (fig. 3.4) representa el número de patrons correctament recuperats. Tot i que aquesta mesura no és usual per a memòria associativa, la prenem com un altre criteri qualitatiu, útil per a la majoria d'aplicacions. En ella es pot veure que la recuperació és molt millor per a la dinàmica amb criteri probabilista.

A la figura 3.5 es mostren els resultats, equivalents als presentats per a les dinàmiques seqüencials (3.2, 3.3 i 3.4), obtinguts en la comparació entre una dinàmica paral.lela i una dinàmica seqüencial amb criteri probabilistic. Les propietats de recuperació són similars per a l'overlap final i el percentatge de patrons correctament recuperats, mentre que en el número d'iteracions necessari per assolir un estat estable, la dinàmica seqüencial mostra una característica amb un grau de linealitat superior a la dinàmica paral.lela. En aquest cas, per a similar número d'iteracions, la influència de la complexitat dels càlculs sobre el temps total de procés, penalitza fortament la dinàmica paral.lela.





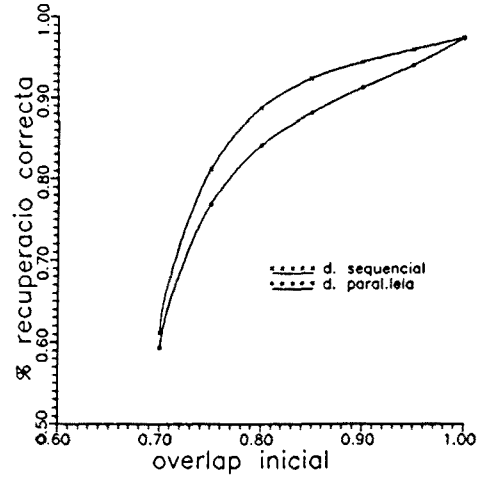


Figura 3.5. Mesures d'overlap final (a), número d'iteracions (b) i percentatge de recuperació correcta (c) respecte d'overlap inicial per a les xarxes, paral.lela i seqüencial amb criteri probabilístic.

Capítol 4

ALGORISMES PER A LA DINÀMICA SEQÜENCIAL AMB CRITERI PROBABILISTA

La implementació en silici de xarxes neuronals programables amb un alt número de neurones i una interconnectivitat genèrica, equivalent a permetre una interconnexió total entre neurones, no sembla factible mitjançant estratègies amb estructures analògiques ja que, d'una banda el paral.lelisme intrínsec xoca amb la dificultat d'expandir els xips a nivell de sistema, i per tant, aquests estan reduits a xarxes amb un número de neurona baix (<256), i de l'altra, els models de dinàmica basats en equacions diferencials de primer ordre no dónen propietats òptimes a les xarxes que els utilitzen com acabem de veure en el capítol anterior.

Malgrat això, l'ús de xarxes programables amb un alt número de neurones és interessant des del punt de vista de la resolució de certs problemes relacionats amb la manipulació massiva de dades de les quals en volem extreure unes determinades característiques.

Des d'aquest punt de vista ,i ja que tenim una dinàmica amb unes característiques qualitatives òptimes (probabilística) respecte de dinàmiques equivalents (síncrona, aleatòria i analítica), sembla important poder disposar de sistemes informàtics amb un gran número de neurones, basats en aquesta dinàmica. Aquesta estratègia passa per la implementació amb elements de procés digitals d'aquest tipus de xarxes.

4.1. Algorísmica i implementació RTL

L'avantatge bàsic que obtenim pel fet de treballar amb elements digitals ve donat per l'automatització del procés de disseny dels circuits integrats a partir d'una descripció algorísmica d'alt nivell. Una descripció d'alt nivell, existeix igualment per al disseny de circuits analògics (tractament algebraic del càlcul diferencial basat en transformades de Fourier, Laplace, z,...), però està mancada de l'automatització del procés de disseny, ja que la manipulació de valors analògics dins del xip està més restringida que la de senyals digitals respecte de factors intrínsecs com sorolls, acoblaments, toleràncies de procés, etc.

El llenguatge d'alt nivell associat al procés digital parteix de la descripció algorísmica del problema. A partir d'aquesta descripció és possible implementar l'algorisme mitjançant diferents estratègies, depenent del grau de complexitat temporal i número de recursos utilitzats [Deschamps], mitjançant les diferents arquitectures posibles.

Les definicions de dinàmica donades en el capítol anterior, permeten una implementació algorísmica directa, com per exemple els associats a les dinàmiques analítica i probabilística que es mostren als quadres I i II.

Quadre I. Algorisme associat a la dinàmica analítica.

Pas 1. Càlcul dels valors finals.

$$V_{fi}(t_0) = \Sigma_j \, T_{ij} \, O_j(t_0) \qquad \forall i$$

Pas 2. Repetir fins a la convergència. (Δt_i infinit)

2.1 Determinació del mínim temps fins a una transició.

$$\Delta t_{nk} = {}_{min}\{ \Delta t_i = R_i \, Ln(1-(V_{fi}(t_0)/h_i(t_0))) \}$$

2.2 Canvi d'estat de la neurona seleccionada.

$$O_k \, (t_n+\Delta t_{nk}) = -O_k(t_n)$$

2.3 Actualització dels camps locals i valors estacionaris.

$$h_i(t_n+\Delta t_{nk}) = R_i V_{fi}(t_n) + (h_i(t_n)-R_i V_{fi}(t_n)) \exp(-\Delta t_k/R_i)$$

$$V_{fi}(t_n+\Delta t_{nk}) = V_{fi}(t_n) - 2 \, O_k(t_n) T_{ik}$$

La principal conclusió que podem deduir d'aquesta implementació de la dinàmica de relaxació, és la complexitat lineal **O(N)** dels algorismHx respecte del número de neurones, en el procés iteratiu de l'actualització del valor de l'estabilitat (estratègia probabilística) o del valor final (estratègia analítica) i el canvi d'una única neurona en cada pas; excepte per a la iteració inicial, corresponent al càlcul del producte vector-matriu, que és d'ordre $\mathbf{O(N^2)}$.

Quadre II. Algorisme associat a la dinàmica probabilística.

Pas 1. Càlcul dels camps locals.

$$h_i(0) = \Sigma_j T_{ij} O_j(0) \qquad \forall i$$

Pas 2. Repetir fins a la convergència. ($E_i > 0$)

2.1 Selecció de la màxima inestabilitat. (mínima negativa)

$$E_k = {}_{min}\{ E_i = O_i h_i, E_i<0 \}$$

2.2 Canvi d'estat de la neurona seleccionada.

$$O_k (n+1) = -O_k(n)$$

2.3 Actualització dels camps locals.

$$h_i(n+1) = h_i(n) - 2O_k(n)T_{ik}$$

Aquest algorisme és adequat per a l'estudi de la complexitat, però no està directament orientat al disseny d'una arquitectura. Per aquest propòsit, cal formular-lo amb una major concreció a nivell de llenguatge de transferència de registres (Quadre III).

Els grafs asHxciats als processos elementals que es poden extreure d'aquest algorisme (figura 4.1.a), mostren una estructura típica associada a màquines simples tipus RISC, que només requereix un cicle de lectura de dades, un de procés i un d'escriptura (aquest només està dins de la iteració en el darrer procés).

Unitilitzant aquesta estructura, es pot augmentar la velocitat de procés utililitzant tècniques de pipeline (figura 4.1.b), mitjançant les quals, diferents cicles corresponents a una instrucció es solapen amb cicles corresponents a les instruccions anterior i posterior. Això permet de reduir el número de cicles per operació fins als valors esmentats a la figura.

Pel que fa als recursos, calen com a recursos elementals un multiplicador i un sumador, a banda de recursos menys costosos del tipus comparadors, registres, etc.

Quadre III. Algorisme RTL per a la dinàmica probabilística.

```
for i:= 0,N-1                                   /* càlcul del producte vector-matriu */
{
        R3:=0;
        for j:= 0,N-1                           /* càlcul del camp local (neurona j) */
                {
                R1 := M(Oj);
                R2 := M(Tij);
                R3 := R1 * R2 + R3;
                }
        M(hi) := R3;
}
while tran = 1;
{
        tran:=0; R3:=0;
        for i:=0,N-1                            /* detecció i elecció de transicions */
        {
                R1 := M(Oi);
                R4 := M(hi);
                if (R1*R4<R3) then
                {
                        R3 := R1 * R4;
                        tran:=1;
                        k:=i;
                }
        }
        R1 := M(Ok);                            /* canvi d'estat de la neurona amb transició */
        R3 := -R1;
        M(Ok) := R3;
        for i:=0,N-1                            /* actualització dels valors dels camps locals */
        {
                R2 := M(Tik);
                R4 := M(hi);
                R3 := R4 + 2 * R2 * R3;
                M(hi) := R3;
        }
}
```

/* Els R_i són registres de dades de la unitat de procés */

/* Els M() continguts de memòria */

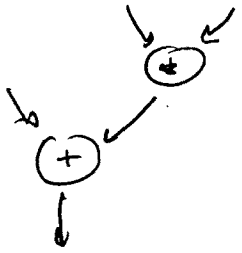
(a) Operació

(b) Pipeline

Càlcul del camp

Instant t

Instant t+1



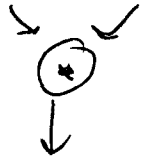
$R_1:=M(O_j),R_2:=M(T_{ij})$

$R_3:=R_3+R_1*R_2$ | $R'_1:=M(O_{j+1}),R'_2:=M(T_{ij+1})$

$R'_3:=R'_3+R'_1*R'_2$

1 cicle/operació

Selecció d'estabilitat



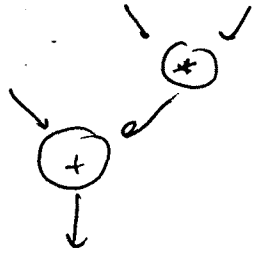
$R_1:=M(O_i),R_4:=M(T_{ij})$

$R_3:=R_1*R_4$ | $R'_1:=M(O_{j+1}),R'_4:=M(T_{ij+1})$

$R'_3:=R'_1*R'_4$

1 cicle/operació

Actualització d'estabilitats



$R_4:=M(h_i)$

$R_3:=M(T_{ik})$

$R_3:=R_4+2R_2*R_3$ | $R'_4:=M(h_{i+1})$

$M(h_i):=R_3$ | $R'_3:=M(T_{i+1k})$

$R'_3:=R'_4+2R'_2*R'_3$

$M(h_{i+1}):=R'_3$

2 cicles/operació

Figura 4.1. (a) Grafs dels processos bàsics corresponents a l'algorisme RTL per a la dinàmica probabilística i (b) pipeline suportat.

El caràcter iteratiu del processat sobre neurones i sinapsis simplifica la unitat de control fins arrivar a un petit autòmat de control de fluxe juntament amb un comptador per a la selecció de les adreces de memòria.

Aquesta estructura és sensible de ser paral.lelitzada augmentant el número d'unitats de procés i mantenint una única unitat de control lleugerament més complexa.

5.2. Arquitectura del sistema: Restriccions en els pesos.

Un altre dels problemes fonamentals a l'hora d'implementar xarxes neuronals totalment interconnectades i programables, ve donat per la quantitat de memòria necessària per emmagatzemar els pesos que és igualment quadràtica respecte del número de neurones.

En el disseny de coprocessadors neuronals o neurocomputadors cal establir, donada una quantitat de memòria fixa al sistema, un compromís entre la precisió dels valors dels pesos i el número de neurones de la xarxa.

Aquest aspecte suposa un inconvenient important per a les realitzacions analògiques programables que tenen la memòria del pesos a l'interior del circuit, perquè obliga a dedicar una part important de la superfície del circuit únicament a memòria. La proporció en superfície augmenta si es volen emmagatzemar pesos analògics o digitals amb més precisió, mentre que en moltes ocasions aquest increment no suposa un guany important en les característiques del sistema, tal i com es presenta al apèndix 1.

Les implementacions digitals utilitzen memoria comercial fabricada amb tecnologies d'alta escala d'integració amb processos no estandard que permeten capacitats elevades. Comercialment es poden adquirir xips de 1 M bit SRAM o 4 Mbit DRAM. Aquesta memoria és externa al processador i per tant els xips a dissenyar poden dedicar-se específicament a la implementació dels algorismes corresponents.

Tot i aquest avantatge, a l'hora de concebre el sistema ens situem ràpidament en condicions d'utilitzar quantitats de memòria elevades (megabits), degut a la dependència quadràtica entre el número de neurones i el de sinapsis.

Adicionalment, la precisió utilitzada per als pesos és un factor multiplicatiu respecte de la quantitat de memòria necessària.

La relació que lliga aquests paràmetres pot veure's a la figura 4.2, la qual mostra una avaluació de la quantitat de neurones implemetables en una xarxa per a una quantitat fixa de memòria disponible en funció de la precisió dels pesos de les sinapsis. Aquesta avaluació està realitzada sobre les quantitats de memòria més usuals en sistemes informàtics de baix nivell (microcomputadors i estacions de treball): PC/XT amb una configuració bàsica de 512 Kbyte, PC/AT amb 1 Mbyte, PS2/70 amb 4 Mbit i finalment una estació de treball SUN4 amb 16 Mbyte.

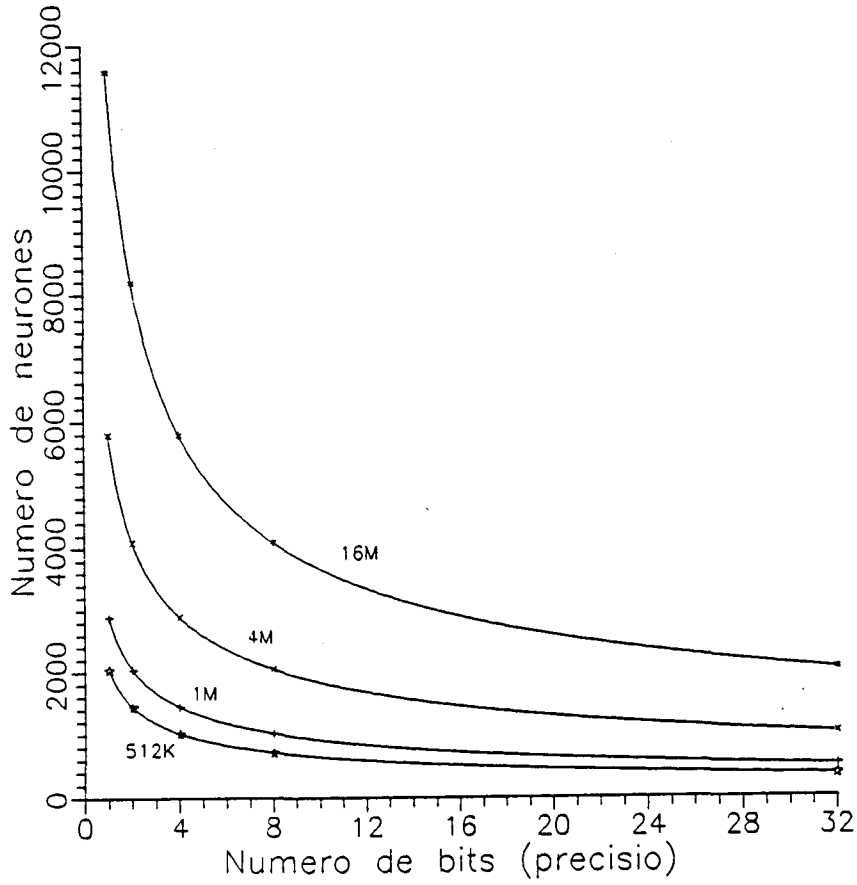


Figura 4.2. Avaluació del número de neurones d'un sistema que implementa una xarxa neuronal totalment interconnectada en funció de la memòria del sistema i de la precisió de les sinapsis.

La decisió de l'alternativa a seguir, depén fortament del tipus de problemes que es pretén resoldre així com de la representació de les dades involucrades. La qüestió de la codificació d'aquestes dades continua essent un dels punts calents en el desenvolupament de les xarxes neuronals. Respecte d'això cal recordar que les xarxes neuronals naturals codifiquen totes les dades en cadenes d'impulsos "digitals" de freqüència variable.

L'objectiu perseguit en la nostra realització ha estat la maximització del número de neurones de la xarxa a canvi de disminuir la precisió dels pesos.

Aquesta decisió s'ha pres per dues raons bàsiques:

(i) Permet incrementar el grau de paral.lelisme simplificant la complexitat del hardware i per tant incrementar la velocitat de relaxació.

(ii) El procés d'aprenentatge s'encarrega d'obtenir pesos de baixa resolució a partir de pesos d'alta resolució, encara que això fa més lenta aquesta fase (Apèndix 2).

D'aquestes consideracions triem els valors característics de la nostra xarxa:

. Número de neurones totalment interconnectades: 4096.
. Número de bits per als pesos: 1 {+1,-1} o 2 {+1,0,-1}.

Si bé aquesta decisió no modifica l'algorisme associat al sistema, donat que tindrem molts processos que involucraran un o dos bits, podem integrar en un xip més d'una unitat de procés, augmentant la velocitat del sistema.

En canvi, la discretització dels pesos introdueix un procés nou: la normalització dels pesos per col.lumnes, com a conseqüència de la formulació associada a les estabilitats. Al capítol anterior, es considerava que la normalització la realitzava la fase d'aprenentatge del sistema en el procés d'obtenció de pesos.

$$h_i' = \frac{\sum_{i=0}^{n} T_{ij} O_j}{\sum_{i=0}^{n} |T_{ij}|} \qquad (4.1)$$

En el cas de pesos discrets $T_{ij} \epsilon \{+1,-1\}$, la normalització no és necessària ja que el número de sinapsis associades a una neurona és igual per a totes.

$$\sum_{i=0}^{n} |T_{ij}| = n \tag{4.2}$$

Però si la representació dels pesos és del tipus $T_{ij} \epsilon \{+1,0,-1\}$ i utilitza dos bits: T^1_{ij} que indica si la connexió és activa o no i T^0_{ij} que indica si és inhibidora o excitadora, cal realitzar el procés de normalització ja que la probabilitat d'un canvi d'estat depén del número de sinapsis connectades a cada neurona.

$$h'_i = \frac{\sum_{i=0}^{n} T^1_{ij} \overline{(T^0_{ij} \oplus O_j)}}{\sum_{i=0}^{n} T^1_{ij}} = \frac{n_u - n_d}{n_u + n_d} = \frac{2n_u - (n_u + n_d)}{n_u + n_d} \tag{4.3}$$

Com a conseqüència, l'algorisme es modifica segons el quadre IV.

El processat en paral.lel només es permet per als càlculs bit a bit, és a dir, per al primer pas d'obtenció del camp local. En aquest cas, es calcularan **m** termes del producte vector-matriu simultàniament (també es reflexa a l'algorisme del quadre IV). En el segon pas de l'algorisme, les dades manipulades són principalment enters i per tant, el paral.lelisme no s'utilitza.

La figura 4.3.a mostra els grafs associats als processos elementals de l'algorisme. Per a la fase quadràtica, tot i la implementació digital, es recupera una estructura neuromorfa, amb **m** processadors d'entrada (sinapsis) i una única sortida (axó). També es mostra el pipeline implementable (figura 4.3.b).

Figura 4.3.- Graf de fluxe de la part del producte vector-matriu d'un sistema amb processadors paral.lels a nivell de bit (quadres IV i V).

Quadre IV. Algorisme adaptat a pesos discrets {+1,0,-1}. (en cursiva els passos que es poden estalviar per a pesos {+1,-1}).

```
for i:= 0,N-1                                   /* càlcul del producte vector-matriu */
{
        R4:=0;R5:=0;                            (*)
        for j:= 0,N-1,m                         /* càlcul del camp local de la neurona j */
                {
                R1 := M(T1ij);                  (*)
                R2 := M(T0ij);
                R3 := M(Oj);
                R4 := R4 + Σl(R1 * (R2 xor R3)); (**) (***)
                R5 := R5 + Σl(R1);              (*) (***)
                }
        M((nu-nd)i) := R4
        M((nu+nd)i) := R5;                      (*)
}
while tran = 1
{
        tran:=0; R4:=0;
        for i:=0,N-1                            /* detecció i elecció de transicions */
        {
                R3 := M(Oi);
                R6 := M((nu-nd)i);
                R7 := M((nu+nd)i);              (*)
                if (R3*R6/R7<R4) then           (**)
                {
                        R4 := R3 * R6/R7;       (**)
                        tran:=1;
                        k:=i;
                }
        }
        R3 := M(Ok);                            /* canvi d'estat de la neurona amb transició
        R4 := -R3;
        M(Ok) := R4;
        for i:=0,N-1                            /* actualització dels valors dels camps locals
        {
                R1 := M(T1ik);                  (*)
                R2 := M(T0ik);
                R6 := M((nu-nd)i);
                R3 := R6 + 2 * R1 (R2 xor R3);  (**)
                M((nu-nd)i) := R3;
        }
}
/* (*) Passos que s'eliminen i (**) Passos amb modificacions per a pesos {+1,-
1} */
/* (***) El sumatori fa referència a que es processen m bits en paral.lel */
```

A la segona fase de l'algorisme de relaxació, actualització dels camps i selecció de la neurona l'estat de la qual canvia, es realitzen operacions sobre enters, tant d'entrada com de sortida.

Com a resultat global estem utilitzant un processat paral.lel per al cas en el qual la complexitat és quadràtica mentre que quan la complexitat és linial és el processat seqüencial. Això en porta a una estimació del temps de procés del tipus:

$$t = t_c\left(\frac{n^2}{m} + kf(\beta)n\right) \qquad (4.4)$$

on t_c és el temps de cícle del sistema i k és una constant que depén de la implementació final (per exemple en el cas del pipeline esmentat anteriorment k=3). Unicament el segon terme de l'equació depén del número d'iteracions que ha de fer la xarxa per assolir un estat estable, mitjançant la funció $f(\beta)$, on β és la distància de Hamming entre l'estat presentat a la xarxa i l'emmagatzemat. Aquest valor el podem obtenir de la gràfica de la figura 3.3 del capítol anterior.

El compromis entre ambdós termes vindrà donat per l'estadística associada al número d'iteracions, que depén al seu torn de l'aplicació per a la qual l'utilitzem.

5.3. Restriccions en la realització de recursos de càlcul VLSI.

Un cop tenim definit l'algorisme òptim per a la nostra dinàmica cal implementar, amb el mínim cost en superfície de silici i la mínima redundància necessària, els recursos de càlcul corresponents a les operacions involucrades.

En aquest sentit, la reducció del rang de pesos de les sinapsis redueix la complexitat dels recursos, tant en superficie com en temps de càlcul, ja que permet substituir el multiplicador, necessari per realitzar els termes producte $T_{ij}O_j$ de la primera fase i $T_{ik}O_k$ de la segona, per operacions lògiques (AND i XOR).

El paral.lelisme introdueix al seu torn únicament un nou recurs: el comptador de número de uns (equivalent a la integració neuronal). Per a la implementació d'aquesta operació utilitzem un recurs de càlcul dissenyat per a la

detecció i/o correcció d'errors en circuits autocomprobables: el comprobador autocomprobable per al codi Berger.

El codi Berguer es caracteritza perquè totes les paraules-codi son separables en part d'informació i part de codi, i perquè permet la detecció d'errors unidireccionals, que formen un dels tipus d'errors més freqüents en circuits digitals MOS.

El disseny del comprobador Berger per a un número de bits de la paraula d'entrada fixe es desenvolupa en detall a l'apèndix 2.

Els càlculs associats a sumes i restes es realitzen mitjançant un únic recurs programable. Aquest recurs es pot dissenyar amb una estructura bit-slice, també detallada a l'apèndix 2.

L'altre recurs de complexitat elevada que cal implementar és **el divisor** necessari per l'operació de normalització.

La implementació de divisors en VLSI amb lògica combinacional ha estat un tema de recerca important degut a l'ús freqüent d'aquesta operació en processadors. Fins a la data no s'ha aconseguit una implemenatció satisfactòria i s'utilitza, en la majoria de cassos, implementacions algorísmiques de complexitat temporal linial respecte del número de bits del quocient.

Implementacions cel.lulars similars a les dels multiplicadors, són poc eficients perquè necessiten una quantitat de portes lògiques elevada (proporcionalment quadràdica amb el número de bits del divisor).

Adicionalment, en un entorn de disseny semi-custom trobem compiladors de data-path i de multiplicadors en la majoria de llibreries de cel.les de diferents fabricants pero no trobem divisors. Llavors l'alternativa és realitzar-lo en full-custom i caracteritzar-lo a nivell elèctric, per després extreure'n els models lògics per tal de poder-lo simular a nivell de sistema amb la resta de components (a falta d'un llenguatge d'alt nivell -tipus VHDL- que permeti integrar aquest conjunt d'eines).

Tant la implementació algorísmica, que introdueix una penalització temporal important, com la realització full-custom, que requereix un coneixement exhaustiu del modelat associat als diferents simuladors utilitzats, ens han fet plantejar la necessitat de substituir aquest recurs.

Per fer-ho, cal tornar enrera i buscar la funcionalitat associada a la divisió. Aquesta apareix de manera natural en la normalització de les estabilitats de la dinàmica probabilística, quan voliem que la influència del canvi en l'estat d'una neurona, sobre el camp local d'una altra, fos proporcional al número total de sinapsis associades a aquesta darrera.

La probabilitat d'una trancisió és independent del número de sinapsis associades a una neurona, ja que si bé l'efecte d'un canvi sobre una neurona amb un número de neurones baix produeix un efecte més gran, la probabilitat de que aquesta estigui interconnectada amb la neurona que canvia és menor.

Estem interessats en escollir la neurona que canvia per comparació de dues magnituds normalitzades,

$$O_i\left(\frac{n_u - n_d}{n_u + n_d}\right)_i \quad ? \quad O_j\left(\frac{n_u - n_d}{n_u + n_d}\right)_j \tag{4.5}$$

Podem realitzar aquesta comparació sense haver de fer la divisió, mitjançant un desplaçament a l'esquerra del valor del camp local proporcional al número de sinapsis associades a aquesta neurona. Aquest desplaçament equival a treballar amb exponencials i logaritmes en lloc de quocients.

$$2^{d_i}\,(n_u - n_d)_i \quad ? \quad 2^{d_j}\,(n_u - n_d)_j \tag{4.6}$$

Per a N, número màxim de neurones de la xarxa, i $(n_u+n_d)_i$ sinapsis associades a una neurona, la quantitat de desplaçaments a l'esquerra d_i sobre el camp local d'aquesta ve donada per l'expressió:

$$d_i = \log_2 N - \log_2 (n_u+n_d)_i$$

La implementació d'ambdós logaritmes és relativament senzilla, ja que n serà normalment potència de 2, per tal d'aprofitar al màxim els recursos de la màquina, i $\log_2 m$ requereix una estructura complexa.

Donat que l'elecció del camp màxim normalitzat parteix d'un raonament probabilístic, realitzat sobre els conjunt de pesos possibles $\{T_{ij}\}$ que es dedueixen a la fase d'aprenentatge, podem establir una estratègia alternativa a la normalització amb una precisió menor.

En aquesta estratègia, aproximem el logaritme per una correspondència entre l'aparició al bit i per l'esquerra del valor al qual aplique el logaritme, i el valor del logaritme. Aquesta nova funció es pot formalitzar en llenguatge C com,

```
int ourlog(z)
int z
{
        int x,k,fl;
        fl=0; x=1; x<<=12;
        for(k=0;fl!=1;k++)
                if(z>(x|(x>>=1))) (fl=1);
        return(k);
}
```

L'aproximació introduïda dóna una qualitat de recuperació similar, simplificant de manera important el càlcul del logarisme, que ara es pot implementar amb un número més reduït de portes, com es mostra a l'apèndix 2.

Juntament amb el logaritme modificat, cal implementar l'exponencial per a la comparació de magnituds (apèndix 2), que en C pot escriure's com:

```
int ourexp(z,y)
int z,y;
{
        int x,k;
        for(k=0;k<4;k++)
        {
                x=1;
                if( y & (x<<=k)) (z<<=x);
        }
        return z;
}
```

L'avantatge principal d'aquestes implementacions és que utilitzen números enters en el càlcul, simplificant el disseny dels recursos combinacionals. Tant en l'estratègia de normalització per divisió, com en la normalització logarítmica, els valors que hem de guardar en memòria són els dels camps locals sense normalitzar, ja que l'actualització que n'hem de fer és de ± 2, que és un valor no normalitzat. Això comporta guardar també el número de sinapsis associades a

cada neurona per a la normalització. Amb la solució logarítmica, la quantitat de memòria que cal emmagatzemar per aquest segon concepte és menor.

Quadre V. Algorisme equivalent al del quadre IV amb els recursos implementats.

```
for i:= 0,N-1
{
        R4:=0;R5:=0;
        for j:= 0,N-1,m
                {
                R1 := M(T1ij);
                R2 := M(T0ij);
                R3 := M(Oj);
                R4 := R4 + Berger(R1 * (R2 xor R3));
                R5 := R5 + Berger(R1);
                }
        R4 := R4-R5/2;
        R5 := ourlog(R5) ;
        M(hi0) := R4;
        M(hi1) := R5;
}
while tran = 1
{
        tran:=0; R4:=0;
        for i:=0,N-1
        {
                R3 := M(Oi);
                R6 := M(hi0);
                R7 := M(hi1);
                if ( ourexp(R6,R7)*R3 <R4 ) then
                {
                        R4 := ourexp(R6,R7) * R3;
                        tran:=1;
                        k:=i;
                }
        }
        R3 := M(Ok);
        R4 := -R3;
        M(Ok) := R4;
        for i:=0,N-1
        {
                R1 := M(T1ik);
                R2 := M(T0ik);
                R6 := M(hi0);
                R3 := R6 + R1 (R2 xor R3);
                M(hi) := R3;
        }
}
```

Amb aquestes modificacions obtenim l'algorisme final que implementem, mostrat al quadre V, i que manté l'estructura neuromorfa del graf de la figura 4.3.

En les simulacions realitzades, la diferència que apareix en els tres cassos de normalització és mínima. Això, es degut a que aquestes simulacions estan realitzades amb una funcionalitat de memòria associativa utilitzant una regla de Hebb. En aquestes condicions, la diferència en el número de sinapsis associades a una neurona per a la xarxa sencera, serà molt poca. Si a això hi afegim l'estadística realitzada sobre diferents vectors inicials i per a diferents matrius, l'efecte es dilueix molt més.

Capítol 5

COPROCESSADOR NEURONAL

En el capítol anterior hem realitzat la correspondència entre el model matemàtic i l'algorisme que ha d'implementar el xip. Aquest ha estat dissenyat fent ús d'una estratègia de disseny basada en llibreries de cel.les, introduïnt un seguit de restriccions forçades pel nombre limitat de recursos que utilitzem, i intentant de fer òptims els nostres objectius: el número de neurones i la velocitat de procés de la fase de relaxació.

En aquest capítol, es pretén fer la correspondència entre el nivell de descripció algorísmica i la descripció associada als xips: estratègia de disseny, estructura del xip, disseny lògic, característiques dels dispositius i portes lògiques, etc. Al contrari que en cas anterior, aquesta correspondència no estableix restriccions sobre el sistema, sinò que es permeten diferents estratègies de realització dels xips en funció de dos paràmetres d'alt nivell bàsics en VLSI: temps i programabilitat.

El temps en aquest cas no es determina a partir de les característiques del sistema matemàtic ni de l'algorisme, sinó amb les del propi procés tecnològic (temps de resposta dels dispositius). Les característiques dels xips dissenyats depenen d'ambdós, de la velocitat de relaxació i del nivell de programabilitat, a través de la topologia desitjada pel sistema neuronal.

La complexitat del sistema VLSI a implementar, depén directament del grau de compromís escollit entre ambdós paràmetres a l'elecció realitzada. Aquesta complexitat es pot mesurar sobre paràmetres d'un nivell d'abstracció més baix: complexitat de la unitat de control, velocitat de relaxació de la xarxa, grau de programabilitat explícita, throughput del sistema.

Per a la concepció dels xips ens hem proposat únicament dues restriccions: **disseny digital amb estratègia basada en cel.les**, utilitzant al màxim les eines de disseny assistit, i manteniment d'una **arquitectura derivada de l'arquitectura clàssica de von Neuman** (unitat de procés més unitat de control). La modificació sobre aquesta arquitectura ve donada per un grau superior de paral.lelisme en el càlcul de producte vector-matriu.

No cal oblidar tampoc el propòsit dels nostres xips: emular xarxes neuronals el màxim de genèriques per tal de disposar del màxim número de topologíes diferents dins d'un mateix sistema neuronal.

5.1 Realització de topologies

Dels resultats presentats i l'estratègia perfilada en capítols anteriors en podem extreure com a conseqüència, que la nostra xarxa serà òptima en termes d'utilització de recursos i memòria per a xarxes totalment interconnectades. Ara bé, volem que el nostre sistema sigui al mateix temps capaç d'implementar altres topologies que poden ser interessants des del punt de vista funcional.

Aquestes formes alternatives d'interconnexió les aconseguim modificant la distribució de zeros a la matriu d'interconnexions, és a dir, tallant interconnexions entre neurones.

L'aspecte de les matrius resultants pot veure's a la figura 5.1, per a dues de les xarxes més utilitzades: la multilayer perceptron y la xarxa de Kohonen.

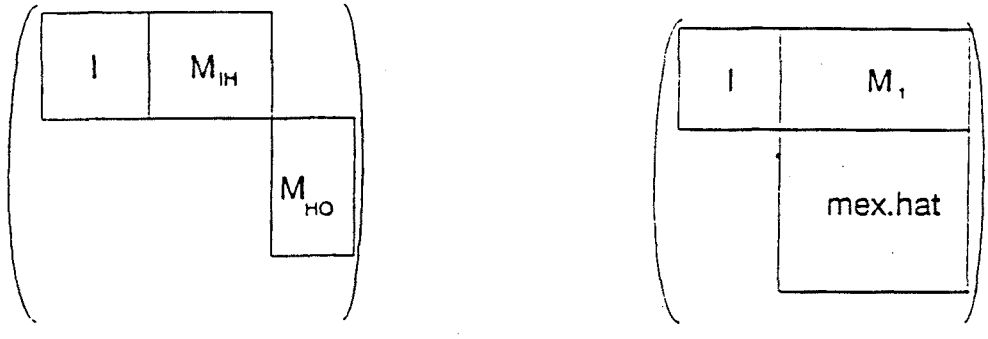


Figura 5.1.- Matrius d'interconnexió per a: (i) una xarxa multilayer perceptron, (ii) una Xarxa de Kohonen.

En el primer cas, la matriu dels pesos s'obtenen utilitzant regles d'aprenentatge de tipus "back-propagation", amb una etapa adicional de conversió de les sinapsis a valors binaris [Pérez91].

Pel que fa a la xarxa de Kohonen, podem veure la combinació de sinapsis que connecten les entrades externes a les neurones i sinapsis de realimentació entre neurones. Per a la xarxa de Kohonen, les regles d'aprenentatge s'utilitzen només per al primer conjunt de sinapsis, mentre que les del segon tenen una topologia fixa donada per la forma del barret mexicà. En el primer cas, els mecanismes de discretització abastarien el procés d'aprenentatge, mentre que en el segon ja han estat realitzades implementacions que discretitzen el barret mantenint les propietats d'inhibició lateral i formació de bombolles que produeixen aquests sistemes [Kohon89].

La generalització d'aquesta representació permet dissenyar també sistemes neuronals, combinació de diferents xarxes encadenades. Un exemple d'això el constituirien les xarxes de tipus Jordan que permeten la síntesi de màquines seqüencials. Aquesta xarxa es pot veure com formada per una multilayer perceptró i una xarxa de Hopfield. A la matriu resultat (figura 5.2), es diferencien perfectament els dos layers.

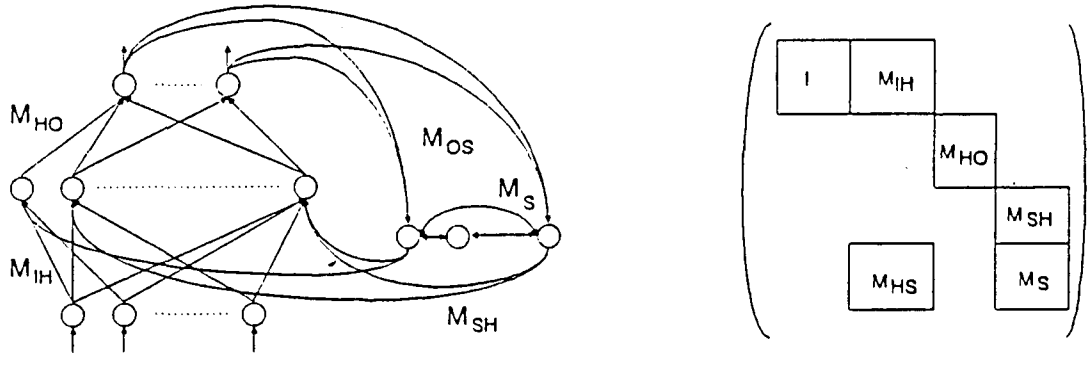


Figura 5.2.- Topologia (i) i Matriu d'interconnexió (ii) per la xarxa de Jordan.

5.2 Dues estratègies bàsiques per a la connectivitat

L'elevat número de zeros que trobem en matrius de topologies diferents de la topologia totalment interconnectada d'Hopfield ens introdueixen una qüestió fonamental a l'hora de decidir quin o quins xips dissenyarem.

El nostre algorisme ens diu com hem de processar les dades (camps locals, estats i interconnexions), però es pot pensar en solucions ad-hoc per a cada xarxa de manera que no es processin les submatrius d'interconnexions que prèviament sabem que contindran zeros.

Les dues estratègies bàsiques són:

(i) Estratègia transparent: Processar qualsevol matriu segons el mateix algorisme de control, com si fos de topologia Hopfield.

(ii) Estratègia especialitzada: Per a cada matriu establir un algorisme de control diferent, optimitzant el temps d'operació.

La decisió de quina estratègia escollir, afecta principament el disseny del processador neuronal, més que no pas la quantitat de memoria, ja que volem mantenir el grau de generalitat, i per tant la possibilitat de programar la màxima interconnectivitat.

L'avaluació del cost de cada alternativa es fa sobre els paràmetres de baix nivell abans esmentats.

5.2.1 Complexitat de la unitat de control

En el cas de l'estratègia transparent, l'algorisme que ha d'implementar la unitat de control és molt simple: Donat el número de neurones de la xarxa, existeix un autòmat que indica l'estat en el qual es troba la xarxa i s'encarrega de controlar el fluxe de les dades: transferències entre memòria i registres, fluxe intern a la unitat de procés, control dels recursos de càlcul programables, etc. mitjançant una estructura de decodificació. La unitat de control conté també, per als estats en que es realitzen processos iteratius, dos comtpadors capaços de comptar fins al número de neurones màxim.

Per a l'estratègia especial, si volem implementar una unitat de control flexible i vàlida per a qualsevol model de xarxa, aquesta serà necessariament complexa. La unitat de control haurà de controlar quin és el número de sinapsis associat a cada neurona, quina és la primera sinapsis activa i per quines neurones és vàlida aquesta informació, per tal de donar els límits del comptador. Això fa que la quantitat de memòria necessària per emmagatzemar aquests valorsawsigui més important que l'associada al camp local (tot i que molt menor que la corresponent a sinapsis) i per tant cal que sigui exterior al xip.

Al mateix temps, per a aquesta estratègia cal augmentar el número d'accessos a memòria i introduir busos nous d'accés a l'exterior, per tal d'aproximar-nos al número d'iteracions mínim donat per l'algorisme.

Aquestes modificacions afecten els dos passos de l'algorisme i per tant l'increment de complexitat és elevat.

5.2.2 Estalvi de temps

L'estalvi de temps és la raó bàsica per la qual es planteja la viabilitat de la segona estratègia. L'estalvi temporal que s'obté de l'estratègia especial, està lligat a les dimensions de les submatrius.

Segons el criteri d'avaluació temporal donat al capítol anterior i suposant que la modificació de la unitat de control no necessita cicles adicionals, les següents expressions donen una avaluació del temps de procés en els tres models de xarxa introduïts: una xarxa d'Hopfield (5.1), una xarxa multilayer perceptron (5.2) i una xarxa de Kohonen (5.3).

$$t = t_c \left(\frac{n^2}{m} + kf(d_H)n\right) \tag{5.1}$$

$$t = t_c \left(\frac{a^2 + ab + bc}{m} + k_1 f_1(d_H)(b + c)\right) \tag{5.2}$$

$$t = t_c \left(\frac{a^2 + ab + b^2}{m} + k_2 f_2(d_H) b\right) \tag{5.3}$$

En aquestes equacions, t_c és el temps de cicle de processador neuronal i d_H és la distància de Hamming del vector presentat, de manera que $f_l(d_H)$ és la funció que dóna el número d'iteracions fins arrivar a un estat estable (figura 3.3 per a la xarxa de Hopfield).

Per a l'estratègia transparent, el temps de procés sempre vindrà donat per l'expressió (5.1), mentre que per a l'estratègia especial tenim un guany en temps tant per a una xarxa sense realimentació, com per a una amb realimentació parcial.

5.2.3 Programabilitat

L'estratègia especial requereix una funcionalitat específica a la unitat de control, programable externament mitjançant els paràmetres anteriorment esmentats. Aquests paràmetres són genèrics per qualsevol tipus de xarxa i la diferenciació de cada xarxa vindrà donada pels valors que poden prendre certes variables de control del fluxe. Així, si volem utilitzar un llenguatje de xarxes d'alt nivell manipulant conceptes com número de neurones, topologies,... haurem de generar una representació intermitja entre aquesta i el processador neuronal de manera que per a cada xarxa es configurin els paràmetres corresponents de la unitat de control. Això implica un esforç adicional a nivell de software que no és necessari si utilitzen una estratègia transparent, en la qual els valors d'interconnexions no determinats per l'algorisme d'aprenentatge són zero.

5.2.4 Throughput

Aquest concepte està íntimament lligat al màxim número de pins d'entrada-sortida que es poden dedicar a dades que és 2 o 3 vegades (per a pesos {+1,-1} i {+1,0,-1} respectivament) el número de processadors **m** que poden funcionar en paral.lel en la primera fase de l'algorisme, que fa el càlcul del producte vector-matriu (figura 4.3).

Com a conseqüència, el temps de càlcul d'aquesta primera fase es redueix de forma inversament proporcional a **m**. Incrementar el número de pins de dades,

implica de manera directa augmentar el paral.lelisme i disminuir el temps de procés.

Aquest aspecte, per un número fixe de pins d'entrada-sortida dóna avantatge a l'estratègia transparent respecte de l'estratègia especial, ja que la programació associada a aquesta segona implicarà la necessitat de dedicar una part dels pins a la unitat de control.

Una primera avaluació del número de pins que podem dedicar a dades, el podem fer si restem del total el número de pins dedicats a altres dades suposant un encapsulat DIL de 64 pins, estandard dels projectes MPC al CNM.

Per a una xarxa amb 2^{12} neurones, són necessaris el següents pins:

- Un mínim de 2 pins d'alimentació, 1 de rellotge i 1 de reset.
- 2 busos d'adreces de memòria de 12 i 9 bits, i 8 bits de selecció de memòria.

Llavors el número de processadors en paral.lel funció de l'estratègia i del conjunt de pesos són a la taula 5.1, per a la realització del qual s'ha suposat que només és necessari un bus de dades de 12 bits adicional per a l'estratègia especial (i es poden utilitzar els busos d'adreces ja existents).

	{+1,-1}	{+1,0,-1}
E. Transparent	16	10
E. Especial	9	6

Taula 5.1. Número de processadors en paral.lel en funció de l'estratègia utilitzada i del conjunt dels pesos.

D'aquest quadre se'n treu la conclusió que les dues estratègies més factibles d'implementació són:

(i) l'estratègia transparent amb pesos {+1,0,-1}.

(ii) l'estratègia especial amb pesos {+1,-1}.

Ambdues estratègies tenen throughputs similars, mentre que de les altres estrategies, l'estratègia transparent amb pesos {+1,-1} permet un grau molt reduït de topologies diferents, i l'estratègia especial amb pesos {+1,0,-1} té un througput molt baix. En aquest cas, l'opció de tenir el zero com a element adicional a la definició de topologia pot, per algunes topologies i regles d'aprenentatge, no aportar informació adicional relevant.

5.2 NDN2

El xip NDN2 (figura 5.3) fou el primer que vam dissenyat. L'estratègia bàsica utilitzada fou la de reduir la complexitat del xip, augmentant la complexitat a nivell de sistema digital.

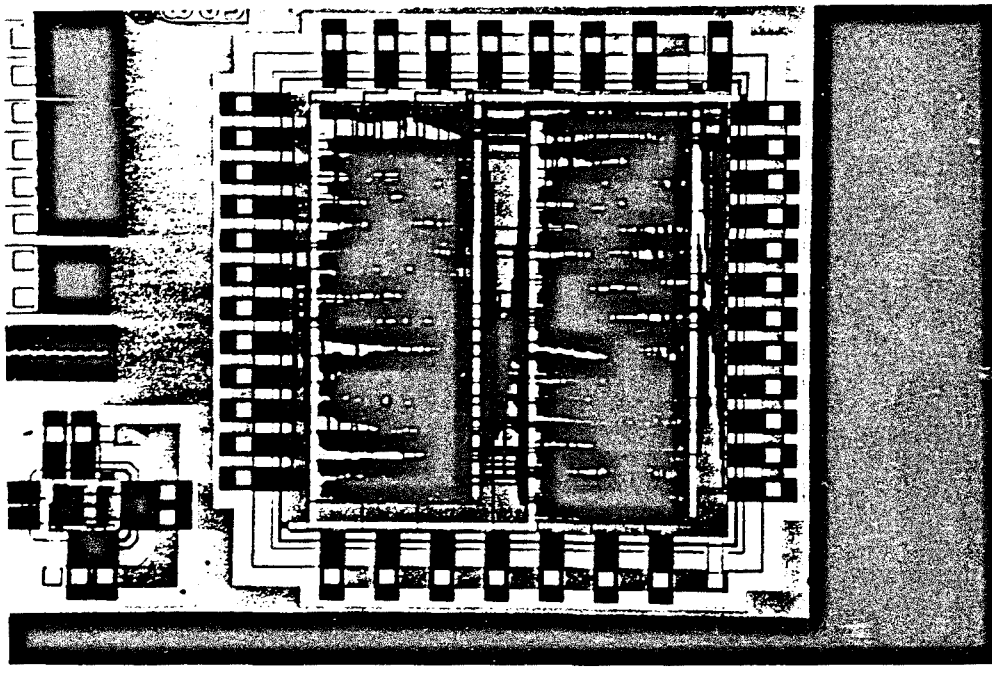


Figura 5.3 Fotografia del xip NDN2.

Les característiques del xip són:

- Implementa únicament la unitat de procés.
- Treballa amb pesos {+1,-1}.
- Permet un paral.lelisme de m = 12 processadors d'un bit.
- Té capacitat per a un màxim de 4096 sinapsis per neurona.
- Ha estat dissenyat amb una estratègia basada en optimized array (fig 5.4), amb el paquet de disseny de CIs SOLO 1400, en la tecnologia CMOS de 2 micres de ES2.

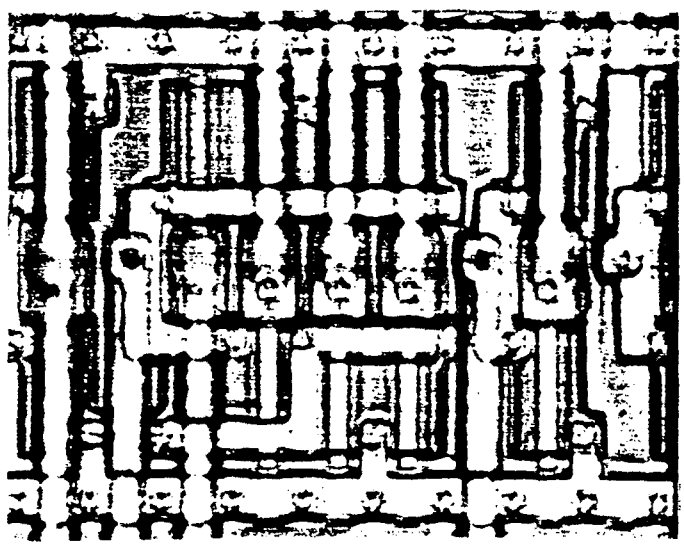


Figura 5.4. Fotografia de cel.les dissenyades amb l'estratègia Optimized Array.

Aquest xip està pensat per seguir l'estratègia especial per la qual, la topologia de la xarxa s'estableix mitjançant la programació de la unitat de control. Aquesta es programarà des del PC, directament a memòria o amb un dispositiu programable comercial auxiliar del tipus LCA (logic cell array), configurat com a unitat de control.

Els avantatges bàsics de no integrar la unitat de control cal buscar-los en dos aspectes que afecten directament el throughput: les restriccions del procés d'encapsulat i l'augment de número de pins.

El procés d'encapsulat ens ha permés fins fa poc temps un rang d'encapsulats DIL limitat a 24, 40 i 48 pins. Això restringia les prestacions que es podien aconseguir mitjançant l'augment del paral.lelisme dels càlculs. En aquest

cas podiem escollir les solucions de reduir el número de senyals a utilitzar o multiplexar l'accés a les dades. Ambdues solucions impliquen una pèrdua de prestacions.

Pel que fa al número de pins, aquest valor també afecta directament el número de processadors que poden funcionar en paral.lel. En aquest cas, la pèrdua en el número de pins es produeix pel fet d'haver de direccionar la memòria externa que requereix un número elevat de pins (P.e. en l'avaluació anterior 29 pins de l'encapsulat de 64, nombre difícil de reduir ja que direccionem el mateix espai de memòria).

La dedicació dels pins del xip següeix el següent repartiment:

- 4 pins per alimentació, terra, rellotge i reset.
- 12 pins per a senyals de comuniació entre la unitat de procés i la de control (11) i a l'inversa (1).
- 24 pins per dades, dividides en dos busos de 12 bits.

L'estratègia d'implementació de l'algorisme es basa en la divisió en operacions més senzilles (Quadre I). La unitat de control, manipula aquestes operacions bàsiques en comptes de controlar directament els senyals associats a registres (senyals de càrrega) i recursos programables (codis d'operació). Aquesta estratègia permet igualment reduir el throughput, ja que substituim els 29 bits d'adreça i selecció de memòria pels 12 associats a la comunicació entre la unitat de procés i la unitat de control.

Funció	Codi d'operació	Cicles/operació
Càlcul $\Sigma_j T_{ij} O_j$	0 0 0	1
Càlcul de $(n_u - n_d)$	1 0 0	4
Selecció $\min(n_u - n_d)$	0 1 0	1
Canvi d'estat	0 1 1	3
Actualització $(n_u - n_d)$	1 1 1	3

Quadre II. Codificació de les operacions que realitza l'algorisme per al NDN2 i número de cicles necessaris per la seva realització.

L'arquitectura interna que utilitza el xip es mostra a la figura 5.5. En ella s'observa que únicament és necessari un registre programable del tipus sumador-restador, un recurs de tipus Berger de 12 bits, quatre registres de 12 bits i lògica combinacional de selecció, decodificació, etc.

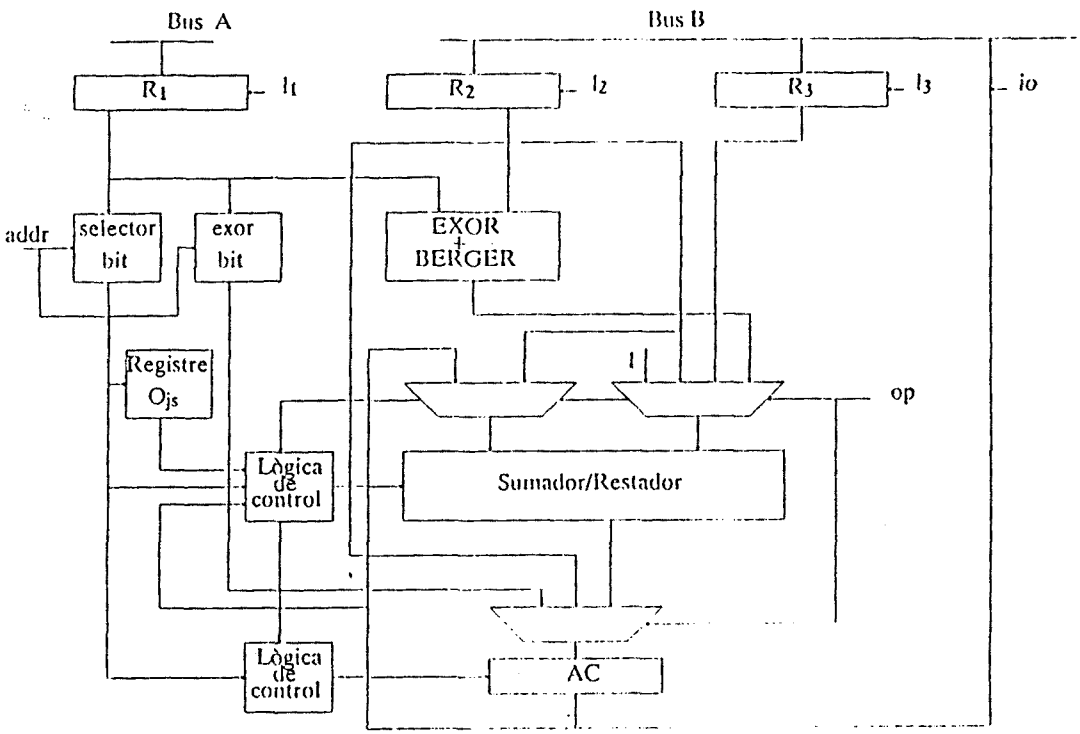


Figura 5.5.- Arquitectura a nivell de registres del xip NDN2.

L'estimació de velocitat que podem fer sobre el sistema digital basat en el xip NDN2, utilitzant les avaluacions de temps per cicle donades per l'algorisme de la xarxa i la utilització del pipeline, ve donada per l'expressió:

$$t = t_c \left[n \left(\frac{n}{12} + 3 \right) + f(d_H)(4n+3) \right] \qquad (5.4)$$

El xip fabricat conté 3392 stages (mesura de ES2 equivalent aproximadament a 3 transistors), en una superfície sense pads de 7.83 mm^2 (2.72 x 2.85), mentre que la introducció dels 40 pads incrementa la superfície fins 14.45 mm^2 (3.74 x 3.87).

El circuit fou testejat amb l'analitzador lògic Tektronix DAS9100. La velocitat màxima mesurada en el test del circuit fou de 5 MHz.

5.3 NDN3

El xip NDN3 (figura 5.6) està dissenyat intentant integrar el màxim número de circuits i sistemes lògics dins del xip, de manera que es redueix la complexitat del sistema resultant a nivell de placa de circuit imprés.

Aquesta filosofia és complementària a la utilitzada en el disseny de l'NDN2 i ens aporta un avantatge important: podem simular a nivell lògic tots els elements del xip i fins i tot una part important de la funcionalitat del sistema, que estarà composat de xips de memòria i de comunicació amb l'exterior per a l'entrada-sortida de dades.

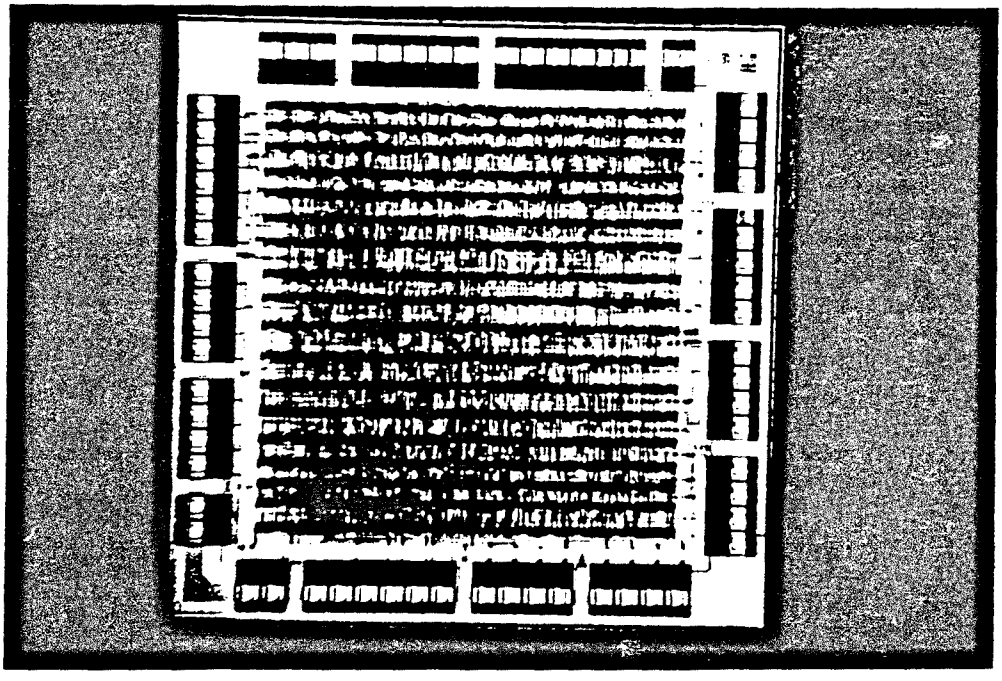


Figura 5.6. Fotografia del xip NDN3.

Les característiques del NDN3 són:

- Implementa la unitat de procés, la unitat de control, la gestió de la comunicació amb l'exterior i la sincronització dels processos.
- Treballa amb pesos {+1,0,-1}.
- Permet un paral.lelisme de m = 8 processadors d'un bit.
- Té capacitat per a un màxim de 4096 neurones, amb un màxim de 4096 sinapsis per neurona.
- Ha estat dissenyat amb una estratègia ASIC basada en standard cells (figura 5.7), amb el paquet de disseny SOLO 2000, en la tecnologia CMOS de 2 micres de ES2, en l'entorn del programa europeu EUROCHIP.

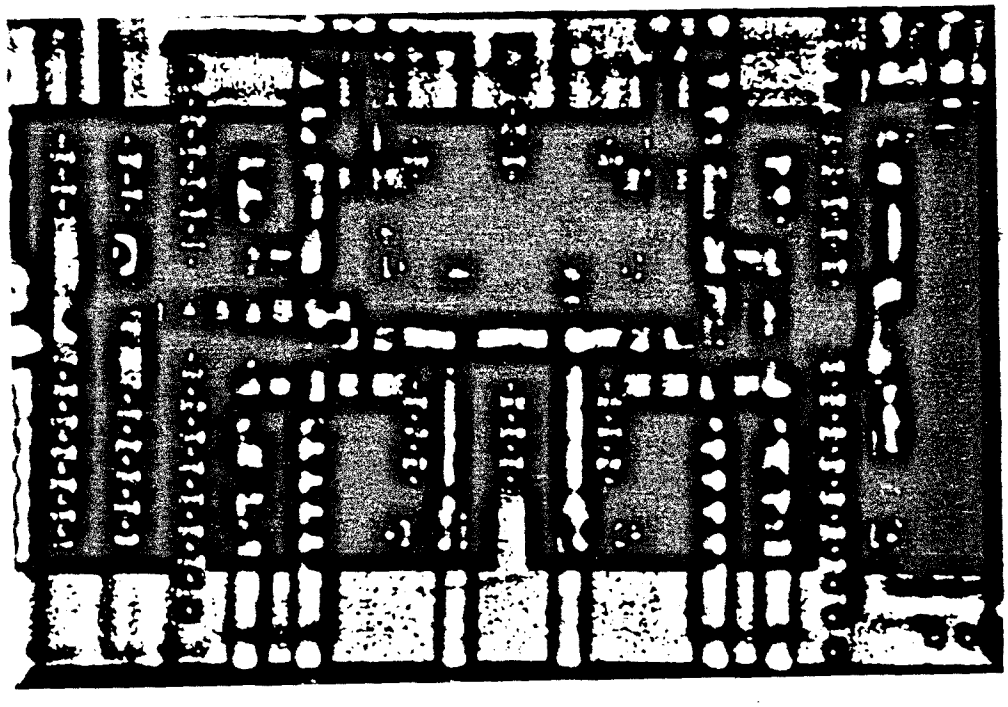


Figura 5.7. Detall d'algunes Standard Cells del xip NDN3.

Aquest xip segueix la que hem anomenat estratègia transparent, en la qual la topologia de la xarxa s'estableix mitjançant la distribució de matrius zero generades per la regla d'aprenentatge.

La unitat de procés implementada (figura 5.8) varia lleugerament respecte de la proposada al final del capítol 4. Les modificacions (Quadre II) introdueixen un nou nivell de pipeline, la primera avaluació de l'estabilitat es realitza simultàniament al càlcul del producte vector-matriu i també es realitzen en el mateix bucle el canvi d'estat de la neurona seleccionada i les actualitzacions dels camps locals.

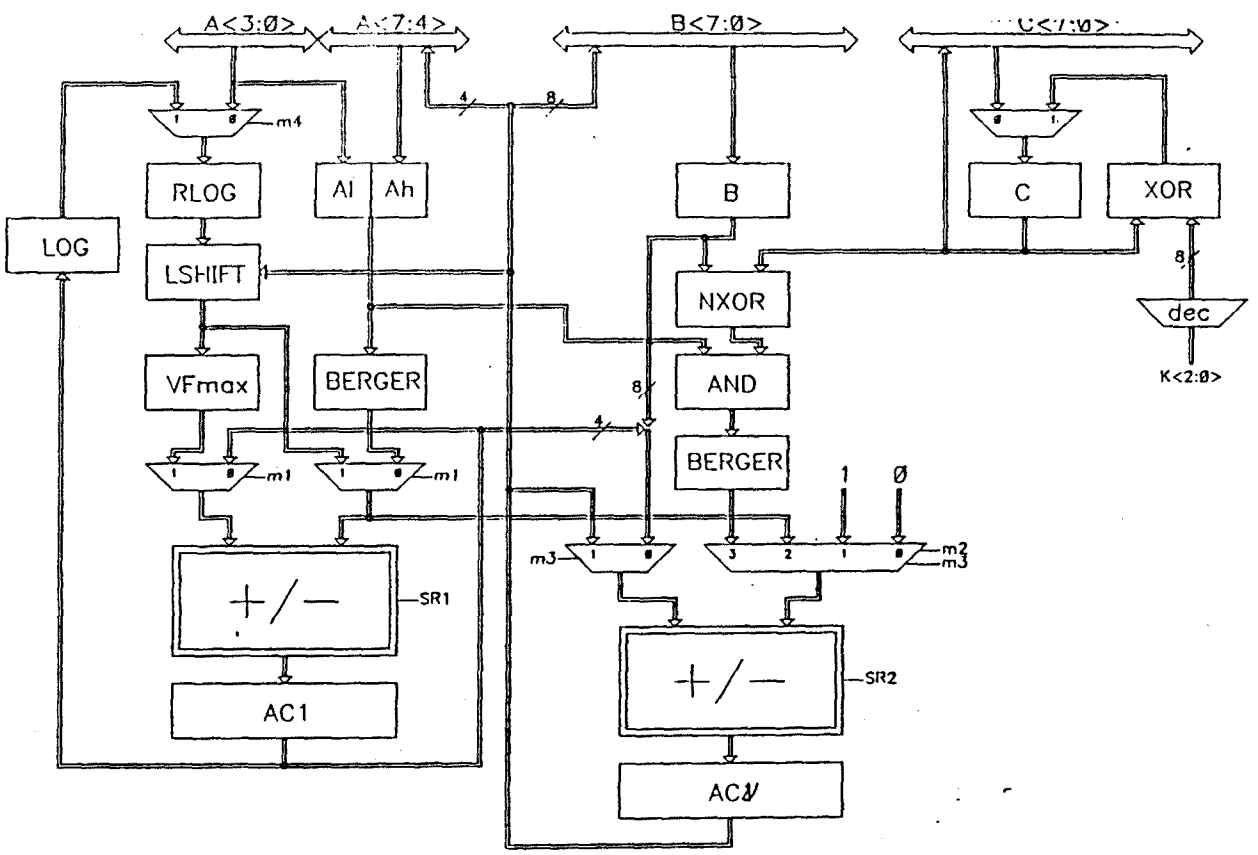


Figura 5.8.- Esquema de la unitat de procés del xip NDN3.

Quadre III. Algorisme implementat per la unitat de procés del xip NDN3.

```
for i:= 0,N-1
{
        R4:=0;R5:=0;
        for j:= 0,N-1,m
                {
                R1 := M(T1ij);
                R2 := M(T0ij);
                R3 := M(Oj);
                R4 := R4 + Berger(R1 * (R2 xor R3));
                R5 := R5 + Berger(R1);
                }
        R4 := R4-R5/2;
        R5 := ourlog(R5) ;
        M(hi0) := R4;
        M(hi1) := R5;
}
while tran = 1
{
        tran:=0; R4:=0;
        for i:=0,N-1
        {
                R3 := M(Oi);
                R6 := M(hi0);
                R7 := M(hi1);
                if ( ourexp(R6,R7)*R3 <R4 ) then
                {
                        R4 := ourexp(R6,R7) * R3;
                        tran:=1;
                        k:=i;
                }
        }
        R3 := M(Ok);
        R4 := -R3;
        M(Ok) := R4;
        for i:=0,N-1
        {
                R1 := M(T1ik);
                R2 := M(T0ik);
                R6 := M(hi0);
                R3 := R6 + R1 (R2 xor R3);
                M(hi) := R3;
        }
}
```

L'elecció de l'estratègia transparent ens permet dissenyar una unitat de control molt simple que realitza el seqüenciament de les 8 operacions bàsiques de la unitat de procés, generant les condicions dels bucles amb comptadors programables.

El graf associat a l'autòmat de la UC es mostra a la figura 5.9, i està composat d'únicament 3 registres i la lògica combinacional associada (Apèndix 2).

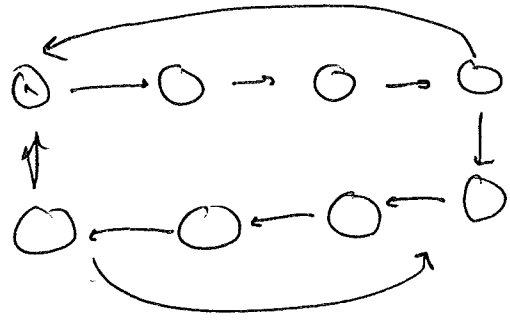


Figura 5.9.- Graf de l'autòmata implementat a la unitat de control.

A diferència del que succeïa amb l'NDN2, la tasca de gestionar la memòria es fa des de dins del xip i per tant, cal generar les adreces i senyals de selecció de bancs de memòria per a cada pas de l'algorisme. Les adreces es generen directament amb els comptadors de la unitat de control, mentre que els senyals de selecció de xip es generen a partir de l'autòmat de la unitat de control.

La dificultat del procés ve donada, no tant per la generació dels esmentats valors lògics, sinó pel format que hem de donar als senyals per tal de complir les especificacions de les memòries comercials utilitzades. A diferència del que passa quan es dissenyen mòduls en full-custom, en el qual cas s'ha de sintetitzar o extreure els valors per al simulador lògic amb el qual es dissenya a nivell de porta, els fulls d'especificacions de les memòries comercials porten tots els valors temporals necessaris a nivell lògic.

L'únic que cal és introduir-los al model de memòria que te el simulador. Aquest procés el vam realitzar sintetitzant un mòdul de memòria de les mateixes dimensions que el que haviem d'utilitzar, substituïnt els valors temporals associats pels valors donats als fulls d'especificacions. D'aquesta forma, vam poder simular el funcionament no només a nivell de xip, sinò a nivell de sistema. Cal esmentar,

que la memòria utilitzada és RAM estàtica la qual cosa fa que l'únic rellotge del sistema és el que sintetitza l'NDN3 amb un cristall.

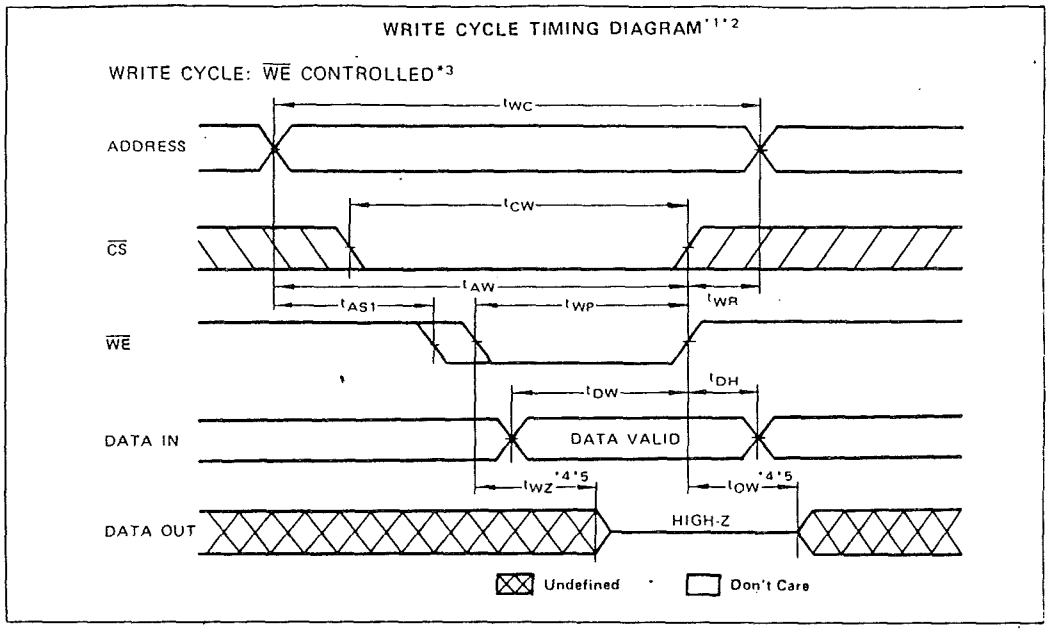


Figura 5.10.- Restriccions típiques entre els senyals de les memòries estàtiques.

Les restriccions temporals entre els diferents senyals de les memòries estàtiques: lectura/escriptura, adreces de memòria i selecció de xip, són importants (figura 5.10) i no varien coincidint amb flancs de rellotge, sinò que estan retardades un cert temps.

En el cas de tenir un únic cicle d'accés a memòria (lectura/escriptura) per cícle de rellotge del processador, cal realitzar els retards mitjançant estructures combinacionals. Aquesta estratègia no és aconsellable ja que variacions de procés temperatura o tensió d'alimentació poden afectar de manera molt important el valor d'aquest retard (més d'un 100%). Això ens porta a treballar amb un rellotge de freqüència quatre cops més elevada a la interna del xip, de la qual n'extraiem quatre fases que ens permeten generar el protocol correcte entre els diferents senyals de control de les memòries.

La concepció del sistema dissenyat a partir de l'NDN3 com un sistema síncron amb un rellotge intern, té com a principal avantatge la transportabilitat respecte del sistema al qual el volem connectar. Les prestacions del sistema no depenen de la velocitat del host al qual està connectat, la qual cosa pot portar

problemes quan aquesta és massa elevada, però introdueix el problema de la comunicació entre dos sistemes síncrons.

Aquesta comunicació ha d'estar controlada per un àrbit, que estableixi els protocols corresponents per a la gestió, tant de la comunicació directament, basada en senyals d'interrupció, com de la transferència de dades entre el banc de memòria del host i la interna de la placa.

L'accés a la memòria interna del host directament pel processador neuronal és impensable, tant en termes de transportabilitat, com d'arquitectura del propi host ja que normalment no es poden controlar de manera senzilla els perifèrics del host. Cal un coneixement profund del processador i una adaptació a aquest, perdent-se una part important de la generalitat del coprocessador.

La solució que hem utilitzat passa un altre cop per reduir el treball de disseny de CIs no estàndar ni innovador. Per això hem utilitzat una **memòria de doble port**, xip estandard de comunicació entre sistemes síncrons. El xip que la implementa conté un sol bloc de memòria estàtica al qual es pot accedir des de dos ports (esquerre i dret) de manera independent tant per a la lectura com per a l'escriptura. Per als cassos en que s'accedeix a una mateixa posició de memòria pels dos ports simultàniament, es disposa d'un àrbit intern que estableix la prioritat per un d'ells generant un senyal d'ocupat (busy) per a l'altre.

La pròpia memòria de doble port, a més de les tasques esmentades que utilitzem per a la **transferència de dades**, pot ser utilitzada com a **gestor de les interrupcions**. Existeix una posició de memòria per cada port tal que quan s'escriu des d'un port (esquerre/dret) es genera una interrupció en l'altre (dret/esquerre). El senyal d'interrupció en un port es reseteja quan es llegeix la posició de memòria que ha activat la interrupció.

La utilització de la memòria de doble port en canvi, no suposa cap pèrdua de generalitat, ja que el nostre sistema neuronal i el host estan separats per aquesta memòria, esdevenint independents. La gestió a la banda del host respecte a l'espai de direccionament i gestió de les interrupcions és específica d'aquest i per això, cal conèixer les característiques corresponents a nivell de sistema.

La introducció de la memòria de doble port dins el sistema, ens ha determinat la gestió de les comunicacions, les quals hem introduït dins el xip

NDN3. L'esquema bàsic fa independents totes les operacions, establint un esquema per a la comunicació tal que per a cada operació s'envia una paraula-codi d'operació, en un sentit quan s'inicia el procés i en l'altre quan s'ha finalitzat (figura 5.11).

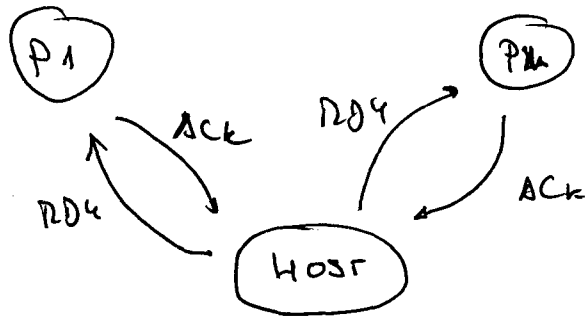


Figura 5.11.- Graf associat a la comunicació entre el host i el NDN3.

El xip ha estat encapsulat en un estandard LCC de 84 pins. La dedicació dels pins del xip segueix el següent repartiment:

- 12 pins per alimentació (4) i terra (8).
- 4 pins per a rellotge i reset.
- 36 pins d'adreces de memòria (30) i senyals de selecció i escriptura (6).
- 24 pins per dades, dividits en tres busos de 8 bits.
- 1 pin d'interrupció des de la memòria de doble port.

L'estimació de velocitat que podem fer cisobre el sistema digital basat en el xip NDN3, utilitzant les avaluacions de temps per cicle donades per l'algorisme de la xarxa i la utilització del pipeline, ve donada per l'expressió (5.5).

$$t = t_c \left(n \left(\frac{n}{8} + 3 \right) + f(d_H) 4n \right) \qquad (5.5)$$

El xip fabricat conté 1294 cel.les standard, en una superfície amb pads de 28.8 mm^2 (5.3 x 5.3). La dimensió i número de transistors de les cel.les estandard és variable. Hem realitzat una estimació del número de transistors del xip sense

els pads, a partir de les portes lògiques utilitzades, de aproximadament 15.000 transistors.

L'NDN3 fou dissenyat per funcionar amb un temps de cicle intern de 100ns (10MHz) i extern de 25ns (40MHz). Aquest cicle està restringit per mínim temps d'accés de les memòries estàtiques d'un megabit de capacitat.

El test del circuit va ser realitzat amb la màquina de test Tektronix LV500. El pattern de test es va obtenir a partir dels vectors de simulació amb el programa Tekwaves.

En placa de circuit imprés, hem aconseguit fer funcionar el sistema a una velocitat de 48 MHz de cicle extern (12MHz de cicle intern).

5.4 NDN3M

El xip NDN3M (figura 5.12) és una derivació del xip NDN3, en la qual hem intentat reduir el número de pins fins a baixar a l'estandar d'encapsulat del CNM, 64 pins DIL.

Aquest propòsit s'aconsegueix introduint la memòria dels estats $M(O_i)$ dins del xip. Aquesta memòria té una capacitat baixa, 4 Kbit i ocupa una superfície de 3.5 x 2.1 mm^2. La introducció d'aquesta memòria no suposa un cost superior en temps, ja que el procés de càrrega és el mateix que si fos externa. L'avantatge principal d'aquesta estratègia és que permet augmentar el throughput del sistema, ja que s'aconsegueix la mateixa velocitat amb una reducció en el número de pins.

Aquest guany seria màxim si en comptes d'emmagatzemar els estats de les neurones, emmagatzemèssim els camps locals, ja que aquests valors són interns de l'algorisme i per tant no necessiten ser carregats exteriorment. Aquesta possibilitat no es pot contemplar en aquesta tecnologia i amb memòria estàtics, degut a que la superfície de silici ocupada és 16 cops superior a la implementada i per tant fora del rang de superfície disponible.

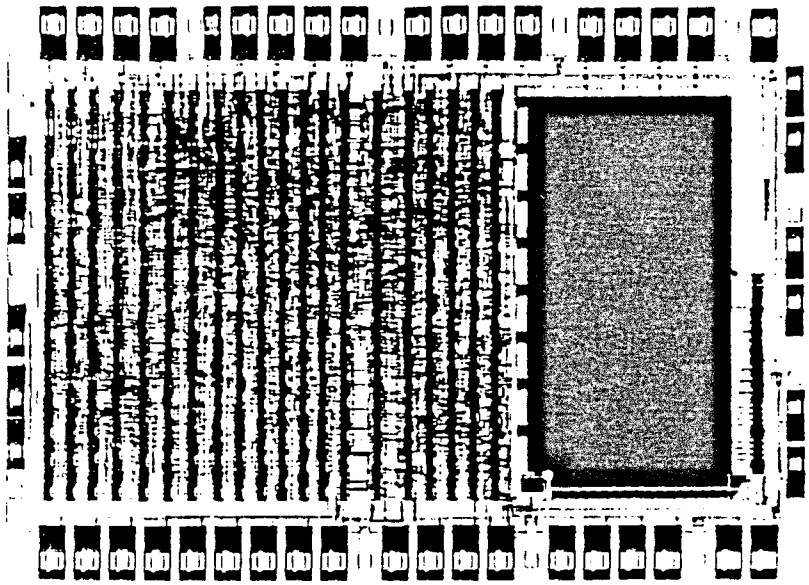


Figura 5.12. Fotografia del xip NDN3M.

La unitat de procés conté la memòria local a la qual s'accedeix mitjançant el registre C, que en el xip NDN3 anava a l'exterior directament.

Tant la unitat de control com els blocs de comunicacions i sincronisme, són els mateixos que per al NDN3.

El xip ha estat encapsulat en un estandard DIL de 64 pins. La dedicació dels pins del xip segueix el següent repartiment:

- 12 pins per alimentació (4) i terra (8).
- 4 pins per a rellotge i reset.
- 26 pins per a adreces de memòria (21), i els corresponents senyals selecció i escriptura (5).
- 16 pins per dades, dividits en dos busos de 8 bits.
- 1 pin d'interrupció des de la memòria de doble port.

El xip fabricat ocupa una superfície amb pads de 35.0 mm^2 (7.0 x 5.0 mm). Les condicions de simulació i test han estat les mateixen que per al NDN3. En aquest cas, el xip no va funcionar. La causa va ser que per la necessitat de reduir superfície (dels 6 x 8 mm^2 inicials a 5 x 7 mm^2) per tal d'encabir-lo dins el dau del

MPC, varem haver de realitar modificacions manuals sobre el layout: modificació de pins longitudinals per transversals, reducció de les dimensions de les línies d'alimentació, etc.

En algun d'aquests processos vam produir un error de desplaçament relatiu entre els dos metalls d'un contacte corresponent a una de les entrades de dades, (figura 5.13). Aquest error es detectava a nivell lògic en el test del circuit.

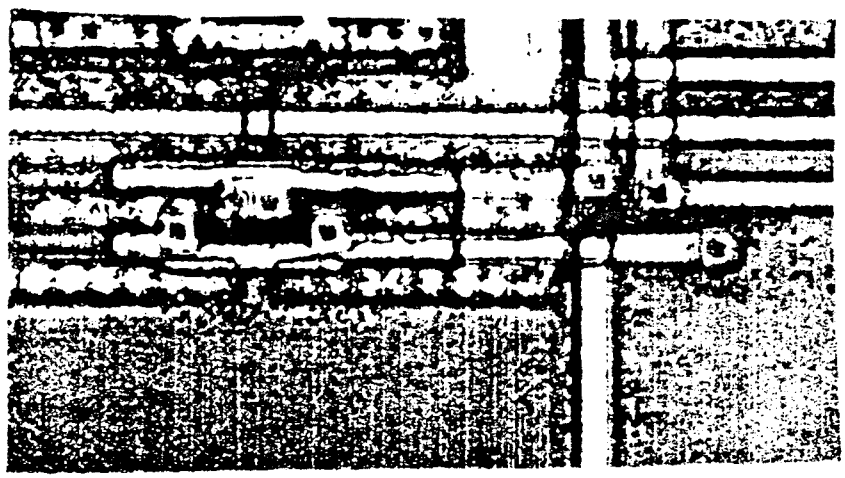


Figura 5.13. Detall de la falta física al xip NDN3M.

La conclusió més important sobre aquesta estratègia d'implementació, és la possibilitat d'incrementar de manera elevada el paral.lelisme del sistema. Amb el mateix número de pins que l'NDN3, els busos de dades podien haver estat de 16 bits, amb la qual cosa tindriem $\mathbf{m = 16}$ processadors funcionant en paral.lel.

En aquest cas doblariem la velocitat de relaxació de la xarxa per al càlcul del producte vector-matriu.

5.5 Placa de PC

Un cop dissenyats els xips, ens hem plantejat de realitzar el sistema digital complet. Aquest sistema està format per una placa de circuit imprés, connectable a un ordinador personal PC compatible.

La decisió de treballar amb el PC està basada en la disponibilitat d'obtenir informació de l'entorn PC: espais de direccionament de memòria, mapa d'assignació d'interrrupcions, format temporal dels cicles de lectura/escriptura, etc. així com en la versatilitat de les eines de desenvolupament de software de baix nivell amb eines d'alt nivell (Turbo C,...).

L'NDN3 és el xip al voltant del qual hem dissenyat la placa de PC. La part principal d'aquesta placa es composa de xips de memòria: l'associada als tres bancs de dades (pesos sinàptics, estats neuronals i camps locals) i la memòria de doble port, a més de lògica auxiliar per a amplificació i decodificació de senyals.

La memòria dels pesos ens ha imposat una primera restricció. Per tal d'emmagatzemar els pesos corresponents a 4096 neurones amb precisió {+1,0,-1}, calen 32 Mbit de memòria. Podem disposar d'aquesta memòria en xips de 1 Mbit SRAM (128k x 8), però tanmateix la superfície ocupada pels 32 xips és superior a la disponible a la placa estandard de conexió a un slop d'expansió del PC. Això ha fet que a la placa tinguem **un màxim de 2048 neurones totalment interconnectades**, que utilitzen 8 Mbit de memòria RAM estàtica.

La memòria per als camps locals i els estats és considerablement menor, i s'ha implementat per als primers en 2 xips de 8k x 8 i per al segon en un xip de 1k x 8.

La memòria de doble port utilitzada te una capacitat de 2k x 8 bits.

En la lògica adicional necessària, cal comptar amb els buffers unididireccionals ('244) per a les adreces del PC i alguns dels senyals de control, i els bidireccionals ('245) per a les dades. També és necessari decodificar l'adreça del PC, corresponent a la memòria de doble port, que se situa a les posicions reservades per expansió i aplicacions, adreces C0000 a DFFFF, i controlar l'accés a bus del PC de la memòria de doble port mitjançant els senyals de lectura-escriptura d'aquest. Els senyal d'interrupció és programable en el rang IRQ6-9.

A la figura 5.14 es pot veure una fotografia de la placa de circuit imprés fabricada amb els components muntats.

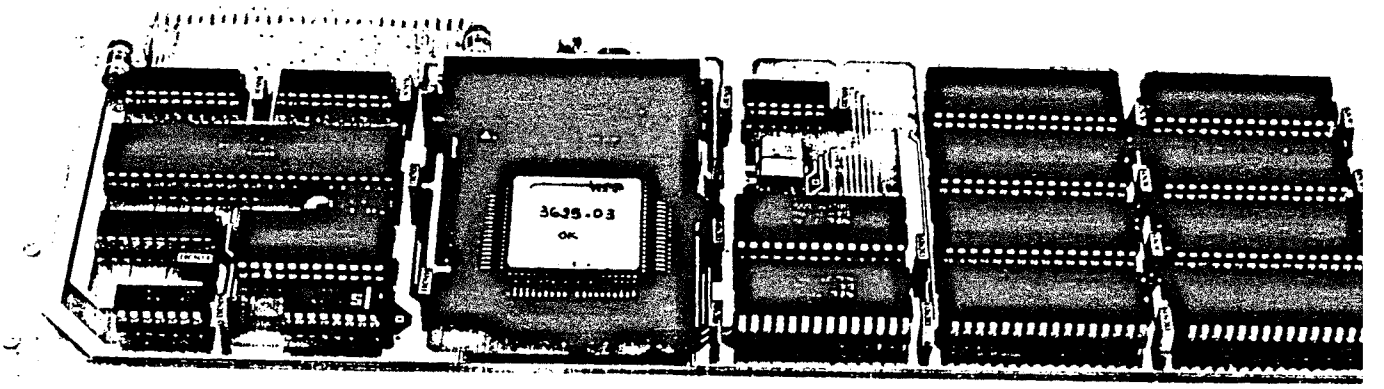


Figura 5.14. Fotografia de la placa de PC.

La velocitat de relaxació d'una xarxa d'Hopfield en placa la calculem considerant tot el procés de transferència de dades:

- Carregar l'estat nou de la xarxa.
- Deixar relaxar la xarxa fins a un estat estable.
- Recuperar el valor obtingut.

Aquest plantejament suposa que els pesos han estat carregats prèviament, i que són fixes en iteracions successives, per a la mateixa aplicació. Un canvi en els valors dels pesos, vindrà donat normalment per un nou procés d'aprenentatge amb uns temps de càlcul molt superiors a la transferència dels pesos.

Els temps que obtenim per a les operacions de transferència de dades per a 2048 bits (256 bytes) i 4 cicles de rellotge per operació de lectura/escriptura en un PC de 8 MHz, són:

$$\mathbf{t = 256 \text{ accessos} * 4 \text{ cicles/accés} * 125 \text{ ns/cicle} = 125 \text{ us}}$$

El temps de relaxaxió de la xarxa, si suposem que es realitzen 205 iteracions (10% número de neurones), segons l'expressió (5.5) és:

$$\mathbf{t = 80 \text{ ns/cicle} \ (\ 2048(2048/8+3) + 205*4*2048) = 176 \text{ ms.}}$$

Aquesta contribució és molt més elevada que l'anterior donada la quantitat d'operacions realitzades, i ofereix una mesura del temps de procés de la xarxa.

La dependència aproximadament lineal del número d'iteracions respecte de la distància de Hamming, fa que aquest número no sigui fixe i que varii en funció de les dades que introduim a la xarxa.

Amb aquest valor de temps es pot calcular la velocitat de procés de la xarxa en actualitzacions/segon. Aquesta magnitud, mesura el número de productes sinàptics $T_{ij}O_j$, equivalents als realitzats amb un producte vector matriu, processats per unitat de temps.

$$v = \frac{205\ (2048)^2 \text{ p.s.}}{0.176 \text{ seg.}} = 5\ 10^9 \text{ upd/sec.}$$

Aquesta velocitat situa el coprocessador NDN3 entre els valors de velocitat aconseguits per implementacions analògiques(10^{11}) i les digitals comercials (10^7).

5.5.1 Tipus i rang de les aplicacions

Les principals característiques que determinen el tipus d'aplicacions que pot soportar la versió actual del nostre coprocessador neuronal són:

- El màxim número de neurones: 2048.
- El temps de relaxació de la xarxa: <200 ms.
- El tipus de neurones: bivaluades.

En el procés de disseny del nostre coprocessador hem intentat que fos útil per a aplicacions en temps real. Els temps aconseguits al voltant dels 200 ms ens permeten pensar que aquesta és una estratègia correcta per a molts cassos.

Per a les aplicacions que no requereixen temps real, la qüestió bàsica passa a ser tant el número de neurones com la seva precisió (si les dades es poden representar amb un sol bit). Pensem que alguns tipus de problemes d'optimització i classificació relacionats amb valors lògics, poden formular-se d'aquesta forma.

Per a aplicacions relacionades amb imatges, el temps de relaxació del nostre coprocessador entra dins del que es considera temps de processat correctes (≈100ms), al mateix temps que la finestra màxima de que disposem, 45 x 45 pixels en blanc i negre, permet pensar en certes aplicacions d'identificació de figures simples (corbes, números,...).

Aplicacions del tipus processament de la veu i control de processos, requereixen uns temps de procés lleugerament menors que el donat pel nostre sistema (≈10ms), de manera que en aquest camp i en el seu estat actual, el coprocessador és útil per a aplicacions que no requereixen temps real. Per tal de fer aquest enllaç, cal estudiar en profunditat la codificació de les dades que permet el treball amb valors discretitzats.

5.5.2 Exemple d'aplicació: Reconeixedor de Caràcter Manuscrits.

L'aplicació que estem desenvolupant, un reconeixedor de caràcters manuscrits, s'ha escollit com un exemple d'aplicació de la nostra placa. No pretenem obtenir però, un reconeixedor de característiques òptimes, ja que per a això cal estudiar profundament els preprocessos gràfics necessaris i avaluar el que millor treballaria amb la nostra placa.

Aquesta aplicació es realitza sota unes certes restriccions, que limiten la complexitat del problema i el fan més resoluble amb mètodes simples. Aquestes restriccions són, per una banda el reconeixement de lletres majúscules i números i per l'altra, la utilització d'un formulari que ens estableix de manera rígida les posicions que ocuparan.

Aquest sistema és útil per aplicacions tals com l'automatització de l'entrada de dades de formularis de qualsevol tipus (administratius, adreces,...). El tipus de formulari utilitzat permet generar l'estructura de dades a alt nivell, adient a l'aplicació desitjada.

L'estructura de base sobre la qual treballem és la lletra. El procés d'obtenció de lletres es fa a partir d'un posicionament relatiu respecte d'unes marques previstes al formulari. Un cop identificada la posició de les lletres, s'extreuen i es realitza un escalat que fa que la lletra ocupi tota la matriu prevista per l'scanner.

Llavors, es fa un preprocés consistent en escombrar la lletra segons les quatre direccions de l'espai. Aquest procés permet transformar les característiques de contorn en característiques superficials, mentre que el fer-ho en les quatre direccions produeix que aquesta transformació sigui aproximadament independent de la seva posició relativa.

Els processos detallats fins ara es realitzen en software, en un entorn obert al qual es pot donar totes les característiques per fitxer (format del formulari, característiques de l'escombrat a l'scanner, lletres i números de l'alfabet que cal aprendre,...).

El que s'introdueix a la xarxa neuronal és el resultat d'aquest quatre escombrats. En la xarxa neuronal es realitzen dos processos: un primer procés de **recuperació**, treballant com a memòria autoassociativa i un segon procés de classificació, del qual se n'obté el codi corresponent a la lletra recuperada. Finalment, aquest codi es tranforma a codi ASCII i s'emmagatzema a l'estructura corresponent.

En aquest moments les dimensions de la finestra de treball són de 16x16 pixels, és a dir 400 bits, que després dels escombrats es converteixen en 1600 bits. La resta de bits fins a 2048 s'utilitzen per a l'obtenció del codi de sortida.

El processat de la xarxa neuronal és ràpid comparat amb els processos de lectura d'scanner i preprocés, tot i que el número de càlculs que es realitzen és elevat.

La realització dels escombrats en les quatre direccions dóna també uns patterns per als quals el número de 0's i el número de 1's són molt similars. En aquestes condicions, la capacitat d'emmagatzemament és màxima.

L'aprenentatge de la xarxa es realitza en aquesta primera versió amb la regla de Hebb. Aquesta regla amb pesos discretitzats per clipping, permet un emmagatzemament màxim de 0.1*1600 = 160 patters. Treballant amb lletres majúscules i números, tenim un alfabet molt més reduït, 38 símbols, que ens possibilita introduir més d'una representació per símbol, en algún d'ells, i treballar encara molt per sota del límit teòric, a α=0.05, la qual cosa millora les característiques del procés de recuperació.

5.6 Comparació amb altres estratègies

Les tres estratègies bàsiques respecte de les quals comparem el nostre sistema són: realitzacions analògiques, realitzacions digitals sistòliques i neurocomputadors.

Els paràmetres respecte dels quals realitzem la mesura són: número de neurones, temps de relaxació i precisió dels pesos.

5.6.1 Realitzacions analògiques.

Hem vist en el capítol 2 que les realitzacions analògiques de xarxes neuronals programables amb pesos discrets, tenen un número de neurones limitat: 256, utilitzant tecnologia sofisticada (0.9 micres). Aquest límit no pot crèixer en ordres de magnitud, ja que les realitzacions analògiques no són expandibles de manera directa.

Respecte d'això, la nostra xarxa presenta un número de neurones i sinapsis per neurona molt més elevat, al mateix temps que pot crèixer en funció de l'increment de memòria externa a utilitzar.

A més, les implementacions analògiques utilitzen l'estratègia transparent, de manera que existeixen físicament les interconnexions entre les neurones. L'utilització d'una estratègia específica dins del nostre sistema, pot ajudar a reduir

la demanda de memòria, o el què és equivalent, augmentar el número de neurones per a certes xarxes.

En termes de velocitat de procés, els valors corresponents a les realitzacions analògiques, 10^{11}-10^{12} actualitzacions per segon, són encara més potents que les digitals, entre elles la nostra, encara que les propietats de la dinàmica assíncrona associada són de menor qualitat.

5.8.2 Realitzacions digitals sistòliques

En el camp digital, l'estratègia alternativa que sembla més potent és la estratègia de realització de productes vector-matriu síncrons de manera sistòlica.

La comparació la realitzem respecte del processador equivalent en termes de superfície i rang dels pesos.

Així, un processador sistòlic per a pesos d'un bit, treballant amb **N neurones**, necessita de **2N-1 processadors** per realitzar un producte vector-matriu en **4N-1 passos** [Mead86].

La cel.la del processador bàsic associat al sistòlic a nivell lògic es pot veure a la figura 5.15. Aquesta, consta d'un registre de 12 bits per als camps locals (resultat de producte vector-matriu) i un de 1 bit per l'estat. Ambdós van circulant a l'estructura sistòlica a la qual entren els pesos. Com a lògica combinacional, són necessaris una XOR per al càlcul del terme $T_{ij}O_j$ i un incrementador/decrementador del valor del camp local.

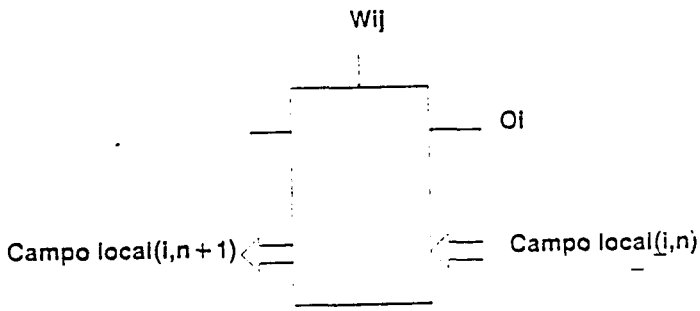


Figura 5.15.- Cel.la bàsica d'un multiplicador sistòlic d'un bit a nivell lògic.

Una estimació sobre les dimensions de la cel.la, com si fos realitzada en full-custom, a partir de les dimensions de diferents cel.les de llibreria (també realitzades en full-custom), ens porta a 0.25 mm^2 per processador. Si considerem la mateixa superfície que l'NDN3, **20 mm^2** sense la superficie ocupada pels pads i suposant que al voltant d'una quarta part de la superfície serà ocupada per la part de control, comunicacions i sincronisme igualment necessaris, obtenim una superfície neta pel processador sistòlic de 15 mm^2, en el qual hi tenen cabuda **60 processadors** d'un bit.

Amb 60 processadors, podem realitzar el càlcul en submatrius de 30 x 30 sinapsis, de les quals n'hi ha 68 x 68 en una xarxa de 2040 neurones totalment interconnectades.

Llavors, el temps total per iteració d'una xarxa de 2040 neurones amb un rellotge de 12 MHz és:

$$t = (4*30-1) \text{ cicles/subm} * 68^2 \text{ subm} * 80 \text{ ns/cicle} = 44 \text{ ms.}$$

Aquest temps caldrà multiplicar-lo pel número d'iteracions necessaris per relaxar la xarxa. En el cas equivalent a l'estimació realitzada anterioment, 205 iteracions (la qual no és coherent amb el que es mostra a la figura 2.5 (c) presentada al capítol 2, el temps total de relaxació de la xarxa seria de:

$$t = 205 \text{ it} * 44 \text{ ms/it} = 9 \text{ seg.}$$

Aquest valor dóna una velocitat de relaxació de 10^7 actualitzacions per segon, molt per sobre de la velocitat obtinguda amb l'NDN3, i reflexa el fet que la complexitat dels càlculs en aquest cas és quadràtica respecte del número d'iteracions.

5.8.3 Neurocomputadors

Respecte dels neurocomputadors, les comparacions són més difícils de realitzar, ja que aquests treballen normalment amb precisions elevades per als pesos (típicament amb valors reals), necessàries per algunes aplicacions, i amb

quantitats de neurones més elevades encara que amb connectivitat reduïda. Com a contrapartida, les velocitats de relaxació solen ser més elevades.

El quadre IV mostra una llista dels numeros màxims de sinapsis i neurones per a diferents neurocomputadors comercials. Es pot observar que, mentre que el número de neurones és elevat, el número de sinapsis no és quadràtic respecte d'aquest. La xarxa totalment interconnectada que és possible realitzar amb aquests neurocomputador queda molt lluny d'aquest número de neurones [Trele89].

COMPANY	NEUROCOMPUTER product	CAPACITY virtual PEs	CAPACITY interconnections	SPEED training	SPEED recall
Hecht-Nielsen Neurocomputer	ANZA ANZA Plus	30K 1M	480K 1.5M	25K 1.5M	45K 6M
Human Devices	Parallon 2 Parallon 2X	10K 91K	52K 300K	15K 15K	30K 30K
Science Applicat. Intl Corp.	SIGMA	1M	1M	2M	11M
Texas Instruments	ODYSSEY	8K	250K	2M	-
TRW	Mark III (multiple boards) MARK IV	8K 65K 236K	400K 1.13M 5.5M	300K 500K 5M	- -

Quadre III. Valors màxims de neurones i sinapsis dels neurocomputadors comercials.

La comparació respecte de la velocitat és clarament favorable a la nostra estratègia que inclou el processament a nivell de bit, i un cert grau de paral.lelisme respecte d'operacions més lentes de processament de números reals (sumes, productes, funcions de transferència no linials,..). A la figura 5.16 es mostra la situació dels diferents processadors estudiats sobre uns eixos que relacionen el número de neurones implementat i la velocitat de relaxació [Ramac90].

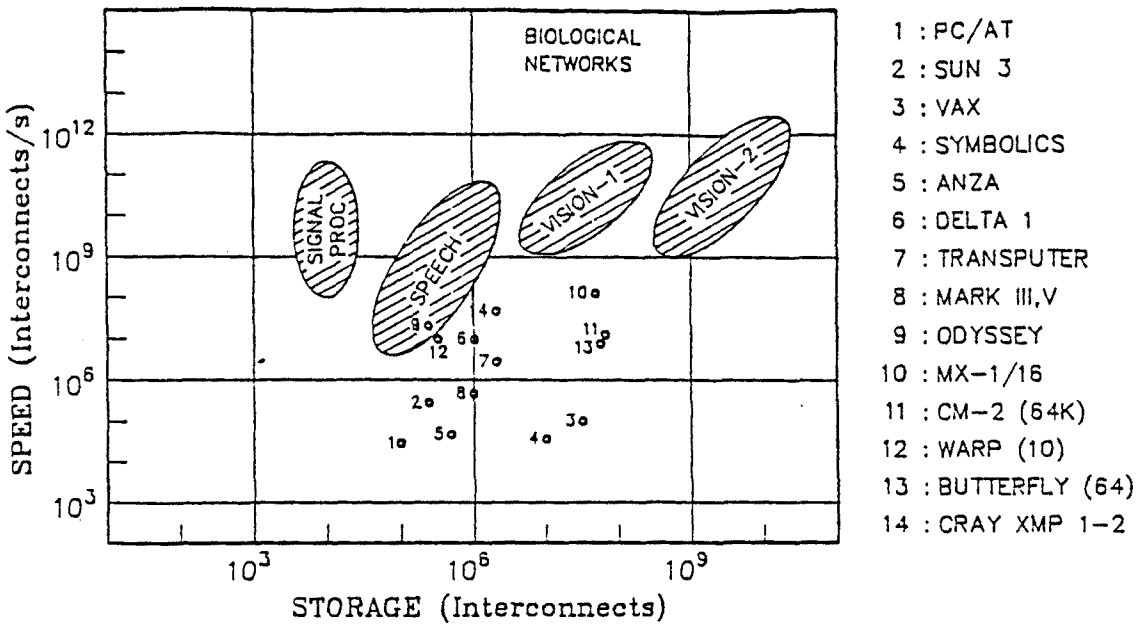


Figura 5.16.- Situació dels diferents processadors estudiats respecte d'una gràfica de velocitat de relaxació respecte de número de neurones.

Capítol 6

AMPLIACIONS FUTURES

El treball que es presenta en aquesta tesis doctoral, compleix un objectiu concret: **la formulació matemàtica, el disseny d'algorismes i les realitzacions a nivell de xips VLSI i de sistema (PCB, connexió a PC) de xarxes neuronals programables d'alta velocitat.** El desenvolupament realitzat en el treball ha estat presentat en els capítols anteriors (fonamentalment als capítols 3, 4 i 5).

De cara al futur s'obre una quantitat elevada de camps a diferents nivells d'abstracció sobre els quals fer millores. L'objectiu d'aquestes millores serà bàsicament el mateix: incrementar la velocitat i la capacitat de processament. Segons les anàlisis realitzades, aquestes millores es podrien aconseguir de diverses formes: augmentant el número de pins del circuit, arrivant a un compromís entre l'estratègia transparent i l'especial, integrant una major quantitat de memòria amb tecnologies de més alta escala d'integració, utilitzant arquitectures paral.leles.

Aquesta aproximació es complementaria amb millores a nivell d'entorn, com ara permetre thresholds per a les neurones com a grau de llibertat adicional que no ocupa una quantitat elevada de memòria i la implementació d'algorismes d'aprenentatge eficients i ràpids, la qual cosa voldrà dir amb hardware específic, bàsicament per al procés de conversió de pesos continus a bipolars (veure apèndix 1). En la implementació d'algorismes d'aprenetatge amb pesos continus difícilment superariem les velocitats dels neurocomputadors.

El punt de referència d'aquestes millores ha d'estar donat pel rang d'aplicacions, intentant anar cap al camp industrial.

El que presentarem en aquest apartat final és un esbós de les idees sobre les quals tenim pensat de treballar en un futur.

6.1. Aplicacions

Les dues línies que ens proposem són el processament d'imatges i el processament de la veu, ja que en aquest camps les aplicacions en temps real són encara poques, sent com són camps difícils de tractar amb arquitectures clàssiques.

En ambdós cassos pretenem fer un tractament digital de la informació i treballar per tant amb xarxes neuronals programables.

6.1.1 Processament d'imatges

El processat d'imatges ha estat sempre un dels camps previstos d'aplicacio de les xarxes neuronals programables. Exemples com el reconeixement de formes simples, com ara contorns, textures o caràcters manuscrits i de formes més complexes com són imatges de cares, etc. han estat utilitzats per a la demostració de les potencialitats de determinades xarxes neuronals.

Els requeriments d'aquestes aplicacions són diversos en termes de representació de les dades (p.e. blanc i negre, escales de grisos o color), número de neurones (funció del número de pixels i de la porció d'imatge manipulada), velocitat de procés (processos en temps real), funcionalitat (memòria associativa, reconeixement de patrons, classificació), tipus d'aprenentatge (inicial o periòdic), etc.

Per al sistema neuronal que hem dissenyat i en el seu estat actual, el conjunt d'aplicacions a realitzar és limitat per les restriccions que hem escollit. Tot i això, presentem dues aplicacions que en un futur proper preveiem desenvolupar.

6.1.1.1 Sistema per al reconeixement de caràcters òptics.

En el capítol 5 hem esbossat les característiques que ens permeten de pensar en un sistema d'aquest estil: pixels en blanc i negre, array de pixels de dimensions limitades per cada lletra, elevada velocitat de procés. Un sistema d'aquest estil pot aplicar-se per exemple a l'informatització de documents com ara els corresponents a dades de matriculació d'alumnes a la universitat.

El sistema complet constaria d'un preprocés extern per software i una estructura de xarxa multinivell, amb una primera xarxa de discriminació global de característiques diferencials d'una lletra i una segona d'identificació i obtenció del codi ASCII corresponent.

En el cas d'un entorn comercial, el treball adicional està relacionat amb el software de preprocés (obtenció del layout de la pàgina, estructures de dades de sortida, etc.) i de facilitat d'ús a l'usuari (menus, help, etc.), mentre que en l'obtenció de característiques de les lletres hem de millorar els processos de discriminació, dels quals n'esperem aconseguir també una compactació de la informació.

6.1.1.2 Reconeixement de Traces

La detecció de característiques bàsiques a alta velocitat és interessant per a un cert nombre d'aplicacions. En particular, la proposada per en Vicens Gaitan del Laboratori d'Altes Energies que es troba al campus de la nostra Universitat.

El problema plantejat és la classificació de traces de càrregues ionitzades provinents d'un accelerador de partícules. L'objectiu principal és realitzar un preprocés que elimini aquelles traces que no siguin interessants. Això permetria reduir de forma important el conjunt de traces a examinar amb el corresponent estalvi en recursos associats (memòria d'emmagatzemament, temps de processat, facilitat en l'estudi dels resultats).

El mètode proposat [Garri91] associa als valors sinàptics relacions topològiques, de manera que les estructures d'excitació-inhibició estan directament relacionades amb la forma d'aquestes corbes (figura 6.1). Els resultats mostren que les simulacions amb xarxes neuronals són de qualitat similar als obtinguts per mètodes clàssics, de manera que l'acceleració dels càlculs donaria avantatge clar al mètodes neuronals.

L'estudi de la influència de la discretització, la velocitat de funcionament i el número de neurones necessari per al problema concret seran factors a estudiar per comprovar la utilitat del nostre sistema en aquest camp.

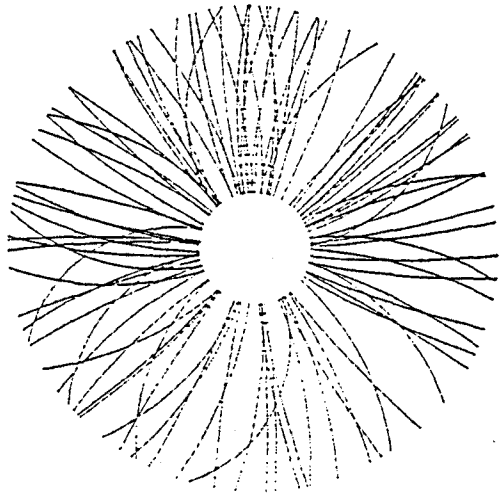


Figura 6.1. Esquema de les traces obtingudes en un detector.

Una altra possible aplicació del nostre sistema, és la detecció i filtrat d'altres corbes de les quals s'en coneix les característiques bàsiques. Aquest tractament pot ser aplicat per exemple a la determinació de paràmetres sobre corbes d'oscil.loscop, tals com la caracterització de la zona subllindar del transistor MOS (figura 6.2) per tal de discriminar la zona de la corba sobre la qual es realitzen els càlculs dels paràmetres associats al model corresponent.

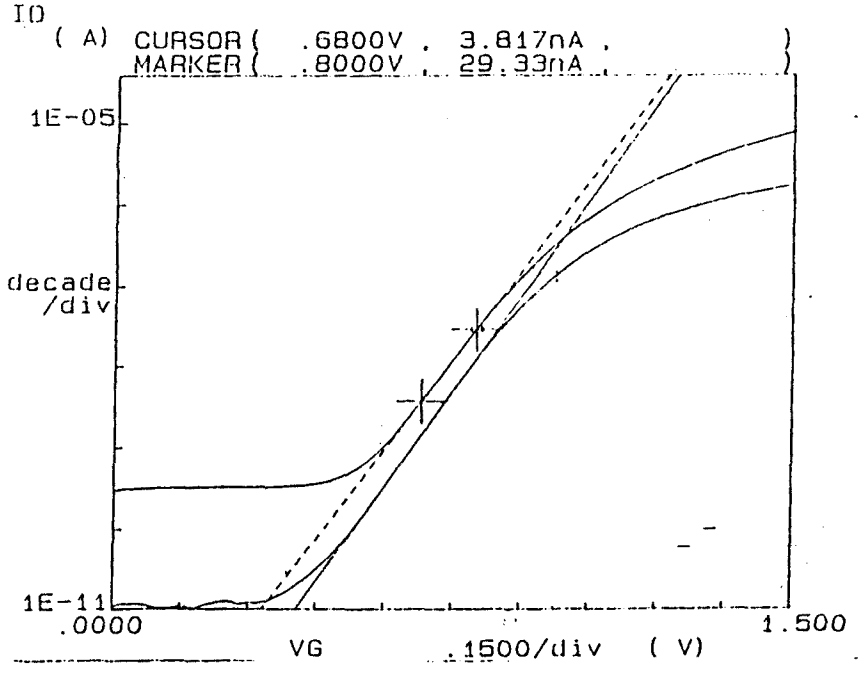


Figura 6.2. Fotografia de la característica subllindar d'un transistor MOS.

6.1.2 Processament de la veu

A diferència del processament d'imatges, l'aplicació del nostre sistema al camp del processament de la veu és molt menys desenvolupada. Tot i això, companys nostres estan treballant en la realització de models coclears [Avell91] que han de permetre emular la funcionalitat de la còclea. Aquests models estan implementats amb filtres digitals i treballen amb números de precisió elevada. La interfase amb el nostre sistema, passa per trobar les codificacions que converteixin el tipus de dades tractades de valors reals a seqüències de pulsos, de manera similar a com es realitza pels ganglis auditius o com en Carver Mead realitza la conversió de sortida del seu xip coclear.

Un cop disposem de la sortida codificada en binari, el següent pas és el reconeixement de seqüències temporals. Per aquesta finalitat es poden utilitzar xarxes neuronals de tipus Elmann o Jordan que poden ser autosincronitzades en funció dels estimuls d'entrada, de manera que en poguem obtenir informació de tipus halofon, lletra o sil.laba. Des d'aquest punt de vista descomposicions freqüencials o paramètriques també podrien ser tractades per la xarxa.

Aquest esquema de primar la redundància sobre la compactació sembla, en principi, més natural des del punt de vista de les xarxes neuronals animals, encara que s'oposa a la teoria clàssica basada en la capacitat d'emmagatzemament limitada i l'ocupació mínima de les línies de transmissió.

6.2. Arquitectura

A nivell d'arquitectura el fet més remarcable és la possibilitat de posar els xips en paral.lel, de manera que hi hagui una assignació de totes les sinapsis, estats i camps locals associats a un conjunt restringit de neurones al mateix xip. Aquesta estratègia genera una arquitectura equivalent a l'arquitectura típica utilitzada en xarxes neuronals digitals (figura 6.3) [Trele89].

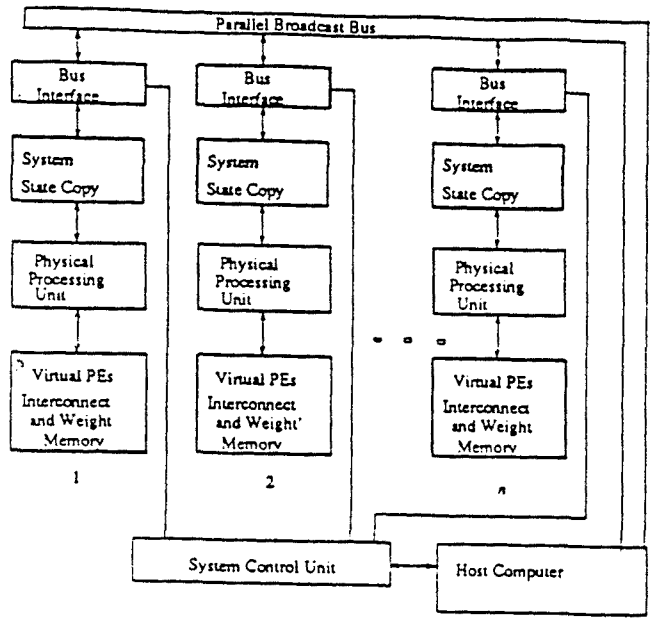


Figura 6.3. Arquitectura d'una xarxa neuronal digital programable.

Les modificacions que caldria fer sobre aquesta arquitectura estan relacionades amb la funcionalitat del nostre algorisme i el distinguirien d'altres sistemes en:

(i) L'estat que entrem a la xarxa és el mateix per a tots els xips que processen en paral.lel durant la primera fase de l'algorisme de relaxació -càlcul del camp local-, i cada xip emmagatzema en memòria local l'estat d'aquelles neurones a les quals està associat.

(ii) L'elecció de la neurona que canvia, és realitzada per un arbitre que recull les màximes inestabilitats de cada xip i retorna l'estat i l'identificador d'aquesta neurona per a l'actualització dels valors dels camps locals a cada xip.

(iii) En l'obtenció de resultats, cada xip treu els corresponents a les neurones que gestiona de forma ordenada.

Aquestes modificacions produeixen un guany en velocitat d'un factor aproximadament igual al número de xips que posem en paral.lel, sense que la complexitat adicional associada a cada xip sigui excessivament elevada. L'arquitectura resultant té l'estructura mostrada a la figura 6.4.

Figura 6.4. Arquitectura paral.lela per al xip NDN4.

D'altres millores realitzables a nivell d'arquitectura són:

- La utilització de **memòria RAM dinàmica** amb la qual cosa augmentarem la capacitat d'emmagatzemament de sinapsis amb el mateix nombre de xips i per tant el mateix espai físic.

- La connexió a un bus de dades de més amplada, encara que hem vist que els temps d'entrada i sortida dels vectors d'estat de la xarxa són baixos en comparació amb els temps de relaxació.

6.3. Xips

Les millores realitzables a nivell de xip estan relacionades amb la voluntat d'augmentar la velocitat de procés, amb l'evolució de la tecnologia i l'ampliació de les llibreries associades.

Per augmentar la velocitat de procés tenim diferents possibilitats. Tal i com hem comentat al apartat 5.1.5, la velocitat de procés del càlcul del camp local (producte vector matriu) depén de forma important del throughput i, per tant, **augmentar el número de pins dedicats al bus de dades.** Una part insignificant dels recursos de càlcul permet augmentar el número de sinapsis processades en paral.lel (m), i reduïr el primer terme de l'expressió (5.5), que és la part independent de l'estat introduït a la xarxa.

A banda d'augmentar el número de pins del circuit, podem utilitzar altres estratègies com la **integració dins el xip d'una part de la memòria externa.** Per això tenim dues alternatives: **la integració dels camps locals,** o **la integració dels estats de les neurones.**

La integració dels camps locals, permet augmentar la velocitat de procés de la segona part de l'algorisme ja que guanyem un cicle de rellotge per iteració en el procés de pipeline. Per això cal que poguem llegir i escriure posicions consecutives de memòria per la qual cosa es fa necessària una memòria de doble port. La capacitat d'aquesta memòria ve donada pel número de neurones que volguem per a la xarxa. Per a **n** neurones, calen $\mathbf{n*(\log_2 n+\log_2(\log_2 n))}$ bits de memòria, que en la tecnologia 1.2 micres de ES2 per a **4096 neurones** implica una superfície de **1.2 cm^2**. Aquest nivell d'ocupació fa que aquesta opció pel moment sigui inviable.

La integració del valor dels estats i l'increment dels busos de sinapsis a 16 bits, permet augmentar també la velocitat de procés del càlcul dels camps locals en un factor 2. La capacitat de la memòria per a **n neurones** és de **n bits**, que en la mateixa tecnologia ocupen una superfície de **3 mm^2**, perfectament assequible dins les dimensions típiques dels circuits dissenyats.

Des d'un altre punt de vista, es pot igualment augmentar de manera important la velocitat d'execució del primer pas de l'algorisme: **permetent un conjunt restringit d'estratègies específiques**, per tal de no processar matrius específiques, com ara matrius de zeros, la matriu identitat, etc.

Les reduccions que es podrien assolir amb aquesta estratègia cal avaluar-les en funció de l'algorisme realitzat, de manera similar a com s'ha fet al capítol 5 amb l'avaluació sobre una xarxa backpropagation i una xarxa de Kohonen de les quals se n'obté les expressions (5.2) i (5.3).

En termes d'implementació, això significa introduir a la unitat de control **un nivell de microprogramació** mitjançant el qual, en funció dels algorismes suportats, es realitza el corresponent seqüenciament, amb un compromís entre la complexitat de la unitat de control i la programabilitat des del host.

6.4. Algorismes

Els desenvolupaments a realitzar en els algorismes són diversos. Les línies bàsiques en aquest sentit estan apuntades en l'apèndix 1, i fan referència a l'obtenció de pesos discrets per a xarxes de les quals en tenim pesos continus, que poden ser obtinguts en altres entorns hardware de més alta velocitat o amb processadors específics.

Assolir un nivell de generalitat suficient en els algorismes de càlcul, ens portaria a una implementació VLSI d'aquests algorismes, ja que en aquests moments, els temps d'aprenentatge associats, per a 2048 neurones, són molt elevats.

Des de l'estat actual del xip, una millora que ens sembla important és **la introducció de valors llindars** per a cada neurona. La despesa en memòria que això representa és reduïda, **n*log2n bits**, que són fàcilment introduïbles en els mateixos
xips de memòria que ocupen els camps locals. Amb la introducció dels llindars aconseguiriem funcionalitats associades a determinades xarxes neuronals (p.e. Hamming, ART, manipulació de funcions booleanes, etc.). La penalització temporal que introdueix el càlcul del camp local només afecta el primer pas de l'algorisme aproximadament com **m/n (<< 1)**.

Per al tractament de funcions booleanes i de certes funcionalitats sembla també interessant poder disposar de **dos tipus de neurones: binaries (1/0) i bipolars (+1/-1)**, que per a determinades funcionalitats (classificació, tranctament de funcions booleanes, etc.) puguin coexistir ambdues representacions en un mateix conjunt de neurones, si tenim en compte que aquest conjunt pot contenir neurones de diferents layers de la xarxa neuronal. Aquesta introducció implica al seu torn doblar la quantitat de memòria associada als estats, **n bits** més (fàcilment integrables dins del xip), mentre que augmenta molt poc la complexitat dels recursos de processament de dades.

Com es pot veure, el rang de variacions que es pot fer és elevat i pensem que en qualsevol cas hauria de venir marcat per les característiques de velocitat,

número de neurones, etc. necessàries per a portar a terme una aplicació determinada.

CONCLUSIONS

Abans de resumir breument el treball desenvolupat en aquesta tesi doctoral recordarem els objectius que ens havíem proposat. L'objectiu bàsic era la realització de xarxes neuronals d'alta velocitat i gran número de neurones. L'alta velocitat està encarada al processament en temps real de la informació tot i que aquest temps real depèn fortament de l'aplicació a la qual la vulguem destinar, mentre que l'elevat número de neurones és necessari donades les propietats col.lectives de les xarxes neuronals. El requeriment de la programabilitat, que hem introduït posteriorment dóna a les nostres realitzacions una flexibilitat important a nivell de sistema.

Al primer capítol fem una introducció als conceptes bàsics associats a les xarxes neuronals. Expliquem com es modelitzen les neurones, com realitzar xarxes mitjançant diferents models d'interconnexió de neurones, i quines són les característiques computacionals associades. Introduïm els algorismes corresponents a algunes de les regles d'aprenentatge més importants. Ens hem fixat especialment en la importància de la topologia de la xarxa per les seves repercusions a l'hora de processar la informació. Mostrem que a part de les funcionalitats complexes associades a les xarxes neuronals també es possible realitzar la síntesi de circuits combinacionals i seqüencials.

En el segon capítol presentem un seguit de realitzacions microelectròniques de xarxes neuronals. En particular, fem un especial esment d'aquelles que introdueixen metodologies noves al VLSI: les xarxes neuronals de preprocés, i les xarxes neuronals programables. Presentem dos circuits realitzats, un conversor analògico-digital de 4 bits que segueix la formulació de Hopfield, i una memòria associativa bidireccional. La resolució del problema del conversor ens presenta la qüestió, fonamental en aquesta tesi, dels problemes associats a la dinàmica de les xarxes. La realització de la BAM ens permet estudiar els límits de les realitzacions

de xarxes neuronals totalment interconnectades analògiques, en el cas de que tots els elements de procés siguin integrats, i en el cas en que es permet una certa multiplexació dels recursos, realitzable amb BAMs. El reduït límit, per que fa a número de neurones ens porta a la realització de xarxes neuronals digitals.

El tercer capítol està dedicat a l'estudi de la dinàmica de relaxació de les xarxes neuronals des d'un punt de vista teòric. Aquest estudi es fonamental per a la realització de xarxes neuronals programables digitals. Estudiem les implicacions de la dinàmica de la xarxa en les característiques qualitatives d'aquesta i fem una comparació de quatre dinàmiques de relaxació diferents: una dinàmica paral.lela i tres dinàmiques seqüencials, que anomenem de criteris aleatori, analític i probabilístic. La dinàmica seqüencial de criteri probabilístic és la que dóna uns millors resultats pel que fa a qualitat de recuperació i la velocitat de relaxació de la xarxa. Aquests paràmetres són fonamentals quan el número de neurones és elevat.

En el quart capítol, desenvolupem els algorismes per al disseny de circuits digitals que implementin la dinàmica seqüencial amb criteri probabilístic tenint en compte d'una banda, les condicions donades pel disseny VLSI, utilitzant una estratègia basada en llibreries de cel.les i mòduls de llibreria estandar, i de l'altra, la restricció de treballar amb pesos discrets per tal de aconseguir un número màxim de neurones amb una quantitat de memòria fixa, i una elevada velocitat de recuperació.

En el capítol cinquè presentem els xips que hem dissenyat, així com les dues estratègies que hem seguit que anomenem, estatègia trasparent i estratègia especial, i que fan referència a la manera de processar les matrius totalment interconnectades, igual per a totes elles o preprocessant certes matrius característiques (zero, identitat, etc.), respectivament. Es presenten els xips NDN2, NDN3, i NDN3M, les seves característiques i les avaluacions de superfície i velocitat de procés. Es mostra també les implicacions a nivell de sistema digital i es mostra la placa de circuit imprès realitzada per la connexió del sistema basat en el xip NDN3 a un PC, sobre el qual estem desenvolupant una aplicació de reconeixement de caràcter òptics. Finalment, es realitza una comparació amb altres estratègies de disseny de sistemes neuronals, realitzacions analògiques,

realitzacions digitals sistòliques i neurocomputadors, que resulta ser satisfactòria pels resultats obtinguts del compromís escollit entre número de neurones totalment interconnectades, 2048, representació dels pesos, {+1,0,-1}, i velocitat de recuperació, 5 10^3 MC/seg.

Finalment, es proposen al capítol sisè possibles estratègies futures d'ampliació i millora d'aquest sistema a diferents nivells: aplicació a sistemes de tractament d'imatge i de veu, introducció de paral.lelisme a nivell d'arquitectura, alternatives de disseny de xips aprofitant les noves eines i tecnologies, i la necessària integració de les regles d'aprenentatge.