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.

Apèndix 1

# REGLES D'APRENENTATGE PER A PESOS DISCRETS

La realització de xarxes neuronals de baixa precisió en els pesos, com el sistema que nosaltres presentem, només té sentit si al darrera hi ha uns algorismes d'aprenentatge capaços de donar pesos discrets per a funcionalitats i xarxes específiques.

Es sabut però que les característiques de les xarxes es degraden quan es perd precisió en els pesos. En aquest apèndix presentarem els estudis realitzats a nivell teòric sobre la capacitat d'emmagatzemament de xarxes totalment realimentades. Posteriorment, es presenten els mètodes utilitzats per arrivar a aquest límit i finalment, s'introdueix un mètode desenvolupat al nostre grup de treball que és extensible a altres tipus de xarxes i funcionalitats.

### A1.1. Capacitat d'emmagatzemament en xarxes de tipus Hopfield

La capacitat d'emmagatzemament $\alpha$ d'una xarxa totalment realimentada realitzant funcions de memòria associativa, es defineix com el número de patrons **p** emmagatzemats a la xarxa dividit pel número de neurones **N**.

Estudis teòrics realitzats per E. Gardner [] demostren que la capacitat màxima que es capaç d'assolir una xarxa neuronal totalment interconnectada amb neurones bipolars (+1/-1) és

$$\alpha_c = 2 \qquad \text{(A1.1)}$$

Aquest valor correspon al cas en que els dominis d'atracció, formats pel conjunt de vectors que dónen com a resultat un dels emmagatzemats, contè únicament el propi patró, essent per tant, una situació funcional poc desitjable.

Aquest límit correspon a una situació per a la qual la precisó dels pesos és infinita, encara que no es dóna la regla d'aprenentatge per la qual és possible obtenir aquests pesos.

Diversos algorismes han intentat arrivar a aquest límit: l'algorisme Minover [Kraut87] és capaç de trobar una solució d'estabilitat òptima encara que

convergeix molt lentament cap a la solució, l'algorisme de Keppler i Abbott [Abbot90] i altres (Edinburgh, Diedrich-Opper, etc.).

L'inconvenient d'aquests sistemes és d'una banda, l'elevada quantitat de memòria necessària per guardar una determinada configuració de pesos i que és, en molts cassos l'element crític en la implementació de xarxes neuronals programables, i de l'altra, la seva elevada complexitat que fa que siguin útils únicament per a xarxes neuronals anb un número de neurones reduït.

### A1.2. Estudis teòrics sobre pesos discrets

La quantitat de memòria necessària per a emmagatzemar les sinapsis marca el límit superior a l'hora d'implementar sistemes neuronals. Es per això que hem buscat un compromís entre la qualitat de recuperació i la capacitat d'emmagatzemament (número de neurones i sinapsis).

El criteri de decisió ha estat basat en estudis sobre el límit teòric associat a les diferents representacions dels pesos de la xarxa. Krauth i Mezard [Kraut89a], han trobat que els límits de les capacitats d'emmagatzemament en funció de la representació utilitzada per als pesos segueixen els valors de la taula I.

| Representació | $\{\pm 1\}$ | $\{\pm 1,0\}$ | $\{\pm 1,\pm 1/2\}$ | $\{\pm 1,\pm 1/2,0\}$ |
|---|---|---|---|---|
| Capacitat | 0.832 | 1.174 | 1.331 | 1.447 |

Taula I. Valors límit de la capacitat d'emmagatzemament en funció de la representació utilitzada per als pesos.

En aquesta taula s'observa que el guany que obtenim d'utilitzar una representació de dos bits en lloc d'un, és baix comparat amb el cost necessari en termes de memòria (es dobla) quan el número de neurones és elevat.

Aquest criteri ha estat bàsic en la decisió de treballar amb sinapsis de baixa precisió i tot i els resultats mostrats a la taula I, hem introduït dos bits per sinapsi, el primer per a la determinació de sinapsi excitadora/inhibidora {+1,-1}, i el segon que indica l'existència o no de sinapsi {1,0}. Com hem mostrat en capítols precedents, aquesta estratègia ens permet generar no només la topologia de xarxa totalment interconnectada sinó qualsevol altra topologia o conjunt de topologies.

### A1.3. Regles d'aprenentatge per a l'obtenció de pesos discrets

Els estudis realitzats sobre emmagatzemament de patrons en xarxes totalment interconnectades (memòria autoassociativa), utilitzen el concepte de condició d'estabilitat, definit com,

$$\Delta_i^\mu - \xi_i^\mu \sum_j T_{ij} \xi_j^\mu \geq K \sqrt{\sum_j T_{ij}^2} \quad \forall i,\mu \qquad \text{(A1.2)}$$

on el paràmetre d'estabilitat **K** està directament relacionat amb la dimensió dels dominis d'atracció de les memòries.

Ambós paràmetres, estabilitat i capacitat màxima, són els paràmetres sobre els quals es mesura el comportament dels algorismes d'aprenentatge per a pesos discrets.

Els algorismes d'aprenentatge realitzats fins al moment segueixen dues estratègies bàsiques: la **búsqueda de solucions a l'espai discret** i la **transformació de solucions de l'espai continu al discret**. La generalització a qualsevol número de neurones i temps de càlcul d'aquests algorismes, així com la possibilitat d'utilitzar recursos d'acceleració dels càlculs per a l'obtenció de les configuracions de sinapsis, són els paràmetres que definiran l'elecció de l'algorisme a utilitzar.

#### A1.3.1 Regles d'aprenentage sobre l'espai discret

L'estratègia bàsica a l'espai discret ha estat l'establiment de mètodes exhaustius o d'optimització per a la busqueda de les solucions als vèrtexs de l'hipercub, avaluant per a cada pas l'estabilitat i elegint el vèrtex d'estabilitat

màxima. Donat que aquest espai no és continu, molts dels mètodes d'optimització no són aplicables (p.e. gradient descent).

Els algorismes més representatius presentats són:

- Algorisme de Krauth i Opper [Kraut89b]

Analitzant l'estabilitat de cadascuna de les $2^N$ configuracions possibles han trobat un valor de capacitat crítica $\alpha_c = 0.82$ per a sistemes petits ($N<25$) amb una estabilitat $K = 0$. Aquest algorisme esdevé intractable en el nostre cas ($N = 2048$ neurones).

- Algorisme de Kohler [Kohle90].

En aquest cas l'aproximació es basa en minimitzar la funció de cost següent,

$$f = \sum_{i,\mu} (\Delta_i^\mu - k\sqrt{\sum_j T_{ij}^2})^2 \qquad \text{(A1.3)}$$

on $\Delta_i^\mu$ ve donat per l'expressió (2). Donat que no treballa en l'espai continu, no pot utilitzar tècniques del tipus descens per màxim gradient. En aquest cas s'utilitza un gradient seqüencial tal que, una sinapsi canvia de signe si l'estabilitat resultant fa baixar la funció energia. Aquest procés no dóna resultats òptims i el límit trobat està a $\alpha_c = 0.4$.

- Algorisme d'Amaldi i Nicolis [Amald89]

Utilitzant el "tabu search" per a sistemes més grans, han establert una capacitat en el rang $0.6 < \alpha_c < 0.9$, per a $N < 81$ neurones. El mètode "tabú search" es basa en ordenar els vectors sobre els quals estudiem l'estabilitat de manera que el pas entre dos dels vectors estudiats es variable: gran per a estabilitats baixes i menor per a estabilitats prop de la crítica. Tot i que no és tant lent com l'anterior, la seva complexitat el fa també intractable quan el número de neurones és elevat.

- Algorisme de Verleysen [Verle89].

Aquest algorisme ha estat elaborat des d'un punt de vista més proper a la realització de circuits VLSI. L'equació (2) es planteja com a funció d'optimització i s'utilitza un mètode de resolució de sistemes lineals (mètode de simplex) per tal de resoldre el sistema amb la restricció de que la solució sigui dins l'hipercub. Amb aquestes restriccions la configuració del conjunt de sinapsis trobada sol ser molt propera als vèrtex de l'hipercub. En aquest cas la realització d'un clipping (discretització) dels valors en funció de la representació (ell utilitza {+1,0,-1}) li permet arrivar a capacitats $\alpha_c = 0.3$. L'avantatge principal d'aquest mètode ve donat per la separabilitat dels càlculs. Aquests no s'han de realitzar a nivell de les sinapsis de tota la xarxa, sinó que es poden aplicar a les sinapsis associades a l'entrada d'una col.lumna. Llavors la quantitat de càlculs associada a l'optimització és menor i, per tant, susceptible de ser utilitzat per a números de neurones elevats.

### A1.3.2 Transformacions de l'espai continu al discret

Una estratègia d'aquest estil presenta com a principal avantatge, el fet que els algorismes d'aprenentatge a l'espai continu han estat optimitzats en funció de la topologia i la funcionalitat de la xarxa (per exemple backpropagation). Per a la majoria d'aquests algorismes d'aprenentatge existeixen acceleradors de simulació, via software o hardware (neurocomputadors), que permeten l'obtenció més o ménys ràpida del conjunt de pesos per a xarxes amb números de neurones elevats. En canvi, la relaxació de les xarxes amb aquestes mateixes eines no és tan ràpida, tal i com s'ha mostrat al capítol 5, com amb la nostra realització.

#### A1.3.2.1. Clipping de la solució contínua

Tradicionalment s'ha utilitzat el clipping o truncament de la solució contínua com l'eina per obtenir el conjunt de sinapsis discretes equivalent. Aquest procés, realitzat per a pesos amb representació bivaluada {+1,-1} o trivaluada

{+1,0,-1}, dóna uns resultats de qualitat inferior als obtinguts per models continus ja que el clipping no dóna la configuració discreta més propera a la contínua, en termes de que els resultats siguin similars, sinò que aquesta pot estar molt allunyada. Com a exemple, el clipping aplicat a la regla d'aprenentatge de Hebbs, redueix la capacitat crítica de $\alpha_c = 0.14$ fins a $\alpha_c = 0.10$.

L'algorisme Minover realitza també un clipping. A partir d'una solució millor, aconsegueix una capacitat límit de $\alpha_c = 0.3$, també força lluny del límit teòric per la raó esmentada.

### A1.3.2.2 Mètodes de transformació complexos

La diferència entre els resultats obtinguts per la solució continua i la solució discreta equivalent, ve donada per la simplicitat del mètode de transformació utilitzat. Al nostre grup han estat desenvolupades metodologies [Perez90,1] que permeten una transformació millor de la solució contínua a la solució discreta.

La base d'aquesta transformació és la formulació del problema com a funció d'optimització contínua sobre la qual hi podem aplicar mètodes d'optimització del tipus descens pel màxim gradient.

Per a l'obtenció del conjunt de sinapsis en l'espai continu, utilitzem la regla d'aprenentage que ens dóna una millor correspondència amb la funcionalitat o topologia escollida. Aquest solució la prenem com a punt de partida en procés d'optimització.

La formulació del problema com a funció d'optimització, parteix del "Interior Penalty Method", que permet la transformació d'un problema de minimització amb restriccions, en una seqüència de processos de minimització sense restriccions.

La funció a minimitzar en cadascun d'aquests passos ve donat per la funció

$$\phi(\bar{X},r) = E(\bar{X}) - rf(\bar{X}) \qquad \text{(A1.4)}$$

en la qual **X** és el conjunt de valors a optimitzar, **r** és el paràmetre de penalti, **E(X)** és la funció a optimitzar i **f(X)** la funció que conté informació sobre les restriccions.

Si la funció $\phi(X,r)$ és minimitzada per a una seqüència decreixent de valors de r, la solució final convergeix a la solució del problema original amb restriccions.

La idea de base d'aquest mètode és que, per a valors alts del paràmetre de penalti la solució ha de ser molt similar a la solució contínua de partida, mentre que en anul.lar aquest paràmetre haurem aconseguit una solució discreta. L'avaluació de l'una a l'altra, es realitza decreixent el paràmetre r de manera que es van obtenin configuracions molt properes entre elles.

La funció a optimitzar es defineix de forma que els seus mínims coincideixin amb els vèrtexs de l'hipercub, equació (5), mentre que les restriccions que han de complir les solucions es poden formular com una derivació de la mateixa regla d'aprenentatge, que tambe han de minimitzar les solucions discretes.

$$E(T_{ij}) = \sum_{i,j} (T_{ij}^2-1)^2 \qquad \text{(A1.5)}$$

La convergència d'aquest mètode necessita de la convexitat dins la regió de l'hipercub d'aquesta funció, encara que en funció de la seqüència de valors escollits per al paràmetre de penalti poden aparèixer problemes si la configuració inicial és molt propera als límits de l'hipercub.

La funció de restricció és pot formular de diferentes maneres. Les que estudiem fan referència als cassos de memòria autoassociativa i de funció de correspondència realitzada amb una xarxa feed-forward.

Per al cas de **memòria autoassociativa**, desitjem fer màxims els dominis d'atracció. Això es reflexa en la formulació de la restricció com

$$f(T_{ij}) = \sum_{i,\mu} \frac{1}{K - \xi_i^\mu \sum_j T_{ij} \xi_j^\mu} \qquad \text{(A1.6)}$$

La convexitat estricta del terme - **r** $f(T_{ij})$ de l'equació (4), és igualment necessària per a la convergència del mètode d'optimització. Durant el procés iteratiu, no es permet violar cap restricció, ja que les fronteres es comporten com a barreres que asseguren que totes les configuracions generades són dins el domini esperat. Aquesta condició fa que el denominador de l'equació (6) sigui sempre negatiu, però el terme és positiu degut al factor **-r**. Com a funció, equival a la branca positiva d'una hipèrbola, el hessià de la qual és una funció convexa de $T_{ij}$.

En aquest cas, l'algorisme pot utilitzar-se per a representacions multivaluades dels pesos, i així mateix l'execució de l'algorisme pot descomposar-se, aplicant-lo al conjunt de sinapsis associat a cada neurona. Aquest darrer punt torna a ser important, tant per la reducció de la quantitat d'informació manipulada, com per la reducció de la complexitat de la funció d'optimització amb la corresponent reducció del temps de càlcul.

La figura A1.1 mostra l'estabilitat de les solucions trobades amb l'algorisme d'aprenentatge respecte de la capacitat de la xarxa quan no es permet autointeracció. La figura A1.2 mostra la capacitat d'ammagatzemament crítica respecte de l'invers del número de neurones, que permet estimar que per $N \to \infty$ la capacitat crítica pren un valor finit.

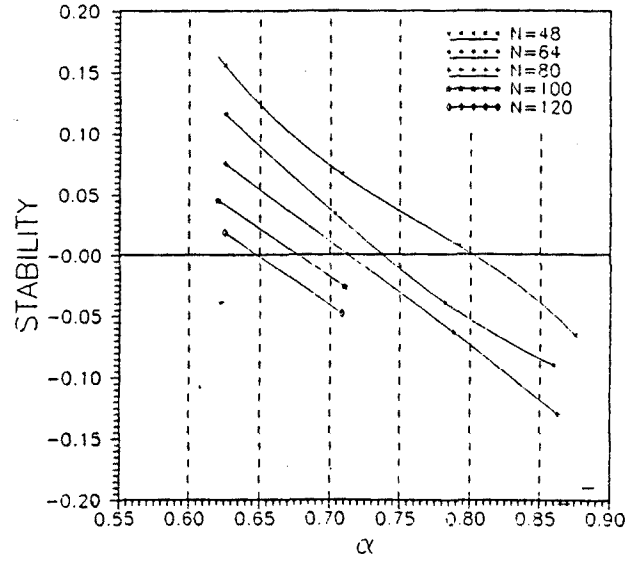



Figura A1.1. Estabilitat de les configuracions $T_{ij}$ trobades pre l'algorisme respecte de la capacitat d'emmagatzemament.

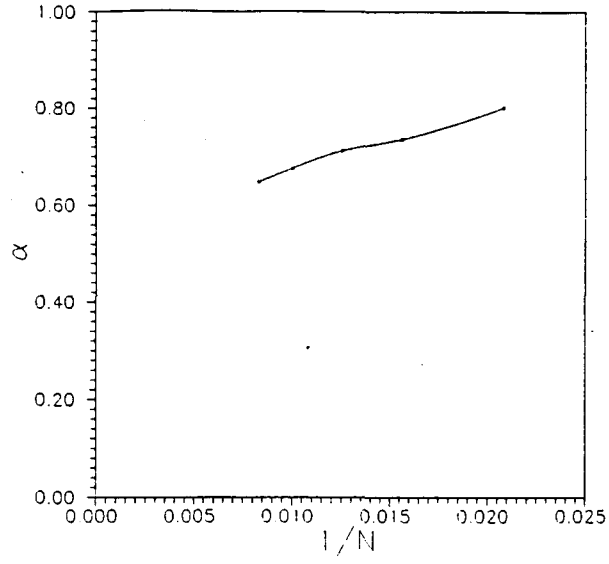



Figura A1.2. Capacitat d'emmagatzemament crítica respecte de 1/N.

En el cas d'utilitzar **xarxes feed-forward** en les quals hi ha una correlació entrada-sortida mitjançant una representació interna, la funció que conté les restriccións canvia ja que es busquen funcionalitats diferents a la de memòria associativa, com per exemple classificació o reconeixement. En aquest cas, es preten que l'aprenentatge minimitzi la diferència entre el pattern de sortida esperat $T_i^\mu$ i l'obtingut $O_i^\mu$. L'equació es pot formular com,

$$\phi(T_{ij},r) = \sum_{i,j} (T_{ij}^2 - 1)^2 + \frac{1}{\sqrt{r}} \sum_{i,\mu} (O_i^\mu - T_i^\mu)^2 \qquad \text{(A1.7)}$$

Aquesta funció té una topologia complexa que no és globalment convexa si l'arquitectura de la xarxa inclou unitats ocultes. Malgrat això, sabem que certs algorismes com ara el backpropagation tenen el mateix tipus de problemes, però en la practica el seu rendiment és suficientment correcte en un rang d'aplicacions ample.

Aquesta nova formulació pot ser utilitzada també per a representacions multivaluades dels pesos. En aquest cas, no és possible treballar a nivell de

sinapsis associades a una neurona ja que l'estructura multicapa fa que les relacions entrada-sortida corresponents a diferents neurones de la capa externa puguin afectar una neurona de la capa interna.

La taula II mostra el percentatge de vegades en les que s'ha trobat solució al problema de la XOR per a pesos continus, per al cas desenvolupat i per una variació d'aquest cas en la qual es permeten pesos {+1,0,1}. En aquest darrer cas, l'eficiència augmenta ja que es permeten més graus de llibertat i les derivades de la funció cost són sensibles a aquest nou valor, de manera que canvia el camí seguit per trobar la solució final.

| *Algorisme* | *Solució* |
|---|---|
| *Continu* | 70.7%±4.5% |
| $T_{ij}=\pm 1$ | 56%±4.3% |
| $T_{ij}=0,\pm 1$ | 77.6%±4.7% |

(A1.8)

Taula II. Percentatge de vegades en les que l'algorisme d'aprenentatge (continu, optimització amb {+1,-1} i amb {+1,0,-1}) ha trobat solució al problema de la XOR.

## Apèndix 2

# CEL.LES VLSI

En aquest apartat, pretenem mostrar les característiques més importants de cel.les bàsiques VLSI que hem dissenyat, bé sigui directament a nivell de transistor i layout, amb el corresponent procés de caracterització (disseny full-custom), o hem utilitzat per a la síntesi dels esquemes dels circuits a nivell lògic, a partir de cel.les de llibreria, el layout i caracterització de les quals ha estat prèviament realitzat (disseny semicustom).

En aquest sentit, a l'hora de fer les avaluacions dels resultats obtinguts és molt important diferenciar aquestes dues estratègies. A nivell de concepció, en ambdues estratègies es treballa a nivell d'esquema, però en full-custom cal generar el layout manualment (per la qual cosa s'aconsegueix un nivell de compactació més elevat), mentre que en semicustom la generació és automàtica. A nivell de simulació de l'esquema, s'observen també diferències importants ja que els esquemes fan referència a nivells d'abstracció diferents, més elevats per a semi-custom de tal forma que permeten realitzar simulacions mes complexes encara que amb menor precisió. També les eines de simulació utilitzades són diferents en ambdós cassos.

### A2.1. Memòria Associativa Bidireccional

El xip que implementa la memòria associativa bidireccional té les característiques de una realització full-custom digital típica: número de dispositius elevat (40.000 transistors), modularitat, aprofitament de les simetries per a la connexió lateral de les cel.les, densitat del connexionat, etc.

El xip consta d'una matriu de (25x25) cel.les de sinapsi i de (25+25) cel.les de neurona, amb els corresponents punts de memòria.

Les dues cel.les bàsiques que hem dissenyat són: la cel.la de la sinapsi i la cel.la de la neurona que inclou la pròpia neurona i el registre de desplaçament que utilitzem per a l'entrada-sortida de dades.

Les simulacions de les cel.les han estat realitzades en SPICE, i el mòdul més gran simulat ha estat una neurona amb el conjunt de sinapsis associat a la seva entrada, que conformen un dispositiu amb un total de 254 transistors.

Per tal d'evitar la dependència del procés tecnològic sobre els paràmetres dels transistors, hem introduït dues fonts de tensió independents, una per a l'alimentació de les cel.les de les sinapsis i una altra per la de les neurones. Amb aquesta darrera, podem controlar el nivell de tensió llindar de la neurona i corregir possibles funcionaments incorrectes.

### A2.1.1 Sinapsi

La sinapsi inclou dues cel.les de memòria RAM estàtica, un multiplexor per a la selecció de la direccionalitat de la matriu i una porta XOR per a l'algebra neuronal.

Esquema i layout de la sinapsi es mostren a la figura A2.1. La dimensió de la cel.la és de 135x117 $\mu m^2$.

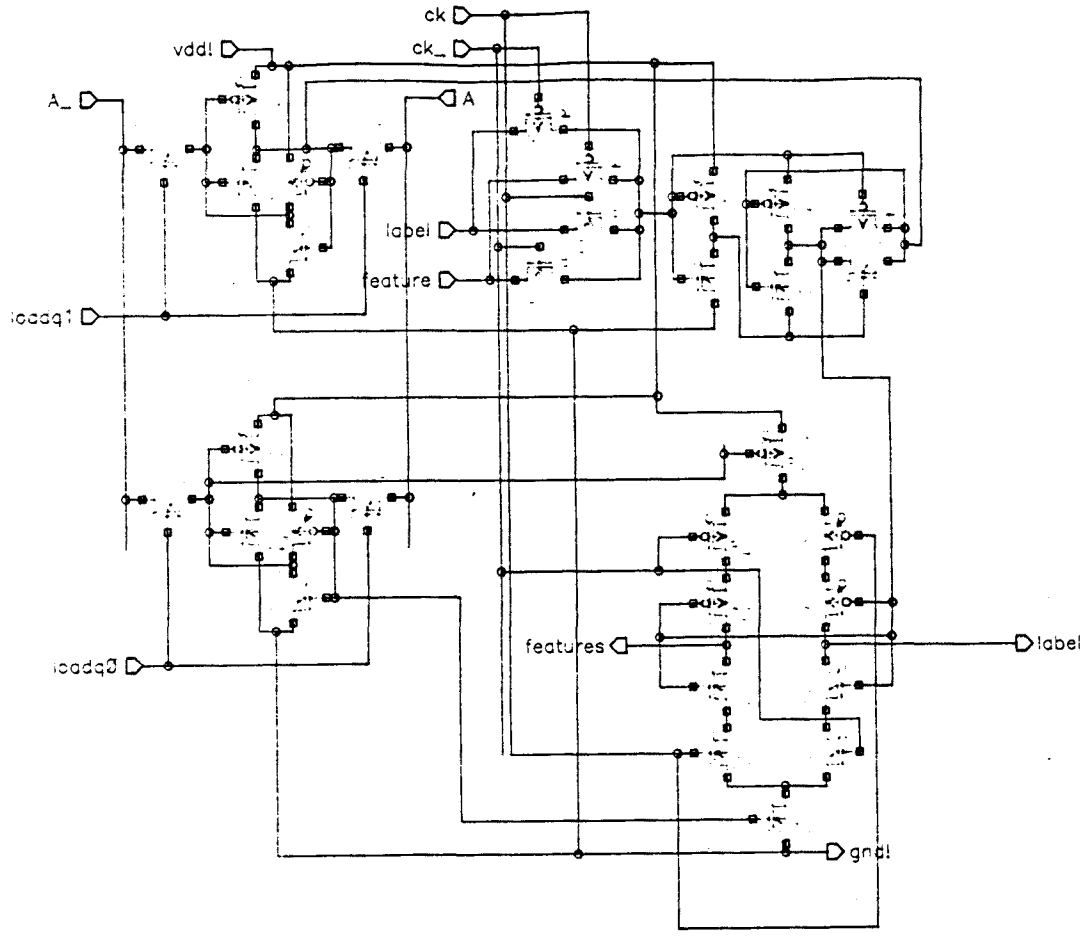
Els transistors de la cel.la que realitzen la funció de resistència sinàptica són els més propers a la sortida, pels quals podem assegurar que treballaran en zona lineal. Han estat dissenyats amb una longitud molt més gran que els altres de la cel.la, als quals estan connectars en sèrie, per tal d'augmentar la resistència respecte dels altres transistors.

## 1.2 Neurona

El dispositiu que fa la funcionalitat de neurona és un buffer composat de dos inversors en sèrie. Cadascun d'aquests inversors ha estat dissenyat amb una tensió llindar de $V_{DD}/2$, essent VDD la tensió d'alimentació dels inversors.

La mateixa cel.la, conté també els registres de desplaçament per l'entrada-sortida de dades, que va serawissenyada amb una estructura sèrie per tal de reduir el número de pads d'entrada-sortida.

L'esquema i el layout de la cel.la es mostren a la figura A2.2.



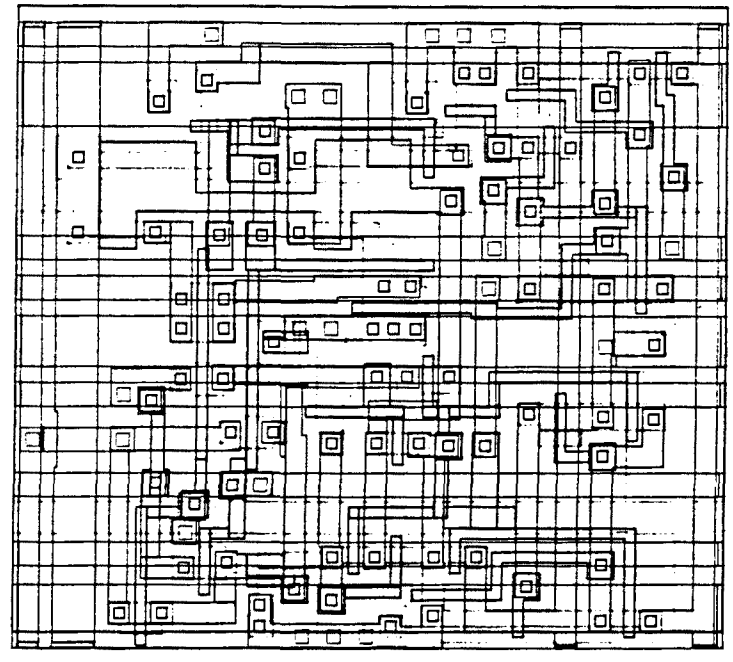
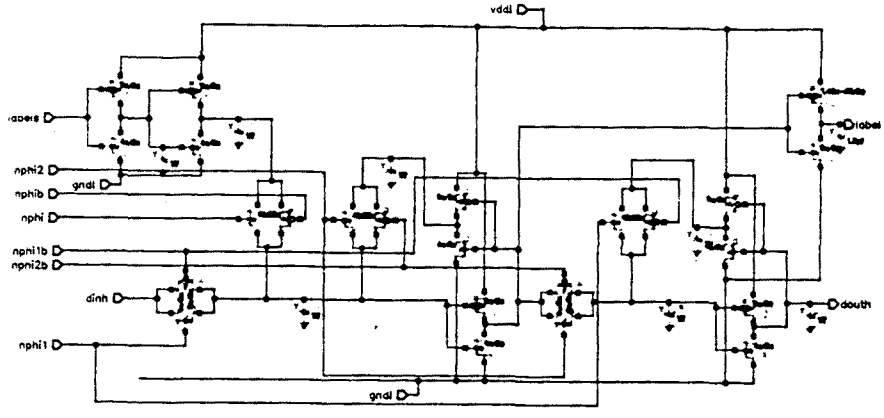



Figura A2.1. Esquema i layout de la cel.la de sinapsi dissenyada per a la BAM.



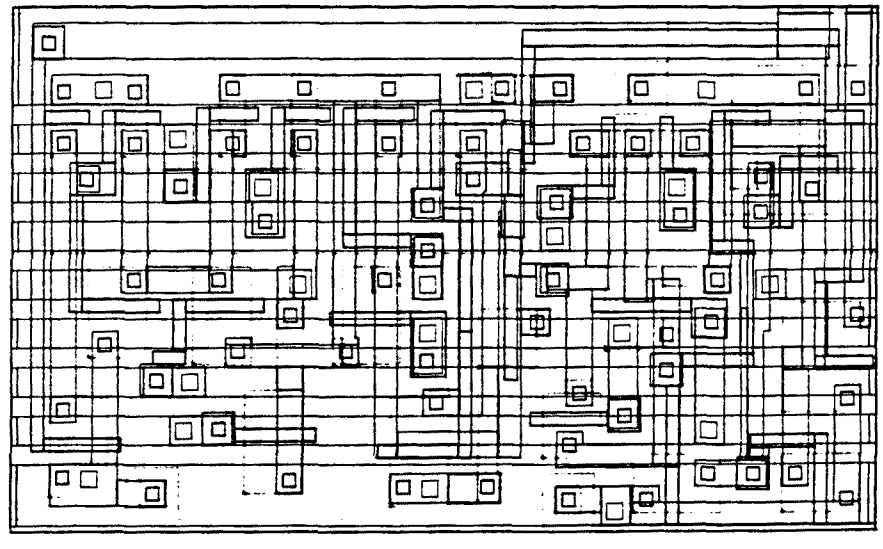



Figura A2.2. Esquema i layout de la cel.la de neurona i registre de desplaçament dissenyaws per a la BAM.

## A2.2. Conversor Analògico-Digital

La realització del conversor correspon també a un disseny full-custom, encara que la característica determinant en aquest cas és el seu funcionament analògic i no pas la necessitat d'una alta densitat d'integració. Per aquesta raó, no hem optimitzat la superfície de silici utilitzada en el disseny de les dues realitzacions del xip.

Aquestes dues realitzacions es diferencien en l'element utilitzat com a resistència, en un cas transistors MOS i en l'altre línies de polisilici.

### A2.2.1 Resistències MOS

La realització amb transistors utilitza la resistència de canal del MOS per a la sintesi del circuit. Aquest presenta l'estructura bidimensional típica de les xarxes neuronals amb realimentació, amb el multiplexor de control de fase del conversor i les sinapsis atacades per cada neurona (que en cada fila ténen del mateix valor de resistència). L'estructura es reflexa tant en la topologia del pla de base (figura A2.3) com al layout final (figura A2.4). Els valors de resistència implementats per transistors MOS estan dins el rang de valors entre 6k25 i 50 kohm.

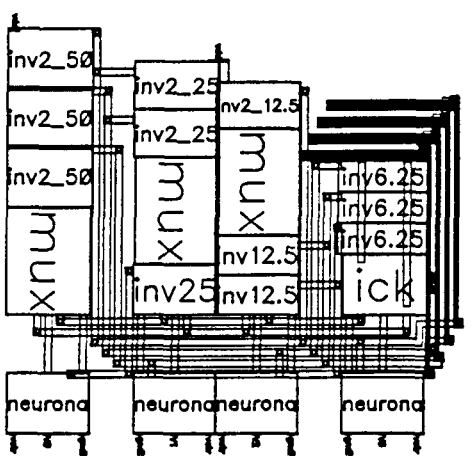



Figura A2.3. Pla de base de la realització amb transistors del conversor A/D.

Les sinapsis de connexió amb l'entrada externa analògica, no poden ser implementades amb resistències MOS, ja que els transistors tenen una **característica inversora no lineal** que no segueix la compleix la formulació a nivell de funcions energia. La utilització simultània dels dos tipus de resistències provoca que la dependència respecte del procés tecnològic sigui més important, ja que no depén només dels processos d'implantació sinó també de les característiques del polisilici.

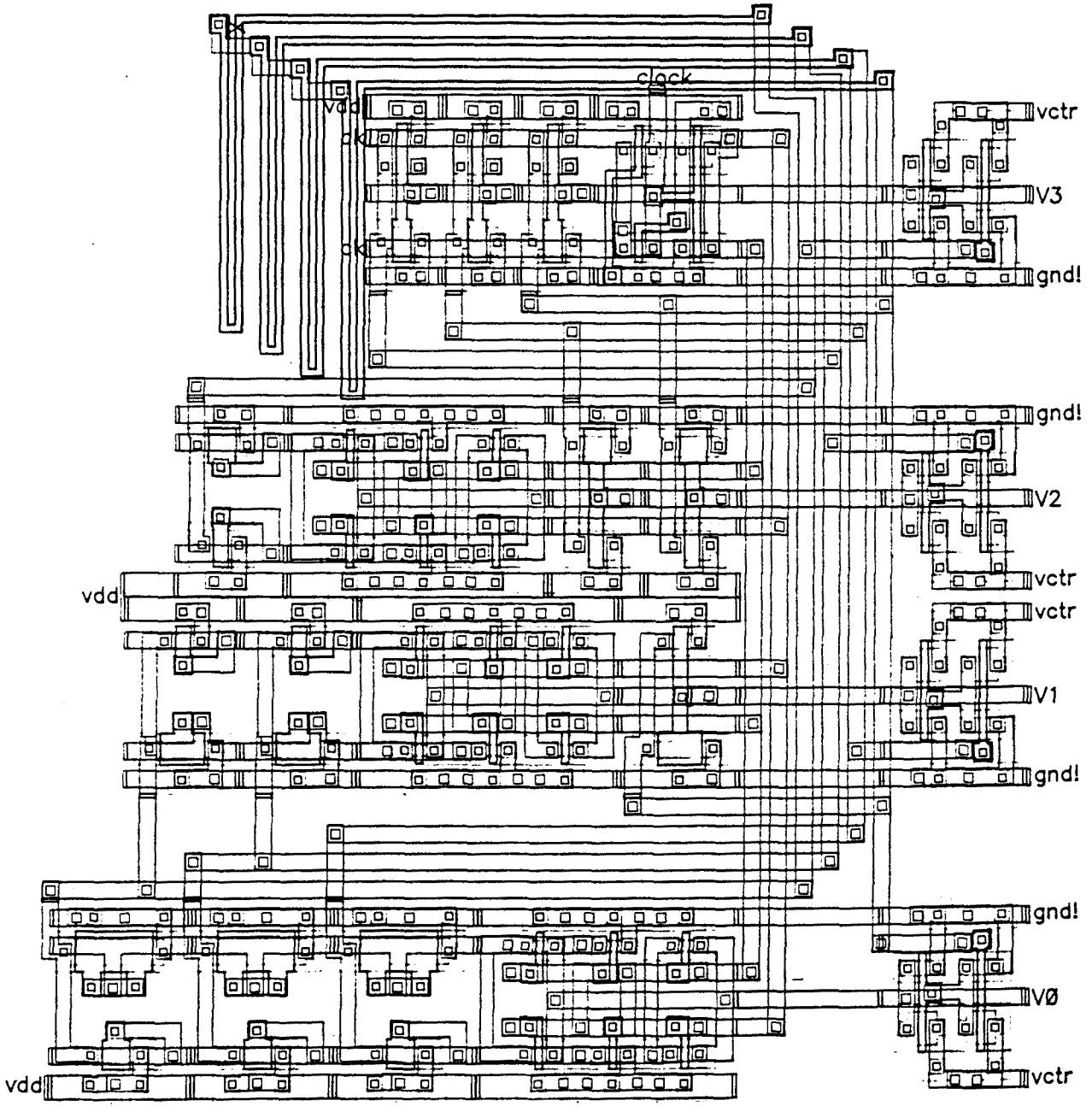



Figura A2.4. Layout de la realització amb transistors del conversor A/D.

### A2.2.2 Resistències de polisilici

En el cas de que totes les sinapsis del conversor siguin realitzades amb línies de polisilici, l'estructura del pla de base del xip canvia completament (figura A2.5): totes les resistències es disposen longitudinalment en la dimensió major del xip, la qual s'obté de la suma de les amplades dels pads necessaris al circuit, i creixent en la dimensió horitzontal per obtenir el valor de resistència desitjat.

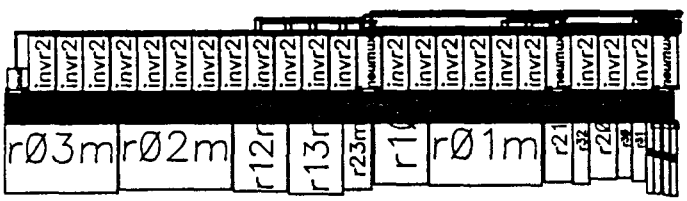



Figura A2.5. Pla de base de la realització amb resistències de polisilici del conversor A/D.

El rang de valors de resistències va de 12k5 a 100K, i és doble que en el cas de resistències MOS per tal de minimitzar el problema d'acoblament d'impedàncies respecte de les sortides de les neurones. Els buffers de sortida de les neurones han estat dissenyats respectant la relació entre impedància de sortida i impedància de les sinapsis, i per tant són més grans per les neurones que ataquen resistències menors (del bit 3 al bit 1).

Un cop assignat l'espai en sentit longitudinal corresponent a cada resistencia, en funció dels valors a implementar (multiples de la resistència bàsica de 6k corresponent a les sinapsis de connexió amb l'entrada externa), es creix en la dimensió transversal. Al layout (figura A2.6) es pot observar que aquesta dimensió es pràcticament igual per a totes les resistències de polisilici, mentre l'efecte dels angles a les línies es menyspreable perquè el número de curbatures de les línies de polisicili es proporcional al valor de resistència implementat.

Com a conseqüència, aquesta realització té una dependència menor respecte de variacions del procés tecnològic, a costa de consumir una superfície major. La caracterització de resistències s'ha realitzat amb un extractor, considerant únicament els efectes de primer ordre (resistències per quadre); altres contribucions de menor importància (resistències de contacte, efectes d'angles, etc.) han estat menypreades.

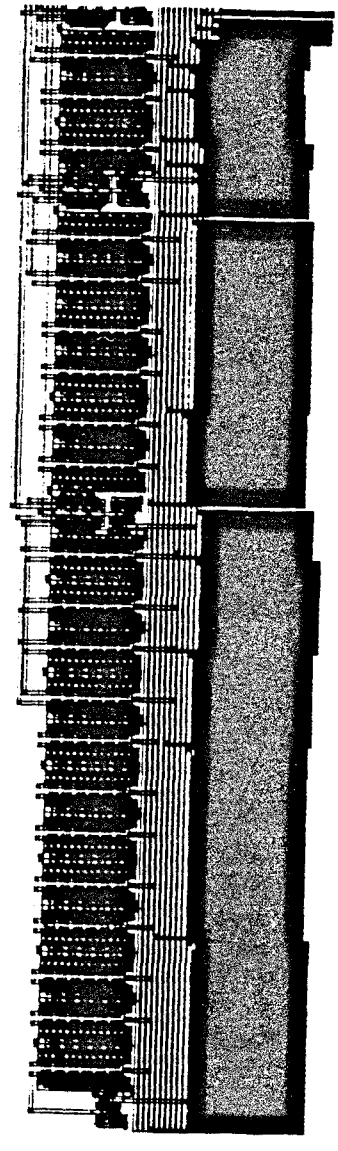


Figura A2.6. Layout de la realització amb resistències de polisilici del conversor A/D.

## A2.3. NDN2

Amb el xip NDN2 deixem d'utilitzar estratègies de disseny full-custom i passem a treballar amb llibreries de cel.les, en aquest cas d'Optimized Array. La funcionalitat neuronal no vindrà donada per la suma de corrents, sinó per la realització dels algorismes presentats als capítols 4 i 5, que implementen la dinàmica seqüencial amb criteri probabilístic.

Els recursos utilitzats en el disseny del xip són recursos estandard (portes lògiques, registres, cel.les de sumadors, multiplexors decodificadors, etc.). L'única estructura específica que hem utilitzat és el comprobador de codi Berger.

### A2.3.1 Comprobador de codi Berger de 12 bits

Els codis Berger són una classe de codis separables (en la paraula codi les parts d'informació i de codi estan separades) caracteritzats perquè codifiquen el número de uns presents a la part d'informació. La quantitat de bits de codi necessaris per $I_1$ bits d'informació, és

$$I_2 = \lceil \log_2(I_1 + 1) \rceil$$

Aquests tipus de codis han estat sovint utilitzats en realitzacions VLSI autocomprovables per la seva capacitat de detecció d'errors unidireccionals, els quals tenen per efecte variar el número de 1s de la paraula codi sempre en el mateix sentit. Aquests errors es poden produir per defectes físics diversos: curtcircuits amb línies d'alimentació o terra (faltes stuck-at), trencament de línies conductores de metall o polisilici, etc.

El disseny de comprobadors de codi Berger, va ser establert per Marouf i Friedman [] l'any 1978 utilitzant cel.les de sumador com a primitives de base. A partir d'aquest formulisme, hem desenvolupat el nostre circuit que compta el número de 1s en paraules de 12 bits, el resultat del qual s'acumula per obtenir al final de la iteració el camp local associat a la neurona corresponent.

L'esquema de la figura A2.7, mostra el disseny del comprobador que consta de 11 cel.les de sumador d'un bit amb un retard equivalent a tres nivells de sumadors.

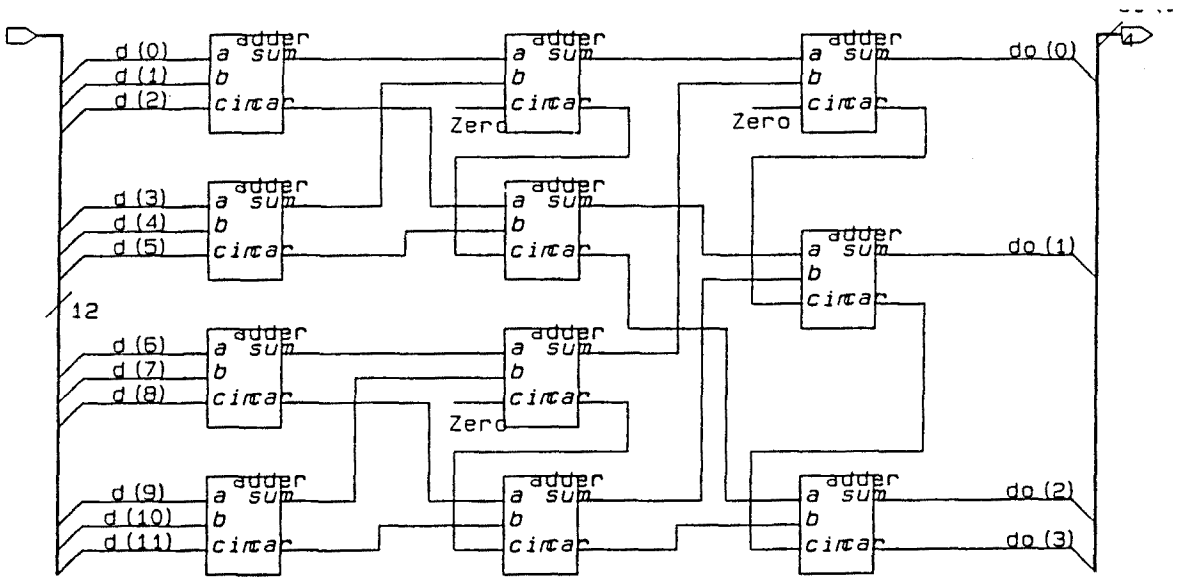



Figura A2.7. Esquema del circuit Berger per a 12 bits dissenyat pel xip NDN2.

## A2.4. NDN3

El xip NDN3 ha estat dissenyat utilitzant una llibreria de Standard Cells. Aquesta elecció es deu a la millora en les característiques temporals associats a aquesta estratègia de disseny, que ens permet un augment de la velocitat de funcionament del xip, i també a les millors prestacions dels programes de CAD associats a aquesta llibreria (SOLO2000) respecte del utilitzat pels Optimized Arrays (SOLO1400): facilitats per a la generació d'estimuls de simulació (llenguatge d'alt nivell) i test (programa de conversió de vectors de simulació), augment de la velocitat de les simulacions lògiques, millora de les característiques dels programes de Placement&Routing, amb la possibilitat de variar certs paràmetres (definició de regions per al Placement, prioritats de nodels per al Routing, ...).

Aquestes característiques permeten abordar l'increment de complexitat necessari per al disseny del xip NDN3, causat per la introducció de la unitat de control, gestió de comunicacions i memòria.

Aquesta complexitat ha fet que siguin molts més els recursos funcionals que hem dissenyat específicament per al xip.

### A2.4.1 Circuit Berger de 8 bits

La realització del circuit Berger de 8 bits no és del tot equivalent a la realitzada pel xip NDN2. La diferència fonamental ve donada per un canvi en el recurs de que disposa la llibreria: una cel.la de sumador de dos bits en lloc d'una cel.la d'un bit, l'oblectiu de la qual és l'estalvi de superfície en ser encadenada dins a sumador. Aquesta conclusió es pot extreure dels valors de fanout de la sortida de carry ($C_{out}$= ff) i capacitat d'entrada de carry de l'etapa anterior ($C_{in}$= ff) i és la raó per la qual hem hagut de posar els buffers que apareixen a l'esquema del circuit (figura A2.8).

La utilització d'aquest recurs, fa que haguem de reformular el mètode de disseny del circuit comprobador a nivell de disseny lògic, la qual cosa és senzilla

per al cas de 8 bits. Els circuit d'entrada i el de sortida corresponen a semisumadors (n'hi ha 4), mentre que els interns són sumadors (2).

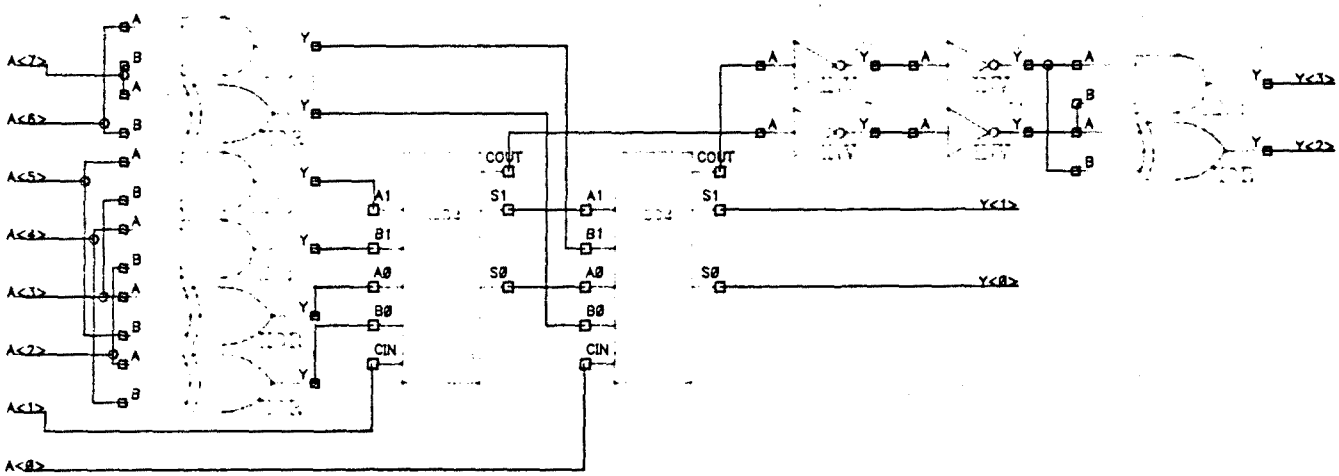



Figura A2.8. Esquema del circuit Berger per a 8 bits utilitzat al xip NDN3.

### A2.4.2 Unitat Aritmètico-Lògica

El mateix element sumador de 2 bits presentat anteriorment l'utilitzem com a base per al disseny de la unitat aritmètico lògica de 12 bits. Tant per l'NDN2 com per l'NDN3, ens hem restringit a la funcionalitat de recurs sumador-restador que hem implementat segons l'esquema de la figura A2.9. En el cas del xip NDN3, ens en calen dos ja que calculem en paral.lel el número de neurones excitadores $\mathbf{n_u}$ i el número total de neurones associades a una sinapsi $\mathbf{n_u+n_d}$, a partir dels valors de les sinapsi $\mathbf{T_{ij}}$ i dels estats de les neurones $\mathbf{O_j}$.

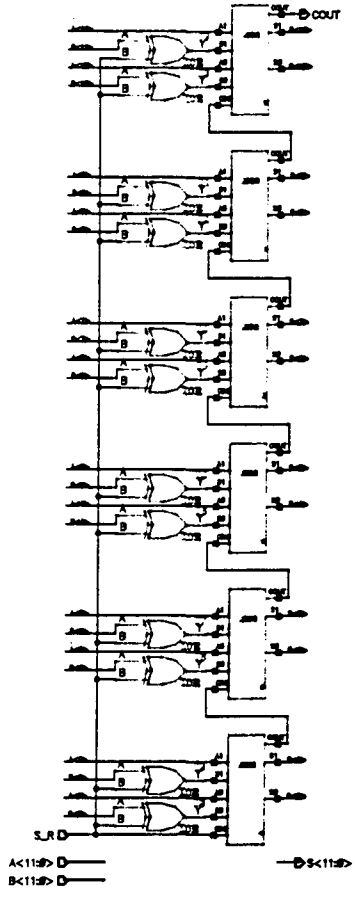



Figura A2.9. Sumador-restador de 12 bits utilitzat al xip NDN3.

Les operacions de resta s'utilitzen per al càlcul del camp local sense normalitzar $n_u$-$n_d$ = $2n_u$ - ($n_u$+$n_d$), encara que s'emmagatzema en memòria aquest valors **dividir per 2** (per tal d'estalviar un bit *i* realitzar les actualitzacions de valor +1/-1). L'operació de resta s'utilitza també per a la comparació de camps locals normalitzats i determinació de la màxima inestabilitat.

La representació dels valors del camp local ha estat realitzada amb números en complement a 2 i per tant calen 13 bits per a la comparació dels camps locals com a resultat de la normalització modificada presentada al capítol 5. D'aquests, el tretzé s'utilitza només com a signe i per tant està implementat exteriorment a la ALU.

### A2.4.3 Logaritme modificat

El càlcul del logaritme modificat del número de sinapsis associades a una neurona $n_u$+$n_d$, permet la compactació d'aquest número a quatre bits. D'aquesta forma, la paraula formada per aquests quatre bits, i els dotze del camp local no normalitzat $n_u$-$n_d$, és una paraula de 16 bits, de manera que es treu un profit màxim tant de les característiques de les memòries (2 xips de 8 bits) com dels busos utilitzats per la manipulació dels valors de les sinapsis (dos busos de 8 bits).

L'esquema de la figura A2.10, implemeta l'expressió donada pel logaritme al capítol 3, realitzada a nivell de portes lògiques.

### A2.4.4 Desplaçament a l'esquerra

El circuit de desplaçament a l'esquerra s'utilitza únicament per a la normalització dels camps locals en funció del número de neurones en la determinació de la inestabilitat màxima. En l'esquema que es presenta a la figura A2.11, utilitzem una estructura de desplaçament basada en la porta CMOS sèrie paral.lel AOI22 de 8 transistors, en lloc de la utilització de 3 portes de 2 entrades (12 transistors).

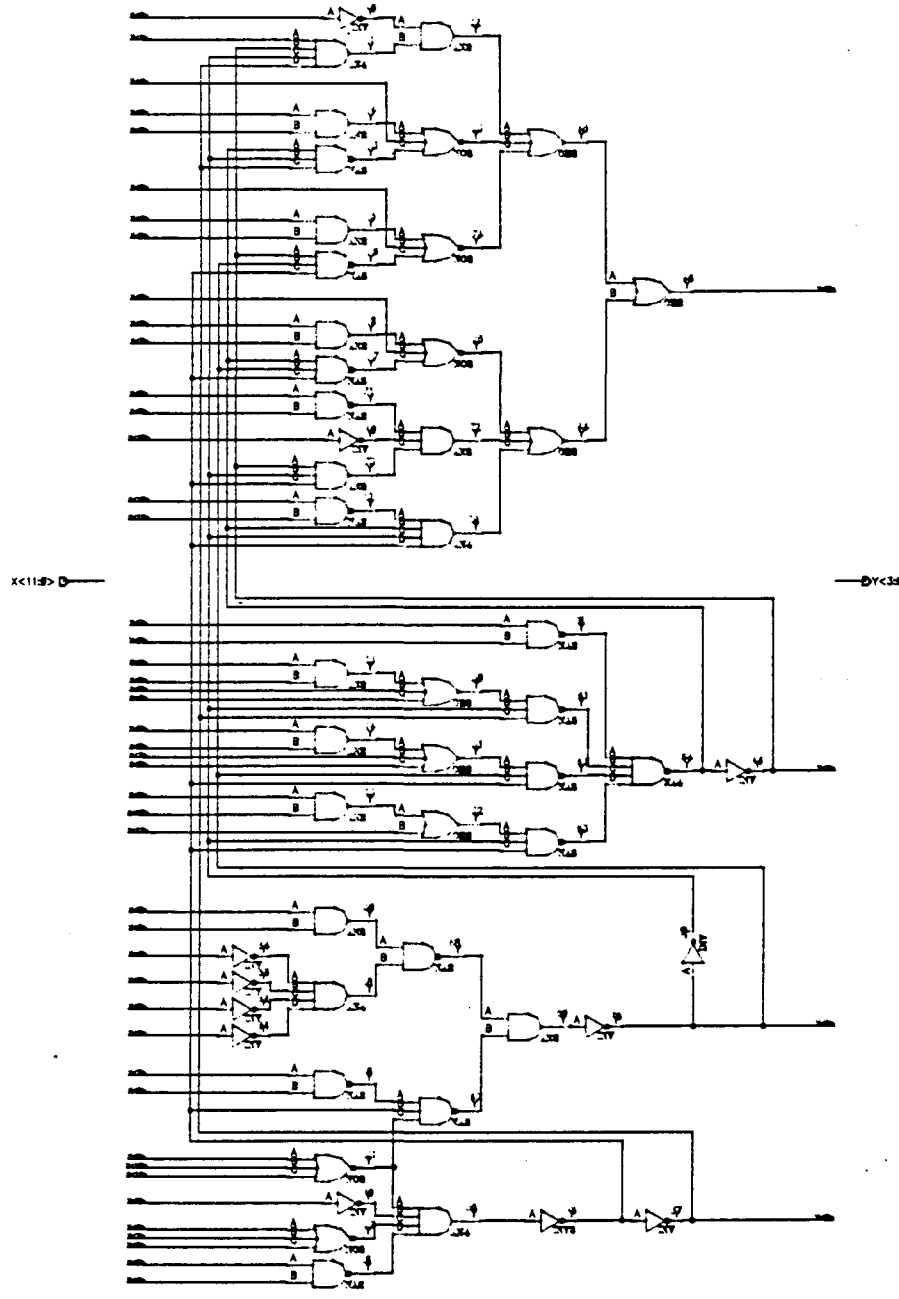


Figura A2.10. Esquema del logaritme modificat.

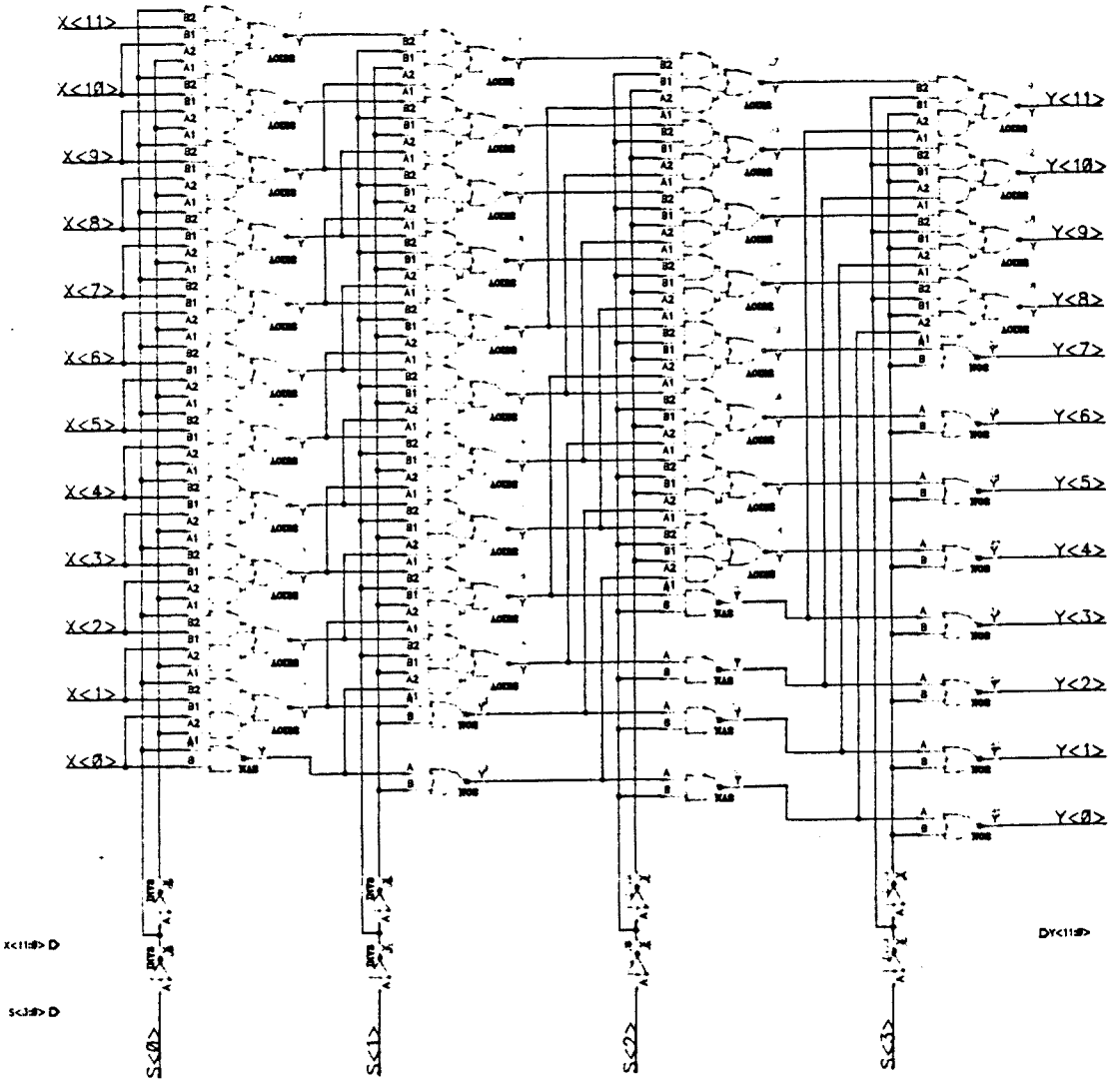



Figura A2.11. Esquema del circuit de desplaçament.

### A2.4.5. Autòmata de control

L'autòmat de 8 estats presentat al capítol 5 l'hem implementat amb 3 flip-flops D, i lògica combinacional, tal i com es mostra a la figura A2.12. La no utilització d'un generador de PLAs, es deguda a que no en disposàvem a la llibreria.

A l'autòmat hi entren les condicions de control del fluxe, i en surten els senyals de control corresponents als estats, que operats amb lògica combinacional, produeixen els senyals de control dels recursos programables, multiplexors i càrrega de registres.

### A2.4.6. Flip-flop D amb reset

El flip-flop D amb reset, és la cel.la bàsica de la qual es composen tots els registres utilitzats, un total de 159 per l'NDN3. A la figura A2.13 en donem tant el full d'especificacions i l'esquema corresponents.

### A2.4.7 Generació de fases de rellotge

La funció d'aquest circuit és l'adequació temporal dels senyals a enviar a l'exterior del xip. Aquests han de complir dues especificacions bàsiques:

(i) L'estructura i ordre dels senyals de control que ataquen les memòries (selecció de xip, senyal d'escriptura i adreces i dades vàlides), donats pels fulls de característiques d'aquestes.

(ii) L'evitar el màxim número de conflictes d'escriptura sobre busos, controlant la direccionalitat dels busos.

Mentre que la primera part es realitza manipulant senyals amb fases de rellotge, el segon el realitzem amb una cadena d'inversors, ja que està referit a l'inici del cicle, i no afecta directament el malfuncionament, sinò que pot portar problemes de consum i degradació.

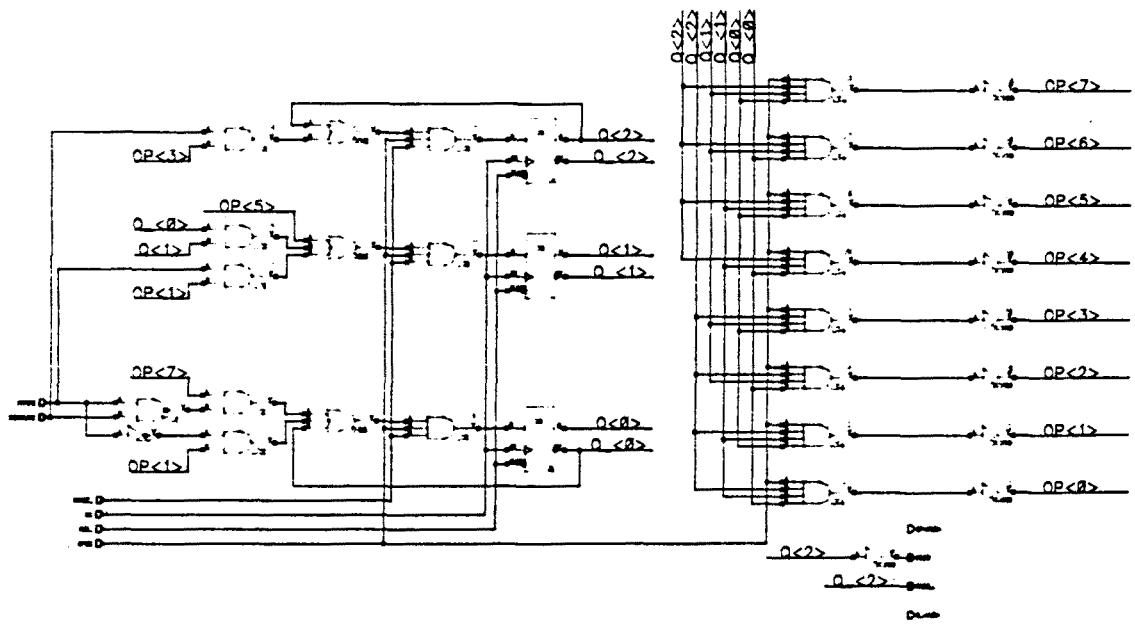


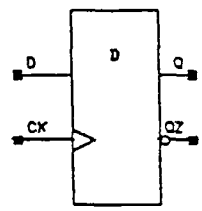

Figura A2.12. Esquema de l'autòmata de la unitat de control.

# ES2 StandardLib REV.1.1

MARCH 93

## D FLIP FLOP D

| D | CK | Q | QZ |
|---|---|---|---|
| H | ⌐¬ | H | L |
| L | ⌐¬ | L | H |
| X | ⌐ | Q0 | QZ0 |
| X | H | Q0 | QZ0 |




| PARAMETER | VALUE | UNIT |
|---|---|---|
| Fanout_Q | 1.2 | pF |
| Fanout_QZ | 1.4 | pF |
| Cin_D | 0.04 | pF |
| Cin_CK | 0.07 | pF |
| Transistors | 28 | |
| Size | 113.5x85.5 | um2 |

| PARAMETER | MIN | TYP | MAX | MIL | UNIT |
|---|---|---|---|---|---|
| fmax | 83 | 83 | 83 | 83 | Mhz |
| D setup | 0.47 | 1.09 | 2.34 | 2.97 | ns |
| D hold | 0.16 | 0.00 | 0.00 | 0.00 | ns |
| CK tw | 0.97 | 1.75 | 3.98 | 4.68 | ns |

| PARAMETER | FROM | TO | MIN | TYP | MAX | MIL | UNIT |
|---|---|---|---|---|---|---|---|
| tplh | CK | Q | 1.57 | 3.08 | 6.36 | 7.82 | ns |
| Δ tplh | CK | Q | 0.54 | 1.03 | 2.02 | 2.48 | ns/pF |
| tphl | CK | Q | 1.42 | 2.79 | 5.75 | 7.11 | ns |
| Δ tphl | CK | Q | 0.52 | 0.85 | 1.47 | 1.75 | ns/pF |
| tplh | CK | QZ | 1.03 | 2.01 | 4.06 | 5.02 | ns |
| Δ tplh | CK | QZ | 0.57 | 1.10 | 2.18 | 2.68 | ns/pF |
| tphl | CK | QZ | 1.14 | 2.28 | 4.69 | 5.76 | ns |
| Δ tphl | CK | QZ | 0.62 | 1.07 | 1.96 | 2.36 | ns/pF |

Figura A2.13. Diverses representacions del flip-flop D.

### A2.4.8 Divisió funcional i pads

La figura A2.14 representa la partició per al disseny del xip en unitat de procés, unitat de control, unitat de comunicacions i unitat de sincronisme. També s'observen els pads utilitzats.

Cal comentar, que seguint les recomanacions de ES2, hem estat molt conservadors a l'hora de situar pads d'alimentació (4) i massa (8) ja que tots els pads excepte un són de sortida o d'entrada-sortida i per tant es poden manipular corrents elevats amb el corresponent risc d'aparició d'efectes indesitjats (degradació de nivells, efecte lach-up,...).

## A2.5. NDN3M

Tal i com hem comentat al capítol 5, la diferència fonamental entre els xips NDN3 i NDN4 és la introducció de memòria externa. Aquesta es caracteritza per haver estat dissenyada en full-custom i tenir una estructura repetitiva complexa (reflexada al compilador d'estructura). El treball d'extracció de paràmetres i caracterització per a simulador és, en aquest cas, molt important per tal de generar models d'alt nivell fiables.

De la memòria utilitzada en presentem: el layout de estructura de repetició bàsica formada per quatre cel.les (figura A2.15), el pla de base utilitzat pel generador d'estructura (figura A2.16) i l'esquema a nivell de blocs funcionals (figura A2.17).

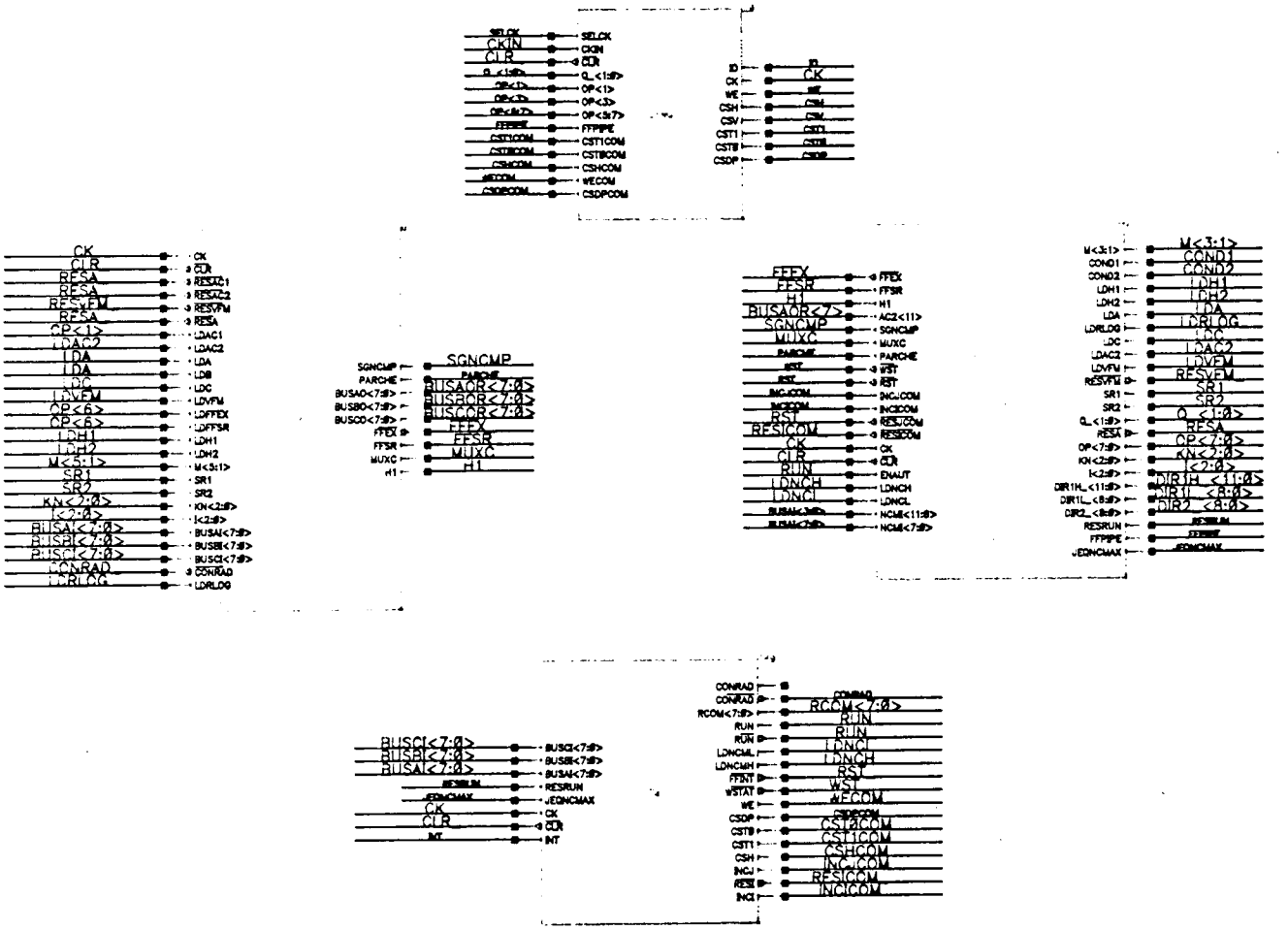


Figura A2.14. Pla de base del xip NDN3.

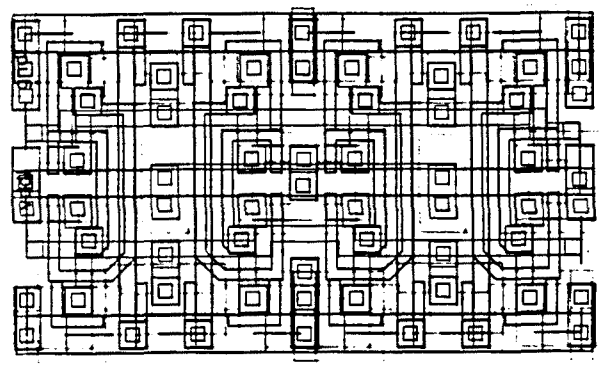


Figura A2.15. Cel.la de memòria utilitzada pel generador de RAM.

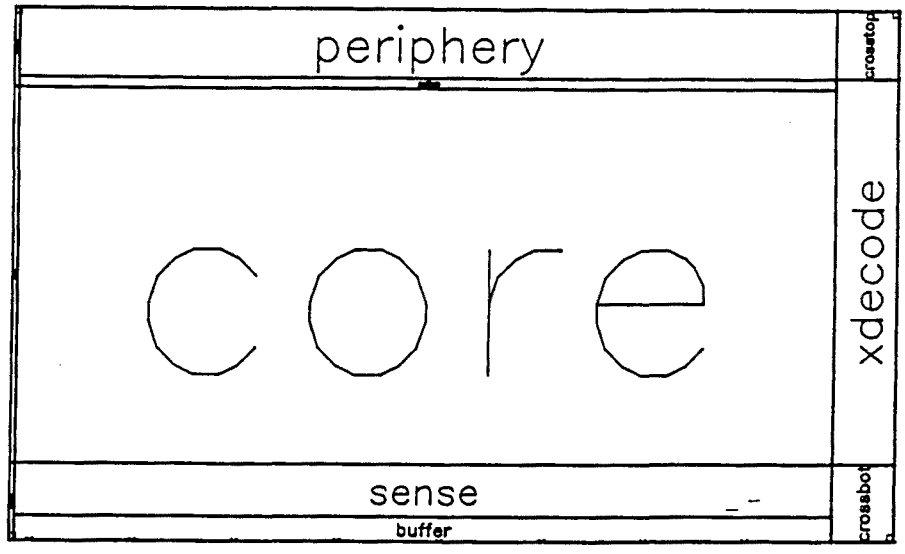



Figura A2.16. Pla de base a nivell de blocs utilitzat pel generador de RAM.

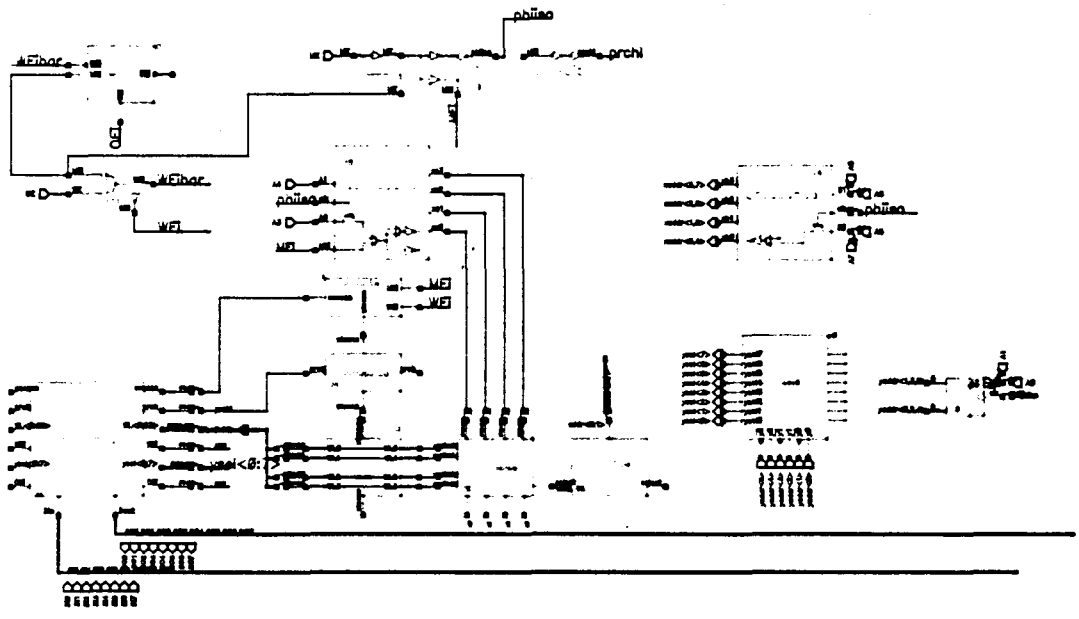


Figura A2.17. Esquema a nivell de blocs funcionals de la meòria RAM4.

# BIBLIOGRAFIA

[Ackle85] "A learning algorithm for Boltzmann machines". D.H. Ackley, G.E. Hinton and T.J. Sejnowsky, Cognitive Science 9 (1985) 147-169.

[Alspe87] "A neuromorphic VLSI learning system". J. Alspector, R.B. Allen. Stanfor Conference on VLSI. MIT Press, 1987.

[Amit87a] "Statistical mechanics of neural networks near saturation". D.J. Amit, H. Gutfreund and H. Sompolinsky, Annals of Physics 173 (1987) 30-67.

[ANZA] "ANZA User's guide". Hecht-Nielssen Corp.

[Biene87] "A neural network for invariant pattern recognition". E. Bienenstock and C. Von der Malsburg, Europhys. Lett. 4 (1987) 121-126.

[Block62] "The perceptron: a model for brain functioning". H.D. Block, Rev. Mod. Phys. 34 (1962) 123-135.

[Bound86] "A statistical mechanical study of Boltzmann machines". D.G Bounds, J. Phys. A: Math. Gen. 20 (1987) 2133-2145.

[Bout89] "A stochastic architecture for neural nets". Van der Bout, D.E., T.K. Miller. IEEE Trns. on Circuits and Systems, May 1989.

[Carra89] "Optimised aechitecture for neural autoassociative memories". J. Carrabina, C.J. Pérez-Vicente, N. Avellana, E. Valderrama. IFIP Workshop on Parallel Architectures on Silicon. Grenoble, 1989.

[Carra90] "Four bits A&D converter based on the Hopfield model". J. Carrabina, E. Valderrama, J.C. Calderón, F. Lisa. Procs. of the First International Workshop on Microelectronics for Neural Networks. Dortmund, 1990.

[Carra91] "Efficient dynamics for networks of hard-limited neurons". (en preparación).

[CNM:1-90] "Un proyecto de red neural". Informe científico 1/90. Centro Nacional de Microelectrónica del CSIC. Marzo 1990.

[Dotse88] "Neural networks: traslation, rotation and scale-invariant pattern recognition". V.S. Dotsenko, J. Phys. A: Math. gen. 21 (1988) L783-L787.

[El-le] "A basic MOS neural type junction: A perspective on neural-type microsystems". N. El-Leithy, R.W. Newcomb, M. Zaghloul.

[Farha85] "Optical implementation of the Hopfield model". N.H. Farhat, D. Psaltis, A. Prata and E. Paek, Applied Optics 24 (1985) 1469-1475.

[Graf86] "VLSI implementation of a neural network memory with several hundreds of neurons". H.P. Graf et al. AIP Conference Procs. of Neural Networks for Computing. Snowbird, 1986.

[Graf87] "A CMOS implementation of a neural model". H.P. Graf, P. de Vegvar. Procs. of the Stanford Conference. MIT Press, 1987.

[Graf88] "VLSI implementation of a neural network model". H.P. Graf, L.D. Jackel, W.E. Hubbard. Computer, March 1988.

[Hopfi82] "Neural networks and physical systems with emergent collective computational abilities". J.J. Hopfield, Proc. Natl. Acad. Sci. 79 (1982) 2554-2558.

[Hopfi84] "Neurons with graded response have collective properties like those of two-state neurons". J.J. Hopfield. Procs. of the National Academy of Science USA. May 1984.

[Hopfi86] "Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit". J.J. Hopfield, D.W. Tank. IEEE Trans. on Circuits and Systems. May 1986.

[IEEEM89] IEEE Micro. Número especial dedicado a "Silicon neural networks". Diciembre 1989.

[Jacke87] "Electronic neural netwrok chips". L.D. Jackel, H.P. Graf, R.E. Howard. Applied Optics, Diciembre 1987.

[Jutte87] "Calcul neuromimetique et traitament du signal, analyse en composantes indépendentes". Tesis Doctoral de C. Jutten. INPG Grenoble, 1987.

[Kante87a] "Potts-glass models of neural networks". I. Kanter, Phys. Rev. A 37 (1987) 2739-2742.

[Kante87b] "Associative recall of memory without errors". I. Kanter and H. Sompolinsky, Phys. Rev. A 35 (1987) 380-392.

[Kohon82] "Self-organized formation of topologically correct feature maps". T. Kohonen, Biol. Cybernetics 43 (1982) 59-69.

[Kohon84] Ver referencia [Kohon89]

[Kohon88] "The neural phonetic typewriter". T. Kohonen. Computer, March 1988.

[Kohon89] "Self organization and associative memory". T. Kohonen. 3ª edición. Springer-Verlag, 1989.

[Kosko87] "Bidirectional associative memories". B. Kosko. IEEE Trans. on Systems, Men and Cybernetics, nº 16, 1988.

[Kosko88] "Bidirectional associative memories". B. Kosko, IEEE Trans. Sys. man and Cybern. 18 (1988) 49-60.

[Kree88] "Recognition of topological features of graphs and images in neural networks". R. Kree and A. Zippelius, J. Phys. A: Math. gen. 21 (1988) L813-L818.

[Linar89] "A programmable neural oscillator cell". B. Linares, E. Sánchez, J.L. Huertas. IEEE Trans. on Circuits and Systems. May 1989.

[Lyon89] "Electronic cochlea" R.F. Lyon, C. Mead. Dentro de [Mead89].

[Mahow89] ""Silicon retina". M.A. Mahowald, C. Mead. Dentro de [Mead89].

[Mead89] "Analog VLSI and neural systems". C. Mead. Addison Wesley, 1989.

[Minsk69] "Perceptrons". M. Minsky and S. Papert, MA: MIT Press, 1969.

[McCul43] "A logical calculus of the ideas immanent in nervous activity". W.S. McCulloch and W. Pitts, Bulletin of mathematical biophysics 5 (1943) 115-133.

[Muell89] "Design and fabrication of VLSI component for a general purpose analog neural computer". P. Mueller et al. Dentro del libro "Analog implementations of neural netwroks". C. Mead and M. Ismail editores. Kluwer Academic Pub., 1989.

[Murra88] "Asynchronous VLSI neural networks - using pulse-stream arithmetic". A.F. Murray, A.V.W. Smith. IEEE Journal on Solid State Circuits, vol 23, Junio 1988.

[Murra90] "Innovations in pulse-stream neural VLSI artihmetic and communications". A.F. Murray et al. Procs. of the First International Workshop on Microelectronics for Neural Networks. Dortmind, 1990.

[Peret86] "Stochastic dynamics of neural networks". P. Peretto, J.J. Niez. IEEE Trans. on Sys. Man and Cyber., vol 16, pp 73-83, 1986.

[Pérez90] "Learning algorithm for binary synapsis". C. Pérez-Vicente. Statistical Mechanics of Neural Networks, pp 167-174. Springer Verlag 1990.

[Pérez91] "Study of a learning algorithm for neural networks with discrete synaptic couplings". Network. (en prensa).

[Perso86] "Collective computational propierties of neural networks". L. Personnaz, I. Guyon, G. Dreyfus. Phys. Rev. A 35, pp 380-392, 1986.

[Perso86b] "A biologically constrained learning mechanism in networks of formal neurons". L. Personnaz, I. Guyon, G. Dreyfus. J. Physique Lett. 46, L359-L365, 1985.

[Rosen58] "The perceptron: A probabilistic model for information storage and organization in the brain". F. Rosenblatt. Psychological Review 65, pp 386-408, 1958.

[Rueck87] "A VLSI concept for associative matrix based on neural netwroks". U. Ruecker, I. Kreuzer, K. Goser. COMPEURO, 1987.

[Rumel85] "Feature discovery by competitive learning". D.E. Rumelhart and D. Zisper, Cognitive Science 9 (1985) 75-112.

[Rumel86] "Parallel distributed processing: Explorations in the microstructure of cognition". Vols. I y II. D.E. Rumelhart, J.L. McClelland. MIT Press, 1986.

[Sage86] "Artificial neural network integrated circuit based on MNOS/CCD principles". J.P. Sage et al. AIP Conference Procs. of Neural Networks for Computing. Snowbird, USA, 1986.

[Silvi86] "VLSI architectures for implementation of neural netwroks". M.A. Silviotti, M.R. Emerling, C. Mead. AIP Conference Procs. of Neural Networks for Computing. Snowbird, USA, 1986.

[Sejno86] "Learning symmetry groups with hidden units: Beyond the perceptron". T.J. Sejnowsky, G.E. Hinton, P.K. Kienker. Physica D 22, pp 260-275, 1986.

[Thako87] "Electronic hardware implementation of neural networks". A.P. Thakoor et al. Applied Optics, Diciembre 1987.

[Trele89] "Neurocomputers". P. Treleaven. International Journal of Neurocomputing, vol 1, 1989.

[Verle89] "A high storage capacity content addressable memory and its learning algorithm". IEEE Trsn. on Circuits and Systems. May 1989.

[Vitto89a] "CMOS integration of Herault-Jutten cells for separation of sources". E. Vittoz, X. Arreguit. Dentro del libro "Analog implementations of neural systems". C. Mead y M. Ismail editores. Kluwer Academic Pub. 1989.

[Vitto89b] "Analog VLSI implementation of a Kohonen map". E. Vittoz et al. Artificil Neural Networks Journées d'Electronique. Lausanne, 1989.

[Vitto90] "Analog storage of adjustable synaptic weights". Procs. of the First International Workshop on Microelectronics for Neural Networks. Dortmund, 1990.

[Wagne87] "Multilayer optical learning netwroks". K. Wagner, D. Psaltis. Applied Optics, vol 26, pp 5077-5080, 1987.