



HAL
open science

Etude de la tolérance aux aléas logiques des réseaux de neurones artificiels

Ammar Assoum

► **To cite this version:**

Amman Assoum. Etude de la tolérance aux aléas logiques des réseaux de neurones artificiels. Autre [cs.OH]. Institut National Polytechnique de Grenoble - INPG, 1997. Français. NNT : . tel-00004913

HAL Id: tel-00004913

<https://theses.hal.science/tel-00004913v1>

Submitted on 20 Feb 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée par

Ammar ASSOUM

Pour obtenir le grade de **Docteur** de

**L'INSTITUT NATIONAL POLYTECHNIQUE DE
GRENOBLE**

(Arrêté ministériel du 30 Mars 1992)

Spécialité : **Signal, Image, Parole**

Etude de la tolérance aux aléas logiques des réseaux de neurones artificiels

Date de soutenance : 4 Avril 1997

Composition du jury :

Président	M. Guy MAZARE	Professeur à l'INPG, Grenoble
Rapporteurs	M. Jean GASIOT	Professeur à l'Université Montpellier II
	M. Joan CABESTANY	Professeur à l'UPC, Catalogne
Examineurs	Mme Mirta GORDON	Chargée de recherches CNRS, Grenoble
	M. Raoul Velazco	Chargé de recherches CNRS, Grenoble

Thèse préparée au sein du laboratoire **Logiciels Systèmes Réseaux**

A ma mère
A ma femme
A mon fils

Et on ne vous a donné que peu de connaissance. (Coran 17.85)

Je voudrais en premier lieu exprimer ma gratitude à Monsieur Raoul VELAZCO, pour avoir dirigé cette thèse avec dévouement, pour ses critiques, ses conseils et sa disponibilité notamment durant les derniers moments de rédaction.

J'adresse mes remerciements à Monsieur Guy MAZARE, Professeur à l'INPG et Directeur de l'ENSIMAG, de m'avoir fait l'honneur de présider le jury de cette thèse.

Je remercie vivement Messieurs Jean GASIOT, Professeur à l'Université de Montpellier II, et Joan CABESTANY, Professeur à l'Université Polytechnique de Catalogne, d'avoir accepté d'être rapporteurs de cette thèse.

Je tiens à remercier Madame Mirta GORDON, chargée de recherches CNRS, pour ses critiques et ses conseils ainsi que d'avoir bien voulu faire partie du jury.

Je suis reconnaissant envers Monsieur Jean-Marie PALAU, professeur à l'Université de Montpellier II pour ses remarques concernant la rédaction de cette thèse.

Je remercie plus particulièrement Monsieur Haissam ZIADE, Maître Assistant à l'Université Libanaise pour ses encouragements et ses conseils tout au long de cette thèse.

Que soient également remerciés ici:

F. BEZERRA et R. ECOFFET, Ingénieurs au CNES.

S. DUZELLIER Ingénieur au DERTS/CERT.

D. BENEZECH, Technicien à la société EREMS.

Je n'oublierais pas tous mes collègues du Laboratoire Logiciels, Systèmes, Réseaux, notamment, C. AKTOUF, O.K. BENKAHLA, M. BENOITHMAN, M. BOUSSEY, X. BOTEY, P. CHEYNET, R. DJEMAL, J.F. GUILLAUD, N.E. IDIRENE, S.E. KANNAT, S.M. KARABERNOU, Y. LATROUS, B. MARTINET, M. OLMOS, M. POKAM, J. RUBIO, N.E. RADI, S. ROCHE, pour leur aide, leur disponibilité et leur bonne humeur et finalement l'équipe de reprographie de Monsieur Daniel IGLESIAS.

Je voudrais remercier aussi mes parents pour leur encouragement et leur aide indirecte sans oublier de réserver ma dernière mais spéciale mention à mon épouse qui m'a soutenu quotidiennement tout au long de cette thèse surtout pendant les derniers jours.

Ammar ASSOUM

Le 4 Avril 1997

RESUME

Avec l'accroissement de la complexité des traitements effectués à bord des véhicules spatiaux et l'utilisation de circuits de plus en plus intégrés, le phénomène d'upset devient de plus en plus critique. En effet, ce phénomène se traduit par le basculement intempestif du contenu d'un point mémoire suite à l'impact d'une particule lourde dans des zones sensibles du circuit. Ses conséquences sont parfois fatales et peuvent conduire à la perte voire à la destruction de l'engin sur lequel il a eu lieu.

Les réseaux de neurones artificiels constituent une nouvelle approche de traitement de l'information. Ils offrent des solutions compactes et rapides pour une large gamme de problèmes, en particulier ceux ayant des contraintes temps réel tel le cas de la plupart des applications spatiales actuelles. Ceci est davantage vrai avec l'utilisation des émulations et des implantations matérielle.

Parmi les propriétés importantes des réseaux de neurones, on peut citer leur tolérance aux fautes qui mesure leur aptitude à exécuter la tâche qui leur est demandée en présence d'informations erronées et de maintenir leur capacité de calcul même si une partie du réseau est endommagée.

L'objectif de cette thèse est d'étudier la tolérance aux fautes des réseaux de neurones face aux fautes de type upset et ceci en vue d'étudier la possibilité de leur utilisation, sous forme matérielle, dans un environnement radiatif tel que l'espace, le but étant de choisir parmi des circuits candidats, ceux qui sont acceptés (ou rejetés) pour des applications spatiales.

Pour ce faire, plusieurs réseaux et plusieurs circuits ont été testés. Les expériences réalisées étaient de type simulation logicielle d'erreurs, injection matérielle de fautes et tests aux ions lourds.

Les résultats obtenus montrent que les réseaux de neurones artificiels sont tolérants aux fautes de type upsets ce qui en fait un bon candidat pour les applications s'exécutant à bord des engins spatiaux.

MOTS CLES

RADIATIONS, SEU, PHENOMENE D'ALEA LOGIQUE, IONS LOURDS, RESEUX DE NEURONES ARTIFICIELS, TOLERANCE AUX FAUTES, ROBUSTESSE.

ABSTRACT

With the increasing complexity of treatments on satellite-borne and the utilisation of highly integrated circuits, the upset phenomenon becomes more and more crucial. Indeed, this phenomenon, caused by a heavy ion strike results in the modification of the information stored in a memory element. Upsets may perturb the operation of satellite-borne applications and can lead to serious consequences on the control of equipment operating in space.

Artificial neural networks (ANN) constitute a new approach in information processing. They offer powerful and compact solutions to a wide range of problems, in particular those with real time constraints which is the case for most of current space applications.

Among the main properties of neural networks we can mention their fault tolerance which measures their capability to perform the desired task under fault conditions (erroneous information) and to maintain their computing ability when a part of the network is damaged or removed.

The goal of this thesis is to study the fault tolerance of neural networks against single event upsets in order to investigate the possibility of their utilisation in a radiative environment such as space. This work aims mainly at choosing, among the tested circuits, those that are accepted (or rejected) for space applications.

Several networks and several circuits have been tested. Three kinds of experiments have been performed: software simulation of errors, hardware injection of faults and heavy ions tests.

Obtained results show that Artificial Neural Networks are tolerant against upsets which make them a good candidate for use in space applications.

KEY WORDS

RADIATIONS, SEU, UPSET PHENOMENA, HEAVY IONS, ARTIFICIAL NEURAL NETWORKS, FAULT TOLERANCE, ROBUSTNESS.

Introduction

L'espace terrestre est irradié par plusieurs types de rayonnements. Les particules qui constituent ces derniers sont divisées en deux catégories : les particules légères et les particules lourdes. Elles ont comme origine le soleil (vent et éruptions), la magnétosphère (ceintures de radiation) et les rayons cosmiques.

Différents types d'interactions peuvent avoir lieu entre ces particules et les composants électroniques. Elles peuvent produire des défaillances sur le fonctionnement de ces derniers et sont dues particulièrement aux effets des ions lourds qui, en traversant les zones sensibles des composants VLSI, peuvent modifier intempestivement l'état des éléments mémoires affectés : c'est le phénomène d'aléa logique ou d'*upset*.

Les réseaux de neurones artificiels, connus généralement sous l'acronyme ANN (Artificial Neural Networks), constituent une approche fondamentalement nouvelle dans le traitement de l'information. Ce sont des systèmes parallèles, adaptatifs et distribués dont le fonctionnement imite celui du neurone biologique, reproduisant ses caractéristiques de base. Les ANN offrent des solutions compactes et rapides pour une large gamme de problèmes, dont certains sont difficiles à traiter par les approches classiques, en particulier les problèmes avec des contraintes temps réel, ou ceux dont la résolution met en jeu des règles inconnues ou difficiles à expliciter ou à formaliser. Parmi les domaines d'applications nous pouvons citer la reconnaissance des formes, le traitement du signal, le contrôle, le traitement d'images, le traitement de la parole, la prévision, la modélisation et l'optimisation.

Parmi les propriétés des réseaux de neurones on peut noter leur tolérance aux fautes, caractéristique souvent citée comme l'un de leurs principaux avantages par rapport à d'autres types de systèmes de traitement de l'information. Cette caractéristique mesure l'aptitude des réseaux de neurones à exécuter la tâche qui leur est

demandée en présence d'informations erronées et de maintenir leur capacité de calcul même si une partie du réseau est endommagée. Cette propriété des ANN est due à leur redondance et à la nature distribuée des traitements qu'ils effectuent. En effet, chaque neurone contribue à l'établissement de la réponse finale, ce qui a pour conséquence qu'en présence de dégradations (permanentes ou transitoires), les fautes vont être supportées par tous les neurones du réseau au lieu de l'être par le neurone concerné, conduisant ainsi à une dégradation progressive plutôt qu'à une défaillance catastrophique.

Le secteur spatial peut être considéré comme un champ important d'applications des ANN. En effet, la plupart des traitements effectués à bord des engins spatiaux appartiennent aux domaines d'applications neuronales cités ci-dessus. Par conséquent, l'étude de la possibilité d'utilisation des réseaux de neurones dans l'espace, notamment sous forme d'implantations et d'émulations matérielles, s'avère nécessaire. Le but principal de la présente thèse est donc l'étude de la tolérance aux fautes des réseaux de neurones face aux fautes de type upset et ceci en vue d'étudier la possibilité de les utiliser dans un environnement radiatif tel que l'espace. En effet, malgré les nombreux travaux de recherche publiés dans le domaine de la tolérance aux fautes des réseaux de neurones et malgré les différents modèles de fautes testés (bruit gaussien, suppression d'unités, etc.), aucun travail publié n'a pas pu être trouvé sur les effets des upsets sur le comportement des applications neuronales. L'un des principaux objectifs de cette thèse est la mise en œuvre d'expériences permettant de confirmer ou d'infirmer la possibilité d'étendre la tolérance aux fautes des ANN aux fautes de type upset.

Différents types de circuits digitaux ont été choisis pour réaliser ces expériences. Ils sont groupés en deux catégories :

1. Des circuits digitaux dédiés aux réseaux de neurones. Trois circuits différents ont été testés : le premier est un prototype expérimental d'un circuit neuronal à usage général, le deuxième est un circuit commercial dédié à l'accélération des calculs neuronaux, enfin, le troisième est un circuit neuronal spécifique (ASIC) dédié à la reconnaissance de caractères.
2. Des circuits standards à usage général. Il s'agit d'un processeur CISC de type DSP (processeur de signaux) et d'un processeur RISC de type transputer.

Quant aux méthodologies utilisées pour évaluer la tolérance aux fautes des différents réseaux de neurones implantés (ou émulés) sur ces circuits, elles sont les suivantes :

- simulation logicielle de fautes sur les données et paramètres en relation avec le fonctionnement du réseau testé ;
- injection matérielle des fautes sur les zones de stockage des données et/ou du code correspondant au programme de test à exécuter par le circuit ;
- test aux ions lourds, qui consiste à exposer les circuits cibles à un faisceau de particules pendant qu'ils exécutent le programme correspondant à l'ANN implanté.

Les deux premières méthodes de test “hors faisceau” permettent d'obtenir un ordre de grandeur sur la sensibilité du réseau évalué face aux fautes affectant l'information accessible à l'utilisateur ; tandis que la troisième technique permet de tester l'application entière, en particulier les zones sensibles du circuit sous test inaccessibles à l'utilisateur via l'exécution d'un programme (par exemple les cellules mémoires du contrôleur, les registres tampons du chemin de données).

Le travail de la présente thèse consiste donc à évaluer la robustesse des implantations digitales des réseaux de neurones artificiels ; le but ultime étant l'obtention de données expérimentales permettant éventuellement de se passer de toute autre approche de tolérance aux fautes dans un système embarqué à bord d'engins spatiaux.

Cette thèse est organisée en quatre parties. Dans la première (chapitre 1) sont présentées les principales caractéristiques des radiations dans l'environnement spatial, leurs effets sur les circuits intégrés, les différents moyens de prévention et de prévision correspondants ainsi que la mise en œuvre d'expérimentations aux ions lourds. La deuxième partie (chapitres 2 et 3) est consacrée aux réseaux de neurones : elle en donne les principes, expose les différents types d'implantations et domaines d'applications existants et décrit l'état de l'art sur leur propriété de tolérance aux fautes. La troisième partie (chapitre 4) décrit les méthodologies de test utilisées tandis que la quatrième et dernière partie (chapitres 5, 6 et 7) présente une synthèse des résultats obtenus.

Chapitre 1

Effets de l'environnement spatial sur les circuits intégrés

Résumé du contenu du chapitre

Différents types de rayonnements sont rencontrés dans l'espace et peuvent avoir des effets critiques sur les circuits intégrés montés sur des équipements embarqués à bord des véhicules spatiaux. Ce chapitre donne les différents types des radiations présentes dans l'espace ainsi que les modes de défaillances qui en résultent sur les composants digitaux. La mise en œuvre du test aux ions lourds sera également décrite (simulation du milieu radiatif, systèmes de test et stratégies correspondantes).

1.1 Environnement spatial et électronique embarquée

L'espace terrestre est irradié par plusieurs types de rayonnements dont les caractéristiques, les origines et les répartitions dans l'espace et dans le temps sont variables. Les sources de ces rayonnements sont :

- le vent et les éruptions solaires ;
- les zones de piégeages de la magnétosphère (ceintures de radiations) ;
- les rayons cosmiques.

Compte tenu de leurs effets, les rayonnements peuvent être groupés en deux catégories : les particules légères et les particules lourdes.

1. Les particules légères. Elles proviennent du soleil [Bourrieau91] et des zones piégées de la magnétosphère (“ceintures de radiations” ou “ceintures de Van Allen”) [Holmes85] et sont composées essentiellement de protons, électrons, neutrons et de particules α .
2. Les particules lourdes. Ce sont des noyaux d’atomes ayant une très haute énergie et un numéro atomique Z supérieur à 2. Elles proviennent essentiellement des étoiles (le rayonnement cosmique) [Capdevielle84] et des éruptions solaires [Chenette84].

1.2 Effets du rayonnement sur la matière

Le paragraphe précédent a illustré brièvement les radiations présentes dans l’environnement spatial. Parmi les radiations, celles ayant une haute énergie peuvent agir sur les atomes ou molécules constituants de la matière des équipements embarqués à bord des véhicules spatiaux [Gussenhoven93]. Cette action se manifeste, dans le cas des circuits intégrés, par trois types de phénomènes :

1. Dérives des paramètres électriques (tension de seuil par exemple) dues à la dose cumulée ;
2. Changement d’état, ou aléa logique, réversible et non destructif, affectant en priorité les points mémoire (phénomène d’upset “SEU”);
3. Verrouillage de l’alimentation dû à la mise en conduction d’un thyristor parasite et pouvant conduire à une destruction thermique du composant (phénomènes de latchup “SEL”).

Ces phénomènes diffèrent par la nature des particules qui les génèrent. On considère que les effets de dose cumulée sont dus principalement aux particules légères. Les phénomènes d’upset et de latchup sont provoqués par un dépôt local d’énergie dû à l’ionisation tout au long du parcours d’un ion lourd (création de paires électron-trou).

1.2.1 Effets de la dose cumulée

On appelle dose, l'énergie déposée dans le matériau par unité de masse. Les unités couramment utilisées sont le Rad et le Gray (Gy).

$$1 \text{ Rad} = 10^{-5} \text{ J/g} \text{ et } 1 \text{ Gy} = 1 \text{ J/Kg} = 100 \text{ Rad}.$$

Il s'agit de l'accumulation de charges au niveau des oxydes (SiO_2) et des interfaces SiO_2/Si (création d'états d'interface) [Mc Lean84]. Ces charges sont dues à la création des paires électron-trou suite à des ionisations provoquées par des passages de particules chargées. Ces charges conduisent à des effets tels que la diminution de la mobilité des porteurs, la création d'un champ électrique parasite et la modification des propriétés des jonctions d'isolement [Bessot93].

Les effets des doses sont cumulatifs et peuvent conduire à une perte progressive puis totale de fonctionnalité du composant. La tenue à ces effets déterminera donc la durée de vie du composant, voire du système embarqué en général.

1.2.2 Effets singuliers induits par les ions lourds

Contrairement aux dérives paramétriques durables produites par les particules légères, les effets des ions lourds peuvent se traduire par deux types de dégradations transitoires déclenchés par le passage d'une seule particule très ionisante dans une zone sensible du circuit :

- un changement intempestif d'état, effet appelé Single Event Upset (SEU) ;
- le passage à un état de basse impédance pouvant conduire à la destruction thermique du composant, effets appelés, selon la technologie utilisée, latchup (SEL), snapback ou burnout.

Les caractéristiques importantes dans le cas des effets des ions lourds sont le pouvoir d'arrêt et le flux de particules.

1.2.3 Pouvoir d'arrêt, notion de LET

Le pouvoir d'arrêt d'une particule, appelé LET (Linear Energie Transfer), est la perte d'énergie par unité de longueur parcourue dans le matériau ($\frac{\partial E}{\partial x}$). L'unité de mesure utilisée est le MeV/cm (ou le MeV/g/cm^2 en divisant par la masse spécifique du matériau). Le pouvoir d'arrêt dépend de la nature et de l'énergie de la particule incidente ainsi que du milieu traversé. Il est d'autant plus grand que les numéros

atomique des ions incidents et des atomes du milieu traversé sont grands. Par exemple, le LET d'un ion de Fluor (^{19}F) ayant une énergie de 103 MeV et traversant du Silicium est de 4.2 MeV/mg/cm^2 . A cette notion de LET, s'attache la notion de parcours dans le matériau. Le parcours dans la matière est d'autant plus grand que la particule est légère et a plus d'énergie.

1.2.4 Le phénomène SEU

Le basculement intempestif de points mémoire [Petersen83a] est parmi les effets les plus critiques des ions lourds. Il s'agit de la modification de l'information logique stockée dans un circuit de rétention d'information (bascule, registre, mémoire). Ce phénomène est non destructif et réversible : dans le cas où l'erreur est détectée, elle peut être corrigée. En effet, le point mémoire modifié pourra être restauré par le processus normal d'écriture et le composant reste opérationnel. On parle alors d'aléa logique ou en général d'upset. Les ions lourds peuvent également affecter des circuits dépourvus de cellules bistables (circuits combinatoires) mais ceci est moins probable. En effet, un signal parasite induit par l'impact d'une particule peut être faussement interprété comme un signal utile : une information logique en cours de traitement ou de propagation dans un circuit peut être alors modifiée.

Afin de décrire le mécanisme d'upset, considérons le cas d'un bistable CMOS (FIG. 1.1), les mécanismes correspondants à d'autres technologies étant décrits en [Pickel83].

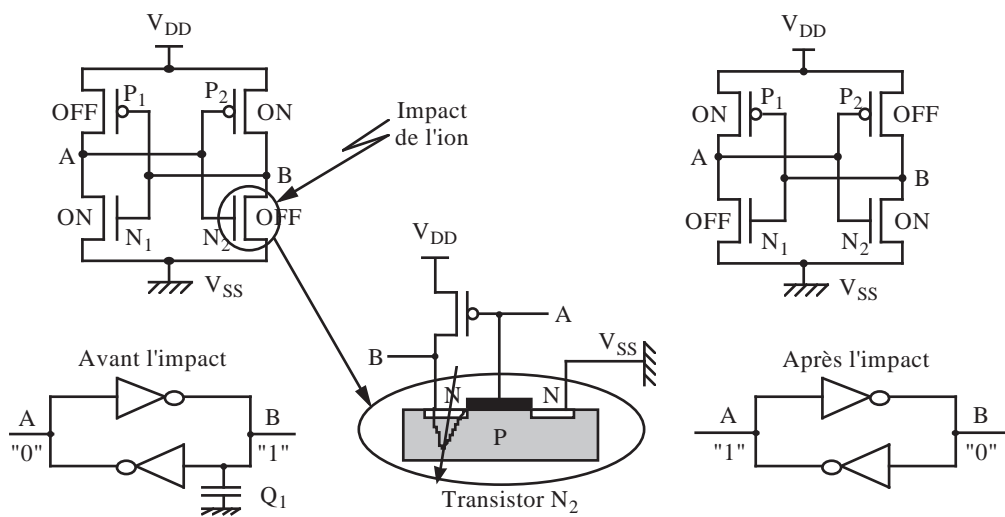


FIG. 1.1 : Le phénomène d'upset - cas d'un bistable CMOS

Supposons qu'un ion traverse la jonction du transistor bloqué N_2 . La charge déposée va créer, au niveau de ce transistor, un canal par lequel elle s'écoulera. L'impulsion de courant produite au nœud B sera transmise à l'autre moitié du bistable, si elle fournit la tension nécessaire pour rendre le transistor P_1 passant, changeant donc l'état du nœud A. Si cet état est maintenu suffisamment longtemps, il sera mémorisé et le basculement logique aura lieu.

L'évolution technologique en matière d'intégration des circuits entraîne une permanente diminution de la charge minimale nécessaire au stockage de l'information dans un point mémoire (charge critique) [Petersen82]. De ce fait, une proportion plus importante d'ions lourds sont susceptibles de provoquer des upsets. En plus, la sensibilité au phénomène d'upset dépend de la dimension de la zone sensible, qui est une caractéristique de la technologie et de l'intégration choisie [Cunningham84].

1.2.5 Notion de LET seuil et de section efficace

Deux grandeurs sont utilisées pour quantifier la sensibilité des circuits électroniques aux effets des radiations : le seuil de sensibilité (LET seuil) et la section efficace.

Seuil de sensibilité

Pour qu'un upset puisse se produire, il faut que le dépôt d'énergie soit supérieur à la quantité de charges qui provoque le changement d'état. Le LET seuil, exprimé en MeV/cm ou $MeV/mg/cm^2$, représente le pouvoir d'arrêt minimum d'une particule pouvant causer une perturbation. Dans la plupart des cas, le LET seuil est obtenu par expérimentation.

Section efficace

Si un circuit est soumis à un faisceau de particules ayant une fluence (nombre de particules incidentes par cm^2) F_p et un pouvoir d'arrêt L_i , manifeste N_e erreurs, sa section efficace sera définie comme étant le quotient $\sigma = \frac{N_e}{F_p}$ (cm^2). Dans la littérature on trouve des sections efficaces par composant et par bit. Dans le dernier cas, il est possible de comparer la sensibilité des circuits ayant des tailles différentes des zones de stockage. En général, c'est la courbe $\sigma = f(LET)$ (section efficace vs. LET) qui caractérise la sensibilité d'un circuit donné. La figure 1.2 donne la forme typique d'une telle courbe. La section efficace à la saturation représente la surface totale des nœuds

sensibles du composant.

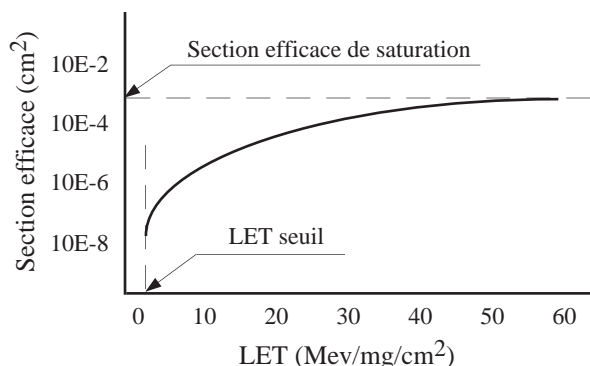


FIG. 1.2 : *Forme typique de la courbe section efficace vs. LET ($\sigma = f(LET)$)*

1.3 Moyens d’actions face aux effets des particules chargées

La théorie de la possibilité d’interaction des radiations avec les systèmes embarqués (upsets, latchups et autres) a été introduite dans les années 70, après l’observation d’anomalies dans le comportement des engins spatiaux [Binder75]. Le phénomène d’upset ne fut sérieusement analysé qu’en 1978 après que des upsets sur des mémoires dynamiques causés par des particules α aient été observés, dans des conditions de laboratoire [May79]. Depuis, le nombre de problèmes rencontrés en vol et dus aux effets de radiations sur les circuits embarqués, n’a cessé d’augmenter [Barillot96, Normand96].

Les conséquences de ces anomalies peuvent être cruciales et préjudiciables au bon déroulement d’un projet spatial, et peuvent même conduire à la perte de l’engin sauf si des moyens d’actions adéquats sont mis en place.

L’augmentation de la complexité de l’électronique embarquée à bord des satellites (télécommunications, traitement du signal, d’images, etc.) fait que la demande de circuits performants (processeurs, mémoires, périphériques, etc.) est de plus en plus élevée. Mais compte tenu de l’évolution technologique dans le domaine de la micro-électronique et de la fabrication des circuits intégrés, la tendance caractéristique des nouveaux circuits (y compris les circuits exigés pour les applications spatiales) va dans le sens d’une augmentation de la rapidité, d’une diminution de la consommation et d’une réduction des dimensions.

Cependant, malgré les nombreux avantages offerts par cette évolution technologique, un inconvénient majeur se manifeste dans la nouvelle génération des circuits intégrés. Il consiste en l'augmentation de leur sensibilité face aux phénomènes induits par les radiations spatiales. Ceci peut être résumé comme suit :

1. La sensibilité aux upsets augmente avec la diminution de la constante de temps des circuits (augmentation de la fréquence maximale de fonctionnement) puisque le basculement transitoire dû à l'impact d'une particule aura le temps nécessaire pour être mémorisé [Gagné90].
2. La diminution de la consommation fait augmenter la sensibilité des composants vu que la charge critique Q_c diminue linéairement avec la tension d'alimentation V_{DD} [Diehl83]. Ceci a été confirmé, pour le cas des circuits mémoires statiques, à l'aide d'essais aux ions lourds ainsi qu'avec des simulations sur un modèle de ces circuits [Kolasinski81].
3. La charge critique Q_c d'un circuit est proportionnelle à sa dimension caractéristique (en technologie MOS, elle correspond à la largeur du canal d'un transistor). Donc, la réduction des dimensions va diminuer la charge critique et par suite augmenter la sensibilité du composant [Brucker84].

D'après ce qui précède, vu les coûts énormes des missions spatiales et compte tenu de la gravité des effets des ions lourds sur l'électronique, des moyens d'action face aux effets de radiations s'avèrent nécessaires. Nous distinguons les moyens de prévention les moyens de prévision.

1.3.1 Moyens de prévention

Les effets des radiations sur un système embarqué peuvent être minimisés par trois types d'actions différents :

1. Elimination des radiations à l'intérieur du satellite (blindage).
2. Réduction de la sensibilité des composants (durcissement).
3. Tolérance des erreurs au niveau du système (sûreté de fonctionnement).

L'inconvénient de ce type d'approche est la pénalisation en temps et/ou en encombrement [Siewiorek84] bien qu'elles permettent l'obtention d'une immunité plus ou moins complète face aux effets des radiations.

1.3.2 Moyens de prévision

Il s'agit de calculer le taux d'erreurs en mission pour les composants candidats, à partir d'une simulation au sol et d'un calcul de prédiction. Ceci permet de ne choisir que les circuits ayant un taux d'erreur acceptable, ou dans le cas où les circuits sont sensibles, de prévoir des actions adéquates pour diminuer les risques en orbite.

Simulation d'upsets au sol

Cette méthode, appelée aussi "test aux ions lourds", consiste à exposer le circuit sous test à un rayonnement aussi représentatif que possible de celui qu'il rencontrera dans l'application finale (durant une mission spatiale). Pendant l'irradiation, le circuit sous test est stimulé par une séquence de test selon une stratégie destinée à déterminer sa sensibilité aux rayonnements. Cette dernière sera calculée en évaluant les réponses du circuit à la séquence de test. Elle se concrétise généralement par une courbe de section efficace $\sigma = f(LET)$ similaire à celle de la figure 1.2.

Prédiction du taux d'événements

Différentes solutions existent pour calculer le taux d'événements en orbite pour un circuit donné. Ceci se fait à partir des courbes des sections efficaces du circuit et des modèles de l'environnement spatial dans lequel est sensé évoluer l'application [Bourrieau88, Petersen83b]

1.4 Mise en œuvre de test aux ions lourds

L'évaluation du comportement des circuits intégrés, notamment les circuits programmables, sous ambiance radiative soulève trois problèmes :

1. Simulation de l'environnement spatial pour le test des circuits sous un faisceau de particules (ions lourds, électrons, protons, etc.) similaires à celles rencontrées dans l'espace.
2. Développement de l'environnement digital/électronique nécessaire à l'application des stimuli et à l'évaluation des résultats durant l'exposition des circuits sous test au faisceau de particules.

3. Développement d'une stratégie de test destinée à déterminer la sensibilité des circuits sous test.

1.4.1 Moyens de simulation des SEU

Différents moyens de simulation existent pour la caractérisation des circuits intégrés exposés aux radiations. Il s'agit des tests aux ions lourds [Koga96] réalisés à l'aide de :

- sources à fission radioactive ;
- accélérateurs de particules lourdes ;
- accélérateurs à protons ou synchrotrons ;

1.4.2 Système de test

1.4.2.1 Fonctions principales

Un système adapté à la mise en œuvre des tests aux ions lourds doit assurer les fonctions suivantes :

1. La protection du circuit sous test contre les effets des latchups puisque celui-ci risque d'être détruit thermiquement suite à un verrouillage de l'alimentation. Cette protection est généralement réalisée par une surveillance permanente de la consommation du circuit étudié.
2. La détection et l'identification des différentes catégories d'erreurs dues aux up-sets, notamment dans le cas de circuits complexes tels que les microprocesseurs.

1.4.2.2 Architecture adoptée

L'approche de test que nous avons adoptée dans la thèse consiste à utiliser un montage proche de celui d'une application réelle dans lequel le circuit sous test se trouve dans un environnement "naturel" de fonctionnement. Les stimuli de test sont stockés dans une mémoire, qui peut être celle du testeur, sous forme de code objet, donc directement exécutable par le circuit sous test. Ceci sous-entend que ce dernier soit capable d'exécuter un jeu d'instructions (cas des processeurs par exemple). Dans le cas où le circuit de test n'est pas un processeur, un processeur "hôte" ou "maître"

lui a été associé. Ce dernier se charge de l'exécution du programme de test et de la récupération des résultats.

1.4.3 Test d'upset

Dans le cadre de la thèse nous nous sommes intéressés seulement au test "d'upset". Ce dernier a pour objectif d'étudier le comportement du circuit cible face aux effets des radiations. Il consiste en particulier, à détecter et à caractériser les erreurs dues à des upsets afin d'évaluer la sensibilité du circuit sous test. Les séquences de test générées doivent stimuler les parties sensibles impliquées dans les fonctions étudiées. Quant aux expériences, elles peuvent être effectuées en environnement réel ou en environnement simulé.

Deux types de tests d'upsets peuvent être distingués : le *test statique* et le *test dynamique*.

1.4.3.1 Test statique

Ce type de test concerne les circuits qui ont pour fonction la mémorisation des valeurs logiques. Parmi ces circuits on peut citer les mémoires statiques et dynamiques, les registres, etc. que ce soit sous forme de composants individuels ou bien sous forme de bloc dans un circuit complexe (un microprocesseur par exemple). Dans ce dernier cas, le test statique est parfois appelé "test de registres".

Le programme de test correspondant à l'évaluation de la sensibilité aux upsets consiste, dans ce cas, à initialiser les éléments mémoires accessibles du circuit avec des données spécifiques, puis à observer leur contenu après une durée d'attente permettant l'exposition du circuit cible aux radiations. Les valeurs observées seront ensuite comparées aux valeurs initiales afin de détecter des différences (erreurs) dues aux effets des radiations bombardant le circuit.

1.4.3.2 Test dynamique

Les circuits programmables complexes tels que les processeurs sont généralement composés d'éléments de stockage de nature différente (mémoires, latches, registres, accumulateurs, etc.). Certains parmi ces éléments sont accessibles directement par l'utilisateur (mémoires, registres d'adresses et de données), d'autres le sont indirectement (compteur du programme, registre d'instruction) ou sont même inconnus à

l'utilisateur (latches de la partie contrôle, registres temporaires de la partie opérative, etc.).

D'autre part, le test des registres d'un processeur ne suffit pas à l'évaluation de sa sensibilité aux upsets. En effet, un programme correspondant à ce type de test ne stimule qu'une partie des points mémoires contenus dans le circuit [Shaeffer91]. D'où la nécessité d'un autre type de test, appelé "test dynamique", qui permet d'obtenir une couverture des points mémoires plus importante que dans le cas d'un test de registres. Par exemple par l'observation du résultat d'une opération sur l'unité arithmétique et logique (UAL), les différents éléments impliqués dans le fonctionnement de cette dernière seront indirectement testés. En effet, un upset sur ces points peut modifier le résultat observé et peut donc être détecté bien qu'il soit impossible de localiser de manière précise l'élément affecté.

1.4.3.3 Entre le test statique et le test dynamique?

Une comparaison entre les résultats des tests effectués suivant les deux approches statique et dynamique, pour différents circuits et différents types d'applications, montre que la sensibilité découlant d'un test dynamique (mesurée en termes de section efficace) est inférieure à celle qui découle d'un test statique et la différence correspondante peut atteindre parfois deux ordres de grandeur [Estrémé93]. Pour cela, en pratique, lors de tests de circuits programmables, les deux approches sont utilisées. Le test statique permet d'évaluer la sensibilité des différents éléments de mémorisation du circuit accessibles à l'utilisateur, tandis que le test dynamique sert à déterminer la sensibilité d'une application exécutée par le circuit.

1.5 Environnement de test et approches adoptées

Le travail effectué au cours de la présente thèse consiste en le test des circuits digitaux complexes implantant (ou émulant) des réseaux de neurones artificiels face aux effets induits par un environnement radiatif, notamment les upsets. En effet, les réseaux de neurones sont de plus en plus utilisés dans différents domaines d'applications y compris potentiellement le domaine spatial [CNES93]. Il s'avère donc nécessaire d'étudier leur comportement sous radiations. Les approches adoptées pour

cette fin sont les suivantes :

1. La simulation de l'environnement spatial a été réalisée par une source à fission radioactive¹ et par un accélérateur de particules lourdes².
2. La mise en œuvre des tests aux ions lourds a été effectuée par un système de test spécifique. Il s'agit du testeur FUTE16 (Functional and Upset TEster) [Velazco92].
3. Les deux stratégies de test d'upset, statique et dynamique ont été appliquées pour chacun des circuits testés.

Conclusion

Des radiations sont présentes dans l'environnement spatial et sont dues essentiellement à deux types de particules : les particules lourdes et les particules légères. Elles interagissent avec les composants électroniques embarqués et peuvent conduire, sous certaines conditions, à leurs défaillances fonctionnelles. Ces dernières résultent en particulier des effets de la dose cumulée et des effets singuliers des ions lourds et peuvent être catastrophiques sur le système entier, notamment si la partie de l'engin affectée par les particules est vitale (système de contrôle par exemple). D'où la nécessité d'étudier les différents moyens d'actions, au niveau des composants ou au niveau système, permettant de faire face aux effets des rayonnements. Deux techniques sont utilisées, la prévention et la prévision, la première consiste à minimiser la probabilité d'apparition de ces effets tandis que la seconde consiste à calculer le taux d'erreurs en mission des composants candidats, à partir d'une simulation au sol (test aux ions lourds) et d'un moyen de calcul de prédiction. Les tests aux ions lourds nécessitent de disposer de moyens de simulation de l'environnement radiatif, des systèmes de test permettant d'interfacer le circuit sous test ainsi que des stratégies de test destinées à déterminer sa sensibilité. Les circuits testés dans cette thèse sont ceux sur lesquels il est implanté ou est émulé des réseaux de neurones artificiels. Pour cela, un rappel sera fait sur leurs principes, leurs applications et leur propriété de tolérance aux fautes avant la description du travail réalisé et les résultats correspondants obtenus.

1. Il s'agit de l'équipement CIRIL construit à base d'une source radioactive de Californium Cf252 et développé par l'ONERA/CERT/DERTS (Toulouse).

2. L'accélérateur Tandem Vann de Graaf de l'institut de physiques nucléaires – IPN (Orsay).

Chapitre 2

Réseaux de neurones : principes, implantations et applications

Résumé du contenu du chapitre

Les réseaux de neurones artificiels sont des systèmes parallèles et distribués de traitement de l'information inspirés par le fonctionnement du cerveau. Dans la première partie du présent chapitre seront décrits leurs principes de fonctionnement, les différentes architectures, les familles principales d'algorithmes d'apprentissage disponibles ainsi qu'un résumé des principaux avantages des réseaux de neurones en tant que systèmes de traitement de l'information. La deuxième partie sera consacrée aux implantations et aux émulations matérielles des réseaux de neurones (principaux circuits et systèmes existants), tandis que la dernière partie en résume les principaux domaines d'applications et secteurs d'activités.

2.1 Introduction

Les réseaux de neurones artificiels (Artificial Neural Networks – *ANN*) [Haykin94, Rumelhart86b] constituent une approche fondamentalement nouvelle dans le traitement de l'information. Ce sont des systèmes parallèles, adaptatifs et distribués dont le fonctionnement imite celui des réseaux de neurones biologiques tout en reproduisant leurs caractéristiques de base :

1. la connaissance est acquise par le réseau à travers un processus d'apprentissage ;
2. les connexions entre neurones, appelées poids synaptiques, sont utilisées pour le stockage de la connaissance.

Du point de vue structural, un réseau de neurones est composé d'un certain nombre d'unités de traitement simples appelées *neurones formels* ou *artificiels*. Ces derniers sont connectés entre eux de façon à produire la réponse correspondant aux entrées reçues par le réseau. Plusieurs modèles de neurones artificiels ont été développés, s'inspirant du principe de fonctionnement du neurone biologique qui assure essentiellement les fonctions suivantes :

- réception des signaux provenant des neurones voisins ;
- intégration de ces signaux ;
- génération d'une réponse ;
- transmission de celle-ci à d'autres neurones.

L'approche neuronale, parfois appelée "connexioniste", s'oppose à l'approche symbolique basée sur l'hypothèse sur laquelle le raisonnement modélisant la pensée est une combinaison de symboles soumise à des règles logiques. Elle privilégie les avantages suivants :

- l'activité parallèle et en temps réel de nombreux composants ;
- la représentation distribuée des connaissances ;
- l'apprentissage par modification des connexions.

2.2 Bref historique

Ce fut durant les années 40 que les réseaux de neurones sont devenus le sujet d'une recherche scientifique proprement dite. Cependant, jusqu'au milieu des années 80, les progrès réalisés dans ce domaine étaient freinés par le coût élevé des ordinateurs puissants capables d'exécuter les algorithmes correspondants. En effet, de tels algorithmes nécessitent une grande capacité de calcul, et par conséquent, demandent des ressources matérielles importantes, que ce soit à la phase de conception ou à celle d'utilisation du réseau. Depuis lors, des progrès considérables ont été réalisés dans le domaine de la conception digitale permettant la construction d'ordinateurs puissants et performants pouvant remédier aux problèmes de vitesse et de ressources ayant été posés auparavant. Nous assistons alors à un intérêt accru pour la technique des

réseaux de neurones, profitant de sa pluridisciplinarité et de l'échange de nouvelles idées et techniques entre les divers domaines de la science : physique statistique, analyse fonctionnelle, statistiques, théorie des systèmes dynamiques, etc., qui ont fait qu'on assiste à une mutation conceptuelle dans ces domaines. Cette mutation est rendue possible grâce aux efforts conjugués de la biologie, des sciences cognitives et des sciences de l'ingénieur, vu qu'elles sont toutes concernées d'une façon ou d'une autre par cette approche. L'étude des réseaux de neurones est donc devenue une discipline bien établie avec une base théorique solide. D'autre part, les possibilités d'exploitation de cette approche dans le domaine industriel étaient riches et les réseaux de neurones ont marqué, par conséquent, un nombre considérable de succès commerciaux [Angéniol90].

2.3 Modèle de base du neurone : *le neurone formel*

Le modèle couramment utilisé du neurone formel est le suivant : un neurone est une unité de calcul élémentaire recevant ses entrées du milieu extérieur et/ou d'autres neurones et transmettant sa sortie à d'autres neurones et/ou au milieu extérieur. Ces entrées sont pondérées par un *poids synaptique* qui modélise l'intensité des connexions (*synapses*) entre les neurones. Chaque neurone additionne ses entrées, préalablement multipliées par les poids associés, et génère la sortie correspondante à travers une fonction non-linéaire appelée *fonction d'activation*.

Un neurone artificiel, qu'il soit émulé de manière logicielle ou implanté matériellement, comprend donc les éléments suivants :

1. L'ensemble des *connexions (synapses)* permettant la réception des signaux d'entrée et la transmission du signal de sortie. Chacune des connexions est caractérisée par un *poids synaptique*, de façon à ce que le signal transmis par un neurone source soit multiplié par le poids associé à la connexion avant d'être reçu par le neurone destination. Les poids synaptiques peuvent être entiers ou réels, positifs ou négatifs selon que la synapse correspondante soit excitatrice ou inhibitrice respectivement.
2. L'*additionneur* qui réalise la somme des signaux d'entrée pondérés par les poids synaptiques.

3. La *fonction d'activation* qui est une fonction non linéaire de saturation servant à limiter l'amplitude du signal de sortie du neurone. Plusieurs types de fonctions d'activation peuvent être utilisés, les plus courants sont donnés sur la figure 2.1.

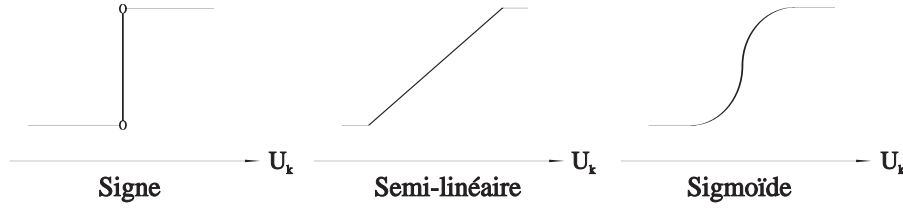


FIG. 2.1 : *Fonctions d'activation d'un neurone*

Du point de vue mathématique, un neurone k est décrit par les 2 équations suivantes :

$$U_k = \sum_{j=1}^n W_{kj} \cdot x_j \quad (2.1)$$

$$y_k = \phi(U_k - \theta_k) \quad (2.2)$$

où

- les x_j et y_k représentent respectivement les signaux d'entrée et la sortie du neurone ;
- les W_{kj} sont les poids des connexions reliant le neurone j (source) au neurone k (destination) ;
- U_k est la somme des signaux d'entrée x_j pondérés par les poids W_{kj} ;
- ϕ est la fonction d'activation ;
- θ_k est le seuil du neurone, il peut être considérée comme une entrée externe ayant pour effet de modifier la valeur d'entrée de la fonction d'activation ;
- $U_k - \theta_k$ est l'entrée totale de la fonction d'activation.

Le schéma de la figure 2.2 décrit le fonctionnement global d'un neurone artificiel.

2.4 Architectures d'un réseau de neurones

Nous entendons par *architecture* ou *topologie* d'un réseau de neurones artificiels la manière selon laquelle les neurones sont organisés. Les structures qui peuvent être

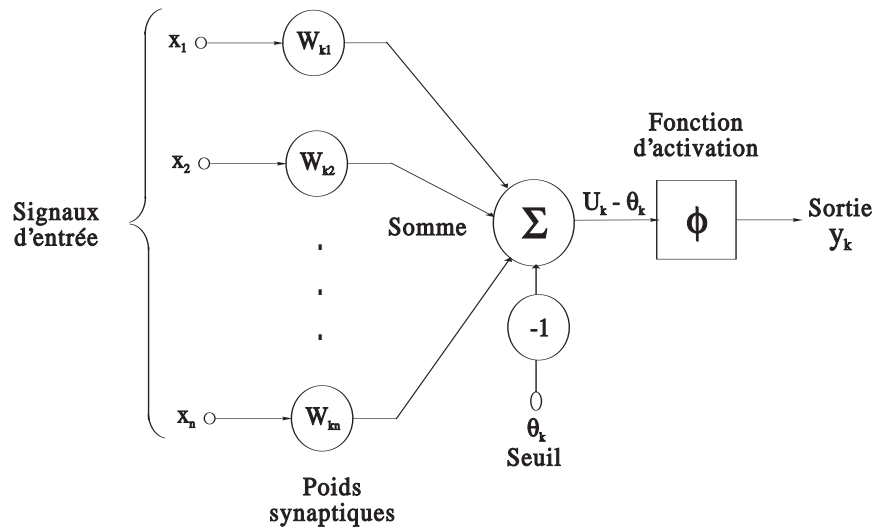


FIG. 2.2 : *Modèle général d'un neurone*

utilisées sont très variées mais beaucoup moins complexes que celles des réseaux de neurones biologiques.

D'une façon générale, l'architecture des réseaux de neurones artificiels peut aller d'une connectivité totale (chacun des neurones du réseau est relié à tous les autres) à une connectivité locale où les neurones ne sont liés qu'à leurs plus proches voisins. Il est courant d'utiliser des réseaux à structure régulière pour faciliter leur utilisation (FIG. 2.3, 2.4 et 2.5).

Deux classes différentes d'architectures de réseaux de neurones peuvent être distinguées :

1. Les réseaux *proactifs* (*feed-forward*).
2. Les réseaux *récurrents*.

2.4.1 Les réseaux proactifs

Cette classe se distingue par l'absence de toute boucle de rétroaction de la sortie vers l'entrée, d'où l'appellation "feed-forward". En d'autres termes, la propagation des signaux s'y fait uniquement dans le sens de l'entrée vers la sortie.

Ce type de réseaux comprend deux groupes d'architectures : les réseaux monocouches et les réseaux multicouches. Ils diffèrent par l'existence ou non de neurones intermédiaires appelés *neurones cachés* entre les unités d'entrées et les unités de sorties appelées *noeuds sources* ou *nœuds d'entrée* et *nœuds de sortie* respectivement.

2.4.1.1 Réseaux proactifs monocouches

Ce type de réseaux possède une *couche d'entrée* recevant les stimuli à traiter par l'intermédiaire des nœuds sources. Cette couche se projette en une *couche de sortie* composée de neurones (nœuds de calcul) transmettant les résultats du traitement au milieu extérieur.

La figure 2.3 montre, par exemple, un réseau proactif monocouche à 4 nœuds d'entrée et 3 nœuds de sortie. La désignation monocouche est attribuée à la couche de sortie (nœuds de calcul). La couche d'entrée n'est pas comptée dans ce sens vu qu'il n'y a pas de calcul fait par les nœuds sources, ils servent uniquement à recevoir les signaux d'entrée et à les transmettre à la couche suivante. Un exemple classique de réseau monocouche est le *perceptron* [Rosenblatt58] qui est un réseau proactif monocouche à une seule sortie. Ce type de réseau ne peut réaliser que la "discrimination de classes d'entrées linéairement séparables".

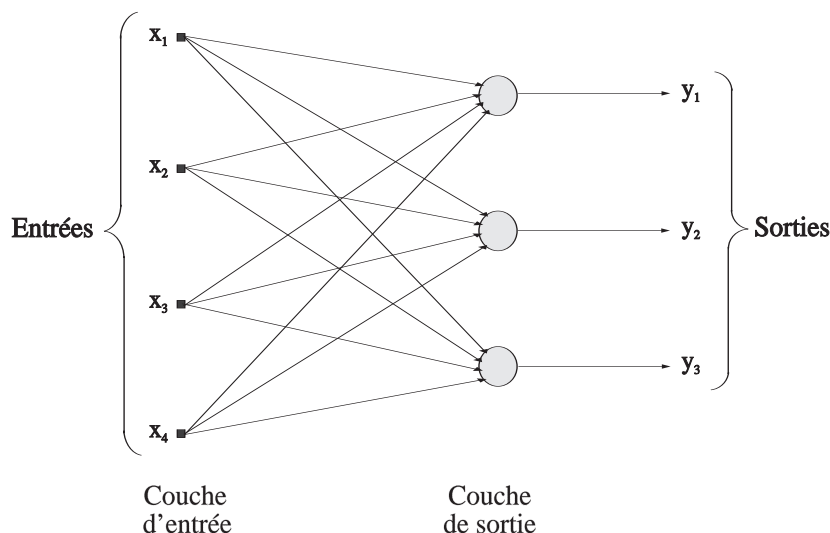


FIG. 2.3 : Réseau proactif monocouche (*perceptron*)

2.4.1.2 Réseaux proactifs multicouches

Ce type de réseaux proactifs se caractérise par la présence d'une ou de plusieurs *couches cachées*, dont les nœuds de calcul correspondants s'appellent *neurones cachés* ou *unités cachées*. Les couches cachées s'interposent entre l'entrée du réseau et sa sortie. Leur rôle est d'effectuer un prétraitement des signaux d'entrée, reçus par la couche d'entrée en provenance du milieu extérieur, et de transmettre les résultats

correspondants à la couche de sortie où sera déterminée la réponse finale du réseau avant qu'elle soit transmise au milieu extérieur.

Ce rôle de prétraitement fait que, en ajoutant une ou plusieurs couches cachées, le réseau est capable d'extraire plus de propriétés statistiques que celles extraites d'un réseau similaire ayant moins de couches cachées. Ceci est utile pour réaliser des fonctions plus complexes que de simples séparations linéaires.

Dans ce type de réseaux, les entrées des neurones d'une couche particulière proviennent uniquement des sorties de la couche adjacente précédente. Les réseaux les plus fréquemment utilisés de cette catégorie sont les perceptrons multicouches (Multilayered Perceptrons – *MLP*) [Rosenblatt62].

Dans la figure 2.4 est montré l'exemple d'un réseau à une seule couche cachée comportant 8 unités d'entrée, 4 unités cachées et 3 unités de sortie (réseau 8-4-3). Ce réseau est dit *complètement connecté* dans le sens que chaque noeud d'une couche est connecté à tous les noeuds de la couche adjacente suivante. Si éventuellement, des connexions manquaient entre des neurones de deux couches voisines, le réseau serait dit *partiellement connecté*.

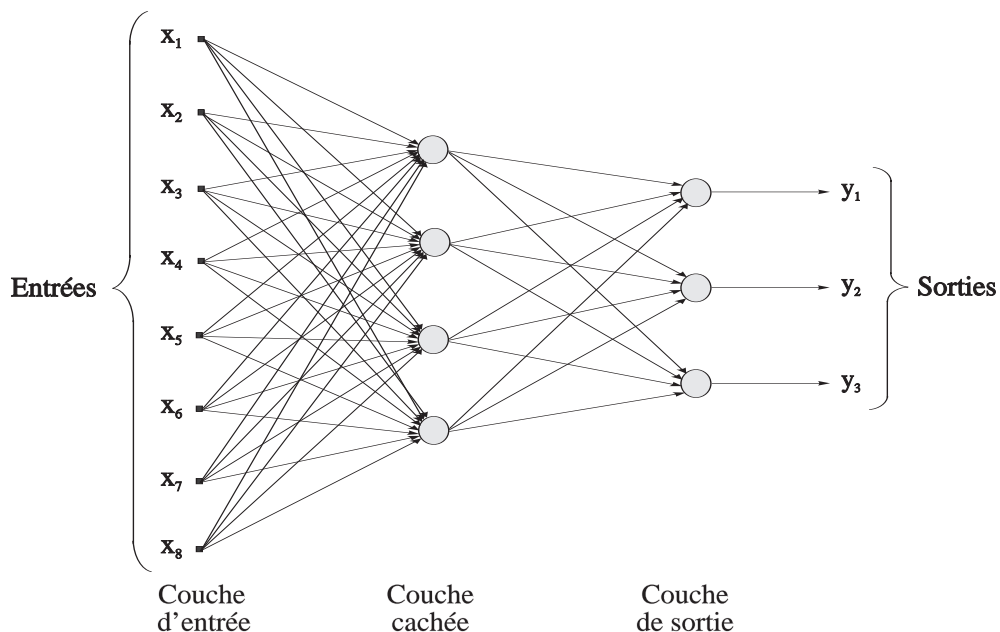


FIG. 2.4 : Réseau proactif complètement connecté avec une seule couche cachée

2.4.2 Réseaux récurrents

Les réseaux *récurrents* se distinguent des réseaux proactifs par le fait qu'ils contiennent au moins une boucle de contre-réaction des nœuds de sortie vers les nœuds d'entrée (ou, au moins d'une couche vers une couche précédente, adjacente ou non).

Dans la figure 2.5 est illustré l'exemple d'un réseau de neurones récurrent ayant 2 unités d'entrée, 3 unités cachées et 2 unités de sortie. Dans ce réseau, les connexions de rétroaction proviennent aussi bien des unités cachées que des unités de sortie.

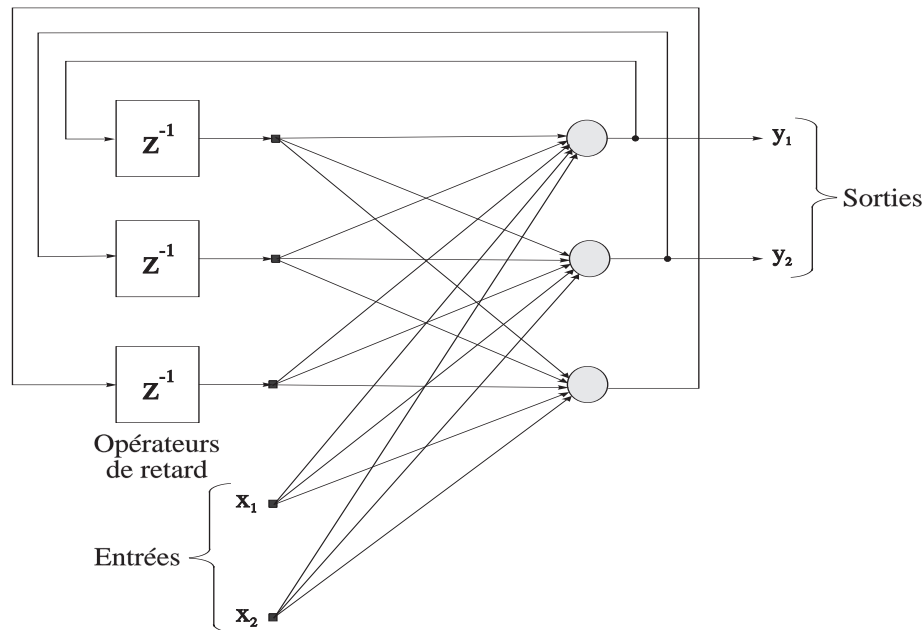


FIG. 2.5 : Réseau récurrent avec neurones cachés

Contrairement aux réseaux proactifs, les propriétés dynamiques du réseau récurrent sont importantes. En effet, la présence des boucles de contre-réaction implique l'insertion, dans le graphe correspondant à la topologie du réseau, de branches particulières composées d'*opérateurs de retard* (les blocs z^{-1} de la figure 2.5). Ces derniers résultent en un comportement dynamique non linéaire du réseau dû à la nature non linéaire des neurones. Dans certains cas, les états des neurones subissent un processus de relaxation faisant évoluer le réseau vers un état stable dans lequel ces états ne peuvent plus changer. Dans d'autres cas, le changement des états des neurones de sortie est important de façon à ce que le comportement dynamique constitue la sortie du réseau. Des exemples de ce type de réseaux sont le réseau de Hopfield [Hopfield82] et la machine de Boltzmann [Ackley85].

2.5 Apprentissage d'un réseau de neurones

Parmi les propriétés importantes des réseaux de neurones, nous pouvons citer leur capacité à *apprendre* de leur environnement et à *améliorer* leur performance par apprentissage. Ce dernier se fait à travers un processus itératif d'ajustement appliqué aux poids synaptiques et aux seuils. Intuitivement, le réseau devient de mieux en mieux informé sur son environnement après chaque itération du processus d'apprentissage. Le type d'apprentissage est déterminé par la manière selon laquelle les paramètres du réseau (poids et seuils) sont changés.

Le processus d'apprentissage implique la séquence suivante :

1. Le réseau de neurones est *stimulé* par son environnement.
2. Le réseau de neurones *subit des changements* suite à ce stimulus.
3. Le réseau de neurones *répond d'une façon nouvelle* à l'environnement à cause du changement qui a eu lieu dans sa structure interne.

L'ensemble des règles permettant d'adapter un réseau de neurones à un problème donné est appelé *algorithme d'apprentissage*. Plusieurs algorithmes d'apprentissage peuvent être trouvés dans la littérature, chacun ayant ses propres avantages et inconvénients. Fondamentalement, les algorithmes d'apprentissage diffèrent par la méthode d'ajustement des poids synaptiques et des seuils, mais la plupart d'entre eux sont basés sur la minimisation d'une fonction d'erreur appelée fonction de coût.

L'apprentissage se fait à partir d'une base d'exemples du problème traité. Il s'agit essentiellement de modifier les paramètres du réseau (notamment les poids synaptiques et seuils) pour qu'il donne la bonne réponse aux stimuli d'entrée de la base d'exemples qui lui sont présentés.

Deux principales stratégies d'apprentissage peuvent être distinguées :

1. L'apprentissage *supervisé*, appelé aussi *actif* ou *associatif* [Solla90], qui consiste à présenter au réseau des couples (entrée, réponse attendue). A chaque pas d'apprentissage, les poids des synapses sont modifiés selon un algorithme donné pour réduire une fonction d'erreur (moindres carrées par exemple). Parmi les algorithmes les plus répandus correspondant à cette stratégie nous pouvons citer *la règle Delta* [Widrow85] et *la rétro-propagation des erreurs* (back-propagation – *B.P.*) [Rumelhart86a].

2. *L'apprentissage non supervisé* ou *compétitif*, appelé aussi *auto-organisation* [Kohonen90] dans lequel seuls les vecteurs d'entrée sont fournis au réseau. Durant l'apprentissage, les vecteurs d'entrée sont groupés en classes par extraction des propriétés statistiques de l'ensemble des vecteurs d'apprentissage. Contrairement au cas supervisé, il n'y a pas d'ensembles préalables de catégories dans lesquelles les motifs sont classés. Le système doit plutôt développer sa propre représentation des entrées. Les algorithmes les plus connus dans ce type d'apprentissage sont ceux de Kohonen [Kohonen88] et de Hopfield [Hopfield82].

Une fois que la phase d'apprentissage est complétée avec succès, le réseau pourra être utilisé pour obtenir la réponse à des vecteurs d'entrée *inconnus* c'est à dire non rencontrés durant l'apprentissage, ce qui constitue la phase dite de *rappel* ou de *généralisation*. Cette capacité, de réagir correctement à des motifs d'entrée non appris, donnerait aux réseaux de neurones une robustesse intrinsèque face aux fautes survenant sur le motif d'entrée. Par contre, les conséquences des modifications d'autres paramètres du réseau (poids synaptiques, seuils et fonction d'activation des neurones) sont moins prévisibles sans la réalisation d'une étude détaillée, ce qui sera fait dans la suite de ce manuscrit.

2.6 Modes de fonctionnement d'un ANN

Les deux principaux modes de fonctionnement sont le mode parallèle et le mode séquentiel. Ils se distinguent par l'ordre du calcul des sorties des différents neurones ; ce calcul étant réalisé en fonction des entrées à l'instant précédent :

1. Le mode parallèle. A un instant donné, tous les neurones calculent leur nouvelle sortie et la transmettent aux neurones auxquels ils sont connectés en tant que neurones émetteurs.
2. Le mode séquentiel. A un instant donné, un seul neurone calcule sa nouvelle sortie et la transmet aux neurones qui lui sont connectés. Ce mode peut se faire suivant une liste cyclique prédéfinie ou suivant un choix aléatoire du neurone qui exécutera ses calculs.

Des modes mixtes peuvent exister et sont des combinaisons des deux modes précédents.

2.7 Exemple d'un réseau de neurones

Afin d'illustrer le fonctionnement d'un réseau de neurones prenons l'exemple du réseau de la figure 2.6. Il s'agit en effet d'un perceptron multicouche (MLP) ayant deux entrées E_1 et E_2 , deux neurones cachés N_{c1} et N_{c2} et un neurone de sortie N_s . Ce réseau réalise le OU exclusif de ses entrées, l'ensemble des poids W_{ij} et seuils S_i correspondants (TAB. 2.1) ayant été calculé par l'algorithme de retropropagation de l'erreur (BP).

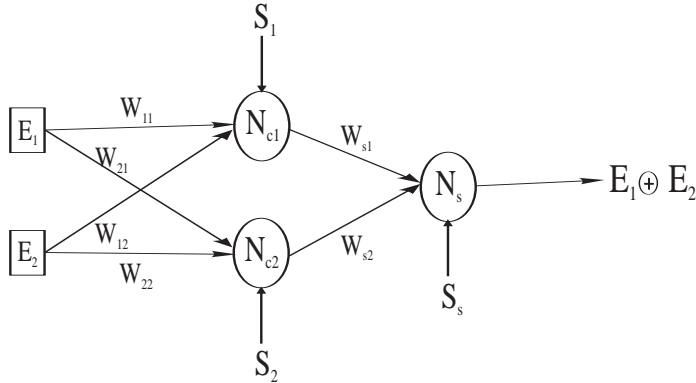


FIG. 2.6 : Réseau d'évaluation du OU exclusif de deux variables

Neurone d'entrée N_1	$W_{11} = 6.45$	$W_{12} = 6.42$	$S_1 = -2.78$
Neurone d'entrée N_2	$W_{21} = 4.30$	$W_{22} = 4.30$	$S_2 = -6.81$
Neurone de sortie N_s	$W_{s1} = 8.78$	$W_{s2} = -9.33$	$S_s = -3.83$

TAB. 2.1 : Valeurs des poids et seuils du réseau XOR

Les sorties des trois neurones N_{c1} , N_{c2} et N_s sont calculées à partir des équations 2.3, 2.4 et 2.5 respectivement.

$$Out_{N_1} = \phi(W_{11}E_1 + W_{12}E_2 + S_1) \quad (2.3)$$

$$Out_{N_2} = \phi(W_{21}E_1 + W_{22}E_2 + S_2) \quad (2.4)$$

$$Out_{N_s} = \phi(W_{s1}Out_{N_1} + W_{s2}Out_{N_2} + S_s) \quad (2.5)$$

La fonction d'activation choisie ϕ est de type sigmoïde (eq. 2.9). Les valeurs de sorties correspondant aux différentes combinaisons d'entrées (E_1 , E_2) sont données dans le tableau 2.2.

E_1	E_2	Out_{N_1}	Out_{N_2}	Out_{N_s}
0	0	0.06	0.00	0.03
0	1	0.97	0.08	0.98
1	0	0.97	0.08	0.98
1	1	1.00	0.86	0.05

TAB. 2.2 : Valeurs des sorties des neurones pour différentes combinaisons d'entrée

2.8 Implantations des réseaux de neurones

Les implantations des réseaux de neurones artificiels peuvent être divisées en 3 catégories principales :

- simulation logicielle ;
- émulation matérielle ;
- implantation matérielle.

La différence entre les 2 dernières catégories n'est pas nettement dessinée. Pour cela, dans ce qui suit, nous allons utiliser le terme *simulation logicielle* pour décrire une exécution par programme (notamment par ordinateur). Les implantations des réseaux de neurones sur des machines à usage général monoprocesseur ou multiprocesseurs seront désignées par le terme *émulation matérielle*. En effet, le terme émulation est utilisé, en conception informatique, pour exprimer l'exécution, par un système, des programmes qui ont été écrits pour un autre système [Mallach75]. Enfin, le terme d'*implantation matérielle* sera réservé aux systèmes et circuits neuronaux dédiés (spécifiquement conçus pour exécuter des réseaux de neurones).

2.8.1 Réseaux de neurones : du logiciel au matériel

La plupart des recherches effectuées dans le domaine des réseaux de neurones artificiels, ainsi qu'un nombre non négligeable de leurs applications ont été réalisés par l'intermédiaire des simulations logicielles sur des ordinateurs séquentiels (PC, stations de travail).

Un des principaux avantages de ces *neurosimulateurs* est leur flexibilité. En effet, la plupart d'entre eux, peuvent être utilisées pour différentes topologies (perceptrons

multicouches, réseaux de Hopfield, etc.), et différents types d'apprentissages (supervisé ou non supervisé). Des paramètres tels que le type de la fonction d'activation utilisée, le nombre de couches dans le réseau et le nombre de neurones par couche, peuvent facilement être adaptés aux besoins de l'utilisateur, une bonne partie de l'effort étant en fait consacrée à l'interface homme-machine. En effet, différents types de présentations, numériques et graphiques, permettent de donner à l'utilisateur, un aperçu clair et rapide du comportement et de la performance du réseau simulé ainsi que de l'algorithme d'apprentissage appliqué. L'existence de ces neurosimulateurs, à la fois puissants et conviviaux, peut être considérée comme l'une des causes principales de l'intérêt actuel pour les réseaux de neurones et des progrès énormes qui y sont réalisés.

Cependant, l'utilisation des ordinateurs séquentiels pour la simulation des réseaux de neurones semble être paradoxale. La nature des ordinateurs digitaux séquentiels est, en effet, très différente de celle des réseaux de neurones biologiques qui sont à la base de l'approche neuronale. En effet, un réseau de neurones biologiques peut fonctionner d'une façon non algorithmique en absence d'un modèle précis, alors que les opérations d'un ordinateur digital sont prescrites par un programme logiciel bien précis. Le cœur d'un ordinateur digital, l'unité centrale (CPU), peut exécuter d'une façon précise des tâches relativement complexes, tandis que le neurone, étant un bloc élémentaire d'un réseau complet, effectue des opérations plutôt simples.

D'autre part, le mot CPU (Central Processing Unit) veut dire que le contrôle dans un ordinateur digital est centralisé, ce qui s'oppose au cas des réseaux de neurones biologiques dans lesquels les blocs constituants fonctionnent d'une façon "autocratique" et indépendante. De même la mémoire dans un ordinateur avec ses données localisées dans des positions adressables, est organisée d'une façon différente de celle d'une mémoire associative distribuée utilisant les connexions variables d'un réseau de neurones biologiques (mémoire adressable par le contenu – *CAM*).

En ce qui concerne la vitesse, les ordinateurs digitaux actuels travaillent avec des cycles d'horloge d'environ une dizaine de nanosecondes alors qu'un neurone biologique possède un temps de réponse de l'ordre de la milliseconde¹.

1. La performance d'une application s'exécutant sur un ordinateur digital dépend de la complexité des traitements correspondants et ne peut donc pas être donnée en termes de temps de réponse. De même, la performance d'une application neuronale dépend de la taille du réseau correspondant (nombre de neurones et de connexions). Ceci explique la différence d'échelle lors de la comparaison des performances.

Cet apparent manque de vitesse dans un réseau de neurones biologiques est compensé par le très grand nombre de neurones constituant : un cerveau humain est un réseau parallèle composé d'environ $10^{11} - 10^{12}$ neurones ayant chacun 10^3 synapses environ. Dans un ordinateur digital, la plupart des opérations sont effectuées d'une façon séquentielle en utilisant un multiplexage temporel des ressources.

La simulation de certaines applications de réseaux de neurones est parfois trop exigeante du point de vue temps de calcul et de réponse pour plusieurs raisons :

1. La plupart de ces applications ont des architectures composées de quelques dizaines jusqu'à quelques centaines de neurones connectés par des centaines, voire des milliers de synapses. Ceci se manifeste par le besoin d'effectuer des opérations de sommation et des non linéarités pour les premiers et des opérations de multiplications pour les dernières. Le nombre de ces opérations augmente avec celui des neurones et de connexions, ce qui augmente le temps de calcul aussi bien à l'apprentissage qu'à l'utilisation du réseau.
2. Afin d'apprendre de manière plus efficace les réseaux de neurones, une grande base de vecteurs d'apprentissage (plusieurs milliers d'exemples du problème traité) doit être disponible. Les algorithmes d'apprentissage existants procèdent par une mise à jour itérative des poids synaptiques, par l'ajout ou la suppression systématiques de petits incréments, ce qui pourrait être trop lent surtout si l'on dispose d'une grande base d'apprentissage.
3. Certains algorithmes d'apprentissage ne garantissent qu'une solution localement optimale et nécessitent, par conséquent, plusieurs exécutions, commençant chacune par un ensemble de poids initiaux différents, avant qu'un comportement acceptable soit atteint.
4. Nombreuses parmi ces applications ont des contraintes temporelles faisant que le réseau de neurones doit être utilisé "en ligne".

D'après ce qui précède, les simulations logicielles des réseaux de neurones ne sont pas toujours bien adaptées aux applications exigeant une rapidité de réponse telles que les applications mettant en jeu des contraintes temps réel (traitement de parole, contrôle, etc.), ce qui rend inévitable, l'utilisation des émulations et des implantations matérielles. Ces 2 dernières permettent une meilleure prise en compte du parallélisme des réseaux de neurones (plusieurs processeurs simples travaillant ensemble) et

peuvent s'exécuter plus vite (parfois même de plusieurs ordres de grandeur) que les simulations logicielles [Lindsey95].

L'émulation des réseaux de neurones par des processeurs RISC (Reduced Instruction Set Computer [Patterson85]) est suffisamment rapide pour certaines applications, surtout avec la très grande vitesse offerte par les nouveaux processeurs RISC. Cependant, pour certaines applications, cette vitesse n'est pas suffisante à cause du niveau de performance demandé qui ne peut pas être atteint par les processeurs RISC actuels. D'où le recours aux implantations matérielles VLSI (Very Large Scale Integration) dédiées aux réseaux de neurones. Celles-ci offrent une vitesse suffisante pour exécuter des applications avec des contraintes temps réel assez fortes tout en permettant, selon la conception, d'effectuer l'apprentissage (même pour le cas de réseaux très complexes) de façon plus rapide que celle d'un processeur standard (RISC ou CISC).

Cependant, la conception des composants neuronaux avec, à la fois, un grand nombre de processeurs et une connectivité importante, est difficile et parfois irréalisable avec les technologies existantes. Les circuits VLSI de nos jours peuvent implanter typiquement 50 – 100 neurones interconnectés. Pour construire des implantations matérielles plus grandes, des cartes multicircuits et même des systèmes multicartes sont envisageables. Ceci offre la possibilité d'avoir quelques milliers de neurones interconnectés, mais nécessite l'adjonction de circuits séquentiels de contrôle ainsi que des ports de communications.

Actuellement, nous pouvons trouver des circuits neuronaux aussi bien commerciaux que prototypes (notamment des prototypes universitaires dont la plupart ne sont pas arrivés au stade de commercialisation). La gamme des produits disponibles est large : elle comporte des circuits digitaux, analogiques, hybrides ou optoélectroniques, des cartes accélératrices et même des neuro-ordinateurs multicartes. Cependant, il n'y en a pas beaucoup qui satisfassent les contraintes propres à une spécification particulière (architecture, technologie, etc.). Par exemple, si l'on demande un circuit analogique pouvant implanter un réseau de 50 – 100 neurones et s'exécutant en moins de 10 μs , seulement un ou deux produits sont disponibles. Une information plus détaillée sur les implantations matérielles des réseaux de neurones peut être trouvée en [Holler91, Graf93, Lindsey94, Glesner94].

2.8.2 Caractéristiques des réseaux de neurones matériels

2.8.2.1 Spécifications et critères de performance

Les spécifications principales d'un réseau de neurones consistent en :

- l'architecture du réseau (feed-forward, récurrent, etc.);
- le nombre d'entrées et de sorties;
- le nombre de neurones du réseau;
- le nombre de synapses par neurone;
- le nombre de couches (si feed-forward).

Pour une implantation matérielle, d'autres spécifications viennent s'ajouter aux précédentes, à savoir :

- la technologie utilisée (analogique, digitale ou hybride);
- le nombre maximum de neurones;
- le nombre de connexions supporté par le circuit;
- la précision (en nombre de bits) des entrées et/ou sorties, des poids, des accumulateurs et des multiplieurs

Plusieurs grandeurs sont utilisées pour indiquer la performance matérielle, parmi lesquelles, la plus utilisée est le nombre de connections par seconde (CPS) définie comme étant la vitesse des opérations de multiplication et d'accumulation pendant la phase de rappel. Une autre grandeur, similaire à cette dernière, est utilisée pour indiquer la performance durant la phase d'apprentissage; il s'agit du CUPS (Connection Update Per Second) indiquant le nombre de mise à jour par seconde des poids durant l'apprentissage. Cependant, vu la grande variété des architectures de réseaux de neurones et leurs implantations matérielles, ces mesures ne peuvent pas toujours donner une idée claire sur les capacités matérielles d'un circuit neuronal. Une normalisation de la valeur du CPS (resp. CPUS) par le nombre de poids supporté par le circuit N_w , CPSPW (resp. CUPSPW), a été proposée dans [Holler91] comme étant un moyen efficace pour indiquer la puissance de traitement du circuit (équation 2.6 et 2.7). De même, la grandeur CPPS (Connection Primitives Per Second) a été proposée

dans [Vankeulan94] afin d'indiquer la précision dans la performance du traitement en fonction des nombres de bits nécessaires pour le codage des entrées et des poids (b_{in} et b_w resp.) (équation 2.8).

$$CPSPW = CPS/N_w \quad (2.6)$$

$$CUPSPW = CUPS/N_w \quad (2.7)$$

$$CPPS = b_{in} \times b_w \times CPS \quad (2.8)$$

2.8.2.2 Principales considérations de conception

Les propriétés fonctionnelles des transistors CMOS (Complementary Metal Oxide Silicon), leur facilité d'utilisation et leur faible coût font de la technologie CMOS un bon choix pour les implantations VLSI des réseaux de neurones. Cependant, qu'il s'agisse de la technologie CMOS ou bien d'une autre technologie, certaines considérations de conception doivent être prises en compte lors du développement des circuits neuronaux aussi bien dédiés qu'à usage général. Parmi ces considérations nous pouvons citer :

1. Calcul de la somme des produits. C'est une exigence fonctionnelle commune au fonctionnement de tous les neurones. Elle implique la multiplication de l'entrée de chaque connexion par la valeur correspondant à son poids, ensuite l'addition des entrées pondérées par les poids (équation 2.1).
2. Représentation des données. D'une façon générale, les réseaux de neurones n'ont pas besoin d'une grande précision dans la représentation de données. Cette caractéristique dépend essentiellement de l'algorithme d'apprentissage et de l'application considérée.
3. Calcul de la sortie. La fonction d'activation la plus fréquemment utilisée pour le calcul de l'état d'un neurone est une fonction lisse non linéaire du type tangente hyperbolique appelée sigmoïde (équation 2.9). D'autres fonctions, moins souvent utilisées, sont la fonction semi-linéaire et la fonction signe (FIG. 2.1).

$$\phi_{sigmoïde}(u) = \frac{1}{1 + \exp(-u)} \quad (2.9)$$

4. Complexité d'apprentissage. Chaque algorithme d'apprentissage possède ses propres exigences. Plusieurs algorithmes utilisent des calculs *locaux* pour la modification des poids synaptiques d'un réseau de neurones, ce qui est fortement

désirable du point de vue implantation. D'autres algorithmes, tels que la retro-propagation des erreurs, ne le font pas, ce qui impose une charge supplémentaire sur l'implantation du réseau surtout quand la précision de calcul souhaitée est importante.

5. Stockage des poids. Cette considération représente le besoin du stockage des "anciennes" valeurs des poids synaptiques du réseau. Les "nouvelles" valeurs des poids sont calculées par une mise à jour des anciennes valeurs par l'addition/soustraction d'incrémentés déterminés par l'algorithme d'apprentissage.
6. Coût d'implantation. Les coûts totaux impliqués par l'implantation d'un réseau de neurones doivent être pris en compte lors de la production d'un circuit en Silicium. Les facteurs à considérer sont les suivants [Haykin94]:
 - bande passante demandée au niveau d'entrée/sortie ;
 - consommation électrique ;
 - flexibilité d'utilisation pour différentes gammes d'applications ;
 - choix des technologies d'implantation : analogiques, digitales ou mixtes ;
 - précision des données, en particulier pendant l'apprentissage.

2.8.3 Réseaux de neurones en VLSI

Les circuits neuronaux existants se distinguent par leur technologie d'implantation, leur architecture et par l'intégration ou non du processus d'apprentissage. Dans ce qui suit, nous classerons les implantations matérielles des réseaux de neurones selon la technologie de réalisation. Ceci nous mène à distinguer 4 groupes principaux :

1. implantations digitales ;
2. implantations analogiques ;
3. implantations hybrides (analogiques/digitales) ;
4. implantations optoélectroniques.

Ces approches seront décrites brièvement dans les paragraphes suivants. Des circuits représentatifs pour chacune d'entre elles sont donnés dans l'annexe A avec la bibliographie correspondante. Des descriptions plus ou moins détaillées peuvent être trouvées dans [Lindsey95, Heemskerk95, CERN96].

2.8.3.1 Implantations digitales

Du point de vue conception, la technologie digitale VLSI, parmi les quatre citées, possède les avantages suivants :

- maturité des techniques de fabrications ;
- possibilité de fabriquer des circuits intégrés avec des dizaines de millions de transistors sur une seule puce de Silicium ;
- facilité d'obtenir une précision quelconque.

A ces avantages s'ajoutent, pour le cas des réseaux de neurones :

- stockage des poids en RAM ;
- possibilité d'implanter un grand nombre de neurones identiques pouvant travailler en parallèle, sur un seul circuit, rendant possible l'exploitation du parallélisme inhérent des réseaux de neurones ;
- conception et implantation faciles des circuits neuronaux à cause de la structure régulière des réseaux de neurones et le nombre relativement petit des opérations arithmétiques compliquées, aussi bien durant l'apprentissage que durant le rappel.

Du point de vue utilisation, les circuits neuronaux digitaux sont facilement incorporés dans les systèmes existants et relativement simples à utiliser. Cependant, les opérations digitales sont, en général, plus lentes que celles des systèmes analogiques, surtout dans la multiplication $poids \times entrée$, et les entrées analogiques doivent être converties en numérique.

La majorité des circuits neuronaux existants sont digitaux, la plupart d'entre eux utilisent la technologie CMOS. A l'intérieur de ce groupe, nous pouvons distinguer plusieurs sous-catégories comprenant des architectures à tranches de bits et des circuits multiprocesseurs.

2.8.3.1.1 Architectures en tranches

Suivant le même principe de l'architecture en tranches des processeurs digitaux classiques, les circuits neuronaux en tranches offrent des blocs élémentaires permettant la construction de réseaux de neurones ayant une taille et une précision arbitraire.

De tels circuits fonctionnent à vitesse modérée et n'intègrent pas, en général, de processus d'apprentissage. Parmi les circuits appartenant à cette famille nous pouvons citer le NLX420 de NeuraLogix et le L-Neuro 1.0 de Philips (annexe A.1.1).

2.8.3.1.2 Architectures multiprocesseurs

Une approche plus élaborée pour construire des réseaux de neurones d'une façon matérielle consiste à intégrer plusieurs processeurs élémentaires dans un même circuit. Chacun d'entre eux peut être assimilé à un processeur très simple ayant sa propre UAL, ses propres registres et éventuellement sa propre mémoire cache. Deux architectures principales peuvent être distinguées : l'architecture SIMD (Single Instruction Multiple Data) et celle à tableaux systoliques. Pour la conception SIMD [Flynn66], chaque processeur exécute la même instruction en parallèle avec les autres processeurs, mais pour une donnée différente. Dans les architectures systoliques [Kung80], un processeur accomplit une étape du calcul (en général la même) avant de passer, en pipeline, ses résultats au processeur suivant. Chaque processeur communique uniquement avec ses voisins les plus proches travaillant ainsi sur une information locale (spatiale et temporelle).

A titre d'exemples, les circuits SIMD suivants peuvent être donnés, le N64000 d'Inova et le CNAPS-1064 d'Adaptive Solutions, tandis que pour les architectures à tableaux systoliques, nous pouvons citer le MA-16 de Siemens AG (annexe A.1.2).

2.8.3.1.3 Autres implantations digitales

De nombreux autres circuits neuronaux existent et ne correspondent pas forcément aux catégories décrites ci-dessus, parmi lesquels nous pouvons citer le NISP-MT19003 de Microcircuit Engineering Corporation et le RSC-164 de Sensory circuit (annexe A.1.3).

2.8.3.2 Implantations analogiques

Les opérations de base exécutées par un réseau de neurones, multiplication, addition et association non linéaire, peuvent, en principe, être facilement réalisées par des circuits analogiques très simples :

- la multiplication peut être réalisée par la loi d'Ohm :

$$u = i \times R \tag{2.10}$$

- l'addition peut être obtenue par la loi des courants de Kirchhoff :

$$i_{sortie} = \sum i_{entrée} \quad (2.11)$$

- un simple étage différentiel est capable de réaliser une fonction lisse non linéaire proche de la sigmoïde (équation 2.9).

Les propriétés de l'électronique analogique (surface petite, faible consommation et grande vitesse) représentent un grand avantage pour implanter des réseaux de neurones. Leur inconvénient potentiel pourrait être leur manque de précision, qui est d'une importance secondaire dans le cas des réseaux de neurones, au moins pour la phase de rappel, à cause de leur tolérance aux fautes (chapitre 3). Pour la phase d'apprentissage, le manque de précision peut devenir un problème à cause de la sensibilité du processus d'apprentissage face aux variations des poids. Par exemple, l'algorithme de la retro-propagation de l'erreur suppose l'implantation des poids avec une précision relativement grande. De plus les décalages de tension pouvant avoir lieu dans les implantations analogiques peuvent corrompre l'apprentissage d'une façon significative. D'autres problèmes se posent, tels que le stockage analogique des poids synaptiques des neurones, la nécessité de compenser les variations de température, le contrôle de la tension d'alimentation, dont la résolution n'est pas toujours évidente et fait appel à des outils de conception et de fabrication sophistiqués. Pourtant, et malgré tous les inconvénients précédents, l'électronique analogique reste toujours un bon choix pour implanter des réseaux de neurones. Ceci explique la poursuite des recherches scientifiques et industrielles dans le domaine des implantations matérielles analogiques de réseaux de neurones.

Parmi les circuits neuronaux analogiques, nous pouvons citer l'ETANN d'Intel et le LSANC de l'université de Pennsylvania (annexe A.2).

2.8.3.3 Implantations hybrides

Dans des paragraphes précédents, nous avons vu que les 2 approches d'implantation analogique et digitale, ont, toutes les deux leurs propres avantages et inconvénients. Les points forts en faveur des implantations digitales sont surtout leur précision et leur flexibilité, tandis que la conception analogique peut donner lieu à des circuits très denses, permettant une implantation complètement parallèle de réseaux de grande taille, ce qui se manifeste par des vitesses de traitement élevées. Pour cela,

les concepteurs essaient de combiner ces 2 approches, sous forme d'implantations hybrides plus ou moins optimales, et ceci dans le but d'exploiter leurs avantages et de circonvenir leurs inconvénients [Murray91]. Pour la plupart des circuits hybrides à apprentissage intégré, la majorité du réseau est en analogique (entrées, sorties et traitements intermédiaires), tandis que le processus d'apprentissage, quant à lui, est implanté en digital ce qui permet d'avoir une précision arbitraire dans les poids et une flexibilité permettant le choix éventuel entre différents algorithmes. Parmi les circuits représentatifs de cette catégorie nous pouvons citer l'ANNA d'AT&T et le CLNN-32 de Bellcore (annexe A.3).

D'autre part, l'utilisation des techniques digitales est avantageuse pour les communications entre neurones à cause de l'immunité au bruit des signaux digitaux et leur facilité de transmission et de régénération. Les signaux de sorties des neurones sont modulés avant d'être transmis à d'autres circuits. La technique utilisée est celle à flux impulsionnel (Pulse Stream – *PS*) [Hamilton92] bien connue dans le domaine des télécommunications. Cette technique est, en quelque sorte, inspirée du fonctionnement des neurones biologiques, lesquels interagissent entre eux par l'intermédiaire de trains impulsionnels. Dans cette technique, les signaux analogiques sont représentés sous formes de séries d'impulsions. Ceci se manifeste par de nombreux avantages concernant la consommation électrique, le calcul et la propagation. Les calculs neuronaux fondamentaux sont effectués par des portes logiques simples : portes OU pour l'addition et portes ET pour la multiplication. Les caractéristiques du signal porteur varient en accord avec l'état du neurone transmis. Plusieurs types de modulations impulsionnelles existent selon la partie du signal affectée par la porteuse [Murray91]:

- modulation impulsionnelle d'amplitude ;
- modulation impulsionnelle de phase ;
- modulation impulsionnelle de fréquence.

La technique à flux impulsionnel n'est pas aussi précise que les techniques purement digitales. Cependant [Murray94a] a montré que ceci ne conduit pas à une restriction d'applicabilité dans le domaine des implantations neuronales. Un des circuits représentatifs de cette technique est l'Epsilon (annexe A.3).

2.8.3.4 Implantations optoélectroniques

Vu que dans la plupart des topologies des réseaux de neurones, le nombre de connexions est beaucoup plus grand que le nombre de neurones, l'implantation des connexions devient de plus en plus difficile avec l'augmentation du nombre de neurones du réseau étudié. Les concepteurs des circuits neuronaux doivent faire face à l'augmentation éventuelle de la surface de leur circuit, mais aussi doivent trouver des solutions aux problèmes de diaphonie entre interconnexions (crosstalk), de consommation et de bande passante. Une des solutions possibles est l'utilisation d'un modulateur de lumière spatial programmable. Cette approche permet de construire des réseaux d'interconnexion tridimensionnels à faible consommation avec une bande passante très grande et une diaphonie négligeable.

La plupart des implantations optiques utilisent le modèle de Hopfield [Hopfield82] à cause de la simplicité du réseau (totalement connecté) et de l'absence de dynamique des connexions, l'utilisation d'autres modèles restant toujours possible.

Le principe d'implantation d'un réseau de neurones optoélectronique est illustré sur la figure 2.7 [Daval090].

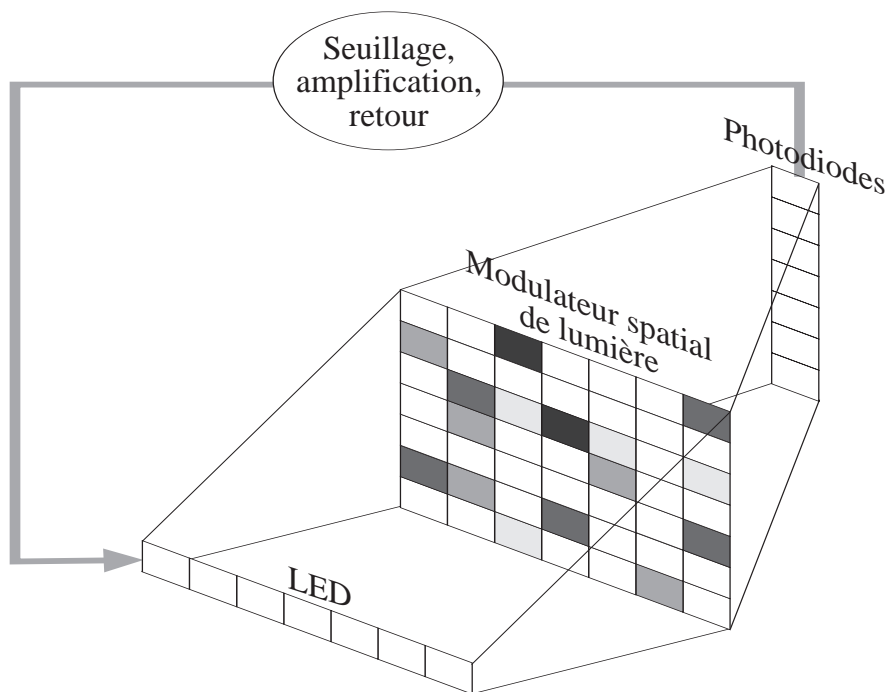


FIG. 2.7 : Principe d'un réseau de neurones optoélectronique

Sur la figure 2.7, les neurones sont représentés par un tableau linéaire de diodes électroluminescentes (LED). Les rayons lumineux provenant de ces LED sont focalisés

par une optique sur un modulateur de lumière : il s'agit d'un masque photographique dont la transparence n'est pas uniforme, et dont la valeur en un point est liée à l'application d'un potentiel électrique en ce point. Cet écran n'est que la matrice des poids implantée sous forme optoélectronique. Sa transparence (valeurs des poids) peut, en particulier, être déterminée à l'aide d'un micro-ordinateur.

Chacune des LED illumine une colonne du masque de la matrice des poids. La lumière qui traverse chaque ligne du masque est projetée sur une diode d'un tableau linéaire de photodiodes. La somme des intensités lumineuses pondérées par les transparences est ainsi calculée, ce qui correspond à la sommation pondérée $\sum \text{poids} \times \text{entrées}$. L'opération de seuillage, quant à elle, est réalisée électroniquement, et un rebouclage sur les LED d'entrée est alors possible (à travers un ordinateur par exemple).

Les réseaux de neurones optoélectroniques modernes utilisent des tableaux de modulateurs (ou parfois de transducteurs) unidimensionnels ou bi-dimensionnels permettant d'atteindre un flux de plus de 1 Gbits/s. Une matrice optique de 100×100 éléments permettrait une performance pouvant dépasser 10 TCPS (10^{13} CPS). Des réseaux pré-programmés peuvent utiliser des masques à transparence fixe. L'apprentissage optique des réseaux de neurones est possible grâce à l'utilisation des modulateurs spatiaux de lumière magnéto-optique (MOSLM) ou ferroélectrique à cristaux liquides (FeCSLM) permettant de modifier la transparence de la matrice des poids.

Parmi les circuits optoélectroniques neuronaux, nous pouvons citer l'EOLC de Mitsubishi (annexe A.4).

2.8.4 Cartes accélératrices de réseaux de neurones

Les cartes accélératrices sont les modèles matériels les plus couramment utilisés dans les applications commerciales des réseaux de neurones. En effet, leur coût est relativement faible, elles sont largement disponibles, simplement montées sur un PC ou une station de travail et généralement accompagnées de logiciels utilitaires puissants et faciles à utiliser (interface avec l'utilisateur, bibliothèques de fonctions, etc.). La différence principale dont les utilisateurs, déjà familiers avec les outils logiciels de simulations des réseaux de neurones, vont se rendre compte est surtout la réduction des temps d'attente habituels dans l'apprentissage et/ou l'exécution de leurs réseaux. Les cartes existantes utilisent aussi bien des processeurs standards (transputers, DSP, etc.) que des circuits dédiés aux calculs neuronaux. L'amélioration de performance est presque d'un ordre de grandeur.

2.8.4.1 Les systèmes utilisant les processeurs standards

Ce sont des systèmes utilisant le plus souvent des processeurs RISC ou des processeurs de traitement de signaux (DSP) pour effectuer les traitements neuronaux en essayant de profiter de leurs capacités de calcul considérables, notamment pour des opérations simples telles que celles utilisées par les neurones (sommation et multiplication). Parmi les principaux systèmes de cette catégorie nous pouvons citer le BrainMaker de California Scientific Software, le NT6000 de Neural Technologies Limited et le XR50 de NeuroDynamX (annexe B.1).

2.8.4.2 Les systèmes utilisant les processeurs dédiés

Ils sont basés sur des cartes construites autour d'un ou plusieurs circuits dédiés aux calculs des réseaux de neurones. Ils offrent la possibilité du développement facile des réseaux de neurones grâce aux outils de développement associés ainsi qu'une vitesse supérieure à celle donnée par les systèmes utilisant des circuits standards. Ceci les rend bien adaptés aux applications du type temps réel. Beaucoup de systèmes commerciaux ont été développés dans cette catégorie, parmi les plus importants nous pouvons citer le SYNAPSE-1 de Siemens AG, le CNAP d'Adaptive Solution, le MY-NEUPOWER de Hitachi et le iNNTS d'Intel (annexe B.2).

2.9 Applications des réseaux de neurones

Comme il a été dit précédemment, les recherches dans le domaine des réseaux de neurones sont relativement nouvelles et n'ont commencé à prendre une réelle importance qu'au milieu des années 80. Ceci pourrait laisser penser que ce domaine n'est pas encore arrivé au stade de maturité lui permettant de concurrencer d'autres systèmes de traitements de l'information. Cependant, une analyse précise de ces recherches montre tout à fait le contraire. En effet, les approches neuronales constituent aujourd'hui une science à part entière avec des fondements théoriques bien établis. Les applications qu'on y trouve touchent pratiquement tous les domaines, en particulier les industries militaires et aéronautiques, le secteur tertiaire (banque, assurance, postes, etc.), les industries pétrolières, chimiques, pharmaceutiques, les compagnies travaillant sur la télévision et les télécommunications, ainsi que le domaine médical.

A titre d'exemple, nous pouvons citer le projet européen Esprit 9811 - SIENA² dans lequel des recherches se font pratiquement dans tous les domaines d'applications actuelles des réseaux de neurones. Des informations en ligne peuvent être trouvées en [SIENA96].

2.9.1 Des raisons d'utiliser les réseaux de neurones artificiels

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans les propriétés qu'ils possèdent et qui devraient permettre de dépasser les limitations de l'informatique traditionnelle, tant au niveau programmation qu'au niveau exécution. Parmi ces propriétés nous pouvons citer :

1. Le parallélisme.
2. La capacité d'adaptation.
3. La capacité de généralisation.
4. La mémoire distribuée.
5. La facilité de construction.

2.9.1.1 Le parallélisme

Cette notion se situe à la base de l'architecture des réseaux de neurones considérés comme des ensembles d'entités coopérant entre eux. L'intérêt d'une telle approche de traitement des données a été mis en évidence par la difficulté rencontrée par les méthodes séquentielles pour traiter des problèmes qui nécessitent des quantités très importantes de données. Le parallélisme permet une rapidité de calcul supérieure mais exige de penser et de poser différemment les problèmes à résoudre.

2.9.1.2 La capacité d'adaptation

Cette propriété se manifeste tout d'abord dans les réseaux de neurones par la capacité d'apprentissage qui leur permet de tenir compte de nouvelles contraintes et de nouvelles données de l'environnement extérieur. En plus, elles se caractérisent dans certains réseaux par leur capacité d'auto-organisation qui assure leur stabilité en tant que systèmes dynamiques.

2. Stimulation Initiative for European Neural Application.

Cette capacité d'adaptation présente un intérêt déterminant pour tous les problèmes évolutifs dont la résolution nécessite la prise en compte de situations inconnues au moment de l'apprentissage. Ceci signifie qu'il faut être capable, premièrement, de se rendre compte de l'évolution dans le problème traité, deuxièmement, d'apprendre à le résoudre d'une nouvelle façon.

2.9.1.3 La capacité de généralisation

L'importance de cette propriété s'est révélée ces dernières années avec les problèmes posés par la constitution de recueils d'expertises pour les systèmes experts. Il devient alors intéressant de disposer d'un système qui puisse, à partir d'ensembles d'exemples, apprendre à trouver des règles sous-jacentes, ou à imiter les comportements, qui permettent de résoudre le problème.

2.9.1.4 La mémoire distribuée

Dans les réseaux de neurones, la *mémorisation* d'un fait correspond à une *carte d'activation* des neurones constituant regroupant leurs états de sortie correspondants. Cette carte est en quelque sorte un codage du fait mémorisé. L'intérêt de cette distribution de la mémoire sur plusieurs entités se manifeste dans la résistance au bruit du problème traité. En effet, la perte d'un élément de cette carte ne correspond pas à la perte d'un élément mémorisé. Alors qu'en informatique classique, une donnée correspond à une case mémoire dont la perte entraîne la perte de la donnée stockée. La destruction d'une entité d'une mémoire distribuée ne modifie que peu les cartes d'activation des neurones.

D'autre part, lorsqu'une connaissance correspond à une donnée bien déterminée stockée dans une case précise, il se pose un problème de gestion de l'ensemble des connaissances. En effet, pour trouver une connaissance particulière, il faut connaître d'une manière précise son adresse et son contenu, ce qui ne permet pas de tenir compte des données bruitées. Il faut alors utiliser un prétraitement des données pour en éliminer le bruit. Cette insuffisance est dépassée dans les mémoires distribuées telles que les réseaux de neurones où l'on peut, à partir d'une donnée bruitée, faire émerger la carte d'activation neuronale de la donnée sans bruit.

2.9.1.5 La facilité de construction

La simulation logicielle d'un réseau de neurones pour une "petite" application est relativement simple et ne nécessite qu'un temps court de développement. Pour des applications "plus complexes", l'utilisation de simulateurs ou de cartes accélératrices, voire des systèmes multicartes pour les applications à fortes contraintes temps réel, se révèle utile. En effet, les progrès qui ont eu lieu dans le domaine de développements logiciel et matériel des réseaux de neurones, notamment l'existence des cartes accélératrices commerciales avec leur interface logicielle, rend facile la tâche des concepteurs surtout quand il s'agit du passage de la simulation à l'implantation sur du matériel.

2.9.2 Des caractéristiques d'une bonne application

A la lumière de ce qui précède, nous pouvons citer quelques unes des caractéristiques des problèmes bien adaptés à une résolution par les réseaux de neurones :

1. Il n'y a pas de modèle précis du problème à traiter et les règles qui permettraient sa résolution sont inconnues ou très difficiles à expliciter ou à formaliser.
2. Nous disposons d'un ensemble d'exemples qui correspondent à des entrées du problème et leurs sorties correspondantes.
3. Le problème fait intervenir des données bruitées ou floues.
4. La durée de vie du système fait qu'il aura à traiter des données non prévisibles au départ.
5. Le problème nécessite une grande vitesse de traitement, cas par exemple des applications temps réel.

Ceci nous amène à dresser la liste suivante des domaines d'applications privilégiés :

1. reconnaissance des formes ;
2. traitement du signal ;
3. vision, parole ;
4. prévision et modélisation ;
5. aide à la décision ;

6. robotique.

Ces domaines possèdent pratiquement toutes les caractéristiques exposées précédemment, c'est pourquoi ils constituent le cœur des applications des réseaux de neurones. Cependant, il ne faut pas oublier le rôle que peuvent jouer les réseaux de neurones artificiels dans l'étude du cerveau en tant qu'instruments de simulation, et dans la recherche sur les neuro-ordinateurs et les nouvelles architectures d'ordinateurs en tant qu'instruments de formalisation et de simulation.

2.9.3 Revue des différentes applications

La littérature des réseaux de neurones est riche en publications qui décrivent des applications neuronales, qu'elles existent sous forme industrielle, sous forme de prototypes et de maquettes, ou qu'elles ne soient envisagées qu'à travers des études. L'exposition détaillée des applications et des travaux qui ont été faits à ce jour sort du cadre de notre travail. C'est pour cela que, dans ce qui suit, nous nous limiterons à la présentation de quelques exemples de travaux réalisés dans les principaux domaines des applications des réseaux de neurones. Cependant, nous allons essayer de détailler les applications qui sont en relation avec notre travail, notamment celles utilisées ou potentiellement utilisables dans l'espace.

Les applications actuelles et potentielles des réseaux de neurones peuvent être classées de plusieurs façons. Les 2 approches principales sont basées sur le classement par "secteurs d'activités" et par "domaines d'applications". Dans la suite de ce paragraphe nous allons décrire brièvement la première approche tout en exposant certains exemples de réalisations effectuées à ce jour. Le second mode de classification est aussi important et permet de choisir parmi les différents algorithmes d'apprentissage (TAB. 2.3). Des applications classées selon cette approche peuvent être trouvées dans l'annexe C.

Les principaux secteurs d'activités des réseaux de neurones sont :

1. Secteur industriel.
2. Secteur financier.
3. Secteur des télécommunications et des réseaux à hauts débits.
4. Secteurs maritime et océanique.

Algorithme d'apprentissage	Rétro Propagation	Hopfield	Machine de Boltzmann	Kohonen
Application				
Classification	•	•	•	•
Traitement d'images	•			•
Aide à la décision	•		•	•
Optimisation		•	•	•

TAB. 2.3 : Types des réseaux pour différentes applications

5. Secteur médical.

6. Secteur de l'environnement.

7. Secteur spatial.

Certains parmi ces secteurs sont plus ou moins liés entre eux et l'on peut donc trouver des applications appartenant à plusieurs secteurs. C'est le cas par exemple de l'application des prévision météorologiques qui fait partie à la fois du secteur de l'environnement et du secteur spatial selon que les mesures sont effectuées sur la terre ou à bord des satellites.

2.9.3.1 Secteur industriel

Les applications des réseaux de neurones dans le secteur industriel sont très diverses. En effet, elles touchent pratiquement toutes les branches de l'industrie actuelle (industrie militaire, aéronautique, pétrolière, mécanique, chimique, pharmaceutique, électronique, électrique, alimentaire, agricole, etc.). Les catégories principales que nous pouvons distinguer sont les suivantes :

- contrôle de qualité [Yong95, Sim95] ;
- contrôle de processus [Kah95, Da Silva95] ;
- robotique [Tso95, Janusz95] ;
- diagnostic et prévision des pannes [Kerezi95, Roehl95].

2.9.3.2 Secteur financier

La complexité des marchés financiers exige l'assimilation et l'analyse de beaucoup de variables. Souvent, des données historiques existent sous forme de séries temporelles de ces variables. Il est alors nécessaire d'incorporer ces séries dans des modèles mathématiques ou des algorithmes dans le but de produire des prévisions d'indication sur les changements attendus des mouvements des prix. Ceci n'est pas toujours facile avec les méthodes classiques de traitement de l'information surtout à cause de la nature "chaotique" du marché financier. Les réseaux de neurones, de par leur nature "floue", ont été utilisés avec succès dans le secteur financier. Parmi les travaux trouvés dans la littérature peuvent être cités :

- la prévision des ventes [Kong95b, Thiesing95];
- l'authentification des signatures [Murshed95, Pottier94];
- la vérification automatique de chèques [Plamondon94];
- la prédiction de séries temporelles [Freisleben95];
- l'attribution de crédits [Barber95, Falcon 96].

D'autres travaux en relation avec la finance et l'assurance ainsi que des documentations en ligne peuvent être trouvés en [Ntl 95].

2.9.3.3 Secteur des télécommunications

Malgré le progrès considérable réalisé dans ce domaine, des défis y subsistent toujours, notamment dans les techniques utilisées pour la caractérisation du trafic (problème de bande passante), l'allocation dynamique de ressources et le contrôle de débit et de congestion [Habib95]. Ceci est dû essentiellement à la complexité des approches utilisées qui fait qu'elles utilisent des approximations importantes qui peuvent affecter la précision de la solution dans des conditions réelles de fonctionnement. Il est difficile pour certaines applications d'obtenir des estimations précises des paramètres statistiques (débit moyen, variance, etc.) ainsi que d'avoir des temps de réponse courts (inférieurs à la microseconde). Les réseaux de neurones peuvent offrir des solutions à ce type de problèmes puisqu'ils ne nécessitent pas une modélisation précise de la tâche à accomplir. En effet, en utilisant un apprentissage adéquat, un réseau de neurones apprendra la relation et produira donc des sorties plus ou moins précises quand on

lui présente un nouveau stimuli d'entrée. Parmi les applications qu'on trouve dans ce secteur on peut citer :

- analyse du signal et séparation des sources [Matsuoka94];
- contrôle d'admission de connexion (CAC) et de congestion (QoS) [Park95, Nordstrom95];
- optimisation [Haitao95];
- élimination du bruit [Klimasauskas89];
- contrôle temps réel des réseaux de télécommunications [Binh95];
- compression d'images [Setiono94].

2.9.3.4 Secteurs maritime et océanique

Les problèmes traités dans ces secteurs sont nombreux, on y trouve par exemple les systèmes des mesures scientifiques, les systèmes de transport, les véhicules sous-marins, etc. Les produits correspondants doivent être fabriqués d'une façon à la fois robuste et optimisée étant donné les conditions sévères de l'environnement de fonctionnement. Ceci se manifeste par le besoin de disposer de technologies sophistiquées afin de permettre à ce type de systèmes de fonctionner d'une façon efficace. Récemment, les réseaux de neurones ont été utilisés avec succès dans ces secteurs dans des applications telles que :

- contrôle de manœuvre [Ogawara95, Yamamoto95];
- diagnostic de fautes [Healy92];
- traitement du signal [Speidel92];
- aide à la décision [Neocleous95].

2.9.3.5 Secteur médical

Les applications qu'on trouve dans ce secteur sont centrées autour de la classification des données, du diagnostic et de la prévision des maladies. A titre d'exemples, peuvent être cités les travaux suivants :

- classification et détection d'éventuelles anomalies dans les données mesurées [Watt95, Yu95, Zheng94];

- classification de chromosomes [Zaradoshti95, Lerner94] et détection des vaisseaux sanguins [Nekovei95];
- pronostic du cancer [Choong94, Choong96] et prévision de rechute chez les malades [Tarassenko96].

2.9.3.6 Secteur de l'environnement

La nature imprévisible de ce secteur permet aux réseaux de neurones, à travers un apprentissage adéquat, de résoudre les problèmes tels que :

- évaluation des risques [Goh94];
- diagnostic des pollutions et analyse chimique [Mirsepassi95];
- prévision météorologiques [Ochiai95, Denoeux95];
- prévision du taux d'écoulement d'une rivière [Ichiyanagi95].

2.9.3.7 Secteur spatial

Plusieurs travaux ont été faits dans le domaine de traitement des données satellites. Les données traitées peuvent être obtenues aussi bien par des mesures directes ou bien indirectement après des prétraitements adéquats des données mesurées. Parmi les applications trouvées dans la littérature, qui sont utilisées ou susceptibles d'être utilisées dans l'espace, nous pouvons citer :

- classification des images prises par le satellite NOAA11 dans le but de produire un plan des mers glacées (sea-ice) [Kilpatrick95];
- détection des cavités "plasmiques" dans la densité des plasmas d'électrons mesurées par le satellite FREJA (expérience F4) [Waldemark95a];
- calibration d'un spectromètre tridimensionnel à composition d'ions (TICS) utilisé au bord du satellite FREJA [Waldemark95b];
- gestion du trafic des communications satellites [Ansari96];
- détection des sifflements protoniques et électroniques au bord des satellites [Minière96];

- contrôle et télérobotique dans les navettes spatiales [Parten90];
- classification des nuages à partir d'images satellites [Kwiatkowska95].

Notons que dans toutes ces applications, il s'agit de traiter, sur terre, des données mesurées à bord des satellites. Cependant, une solution consistant en des traitements in situ semble être beaucoup plus avantageuse que la première. En effet une telle approche permet d'obtenir un gain considérable dans les performances : diminution du flux de données envoyées vers la terre et par conséquent du temps de réponse de l'application pour les traitements en ligne et diminution de la taille des supports de stockage pour les traitements hors ligne. Néanmoins, dans ce cas, un besoin surgit et consiste en l'étude des effets, sur les circuits exécutant l'application neuronale, des rayonnements rencontrés dans l'espace, en particulier le phénomène d'upset, ce qui sera réalisé dans la suite de ce manuscrit.

Conclusion

Les réseaux de neurones offrent une solution attractive à beaucoup d'applications actuelles certaines parmi elles sont difficilement traitables par les moyens classiques. Pour des raisons de rapidité de réponse, demandée par certaines applications, les solutions logicielles ne sont pas suffisantes. Il faudra donc recourir à des réseaux de neurones *matériels*, qu'ils soient émulés par des processeurs standards ou bien implantés sur des circuits dédiés. Quant aux secteurs dans lesquels les réseaux de neurones peuvent être utilisés, ils sont nombreux. On y trouve les industries militaire, pétrolière, chimique, mécanique, biochimique, etc., le secteur tertiaire (banque, assurance, postes, etc.), les compagnies travaillant sur la télévision et les télécommunications, ainsi que le secteur médical et spatial. Ceci révèle l'importance des réseaux de neurones et la nécessité d'étudier leurs propriétés. Parmi ces propriétés, peut être citée leur tolérance aux fautes qui est l'un de leurs principaux avantages par rapport à d'autres méthodes de traitement de l'information. C'est l'étude de cette propriété qui constituera le but de ce manuscrit, son état de l'art sera traité dans le chapitre suivant.

Chapitre 3

Tolérance aux fautes des réseaux de neurones artificiels : état de l'art

Résumé du contenu du chapitre

Les réseaux de neurones sont considérés comme étant tolérants aux fautes à cause de la nature distribuée et non linéaire du traitement qu'ils effectuent et puisqu'ils contiennent souvent un nombre d'unités de calcul, dépassant le minimum demandé (notion de redondance). Dans ce chapitre nous résumons les principaux travaux de recherche réalisés dans le domaine de la tolérance aux fautes des réseaux de neurones.

3.1 Introduction

Un système fiable nécessite qu'une connaissance antérieure existe sur son comportement au cours de sa durée de vie qui peut varier de quelques secondes à plusieurs décades. Dans des applications critiques, telles que le contrôle des engins spatiaux et la détection des pannes dans les réacteurs nucléaires, il est vital que certains aspects du système soient bien pris en compte, parmi lesquels sa durée de vie et sa tolérance aux fautes.

Certaines parmi les applications neuronales exigent une fiabilité élevée au cours de leur durée de vie. Par conséquent, l'étude de la tolérance aux fautes et de la fiabilité des réseaux de neurones ainsi que celle des méthodologies de conception des réseaux tolérants devient un besoin urgent.

Les erreurs peuvent avoir lieu aussi bien dans la phase d'apprentissage du réseau que lors de son utilisation réelle dans une application. Les erreurs ayant lieu durant la

phase d'apprentissage (problème de précision, apprentissage en milieu bruité) peuvent augmenter la durée d'apprentissage mais ont une probabilité minimale d'affecter la performance du système. En effet, de par la nature de l'apprentissage, les composants erronés (nœuds et/ou connexions) sont compensés par les parties non défectueuses du réseau, tout en supposant qu'il y ait suffisamment de redondance. Si, par contre, les erreurs sont détectées dans la *phase de test* du réseau, un réapprentissage avec rajout de nouvelles ressources peut résoudre le problème. Une telle correction n'est pas possible si le développement du système est achevé et si les erreurs ont lieu sur une application déjà en utilisation. Une conception tolérante aux fautes assurerait le fonctionnement correct du système utilisé en présence d'erreurs, ou se manifesterait par une dégradation progressive de performance si les paramètres du réseau changent.

Jusqu'au début des années 90, la tolérance aux fautes des réseaux de neurones a été souvent citée comme étant l'une de leurs propriétés remarquables et l'un de leurs principaux avantages par rapport à d'autres systèmes de traitement de l'information. Ces revendications étaient basées sur la supposition que cette tolérance était due à la manière dont le système fonctionne plutôt qu'à la présence d'éléments supplémentaires (redondance). Les réseaux de neurones biologiques sont connus comme étant remarquablement tolérants aussi bien aux fautes internes qu'au bruit affectant les données sensorielles. Ceci peut être tiré des nombreuses études qui ont été réalisées sur les lésions, et aussi du fait que les données reçues par nos sens sont, dans la plupart des cas, bruitées. Par conséquent, si les réseaux de neurones artificiels sont de bons modèles des réseaux de neurones biologiques desquels ils sont inspirés, alors il est raisonnable de supposer que cette caractéristique des réseaux de neurones biologiques est étendue aux réseaux de neurones artificiels. L'étude de la tolérance aux fautes des réseaux de neurones nécessite la définition des modèles de fautes pouvant survenir ainsi que la définition des moyens pour la mesure de leur fiabilité. La plupart des travaux publiés dans ce domaine, jusqu'à la fin des années 80 présentent des résultats expérimentaux sur les effets des erreurs et/ou bruit sur les réseaux de neurones.

Cependant, depuis le début des années 90, les réseaux de neurones artificiels ont commencé à attirer l'attention par la pluralité de leurs domaines d'applications et par leur performance pouvant concurrencer les méthodes classiques de traitement de l'information surtout dans leurs émulations et implantations matérielles qui peuvent résoudre des problèmes critiques mettant en jeu des contraintes de type temps réel. C'est ainsi qu'on voit démarrer des recherches importantes dans le domaine de la

tolérance aux fautes et de la fiabilité des réseaux de neurones. Les principaux travaux effectués seront décrits dans la suite de ce chapitre.

3.2 Terminologie de fiabilité

Plusieurs termes clefs sont reliés aux aspects fiabilité des systèmes. Une *faute* est une panne physique ou un défaut dans un élément d'une implantation matérielle ou logicielle. Une *erreur* survient sur une information et est, en général, causée par une faute sur un composant particulier. Bien que des fautes ou des erreurs peuvent exister dans un système donné, elles peuvent ne pas être remarquées à travers les actions du système. Quand une déviation par rapport au comportement attendu est remarquée, le terme de *défaillance* est utilisé. La *tolérance aux fautes* (parfois appelée *robustesse*) signifie qu'un système continue à fournir une réponse correcte en présence de fautes. Quant à la *fiabilité*, c'est la probabilité qu'un système fonctionne toujours selon les spécifications qui lui ont été prescrites après une certaine période de temps étant donné que son fonctionnement était correct à l'origine.

Le domaine d'étude de la fiabilité des réseaux de neurones est vaste, vu qu'il comprend des modèles de fautes différents ainsi que plusieurs types d'architectures. Pour que les résultats correspondants soient comparables, il faut qu'une méthodologie soit définie afin de permettre une étude matérielle de la fiabilité des réseaux de neurones de façon à obtenir des résultats généraux. Ceci nécessite une catégorisation des causes d'erreurs pouvant affecter la fiabilité, l'architecture et les composants opérationnels d'un réseau de neurones.

3.3 Domaines d'étude de la tolérance aux fautes

Lors de l'étude de la tolérance aux fautes des réseaux de neurones, deux aspects doivent être étudiés :

- leur capacité à supporter les défauts dans les composants internes (neurones et/ou connexions) ;
- leur capacité à produire des résultats satisfaisants en présence de données bruitées provenant de l'extérieur.

Chacun de ces deux aspects comprend deux domaines différents. Il s'agit essentiellement des effets, probablement différents, de fautes, durant la phase d'apprentissage et la phase de rappel respectivement. Cette hiérarchie de domaines d'exploration est vue à 3 niveaux différents : (1) le modèle abstrait, (2) l'architecture du système et finalement (3) l'implantation concrète dans laquelle la conception architecturale est transformée en réalité par une technologie adéquate.

1. L'examen des modèles abstraits de réseaux de neurones permet de comparer les résultats obtenus pour les différents modèles existants afin d'indiquer leurs mérites. De plus, un tel examen permet de concevoir de nouvelles architectures, de façon à ce que la tolérance aux fautes inhérente au modèle soit exploitée au maximum.
2. L'analyse des architectures existantes est utile dans la production de résultats sur la nature de leur fonctionnement. Elle pourrait probablement conduire à des conceptions plus tolérantes et plus robustes.
3. Au niveau le plus bas (l'implantation), bien que des aspects très précis de la tolérance aux fautes puissent être analysés, peu de changements peuvent être introduits dans le système pour augmenter sa fiabilité.

En passant du niveau d'abstraction aux implantations réelles, la mesure et l'analyse de la fiabilité devient plus facile, mais le système est probablement moins malléable aux changements visant à améliorer cette mesure.

3.4 Considérations sur les modèles de fautes

Un besoin existe pour mesurer, d'une façon significative la tolérance aux fautes présente dans un réseau de neurones. Comme une première approximation, ainsi que pour l'utilisation dans une fonction arbitraire, la définition de différents modèles de fautes sera nécessaire. De même, si une fonction de fiabilité existe pour un réseau de neurones donné, alors ces modèles de fautes peuvent y être appliqués. Cependant, vu que les différentes parties constituantes du système dépendent l'une de l'autre d'une façon complexe, il est probablement difficile de définir une fonction pour la plupart des modèles des réseaux de neurones qui génère la fiabilité totale à partir des fiabilités des unités/connexions individuelles.

Plusieurs modèles de fautes peuvent être définis à différents niveaux de description du système (abstrait, architectural, implantation). Le modèle de faute doit prendre en compte aussi bien les défauts que le bruit interne. Au niveau abstrait, il est possible d’imaginer des unités ou des connexions individuelles coupées ou endommagées, pouvant probablement introduire du bruit dans le signal d’entrée d’une unité par exemple. Ceci semble bien corrélérer avec la perte continue des neurones et/ou synapses dans le cerveau. La généralisation du modèle de faute au niveau architectural est impossible et dépendra de la conception individuelle et au niveau de l’implantation, de la technologie utilisée.

D’autre part, des mesures convenables doivent être définies afin de donner une indication sur l’effet des fautes induites dans un réseau de neurones utilisant des modèles de fautes sur l’un des niveaux cités ci-dessus. Contrairement aux ordinateurs classiques, il est peu probable que les fautes puissent causer une défaillance du système ; nous assisterons plutôt une *dégradation progressive* des performances. Ces mesures peuvent être une fonction de la sortie attendue et de la sortie réelle du réseau, malgré que la gravité relative des différentes sorties erronées doive être prise en compte.

3.5 Exemple d’illustration de la tolérance aux fautes des ANN

Afin d’illustrer le mécanisme de tolérance aux fautes des réseaux de neurones, prenons l’exemple du réseau de calcul du XOR de la figure 2.6 et examinons l’influence des perturbations sur ses entrées et/ou poids. Il s’agit d’un “cas d’école” vu la taille du réseau et la fonction simple qu’il réalise.

D’après les équations 2.3, 2.4 et 2.5, une erreur survenant sur une information liée à un neurone N_i (entrée, poids ou seuil) va se manifester par une variation ΔOut_{N_i} de sa sortie Out_{N_i} (eq. 3.1). Cette variation n’est pas proportionnelle à l’erreur qui l’a provoquée à cause de la nature non linéaire des traitements effectués (somme des produits et fonction sigmoïde).

$$Out_{N_i} \longleftarrow Out_{N_i} + \Delta Out_{N_i} \quad (i = 1, 2 \text{ et } s) \quad (3.1)$$

A titre d’exemple, le tableau 3.1 donne les sorties des neurones N_i en présence de

fautes sur les entrées ¹, le cas de fautes sur les poids étant similaire du point de vue conséquence au niveau de la sortie.

E_1	E_2	Out_{N_1}	Out_{N_2}	Out_{N_s}
0.00	0.20	0.18	0.00	0.10
0.10	0.10	0.18	0.00	0.10
0.10	1.00	0.99	0.11	0.98
0.20	0.90	0.99	0.11	0.98
0.90	1.00	1.00	0.80	0.10
0.90	0.80	1.00	0.62	0.30

TAB. 3.1 : Valeurs des sorties des neurones pour des entrées perturbées

Les chiffres donnés montrent que, malgré les valeurs importantes des poids du réseau (cf. TAB. 2.1) qui vont jouer le rôle d’“amplificateurs” pour les sorties des neurones situés en amont, les erreurs affectant les entrées se manifestent de façon moins grave au niveau de la sortie du réseau. Par exemple une erreur de 0.2 survenant sur l’entrée E_2 (cas de la combinaison d’entrée $(0, 0)$) et qui sera multipliée par $W_{12} = 6.42$ (neurone N_{c_1}) et $W_{22} = 4.30$ (neurone N_{c_2}) résulte en une erreur de 0.12 et 0.00 sur Out_{N_1} et Out_{N_2} respectivement. Ceci se manifeste par une erreur de 0.07 sur la sortie finale Out_{N_s} . Ces résultats dépendent bien entendu des valeurs des poids et doivent être moyennés sur l’ensemble des combinaisons d’entrée possibles, ce qui sera fait, pour des réseaux plus représentatifs dans la suite du manuscrit.

3.6 Facteurs liés à la tolérance aux fautes des ANN

Plusieurs facteurs sont liés, d’une façon plus ou moins étroite, à la tolérance aux fautes d’un réseau de neurones, parmi lesquels nous pouvons citer :

- la redondance ;
- nature distribuée de l’information ;
- format de données ;
- présence des rétroactions ;

1. Les sorties correspondant à des entrées non “corrompues” sont données dans le tableau 2.2.

- notion d'éléments de traitement.

3.6.1 La redondance

La redondance dans un réseau de neurones, aussi bien dans les unités que dans les connexions, a été souvent identifiée comme un facteur produisant un système fiable [Emmerson91]. Les techniques classiques pouvant contribuer à la tolérance aux fautes dans un système à ordinateur ont été comparées en [Moore88] avec des mécanismes apparaissant dans les réseaux de neurones biologiques afin d'en extraire des implications. Moore a montré que les réseaux de neurones biologiques utilisent simultanément une redondance spatiale et temporelle analogues à celles utilisées par les techniques classiques de fiabilité afin d'atteindre un degré élevé de tolérance aux fautes. En effet, les calculs dans les réseaux de neurones biologiques sont répétés de manière continue (redondance temporelle) et les neurones/connexions y sont souvent dupliqués (redondance spatiale).

3.6.2 Nature distribuée de l'information et des traitements

Une des caractéristiques des réseaux de neurones conduisant à leur tolérance aux fautes est la **distribution** de l'information et des traitements partout dans le réseau. En effet, lors du fonctionnement du réseau, toutes les unités et connexions contribuent, d'une façon directe ou indirecte, à l'établissement de la sortie finale du réseau. Par conséquent, la perte d'une unité ou bien d'une connexion particulière n'affecterait pas de manière significative, le fonctionnement du réseau pourvu qu'il ait suffisamment de redondance [Emmerson92].

3.6.3 Format de données

Le format des données à traiter est une considération supplémentaire [Hancock88] dans l'amélioration de la tolérance aux fautes d'un réseau de neurones. Les entrées, les sorties et surtout les paramètres du réseau peuvent être choisies, selon le cas, comme étant des valeurs entières, réelles ou binaires. Le choix du codage peut influencer sur le comportement du réseau étudié en présence d'erreurs puisque la taille des données sensibles ne sera pas la même pour tous les codages.

3.6.4 Présence des rétroactions

La présence éventuelle des contre-réactions dans les réseaux de neurones peut affecter leur fiabilité puisqu'elle va rajouter une redondance temporelle [Moore88]. Cependant, il n'est pas évident de savoir si les erreurs sur les signaux internes causées par le bruit ou par les fautes sur les unités et/ou connexions disparaîtront avec le temps ou bien seront amplifiées.

3.6.5 Notion d'éléments de traitement

La notion d'éléments de traitement (Processing Elements – PE) simples et similaires, associés aux neurones, même pour des conceptions différentes, peut avoir une conséquence sur la fiabilité totale du système, bien que la “taille” d'un PE peut varier d'une implantation à une autre. En augmentant la fiabilité de chaque élément, on peut améliorer davantage la fiabilité totale du système si nécessaire.

3.7 Etat de l'art de tolérance aux fautes des ANN

Comme il a été dit dans ce qui précède, la plupart des travaux publiés dans le domaine de la tolérance aux fautes des réseaux de neurones datent du début des années 90. Les recherches effectuées dans ce domaine jusqu'à nos jours sont bien avancées et ont donné lieu à des méthodes théoriques et expérimentales bien établies pour certains modèles de fautes. Les études réalisées peuvent être classées en 2 classes principales :

1. Etude de la tolérance aux fautes survenant durant la phase de rappel.
2. Amélioration de la tolérance d'un ANN durant la phase d'apprentissage.

3.7.1 Tolérance durant la phase de rappel

Ces travaux visent à étudier la tolérance aux fautes des réseaux de neurones au stade opérationnel. Le but est de déterminer l'influence des erreurs survenant sur les stimuli d'entrée du réseau et/ou les poids des connexions entre les neurones, en supposant que le réseau initial fonctionne correctement.

3.7.1.1 Tolérance aux fautes survenant sur les stimuli d'entrée

Une recherche a été menée en [Stevenson90] dans le but de déterminer la probabilité d'erreur d'un réseau de neurones. Les fautes ont été introduites durant la phase de rappel sous forme de bruit sur les entrées et sur les poids simultanément. Les réseaux étudiés étaient du type "Madaline" [Widrow88] réalisant une fonction d'association entrées/sorties. Une étude théorique ainsi que des simulations logicielles ont montré que pour de faibles taux d'erreurs, ce type de réseaux présente une bonne tolérance. En effet, dans le cas extrême étudié, correspondant à un taux de perturbation de 8% et de 50%, introduit sur les entrées et les poids respectivement, le taux d'erreur n'a pas dépassé 23%.

3.7.1.2 Tolérance aux fautes survenant sur les poids synaptiques

Afin d'améliorer la tolérance des réseaux de neurones face aux fautes survenant sur les poids des connexions, une technique a été développée par [Emmerson92]. Elle consiste à augmenter la redondance du réseau en augmentant la taille de ses couches cachées par une duplication de tous ses éléments constitutants (nœuds et connexions correspondantes). Les éléments dupliqués se partagent les tâches avec les éléments originaux, en particulier avec les éléments les plus sensibles du réseau. Par conséquent, une faute sur un élément (nœud ou connexion), qui était critique pour le réseau original, sera mieux tolérée par un de ses homologues du réseau augmenté. Cette technique est bien entendu coûteuse ; une alternative serait alors de dupliquer uniquement les nœuds cachés les plus sensibles du réseau, ceux-là pouvant être déterminés par analyse de la représentation interne du réseau (par exemple en utilisant la méthode SVD (Singular Value Decomposition) utilisée en [Xue90]).

3.7.2 Tolérance durant la phase d'apprentissage

Les recherches correspondantes sont axées sur l'étude des approches permettant d'améliorer la tolérance aux fautes des réseaux de neurones au stade d'apprentissage. Ces approches consistent à modifier les algorithmes d'apprentissage existants, à concevoir de nouveaux algorithmes ou bien à introduire des fautes sur les entrées et/ou poids durant l'apprentissage. Les principales techniques utilisées sont :

- Réapprentissage.
- Apprentissage avec restriction des poids.

- Apprentissage par injection des fautes artificielles.
- Procédure d'addition/suppression.

3.7.2.1 Réapprentissage

En général, une erreur survenant dans un système à réseau de neurones va se manifester par une dégradation plus ou moins importante. Le réapprentissage peut être considéré comme une méthode pour minimiser la gravité du problème et pour restaurer la précision du traitement. Un réapprentissage a été fait par [Tanaka89] sur un perceptron multicouches (MLP). Les résultats obtenus ont montré la possibilité de regagner des niveaux élevés de précision même après que plusieurs fautes aient eu lieu.

[Bedworth88] a également étudié le réapprentissage pour un MLP qui est considérablement plus complexe que celui de [Tanaka89]. Malgré que la moitié des connexions ait été supprimée, le réseau était toujours capable, via un réapprentissage adéquat, d'atteindre sa performance originale et ceci en un cinquième de son temps d'apprentissage initial.

La capacité de réapprendre un système à base de réseaux de neurones peut produire un système qui peut facilement répondre à des contraintes de fiabilité à long terme, bien que le système responsable du réapprentissage soit le facteur clé dans l'équation de fiabilité du système entier.

3.7.2.2 Apprentissage avec restriction de poids

Dans un réseau de neurones “bien appris”, les connexions qui sont les plus sensibles sont celles qui ont les poids les plus grands puisque ces derniers “amplifient”, d'une façon ou d'une autre, les erreurs survenant sur les différents paramètres internes. En forçant tous les poids à avoir des valeurs faibles, nous pouvons garantir une tolérance élevée pour toutes les connexions. Cependant, la performance du réseau se dégradera si l'on se contente d'une simple réduction des valeurs des poids des connexions après avoir effectué l'apprentissage du réseau, la configuration optimale du réseau étant altérée. C'est la raison pour laquelle la restriction des poids doit être appliquée durant l'apprentissage et ce, en “pénalisant” les poids ayant des grandes valeurs et en rajoutant éventuellement des nœuds supplémentaires. Parmi les algorithmes développés utilisant cette technique, nous pouvons citer celui de [Damarla89]

ainsi que le WRTA (Weight-Restricted Training Algorithm) [Chiu93a]. Notons que, malgré l'augmentation possible du nombre de neurones en utilisant ce type d'apprentissage, la sensibilité du réseau obtenu reste inférieure à celle d'un réseau "normal" à cause de la diminution des valeurs des poids.

3.7.2.3 Apprentissage avec injection de fautes artificielles

[Holmstrom92] et [Matsuoka92] ont montré que le fait de rajouter un bruit artificiel aux entrées d'un réseau de neurones durant l'apprentissage améliore sa qualité en termes de capacité de généralisation et de tolérance. Ce même résultat peut être étendu à l'injection du bruit sur les poids. En effet, [Murray94b] a montré, via une étude théorique, que le terme qui correspond au bruit rajouté et qui apparaît dans la fonction du coût (différence entre la sortie obtenue et la sortie attendue), non seulement améliore la qualité d'apprentissage mais diminue aussi sa durée et joue un rôle de régulateur en distribuant l'information de manière optimale au sein du réseau.

D'autre part, une technique a été développée en [Séquin90] afin d'améliorer la tolérance aux fautes des réseaux de neurones. Elle consiste en la remise à zéro des sorties de certains neurones cachés, aléatoirement choisis, et ceci avant chaque itération du processus d'apprentissage. Une approche similaire a été proposée en [Bolt91] en mettant les poids des connexions à zéro au lieu des sorties des neurones cachés. Cependant, ces 2 méthodes ne sont efficaces que pour le type spécifique de fautes avec lequel l'apprentissage a été effectué. Par exemple, dans la méthode de [Séquin90], la tolérance a été améliorée face à des fautes qui mettent la sortie des neurones cachés à "0" mais pas forcément face aux fautes qui les mettent à "1" (valeur maximale). Le même raisonnement pourra être fait avec la méthode de [Bolt91] qui désactive aléatoirement des connexions.

Une généralisation a été proposée par [Chiu94] afin d'améliorer les 2 approches citées précédemment. Il s'agit de l'algorithme MFTA (Multiple Fault Training Algorithm) qui consiste à injecter différents types de fautes durant l'apprentissage afin d'obtenir un ensemble de poids qui soit robuste simultanément à plusieurs types de fautes. Dans chaque itération du processus d'apprentissage, des neurones cachés ont été choisis aléatoirement et leurs sorties ont été mises à "0" et à "1" alternativement. Au cours de la même itération, des poids de connexions, aléatoirement choisis, ont été perturbés.

3.7.2.4 Procédure de rajout/suppression

Il s'agit notamment de la méthode ADP (Addition/Deletion Procedure), décrite en [Chiu93b] dans le but de pouvoir supporter les changements des valeurs des poids des connexions. Elle comprend 3 étapes qui doivent être répétées jusqu'à ce que la tolérance désirée soit atteinte :

1. Elimination des nœuds non importants.
2. Réapprentissage du réseau réduit ainsi obtenu.
3. Rajout de nœuds redondants.

3.7.2.4.1 Elimination des nœuds non importants

En pratique, dans un réseau de neurones de grande taille, certains nœuds peuvent être éliminés sans affecter la performance totale du système. Ceci est dû au fait que les algorithmes traditionnels d'apprentissage n'assurent pas que les nœuds redondants améliorent la tolérance aux fautes.

Une fois le réseau appris, l'importance de chacun des nœuds cachés peut être mesurée en termes de sa *sensibilité*. Cette dernière est donnée en fonction des sensibilités de toutes les connexions de ce nœud (erreurs moindre carrés). Etant donné une sensibilité de référence ϵ , un nœud n_{ij} est éliminé de la couche cachée j si la sensibilité $S_j(i)$ qui lui correspond est inférieure à ϵ ($S_j(i) \leq \epsilon$). La valeur de ϵ peut être réglée de façon à ce que l'élimination de tous les nœuds ayant une sensibilité inférieure ne se traduise pas par une grande différence de performance entre le réseau ainsi obtenu et le réseau original. Par conséquent, la suppression des nœuds ayant une petite sensibilité donne lieu à un réseau réduit mais ayant une performance comparable à celle du réseau original.

3.7.2.4.2 Réapprentissage du réseau réduit

Comme il a été dit dans le paragraphe précédent, l'élimination des nœuds non importants n'a pratiquement pas d'influence sur la performance du réseau. Cependant un réapprentissage du réseau réduit obtenu est utile afin de réduire les valeurs de la fonction de coût (erreur d'apprentissage) dans les minimums locaux du réseau suite à un décalage éventuel du aux éliminations. Le nombre d'itérations nécessaire pour ce faire est généralement petit (< 10 dans la plupart des cas).

3.7.2.4.3 Rajout de nœuds redondants

Afin d'améliorer la tolérance aux fautes du réseau obtenu à la fin de la deuxième étape, des nœuds supplémentaires sont rajoutés dans les couches cachées, de façon à ce qu'ils partagent la tâche des nœuds critiques du réseau (nœuds ayant la sensibilité la plus élevée). Ce processus est répété jusqu'à ce que l'amélioration apportée en rajoutant des nouveaux nœuds ne soit plus significative. D'autres possibilités consistent à s'arrêter dès que la sensibilité du neurone caché le plus critique soit inférieure à un certain seuil ou bien que le nombre total de nœuds soit égal à celui du réseau original. Cette dernière possibilité permet de comparer 2 réseaux ayant la même taille.

3.7.2.5 Comparaison entre quelques techniques d'amélioration

Une étude comparative a été effectuée par [Chiu94] entre l'algorithme standard de rétro-propagation d'erreur et certains algorithmes d'amélioration de la tolérance aux fautes. Le problème étudié était de type classification, les résultats obtenus, pour des erreurs de poids et/ou nœuds de type mise à "0" (STK 0) et mise à "1" (STK 1), sont résumés sur le tableau 3.2 en termes de pourcentage de sensibilité moyen et pire cas.

Données d'apprentissage / de test					
Algorithme	Classification correcte	Sensibilité moyenne		Sensibilité pire cas	
		STK 0	STK 1	STK 0	STK 1
BP	99/94%	17/20%	14/14%	37/38%	52/50%
WRTA	99/96%	5/6%	16/17%	8/12%	29/30%
Séquin	98/96%	2/6%	31/28%	2/8%	39/36%
Bolt	99/92%	11/13%	18/18%	34/30%	45/42%
MFTA	97/94%	7/9%	13/14%	21/22%	26/26%
ADP	99/96%	4/6%	5/7%	9/12%	7/10%

TAB. 3.2 : Comparaison des sensibilités moyenne et pire cas pour différents algorithmes d'apprentissage, cas des erreurs de type mise à "0" et à "1"

3.8 Discussion

Les expériences qui ont été discutées ci-dessus ainsi que d'autres travaux de recherche trouvés dans la littérature [Kerlirzin93, Phatak95] montrent, aussi bien par

des études théoriques que par des simulations logicielles, la propriété de tolérance aux fautes des réseaux de neurones artificiels. D'autre part, un upset peut être assimilé à une faute particulière affectant un seul bit de l'information concernée. Cependant, malgré les différents types de fautes étudiés dans la littérature, dont certaines correspondent à des défauts matériels (suppression d'une connexion, mise à zéro de la sortie d'un neurone, etc.), nous n'avons pas trouvé des travaux publiés qui traitent les effets des fautes binaires de type upset. Ceci révèle la nécessité de combler cette lacune et d'étudier les effets de ce type de fautes sur les réseaux de neurones afin de juger de la possibilité de leur utilisation sous forme matérielle (implantations et émulations) dans des milieux pouvant perturber l'information en relation avec leur fonctionnement, ce qui est le cas des milieux agressifs. Cette information peut être du type données (stimuli d'entrée et poids synaptiques) ou du type programme (instructions correspondant à l'application exécutée).

Conclusion

Dans ce chapitre, ont été décrits brièvement les principaux travaux de recherche effectués sur la tolérance aux fautes des réseaux de neurones. Ces travaux ont illustré l'existence de cette propriété intrinsèque durant la phase de rappel ainsi que la possibilité de son amélioration durant la phase d'apprentissage. D'autre part, l'utilisation des réseaux de neurones matériels dans des applications s'exécutant dans un environnement agressif tel que l'espace pourrait être un choix adéquat dans certains cas (par exemple fortes contraintes temps réel). Cependant, les effets des fautes de type upset sur les réseaux de neurones n'ont pas été étudiés. Il s'avère donc nécessaire de réaliser ce type d'expériences sur les réseaux de neurones matériels (implantations et émulations). Ce travail constitue l'objectif de la thèse ; la méthodologie, les différentes stratégies de test et les résultats obtenus seront décrits dans le reste du présent manuscrit.

Chapitre 4

Méthodologies de test, circuits et réseaux de neurones choisis

Résumé du contenu du chapitre

La première partie de ce chapitre est consacrée à la description des différentes stratégies de test adoptées dans la thèse pour l'étude de la tolérance aux fautes des réseaux de neurones artificiels et la mise en œuvre correspondante. Dans la deuxième partie sont détaillés les différents circuits cibles ainsi que les réseaux de test choisis. Les programmes de tests quant à eux, sont décrits dans la dernière partie.

4.1 Stratégies de test

Pour étudier les effets des upsets sur les réseaux de neurones, nous avons adopté une démarche expérimentale incluant trois approches différentes :

1. Simulations logicielles.
2. Injection matérielle.
3. Tests aux ions lourds.

4.1.1 Simulations logicielles d'upsets

Les simulations logicielles permettent d'étudier les effets des SEU sur les réseaux de neurones sans le besoin d'un développement matériel spécifique. Cependant, les erreurs de type SEU peuvent être simulées uniquement sur les données nécessaires

pour le fonctionnement du réseau (poids, seuils et stimuli d'entrée respectivement), ceci n'étant pas possible pour le programme puisqu'il est écrit en langage évolué. Cette technique est basée sur l'introduction d'une perturbation (changement d'état d'un des bits) sur la donnée analysée avant de lancer le programme correspondant au calcul neuronal. Ceci permet de délimiter de façon précise les bits pour lesquels l'erreur est tolérée ainsi que les zones où l'erreur est transmise à la sortie du réseau.

4.1.2 Injection matérielle d'upsets

Cette expérience est réalisée sur des cartes prototypes construites autour des circuits sous test. Elle consiste à injecter des fautes de type upset dans les cellules mémoires sensibles du circuit sous test en inversant, avant de lancer l'application neuronale, des bits correspondants. De cette façon, le circuit sous test exécute l'application avec des données ou paramètres corrompus : si le réseau implanté donne une bonne réponse avec la configuration corrompue, l'upset correspondant qui aura lieu durant le fonctionnement du circuit dans un milieu radiatif réel sera toléré, sinon il doit être pris en compte pour le calcul de la section efficace finale de l'application. Cette approche permet de tester, en plus des zones de stockage des données, le code du programme exécuté par le circuit sous test. Cependant, malgré son avantage par rapport aux simulations logicielles (test des données et du programme en même temps), la technique d'injection présente une limitation puisqu'elle ne permet pas de perturber les zones internes du circuit sous test inaccessible par programme (registres tampons, latches, etc.).

4.1.3 Test aux ions lourds

Ce type de test permet d'évaluer la sensibilité aux upsets, d'un circuit cible soumis aux radiations. Nous avons vu précédemment que, pour le cas des circuits de type processeur, l'application d'une stratégie de test statique, conduit à une surestimation de la sensibilité aux upsets du composant. En effet, lors de l'exécution d'un programme de calcul quelconque, des données transitent dans des registres le temps de l'exécution de quelques instructions seulement, le reste du temps le registre considéré ne contenant pas des données significatives, les SEU pouvant l'affecter n'auront pas d'effet sur le résultat du programme. C'est pourquoi, le calcul de sensibilité aux upsets de processeurs est réalisé à partir aussi bien des programmes de test statique,

que des programmes d'application représentatifs de modules d'une application réelle.

Compte tenu que les circuits neuronaux sont des processeurs particuliers, le même type de réflexion pourrait être fait à propos de la non représentativité des sections efficaces tirées de tests statiques. De plus, si la propriété de tolérance intrinsèque aux fautes des réseaux de neurones est valable pour le cas des upsets, alors nous pourrions estimer que, pour les circuits implantant (émulant) les réseaux testés, ceci peut rendre encore plus éloignés les résultats obtenus avec les deux types de stratégie.

Nous adopterons donc pour chacun des circuits neuronaux choisis, une stratégie de test statique qui se traduira par un programme de type test de registres, et une stratégie dynamique, qui sera mise en œuvre par des programmes neuronaux proprement dits, qui sont des programmes typiques d'application utilisant un sous-ensemble plus ou moins large des ressources du circuit sous test (RAM, registres, processeurs, etc.). Comme pour les microprocesseurs, le problème à traiter pour cette deuxième stratégie peut être quelconque, le but étant la mise en évidence de l'ordre de grandeur de la différence entre les sections efficaces obtenues par les deux stratégies.

L'avantage principal d'une stratégie de test aux ions lourds par rapports aux techniques des simulations et d'injection de fautes, est qu'elle permet d'évaluer la sensibilité du circuit sous test dans un environnement proche de celui d'un fonctionnement réel bien qu'elle ne permet pas de donner l'emplacement exact des zones de mémorisation critiques de l'application implantée.

4.2 Extrapolation de la section efficace dynamique

Une des méthodes originales appliquée dans la thèse est la détermination de la section efficace de l'application testée par extrapolation à partir du mapping des bits sensibles aux upsets (résultant d'une expérience d'injection de fautes) et d'une séquence donnée d'upsets (obtenue à partir des traces d'un test statique sous radiations). Ceci nécessite donc la réalisation de deux types de tests:

1. Un test par injection matérielle de fautes. Ce test permet, par l'introduction successive de fautes de type upset dans la zone mémoire occupée par l'application testée, de déterminer le mapping des bits sensibles. Ce mapping représente les emplacements mémoires pour lesquels l'occurrence d'un upset va se manifester par une erreur à la sortie finale du réseau (FIG. 4.1).

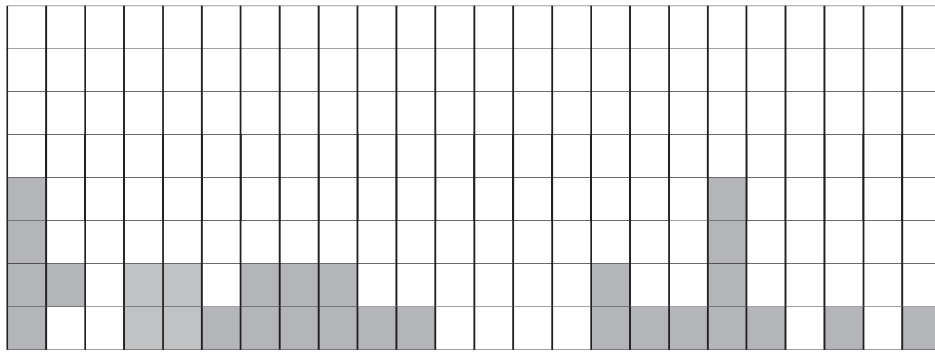


FIG. 4.1 : *Mapping des bits sensibles de l'application testée obtenu suite à un test par injection matérielle de fautes*

2. Un test aux ions lourds de type statique. Ce test permet, en irradiant le circuit sous test, d'obtenir une séquence d'upsets sur la zone mémoire occupée par l'application en cours. Une comparaison de la "carte" de cette séquence (FIG. 4.2) avec le mapping des bits sensibles (FIG. 4.1) permettra de trouver l'emplacement et le nombre des upsets qui donneraient lieu à des erreurs en dynamique. Ces upsets seront ensuite utilisés ainsi que la fluence des particules du test statique pour déterminer la section efficace extrapolée de l'application.

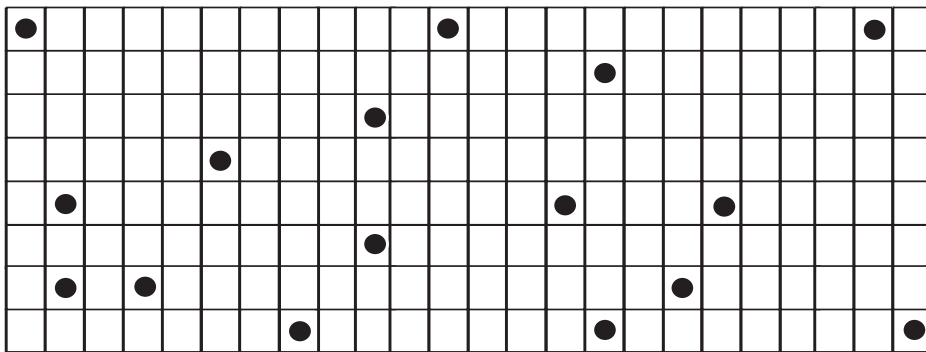


FIG. 4.2 : *Carte d'une séquence d'upset obtenue suite à un test statique*

L'avantage principal d'une telle approche est qu'elle permet de déterminer la sensibilité d'une application quelconque sans avoir besoin d'irradier le circuit pendant son exécution puisque la détermination des bits sensibles se fait hors faisceau. De ce fait, le coût d'un test dynamique utilisant cette technique est considérablement diminué. Cette méthode a été validée sur des applications neuronales: les sections efficaces dynamiques extrapolée et mesurée (obtenue par un test dynamique) sont de même ordre de grandeur.

4.3 Schéma bloc d'un réseau de neurones matériel

Qu'il s'agisse d'une implantation ou d'une émulation matérielle, un réseau de neurones comprend deux blocs principaux (FIG. 4.3) :

1. Une mémoire dans laquelle sont stockés le programme, la fonction d'activation, les poids et seuils, les stimuli d'entrées ainsi que les résultats (intermédiaires et finals).
2. Un bloc processeur(s) chargé(s) de l'exécution du programme neuronal et/ou du contrôle des différents blocs.

Notons que cette architecture n'est pas la seule possible puisqu'il existe des architectures multiprocesseurs dédiées dans lesquelles chaque processeur est doté de sa propre mémoire (programme et données) (FIG. 4.4).

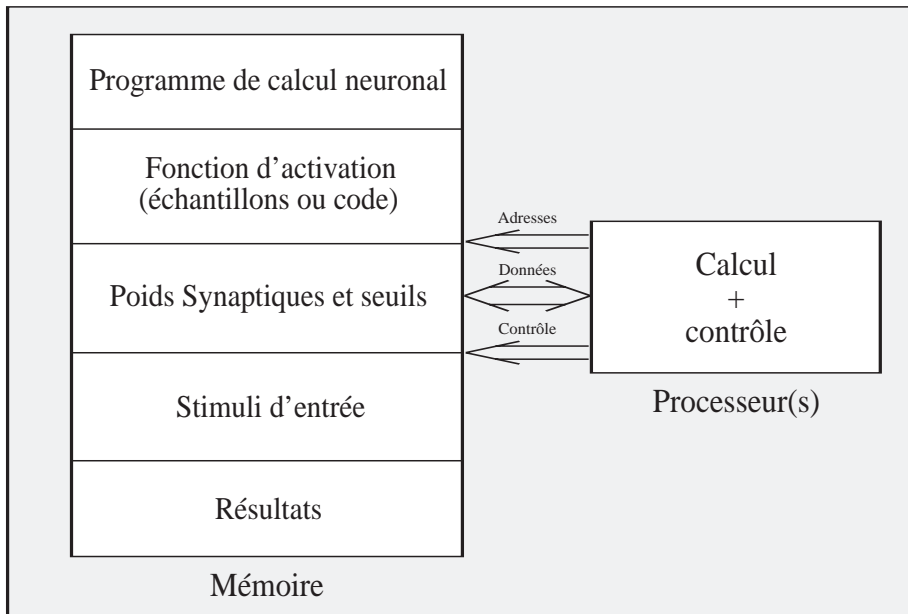


FIG. 4.3 : Schéma bloc d'une implantation/émulation neuronale à une seule mémoire

La zone programme contient le code exécutant les calculs neuronaux. Les poids sont déterminés durant la phase d'apprentissage du réseau et sont codés, ainsi que les stimuli d'entrées, selon le format adopté pour le circuit sous test. Quant à la fonction d'activation, elle peut soit être stockée sous forme échantillonnée (look-up table), soit évaluée par programme. Notons que toutes ces informations peuvent être stockées aussi bien dans des mémoires externes au circuit sous test ou dans des zones internes (mémoire et/ou registres).

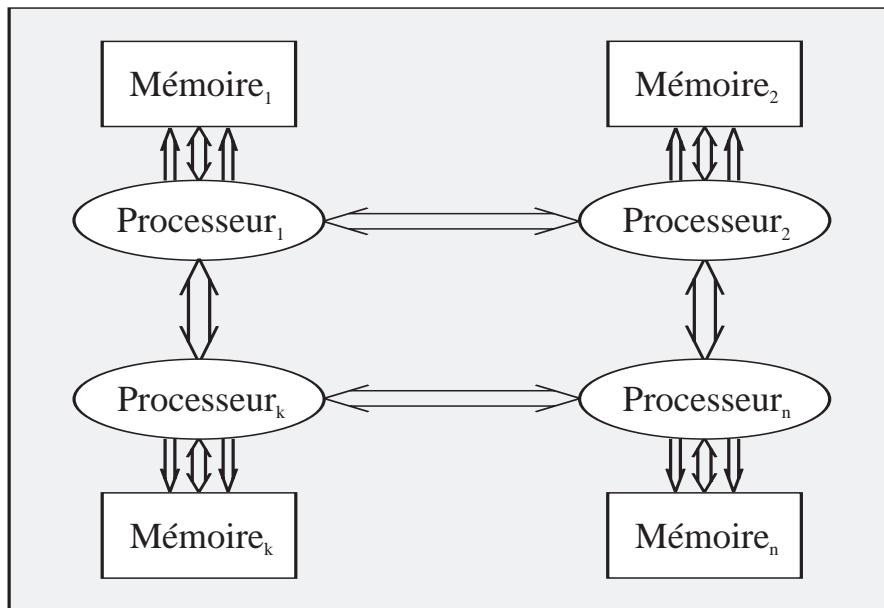


FIG. 4.4 : Schéma bloc d'une implantation/émulation neuronale à plusieurs mémoires

4.4 Mise en œuvre de tests des ANN matériels

Il s'agit essentiellement du montage utilisé pour les expériences de type injection de fautes et tests aux ions lourds. Le testeur autour duquel ces expériences sont mises en œuvre est le FUTE16 [Velazco92] qui a été spécifiquement développé pour le test d'upsets des circuits de type processeurs. Après avoir analysé les protocoles d'accès de certains circuits neuronaux existants, et afin d'éviter les automates de contrôle nécessaires pour s'adapter aux particularités propres à ce nouveau type de composants, nous avons rajouté une interface intelligente entre le testeur FUTE16 et les circuits neuronaux. Cette carte d'interface, construite autour d'un microprocesseur, a pour tâches :

1. Le chargement (dans des RAM internes aux composants sous test ou dans des mémoires externes) des configurations des réseaux implantés et éventuellement des stimuli à traiter dans la phase de rappel ;
2. Le lancement du processus de rappel (et éventuellement d'apprentissage) ;
3. La récupération des résultats.

Ces tâches sont accomplies par un programme, écrit en langage assembleur du microprocesseur maître, et dont le code sera résident dans la RAM de FUTE16. De cette manière les circuits neuronaux à tester seront vus comme des périphériques

toujours d'un même microprocesseur, permettant de récupérer toutes les fonctions de FUTE16 propres au test d'upsets.

Nous avons choisi de construire la carte d'interface entre les circuits neuronaux et FUTE16 autour d'un microprocesseur MC68000. Le schéma donné sur la figure 4.5 représente le principe de la solution adoptée. Les circuits neuronaux possèdent en général un jeu d'instructions simple qui permet de les configurer pour un problème donné, ainsi que d'effectuer les opérations nécessaires à l'évaluation de l'état des neurones du réseau simulé. Le programme correspondant doit être chargé par le 68000 dans une mémoire se trouvant selon les cas, à l'intérieur du circuit neuronal sous test ou à l'extérieur, sur la carte déportée.

Après les chargements correspondant à la phase d'initialisation, la phase de rappel est déclenchée par application d'un signal de type "reset" sur les circuits neuronaux concernés. Durant l'activité du circuit neuronal (exécution de la phase de rappel) le 68000 maître exécute une boucle d'attente, jusqu'à l'activation d'une interruption provenant du circuit neuronal et indiquant la fin des calculs. Les résultats finals et l'état du circuit sous test (CST) sont ensuite récupérés et analysés (comparaison avec des motifs de référence) par le microprocesseur maître. A la fin du programme de test, le 68000 envoie un signal adéquat au testeur afin de lui passer la main.

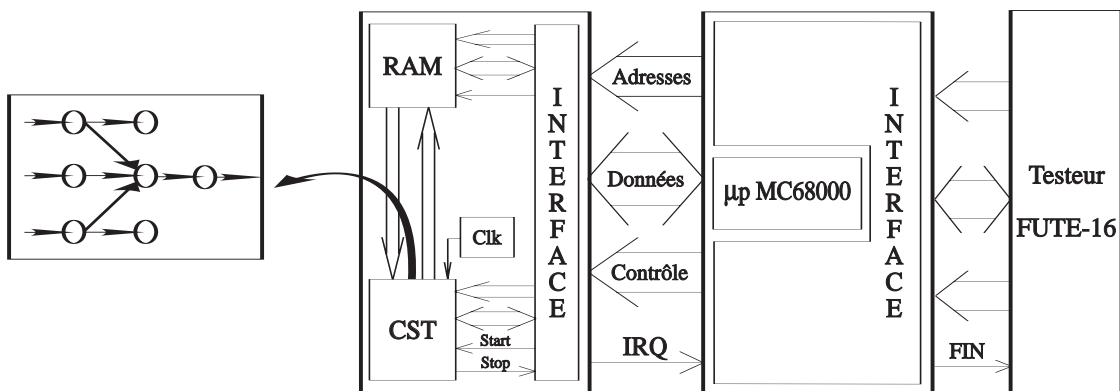


FIG. 4.5 : Interface FUTE16/circuits neuronaux

Un prototype de carte maître 68000 a été conçu, réalisé et validé (annexe D). Le dialogue avec les circuits neuronaux à tester est réalisé via les bus d'adresses, de données et de contrôle, en considérant le circuit à tester comme un périphérique du 68000 (ensemble de zones mémoire adressables par des cycles normaux de lecture/écriture).

4.5 Circuits cibles et zones sensibles correspondantes

4.5.1 Contexte

A l'origine, il était prévu que l'étude se fasse autour d'un circuit neuronal représentatif, choisi parmi les circuits disponibles. Compte tenu de la diversité des circuits neuronaux existants (cf. §2.8.3), ainsi que des risques de verrouillage d'alimentation (latchup) entraînés par les essais SEU sur des circuits en versions CMOS standards, nous avons décidé de choisir plusieurs circuits afin de pouvoir élargir les chances de succès de ce projet.

Trois circuits digitaux dédiés aux réseaux de neurones (l'UTAK, le L-Neuro 1.0 et l'OCR) et deux processeurs à usage général (le DSP96002 et le T225) ont été considérés. Une carte de test a été développée autour de chacun d'eux sauf dans le cas du DSP où l'environnement n'a pas été le même. Les schémas de réalisation sont donnés dans les annexes E, F et G.

4.5.2 Le circuit UTAK

Le circuit UTAK¹ [Castillo91] est un prototype de circuit neuronal à usage général construit autour d'un processeur ayant un jeu de 16 instructions.

Un seul circuit UTAK permet d'émuler des réseaux de neurones relativement simples, ayant un maximum de 3 couches cachées et de 3 neurones par couche. Plusieurs UTAK peuvent être utilisés ensemble (sous forme d'un anneau) afin d'émuler des réseaux de neurones plus complexes (avec plus de couches et plus de neurones par couche).

Les connexions entre UTAK, dans une application qui en contient plusieurs, se font par l'intermédiaire de 3 ports de communication dont la programmation permet d'utiliser une même architecture en anneau pour plusieurs applications.

L'architecture de l'UTAK comporte les unités suivantes :

- une RAM interne de 128 octets où seront stockés les poids synaptiques du réseau de neurones implanté ;

1. Unit for Trained Adaptive Knowledge, conçu et fabriqué par l'Université Polytechnique de Catalogne – UPC (Catalogne) en collaboration avec la société SIDA (Madrid).

- une ROM interne de 256 octets où est stockée la fonction sigmoïde utilisée pour calculer l'activation (sortie) des neurones ;
- deux ports unidirectionnels et un port bidirectionnel pour les communications inter-UTAKs dans une architecture multicircuits ;
- des opérateurs (additionneur, multiplieur, etc.) et des zones de mémorisation (registres d'état, accumulateurs) nécessaires au calcul et au stockage des états des neurones.

4.5.3 Le circuit L-Neuro 1.0

Le L-Neuro 1.0² [Mauduit92] est un coprocesseur digital commercial dédié à l'accélération des calculs neuronaux, et qui permet d'implanter, en utilisant un ou plusieurs circuits, la plupart des applications neuronales existantes. Le L-Neuro est programmable à l'aide d'un jeu de 45 instructions et intègre des automates de rappel et d'apprentissage. A un instant donné, le L-Neuro permet de simuler un neurone ayant au plus 16 entrées.

L'architecture du L-Neuro prévoit l'interfaçage avec une mémoire extérieure contenant une table d'interpolation de la fonction d'activation choisie, nécessaire aux calculs neuronaux. Le calcul de l'état d'un neurone est effectué de sorte à implanter un parallélisme convergent sur les entrées, dans ce type de parallélisme, la sortie d'un neurone est évaluée à chaque top d'horloge en fonction de la valeur de ses entrées au top d'horloge précédent.

4.5.4 Le circuit OCR

Le circuit OCR³ [Jacquet94] est un composant neuronal digital à haute performance permettant d'implanter des réseaux de neurones multicouches. Les algorithmes d'apprentissages supportés sont de type BP (rétro-propagation de l'erreur [Rumelhart86a]) ou LVQ (Learning Vector Quantization [Kohonen88]). Le circuit OCR a été conçu pour être utilisé comme un accélérateur matériel de reconnaissance de caractères mais peut, grâce à sa flexibilité, être utilisé pour implanter d'autres

2. Learning Neurochip, conçu et fabriqué par le Laboratoire d'Electronique de Philips – LEP (Paris).

3. Optical Character Recognition chip, conçu et fabriqué par le laboratoire Systèmes et Circuits Intégrés – SCI (Grenoble) en collaboration avec la société MIMETICS (Paris).

applications. Les calculs des états des différents neurones implantés sont effectués par l'intermédiaire de 5 processeurs dédiés ($P_1 \cdots P_5$).

4.5.5 Le processeur de signaux DSP96002

Le DSP96002 [Motorola91] est un processeur digital 32 bits capable de traiter des nombres en virgule flottante, simple et double précision, selon le standard IEEE 754–1985. Il s'agit d'un processeur monolithique CISC (Complex Instruction Set Computer) à port dual intégrant :

1. Une RAM de 2×512 mots de 32 bits pour le stockage des programmes.
2. Deux ROM où sont stockées les tables d'interpolation des fonctions sinus et cosinus.
3. Un banc de registres de données de 10 mots de 96 bits.
4. Trois bancs de registres d'adresses (8 registres de 32 bits chacun par banc).
5. Six registres de contrôle de programmes de 32 bits chacun.
6. Un contrôleur DMA à double canal.
7. Un émulateur intégré permettant d'accéder aux ressources internes du circuit.

Du point de vue architectural, le 96002 possède 3 unités d'exécution indépendantes capables de fonctionner en parallèle (l'unité arithmétique et logique, l'unité de génération d'adresses et l'unité de contrôle du programme). Cette architecture permet l'exécution simultanée d'opérations telles que : recherche d'instructions, transfert de données, opérations en virgule flottante et mise à jour des pointeurs d'adresses. En plus, ces opérations peuvent être exécutées en parallèle avec l'accès mémoire à travers des canaux DMA.

L'accès aux mémoires internes se fait en un seul cycle d'horloge tandis que plusieurs cycles sont nécessaires pour accéder aux mémoires externes. La structure à port dual permet de connecter le 96002 à des mémoires externes de différents types (SRAM, DRAM, VRAM). En outre, chaque port peut être configuré pour fonctionner en "mode hôte" ce qui permet d'implanter des architectures multiprocesseurs.

4.5.6 Le transputer T225

Il s'agit d'un processeur RISC (Reduced Instruction Set Computer) 16 bits de la famille des transputers [Inmos92]. Le jeu d'instruction réduit des transputeurs bien adapté aux calculs neuronaux (multiplications et additions ayant un code opération compact et un temps d'exécution réduit), en fait un bon choix pour les émulations matérielles des réseaux de neurones. A cet avantage s'ajoute la possibilité d'exécution de processus concurrents et l'échange direct de données entre circuits dans une architecture qui en contient plusieurs, grâce à des liens de communications séries intégrés. Le choix du T225 a été motivé par la disponibilité de données sur son comportement sous radiations.

4.5.7 Zones sensibles aux SEU des circuits cibles

Le tableau 4.1 résume les zones sensibles de chacun des circuits cibles ainsi que leur taux d'accessibilité directe par programme correspondant.

Circuit	RAM interne	Registres	Total	Accessibilité (%)
UTAK	128 × 8 bits	87 bits	1111 bits	94.7
L-Neuro 1.0	1024 × 8 bits	608 bits	8800 bits	99.6
OCR	2622 × 8 bits	770 bits	21746 bits	96.5
DSP96002	512 × 32 bits	1920 bits	18304 bits	100.0
T225	4096 × 8 bits	240 bits	33008 bits	100.0

TAB. 4.1 : Zones sensibles aux upsets des différents circuits cibles

4.6 Choix des réseaux de neurones à tester

Plusieurs réseaux ont été testés; il s'agit de perceptrons multicouches (MLP) dont certains correspondent à des "cas d'école" (évaluation de fonctions booléennes, classification de motifs), d'autres à des problèmes plus complexes (reconnaissance de caractères et détection de sifflements à bord des engins spatiaux).

4.6.1 Réseau de classification de motifs

Il s'agit d'un ANN simple de type perceptron [Rosenblatt58] (FIG. 2.3) réalisant une fonction de classification de motifs, l'ensemble d'exemples d'apprentissage ayant

été choisi aléatoirement. Vu qu'un perceptron ne peut classifier que des entrées linéairement séparables dans l'espace des variables correspondant, les sorties du réseau traité ont été déterminées par un perceptron maître, dont les poids, ont été choisi aléatoirement. L'apprentissage a été réalisé par l'algorithme *minimerror* [Gordon93] qui permet de trouver une solution ayant des performance de généralisation optimales.

4.6.2 Réseaux d'évaluations des fonctions booléennes

Il s'agit en effet d'un réseau calculant 3 fonctions booléennes y_1 , y_2 et y_3 à 3 variables x_1 , x_2 et x_3 (eq. 4.1, 4.2 et 4.3) effectuant respectivement le OU exclusif des entrées, le test de leur parité (1 si le nombre d'entrées à 1 est impair) et le test d'égalité (1 si les 3 entrées sont différentes) (FIG. 4.6). Afin d'étudier l'influence de l'architecture du réseau (nombre de couches) sur la sensibilité aux SEU, nous avons choisi de calculer les mêmes fonctions booléennes en utilisant 3 perceptrons multicouches (MLP) différents avec 1, 2 et 3 couches cachées respectivement (les réseaux MLP1 (3-3-3), MLP2 (3-3-3-3) et MLP3 (3-3-3-3-3) de la figure 4.7). Ces réseaux sont entièrement connectés : les neurones d'une même couche ont leurs sorties connectées à chacun des neurones de la couche adjacente en aval. L'apprentissage des réseaux a été réalisé par l'algorithme standard de retropropagation des erreurs.

$$y_1 = x_1 \oplus x_2 \oplus x_3 \quad (4.1)$$

$$y_2 = \overline{x_1 \oplus x_2 \oplus x_3} \quad (4.2)$$

$$y_3 = (x_1 + x_2 + x_3)(\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \quad (4.3)$$

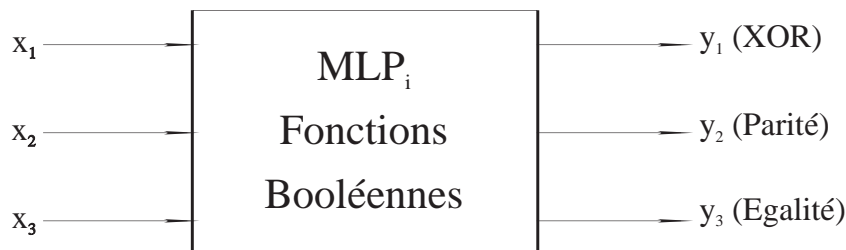


FIG. 4.6 : Fonctions booléennes évaluées

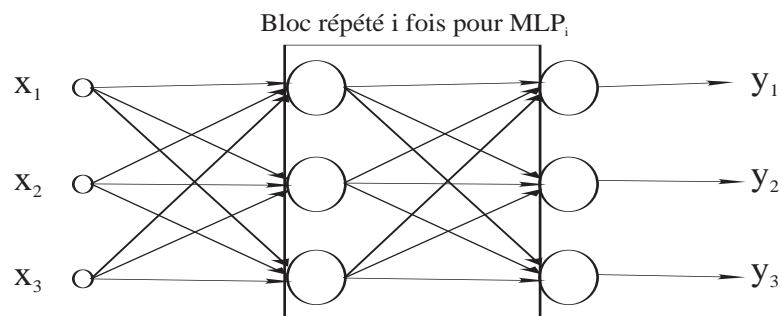


FIG. 4.7 : Réseaux MLP_i correspondant aux fonctions booléennes

4.6.3 Réseau de reconnaissance de caractères

Le domaine de la reconnaissance de caractères est l'un des premiers domaines dans lesquels les réseaux de neurones ont été utilisés. En effet, la grande variation de taille, de position, de police et de style d'écriture rend cette tâche difficile pour les techniques traditionnelles de reconnaissance. Ce type de traitement de l'information est facilement accompli par les réseaux de neurones artificiels grâce à leur propriété d'apprentissage. L'application que nous décrivons dans ce paragraphe est implantée sur le circuit OCR et consiste en la reconnaissance de caractères latins typographiques écrits suivant différentes polices et différents styles. Le schéma du réseau correspondant est donné sur la figure 4.8.

Les caractères à reconnaître sont présentés au circuit sous forme d'une matrice de 24×24 octets, appelée bitmap, dont chaque élément code un niveau de gris sur 8 bits et correspond à un pixel du caractère à reconnaître.

4.6.4 Réseau de détection de sifflements

4.6.4.1 Description du problème

Le rayonnement électromagnétique, associé aux éclairs d'orages atmosphériques, est couramment observé dans la gamme des extrêmement basses fréquences et très basses fréquences (EBF/TBF) par les satellites magnétosphériques. Cependant, les limites des mémoires de bord des satellites et surtout des systèmes de télémétries empêchent une observation continue de ce phénomène. Dans ces conditions, de nombreuses questions restent posées sur les effets des éclairs atmosphériques : augmentation globale du niveau du bruit électromagnétique, précipitations de particules énergétiques et déclenchement d'interaction ondes-ondes et ondes-particules pouvant donner naissance à des émissions continues. En effet, dans la plupart des expériences de

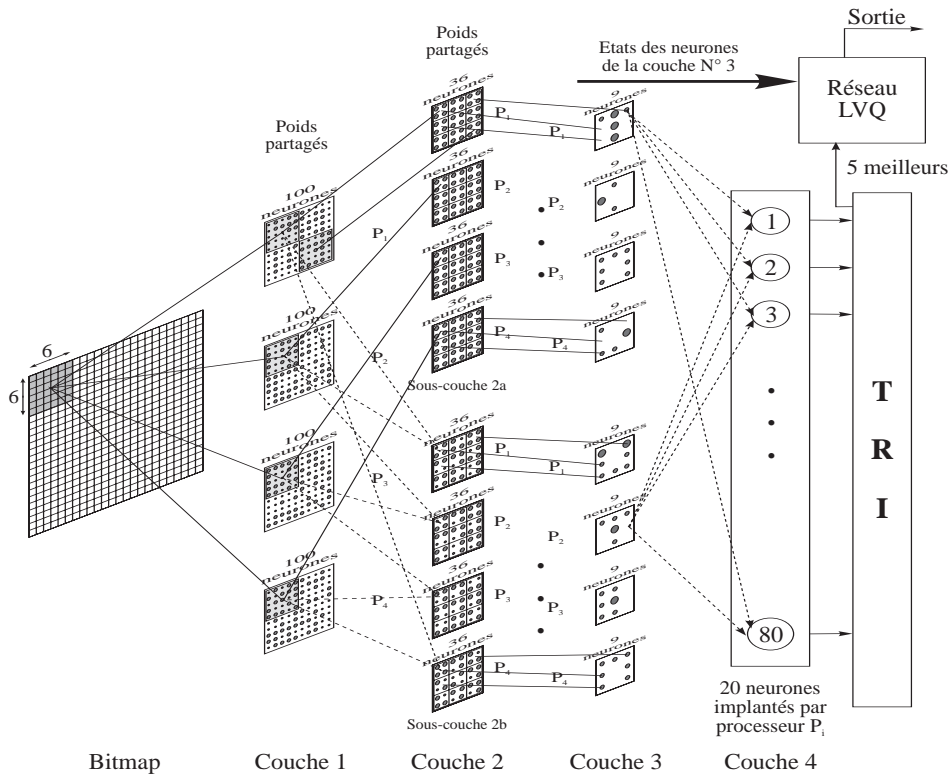


FIG. 4.8 : Réseau de neurones utilisé pour la reconnaissance de caractères

mesures spatiales, les observations sont faites à la sortie des filtres intégrateurs analogiques ou numériques. A l'évidence, ce type de mesure ne conserve pas d'information sur les émissions discrètes. Cependant, celles-ci peuvent être présentes et peuvent jouer un rôle clé dans la génération d'événements observés. Plusieurs phénomènes peuvent être à l'origine de ces émissions mais les données disponibles actuellement ne permettent pas de le confirmer. Parmi ces phénomènes, nous pouvons citer la présence des rayonnements électromagnétiques associés aux éclairs atmosphériques [Hayakawa89] et les ondes électromagnétiques associées aux tremblements de terre [Parrot93]. La plupart des données disponibles sont issues de batteries de filtres et supposent qu'aucun phénomène discret ne vienne perturber la mesure. Cependant, pour les données d'un événement enregistré par le satellite AUREOL-3 (expérience ARCAD-3) et présenté en [Serebryakova92], les formes d'ondes sont disponibles. Leur analyse montre l'existence conjointe d'émissions EBF et TBF à bande étroite, appelés "sifflements", bien qu'elle ne permette pas de statuer sur la présence d'émissions discrètes larges bandes dans les émissions associés aux tremblements de terre.

La solution idéale pour résoudre ce type de problème consisterait à transmettre au sol, en continu, les données formes d'ondes associées à la mesure d'une ou de plusieurs

composantes du champ électromagnétique. Pour une fréquence maximale de 20 KHz, cela correspond à un débit de télémétrie de 40 bits/s (mots de 8 bits) pour chaque composante. Ce débit n'est pas très important pour des observations de quelques minutes. Il devient cependant difficile à gérer pour des observations continues. L'alternative est une réduction, puis une compression, des données à bord. C'est dans cette direction que le travail réalisé par le LPCE⁴ est orienté. Il s'agit de concevoir un système qui identifie et classe des émissions structurées de type sifflement. Le but du LPCE était de réaliser un analyseur qui soit, d'une part, temps réel, c'est à dire que son temps d'analyse soit inférieur au temps de collecte de données nécessaires à l'analyse suivante ; et d'autre part, embarquable sur satellite, c'est à dire que son poids, son volume, sa puissance consommée et sa stabilité soient compatibles avec les conditions de bord.

4.6.4.2 Choix de l'approche neuronale pour la détection des sifflements

Plusieurs approches ont été testées par le LPCE pour aborder le problème de la détection de sifflements : les méthodes classiques de classification notamment la méthode du plus proche voisin et la méthode de K-moyennes, et les approches réseaux de neurones artificiels. Pour les approches classiques, une amélioration a été introduite notamment lors du choix des vecteurs prototypes, dans le but de permettre la reconnaissance des sifflements non calés dans la matrice d'entrée ce qui a augmenté le taux de reconnaissance correspondant. Les performances maximales atteintes pour chacune des approches sont résumées dans le tableau 4.2 extrait de [Minière94] en termes de pourcentage de reconnaissance correctes pour une base de 1000 patterns.

Les résultats donnés dans le tableau 4.2 montrent les performances des diverses approches étudiées et l'avantage de la solution neuronale TDNN par rapport aux autres solutions. C'est la raison principale pour laquelle cette solution a été adoptée par le LPCE pour implanter l'application de détection des sifflements. D'autres raisons qui ont conduit à l'utilisation de l'approche neuronale sont :

- des banques volumineuses de données existent, ce qui permet la constitution de bases de données complètes pour la phase d'apprentissage et pour les tests de validation ;
- une fois la phase d'apprentissage effectuée en laboratoire, un réseau de neurones

4. Laboratoire de Physiques et Chimie de l'Environnement (Orléans).

Approche utilisée	Performance atteinte (%)
Plus proche voisin	75.6
K-moyennes	65.4
Plus proche voisins améliorée	82.1
K-moyenne améliorée	86.2
Perceptron multicouches (1 couche cachée)	70.1
Perceptron multicouches (2 couches cachées)	74.2
ANN partiellement connecté à entrées partagées	78.3
Réseau TDNN à poids et entrées partagés	87.8

TAB. 4.2 : Performances atteintes par les différentes approches testées

nécessite un temps de calcul minimum et constant, ce qui rend possible une analyse temps réel ;

- la même architecture de réseau peut être utilisée pour l'étude de phénomènes différents ; par exemple, le même type de réseau peut être adapté à la détection des sifflements protoniques et électroniques ;

4.6.4.3 Description du réseau EBF

Le LPCE a adopté un type de neurone particulier appelé dans la terminologie anglo-saxonne "Time Delay Neuron" (TDN), décrit et utilisé par [Waibel89] pour la reconnaissance de phonèmes. Ce type de neurone est bien adapté à la structure, dans le domaine de temps/fréquence, des spectres obtenus pour les sifflements puisqu'il permet de suivre l'évolution temporelle du signal d'entrée échantillonné. Le réseau résultant comporte 39 TDN organisés en 3 couches partiellement connectées ayant 32, 6 et 1 neurones respectivement (FIG. 4.9). Les stimuli sont présentés au réseau sous forme d'une matrice 10×5 dont les éléments sont les échantillons de la densité spectrale de puissance normalisée du spectre mesuré du signal à analyser et qui sont pris à 10 instants successifs pour chacune des 5 fréquences sélectionnées. Cette matrice résulte d'un prétraitement effectuée sur les formes d'onde (l'amplitude du signal à analyser en fonction du temps).

Il est clair que la réponse attendue du réseau est binaire (classification des entrées présentées en 2 classes correspondant ou pas à un sifflement). Cependant, la sortie des neurones TDN considérés est une valeur non binaire (entière ou flottante suivant

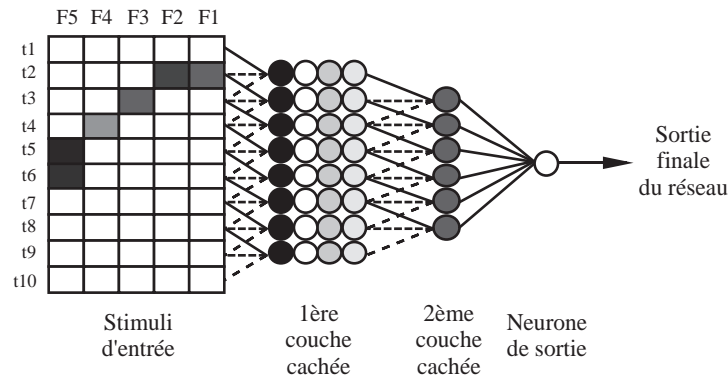


FIG. 4.9 : Réseau de neurones TDNN utilisé pour la détection de sifflements

le format de données adopté (cf. eq. 2.2)). C'est pour cela qu'une conversion de la sortie délivrée par le réseau en une décision binaire est effectuée en la comparant à un seuil de sortie prédéterminé Th_{out} . La valeur du seuil est fixée durant l'apprentissage de manière à rendre maximale la performance du réseau. Dans notre cas, la reconnaissance de sifflements a lieu quand la sortie du réseau appartient à l'intervalle $[Th_{out}, 1]$. La figure 4.10 montre quelques exemples des stimuli d'entrée ainsi que la réponse du réseau (état du neurone de sortie) correspondante. Ces exemples montrent que le sifflement est mieux reconnu quand il est bien centré dans la matrice d'entrée (densité spectrale de puissance échantillonnée), ce qui est le cas des exemples 3 et 4 (FIG. 4.10).

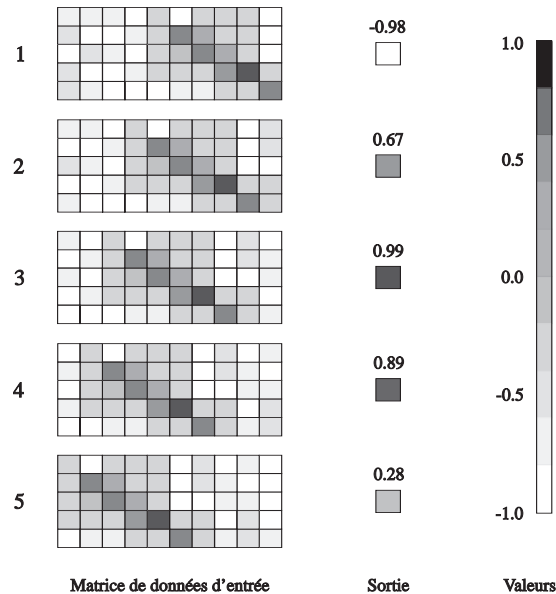


FIG. 4.10 : Exemples de stimuli d'entrée du réseau et sorties associées

L'une des caractéristiques du réseau adopté, dont le schéma est donné sur la figure

4.9 pour le cas de sifflements protoniques (réseau EBF), est que, dans la première couche de neurones, les neurones TDN de la même colonne ont les mêmes poids synaptiques (réseau à poids partagés) alors que ceux de la même ligne ont les mêmes entrées (réseau à entrées partagées). De plus, deux neurones adjacents d'une même colonne ont leurs stimuli d'entrée décalés dans la matrice des entrées, correspondant donc à deux instants successifs d'échantillonnage. Des détails sur l'architecture du réseau ainsi que sur ses performances lorsqu'il est utilisé pour classifier des données réelles sont donnés en [Minière96].

4.7 Programmes de test

Les programmes de test développés pour les différents circuits testés sont les suivants :

Programmes de test de l'UTAK

Test statique

MemStat est le programme de test statique des 128 octets de la RAM interne et des registres accessibles de l'UTAK.

Test dynamique

Il s'agit des programmes d'implantation sur l'UTAK, des 3 réseaux MLP_i ($i \in \{1, 2, 3\}$) d'évaluation des fonctions booléennes. Ces programmes sont écrits en langage assembleur 68000/UTAK (*1_Couche*, *2_Couche* et *3_Couche* respectivement)

Programme de test du L-Neuro 1.0

Test statique

NeuroSta est le programme de test statique des 1024 octets de la RAM synaptique du L-Neuro ainsi que des bancs de registres d'états neuronaux (BNSR).

Test dynamique

Différents réseaux ont été implantés sur le L-Neuro 1.0. Il s'agit du problème d'évaluation des fonctions booléennes (réseaux MLP_i , $i \in \{1, 2, 3\}$) et de l'application de détection des sifflements protoniques (réseau EBF). Les programmes correspondants sont écrits en langage assembleur 68000/L-Neuro 1.0 (*1_Layer*, *2_Layer* et *3_Layer* et EBF_{Lneuro} respectivement).

Programmes de test de l'OCR

Test statique

Le programme *StatRam*, correspond au test de la portion de la RAM interne accessible en lecture/écriture (1 Koctets).

Test dynamique

La reconnaissance d'un caractère par l'OCR, nécessite le chargement de la configuration LVQ dans les 128 Koctets de RAM externe. Cette RAM est accédée par le processeur P_5 du circuit OCR lors de l'activation de la dernière phase de traitement du réseau. Compte tenu des contraintes de capacité mémoire de FUTE16 (taille maximale d'un programme de test égale à 64 Koctets) ces initialisations sont effectuées par quatre programmes de chargement différents appelés $Charg_iRo$, ($i \in \{1, 2, 3, 4\}$). La configuration BP du réseau OCR quant à elle, sera chargée dans les 3 Koctets de RAM interne en même temps que la bitmap du caractère à reconnaître et que la fonction d'activation (sigmoïde) échantillonnée. La carte développée permet la reconnaissance d'un caractère en 10 *msec* et le programme 68000-OCR correspondant s'appelle *ItMap*.

Programmes de test du DSP96002 et du T225

Le même problème a été traité par les deux processeurs. Il s'agit de l'émulation de l'application de détection des sifflements protoniques, le but étant d'étudier le comportement, face aux radiations, des processeurs standards quand ils exécutent des programmes émulant des réseaux de neurones. Les programmes correspondants s'appellent EBF_{DSP} et EBF_{T225} pour le DSP96002 et le T225 respectivement.

L'ensemble des programmes de test et les expériences réalisées sont résumés dans les tableaux 4.3 et 4.4 respectivement.

Conclusion

Dans ce chapitre ont été citées les différentes stratégies de test adoptées dans cette thèse (simulations et injection de fautes, tests aux ions lourds). Les circuits choisis pour réaliser ces tests ont été également décrits. Il s'agit de trois circuits digitaux dédiés aux réseaux de neurones et deux processeurs à usage général (un CISC et un RISC). Plusieurs réseaux ont été sélectionnés et varient entre des "cas d'école" (évaluation des fonctions booléennes) et des applications commerciales (reconnaissance des caractères) et spatiales (détection des sifflements à bord des satellites). Les résultats des tests effectués sur les circuits et les réseaux choisis et pour les différentes stratégies adoptées seront donnés dans les chapitres suivants.

Programme	Description
<i>MemStat</i>	Test statique de l'UTAK
<i>i_Couche</i>	Test Dynamique de l'UTAK, évaluation du réseau MLP_i
<i>NeuroSta</i>	Test statique du L-Neuro 1.0
<i>i_Layer</i>	Test Dynamique du L-Neuro 1.0, évaluation du réseau MLP_i
<i>EBF_{Lneuro}</i>	Test Dynamique du L-Neuro 1.0, détections des sifflements
<i>StatRam</i>	Test statique de l'OCR
<i>Charg_iRo</i>	Test dynamique de l'OCR, chargement des RAM externes [i/4]
<i>ItMap</i>	Test dynamique de l'OCR, reconnaissance d'une bitmap
<i>EBF_{DSP}</i>	Test Dynamique du DSP96002, détections des sifflements
<i>EBF_{T225}</i>	Test Dynamique du T225, détections des sifflements

TAB. 4.3 : Liste des programmes de test des différents circuits cibles

	Simulations logicielles	Injection matérielle	Test aux ions lourds
Perceptron	•		
MLP_i		UTAK, L-Neuro	UTAK, L-Neuro
OCR		OCR	OCR
EBF	•	L-Neuro, DSP et T225	L-Neuro, DSP et T225

TAB. 4.4 : Expériences réalisées pour les différents réseaux choisis

Chapitre 5

Résultats des simulations logicielles d'upsets

Résumé du contenu du chapitre

Les simulations logicielles constituent un outil intéressant pour l'évaluation de la tolérance aux fautes des réseaux de neurones, en particulier les fautes de type upset. Ce chapitre présente les résultats de simulations d'upsets obtenus pour les réseaux correspondant aux problèmes de classification des motifs et de détection des sifflements protoniques.

5.1 Introduction

Comme il a été dit dans le chapitre précédent, l'intérêt des simulations logicielles réside dans le fait qu'il permettent d'examiner la sensibilité d'un réseau de neurones bien qu'elles ne permettent pas de prendre en compte tous les types d'erreurs possibles, en particulier les erreurs survenant sur le code du programme exécuté. De telles simulations ne nécessitent qu'un développement logiciel (programme correspondant au réseau testé) et peuvent s'appliquer pour différents formats de données. Les erreurs simulées sont du type upsets (inversion de bits) simples ou multiples et sont introduites avant le lancement du programme correspondant aux calculs neuronaux. Les résultats obtenus pour les différents réseaux testés sont donnés dans ce qui suit.

5.2 Résultats obtenus pour le perceptron

Vu que la différence entre les résultats obtenus pour les 2 configurations testées (perceptrons ayant 20 et 50 neurones respectivement) n'était pas significative, nous allons présenter les résultats obtenus pour le perceptron à 50 entrées. Deux types d'expériences ont été réalisées et correspondent à la perturbation des motifs d'entrée et des poids synaptiques respectivement. La performance du réseau a été mesurée en termes de sa capacité à donner une bonne réponse pour des motifs d'entrées correspondant à la base d'exemples d'apprentissage du réseau. La taille de cette dernière est donnée en terme du rapport α du nombre de motifs P sur le nombre de poids à déterminer N (eq. 5.1).

$$\alpha = \frac{P}{N} \quad (5.1)$$

L'influence de α sur la capacité de généralisation du réseau est bien établie et est donc un paramètre à considérer. Pour cela les tests réalisés ont été répétés pour différentes valeurs de α . Notons que le choix aléatoire des motifs d'entrée fait que la représentation de la connaissance au sein du réseau est distribuée de manière homogène entre les différents poids synaptiques, contrairement à la plupart des applications dans lesquelles les données sont généralement redondantes, ce qui améliore naturellement la tolérance aux fautes des réseaux appris. Par conséquent, les résultats présentés dans ce qui suit doivent être considérés comme pire cas plutôt que génériques.

5.2.1 Effets des perturbations sur les motifs d'entrée

Dans une première expérience, nous avons perturbé aléatoirement b bits de chacun des motifs d'entrée (base d'apprentissage). Si la réponse du réseau au motif corrompu est identique à l'originale, celui-là est considéré comme étant reconnu. Le taux de reconnaissance ρ , qui correspond au pourcentage de sorties classifiées correctement, est moyenné sur 200 ensembles d'apprentissage différents et sera fonction du nombre b des bits perturbés (FIG. 5.1). Les simulations ont été réalisées pour 6 tailles d'apprentissage différentes, $\alpha = 1 \dots 6$, seulement le cas $\alpha = 6$ est montré vu que les résultats sont quasi-indépendants de la taille de l'ensemble d'apprentissage. La dégradation varie linéairement avec le nombre de bits perturbés. Pour une altération d'un seul bit, le réseau reconnaît correctement 96% des motifs d'entrée, et son taux de reconnaissance diminue presque linéairement avec l'augmentation du nombre b de bits perturbés (eq.

5.2).

$$\rho = 100 \frac{N - b}{N} \quad (5.2)$$

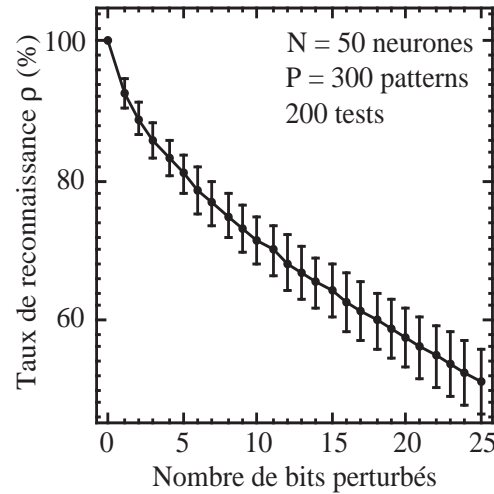


FIG. 5.1 : Taux de reconnaissance ρ vs. le nombre b de bits perturbés dans les motifs d'entrée

5.2.2 Perturbations des poids synaptiques

Dans cette expérience, nous déterminons le taux de reconnaissance ρ pour la perturbation des poids. Les poids sont codés au format entier 16 bits, le bit 15 étant le bit de signe. Pour chacun des perceptrons étudiés, la sortie du réseau a été déterminée après avoir modifié chaque bit de chaque poids. Le taux de reconnaissance ρ , moyenné sur 50 poids de chaque perceptron et sur 50 réseaux, est présenté sur la figure 5.2 en fonction de la position du bit perturbé, pour différentes tailles d'apprentissage. Les modifications du bit de signe affectent sérieusement la reconnaissance, tandis que la perturbation des bits non significatifs (les bits 0 à 7) n'a pas d'influence sur la sortie.

Vu qu'un upset sur les poids peut affecter n'importe quel bit avec la même probabilité, le taux moyen de reconnaissance par bit $\bar{\rho}$ a été évaluée en fonction de la taille de la base d'apprentissage α (FIG. 5.3). Pour le codage entier utilisé, l'effet d'un upset dégrade la performance du réseau selon la taille de l'apprentissage d'une façon approximativement linéaire (eq. 5.3). Plus le processus d'apprentissage est précis (valeur de α grande), plus le taux de reconnaissance diminue en présence de fautes.

$$\bar{\rho} = 87.8 - 2.5 \alpha \quad (5.3)$$

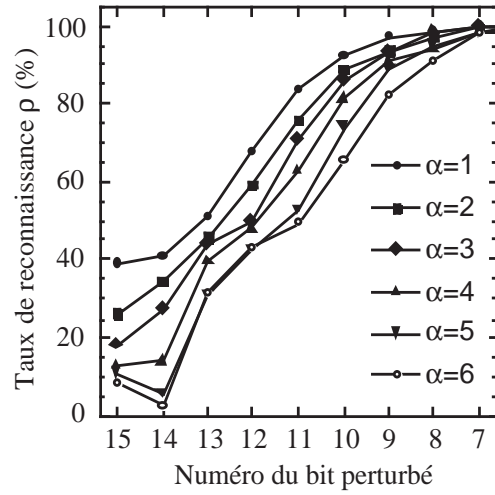


FIG. 5.2 : Taux de reconnaissance ρ vs. le nombre b de bits perturbés dans les poids synaptiques

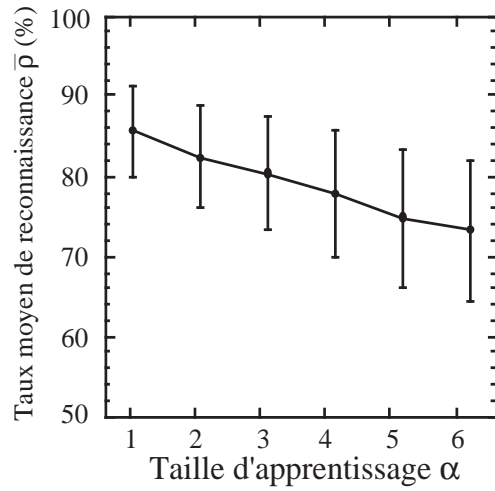


FIG. 5.3 : Taux de reconnaissance moyen $\bar{\rho}$ par faute vs. la taille d'apprentissage α

5.3 Résultats obtenus pour le réseau EBF

Un programme simulant le fonctionnement du réseau de détection de sifflements a été développé en langage évolué (C) et a été lancé avec des données (stimuli d'entrée ou poids) corrompues en observant la sortie correspondante afin de détecter d'éventuelles erreurs.

Vu que les valeurs traitées sont réelles et qu'elles peuvent être stockées dans des zones mémoires sensibles aux upsets, la tolérance aux fautes du réseau simulé peut dépendre du format utilisé pour le codage des données. D'où, deux types de codages différents ont été considérés : un codage sur 8 bits (complément à 2) et un codage flottant sur 32 bits (le format standard IEEE 754–1985). De même, afin d'étudier l'influence de la distribution des poids sur la tolérance aux fautes, 2 versions du réseau EBF, ont été testées. Dans la première version, dite à poids non partagés, il a été supposé que les poids synaptiques et les seuils des différents neurones (même ceux ayant des valeurs égales), sont stockés dans des emplacements mémoires différents. A priori, dupliquer les poids conduit à une sensibilité plus grande : la zone sensible de l'implantation sera environ 7 fois plus grande si les paramètres des neurones sont stockés de manière redondante (597 valeurs plutôt que 79 pour le réseaux à poids partagés). Mais de l'autre côté, les upsets sur les paramètres partagés affecteraient une portion plus grande du réseau.

Vu la grande taille de la base de données disponibles sur les signaux à analyser (1000 motifs d'entrée environ), les simulations ont été effectuées sur un sous-ensemble de 100 motifs arbitrairement choisis dont les 50 premiers correspondent à des sifflements pour un seuil de sortie $Th_{out} = 0.85$.

5.3.1 Résultats obtenus pour la perturbation des stimuli d'entrée

Chacun des bits du stimuli d'entrée a été inversé, un à la fois, avant de lancer l'exécution du programme correspondant à l'application EBF avec l'entrée corrompue. Ceci a été répété pour l'ensemble de 100 motifs choisis. De même, comme la réponse finale du réseau est obtenue en comparant la sortie du neurone de la dernière couche à un seuil déterminé Th_{out} , nous avons effectué les simulations de fautes sur les entrées pour différentes valeurs de Th_{out} [Assoum96b]. Les résultats sont montrés sur la figure 5.4 en termes du nombre moyen des bits sensibles aux upsets pour les 2 formats testés

(8 et 32 bits).

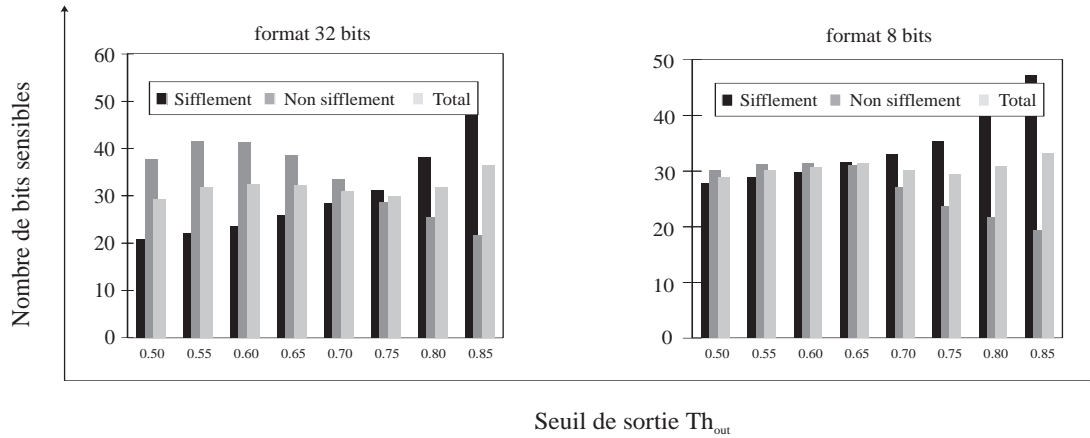


FIG. 5.4 : Nombre moyen de bits d'entrée sensibles aux upsets pour différentes valeurs du seuil de sortie

Pour mieux comprendre les histogrammes obtenus (FIG. 5.4), rappelons que, la sortie finale du réseau EBF (état du neurone de sortie) est comprise entre -1 et 1 à cause du choix de la fonction d'activation (tangente hyperbolique). Un motif d'entrée sera identifié comme sifflement si la sortie finale appartient à l'intervalle $[Th_{out}, 1]$. Les motifs correspondant à des sorties finales appartenant à l'intervalle $[-1, Th_{out}[$ seront identifiés comme des non sifflements. Il est clair que la performance de la classification dépend du choix du seuil de sortie Th_{out} . Les résultats obtenus montrent que la variation de Th_{out} a peu d'influence sur la tolérance aux upsets sur les stimuli d'entrée. En effet, les effets d'inversion des bits pour seulement un petit nombre de bits des pixels d'entrées conduisent à une classification erronée quand nous adoptons différentes valeurs du seuil. Ceci montre que le choix du seuil de sortie, qui peut avoir une influence considérable sur la performance de classification, peut être effectué indépendamment des considérations de la tolérance aux upsets. Cependant, dans le but de rendre maximale la performance du réseau en présence d'entrées corrompues, la valeur $Th_{out} = 0.75$ est la meilleure avec un nombre de bits sensibles inférieur à 30.

5.3.2 Résultats obtenus pour la perturbation des poids

Les effets des upsets sur les poids, pour chacun des motifs testés, ont été également étudiés en balayant exhaustivement toute la zone correspondant aux poids synaptiques du réseau EBF [Assoum97]. Les résultats obtenus, pour les versions à poids partagés et à poids redondants (non partagés), sont donnés sur les figures 5.5 et

5.6 respectivement. Dans ces figures, la performance du réseau est donnée en termes de nombre de fautes introduites qui se manifestent par une erreur au niveau de la sortie finale (bits sensibles aux upsets), tandis que dans les figures 5.7 et 5.8, elle est donnée en terme de pourcentage de sensibilité par bit.

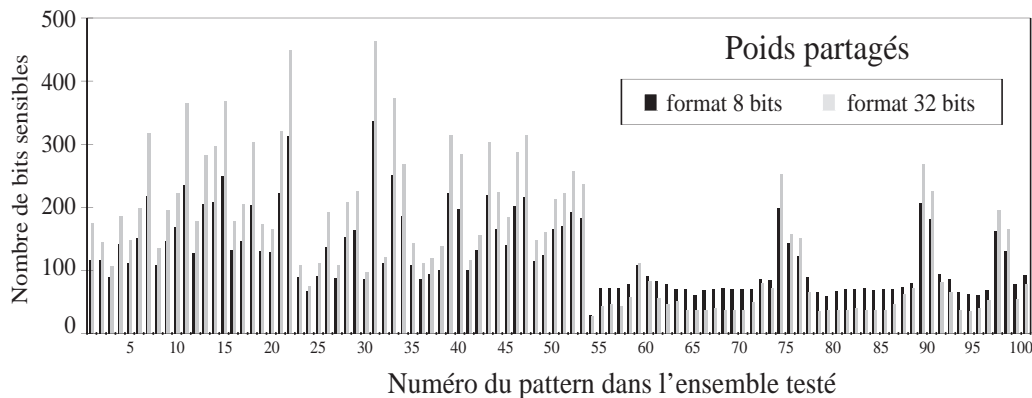


FIG. 5.5 : *Nombre de bits de poids sensibles par motif d'entrée (réseau à poids partagés)*

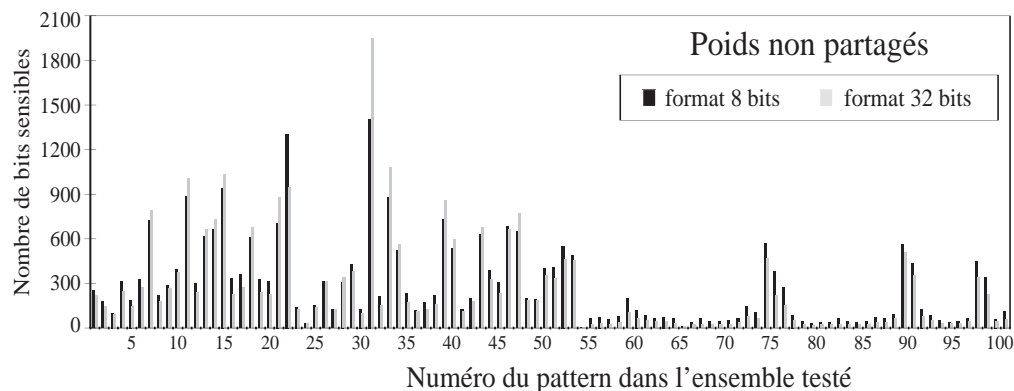


FIG. 5.6 : *Nombre de bits de poids sensibles par motif d'entrée (réseau à poids non partagés)*

Les résultats obtenus dépendent beaucoup du stimuli d'entrée traité. Par exemple, la figure 5.5 montre une variation de sensibilité de 4.7% et 1.1% (entrée #54) à 53.2% et 18.0% (entrée #31) pour les formats de codage sur 8 et 32 bits respectivement. La tolérance face aux upsets de l'implantation du réseau de détection de sifflements est clairement illustrée sur les tracés. En effet, pour l'ensemble considéré de stimuli d'entrée, les pourcentages moyens d'upsets non tolérés, pour les formats 8 et 32 bits, sont respectivement 18.4% et 6.0% pour les poids partagés et 6.0% et 2.2% pour les poids non partagés. Ceci correspond respectivement à 116 et 152 bits pour le premier cas et à 287 et 420 bits pour le second (tab. 5.1).

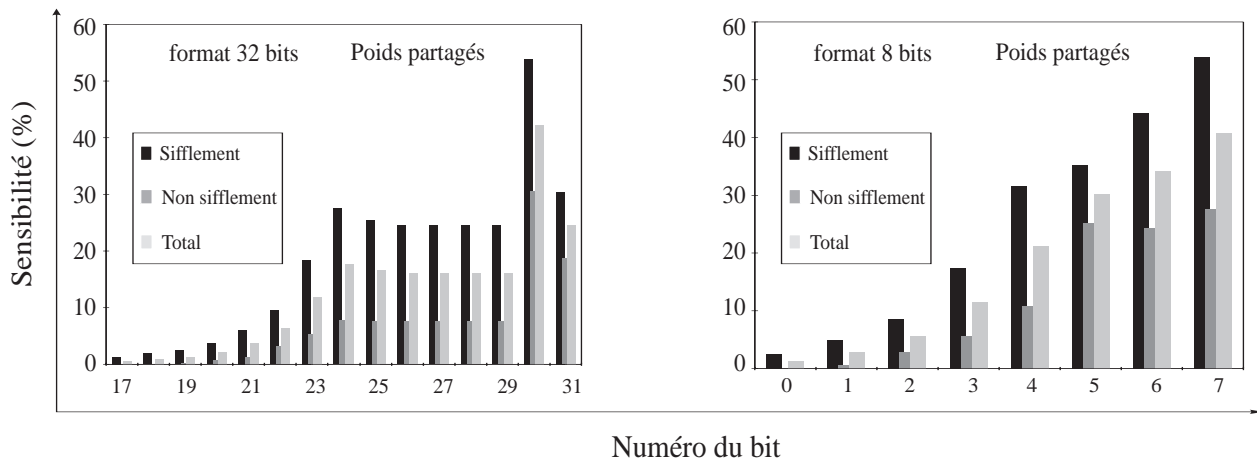


FIG. 5.7 : Sensibilité aux upsets en fonction du bit perturbé (réseau à poids partagés)

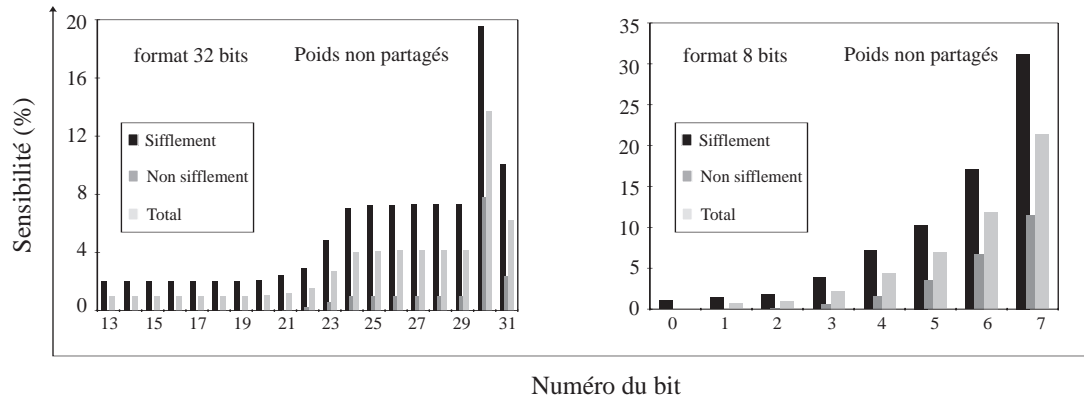


FIG. 5.8 : Sensibilité aux upsets en fonction du bit perturbé (réseau à poids non partagés)

	Poids partagés		Poids non partagés	
	Pourcentage de sensibilité	Nombre de bits sensibles	Pourcentage de sensibilité	Nombre de bits sensibles
Sifflements	24.8/8.7	157/220	9.2/3.9	439/745
Non sifflements	12.1/3.3	76/83	3.0/0.5	143/96
Total	18.4/6.0	116/152	6.0/2.2	287/420

TAB. 5.1 : Sensibilité moyenne aux upsets des réseaux testés pour les formats 8/32 bits

Pour les 2 versions étudiées du réseau EBF, le format 32 bits est plus tolérant que le format 8 bits. En effet, pour le format 32 bits, les perturbations sur plus de la moitié des bits n'ont pratiquement aucun effet sur la performance de classification du réseau simulé (FIG. 5.7 et 5.8). Cependant, vu la différence de taille mise en jeu par les implantations correspondant à chacun des 2 formats (4 fois plus grande pour le format 32 bits), la sensibilité aux upsets doit être prise en absolu (en termes de nombre de bits sensibles) plutôt qu'en relatif (en termes de pourcentage). Ceci conduit à une sensibilité plus grande pour le format 32 bits. Une autre remarque est que le réseau testé est plus sensible quand l'entrée qui lui est présentée correspond à un sifflement. Ceci est dû essentiellement à la nature de la trace de sifflement dans la bitmap d'entrée qui est facilement corrompue et par suite, n'est plus reconnue par le réseau.

Conclusion

Dans ce chapitre ont été présentés les résultats de simulations logicielles d'upsets pour deux réseaux différents. Il s'agit respectivement d'un problème simple de classification et de la détection des sifflements protoniques (réseau EBF). Malgré la différence existant entre les tailles et les architectures des réseaux testés, les résultats obtenus sont très proches et montrent une tolérance aux upsets supérieure à 90% pour les SEU sur les stimuli d'entrée et de plus de 80% pour ceux sur les poids.

Chapitre 6

Résultats de l'injection matérielle d'upsets

Résumé du contenu du chapitre

Ce chapitre présente les résultats obtenus par injection matérielle de fautes dans les différentes zones de mémorisation des circuits testés. L'avantage d'une telle technique pour le test des réseaux de neurones réside dans sa capacité d'accéder à la zone de stockage du programme et de localiser l'emplacement des cellules du circuit sensibles aux upsets.

6.1 Introduction

Comme pour le cas des simulation logicielles, la technique d'injection matérielle d'upsets permet d'estimer la tolérance aux fautes des circuits neuronaux étudiés. Elle est réalisée en absence de radiations et permet d'injecter des fautes de type upset dans les cellules mémoires sensibles accessibles en écriture sur les cartes prototypes. Ceci permet de localiser les positions mémoires qui, affectées par des upsets, conduiront à des erreurs de sortie. Ceci sera représenté dans la suite de ce chapitre, et pour les différents circuits de test, sous forme d'un mapping donnant les positions des bits sensibles pour l'application traitée.

6.2 Résultats obtenus pour les réseaux MLP_i

Les circuits qui ont été testés avec cette technique sont l'UTAK et le L-Neuro 1.0 [Assoum95c]. A titre d'exemple, sont donnés sur les figures 6.1 et 6.2 les mappings

des bits sensibles aux upsets pour le réseau à 2 couches cachées MLP_2 (pour l'UTAK et le L-Neuro 1.0 respectivement). Les cellules mémoires sont identifiées par leurs adresses : numéro du bit (ligne) et numéro de l'octet (colonne).

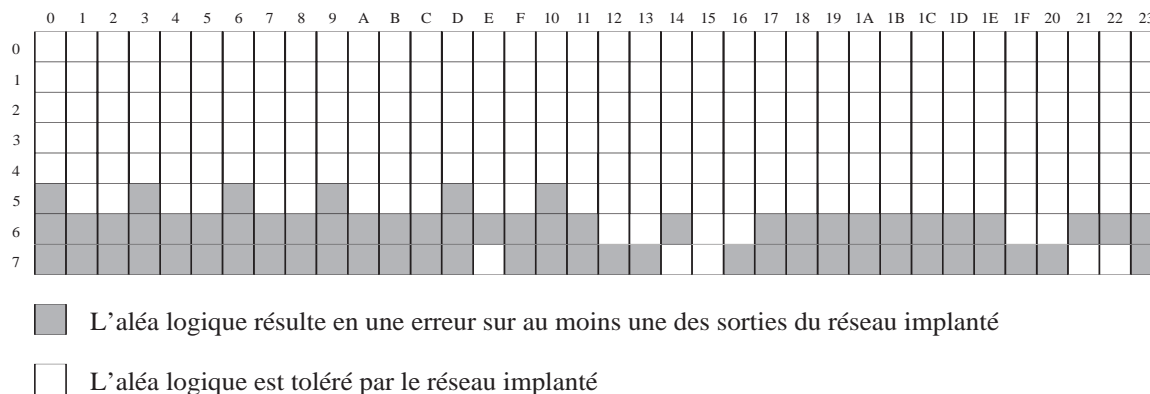


FIG. 6.1 : Mapping des cellules sensibles aux upsets de l'UTAK pour un réseau à deux couches cachées (seuls 67 parmi les 1 Kbits sont sensibles aux upsets)

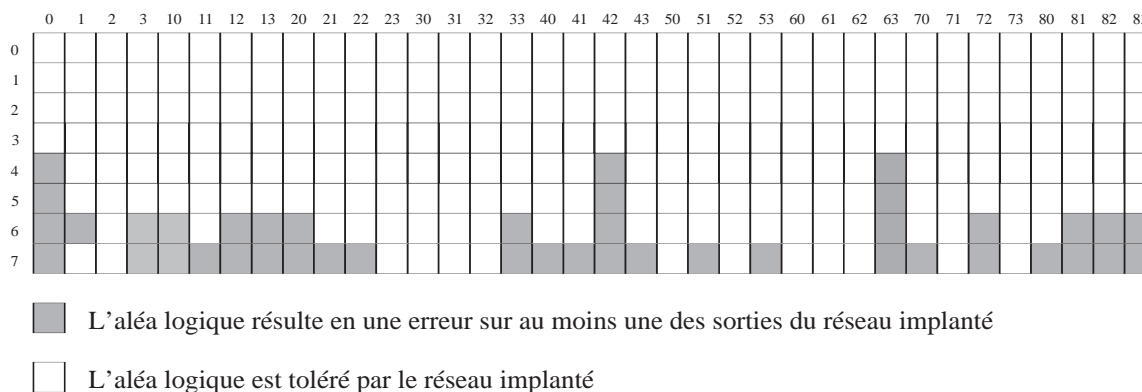


FIG. 6.2 : Mapping des cellules sensibles aux upsets du L-Neuro pour un réseau à deux couches cachées (34 bits sont sensibles parmi les 8 Kbits de la mémoire synaptique)

Le tableau 6.1 donne les résultats d'injections d'erreurs effectuées, en termes de pourcentage de la RAM interne où les upsets injectés résulteront en une erreur sur la réponse du réseau.

D'après ce qui précède, les circuits testés tolèrent une grande proportion des fautes de type upset dans leur mémoire interne utilisée pour stocker les poids synaptiques. Les différences entre les résultats obtenus pour les deux circuits est due principalement à la différence entre les tailles des mémoires des deux circuits (128 et 1024 octets pour l'UTAK et le L-Neuro respectivement). Une autre cause est le format du codage de données. En effet, pour l'UTAK, les poids synaptiques et les seuils des neurones sont

Réseau testé	UTAK	L-Neuro 1.0
MLP_1	6.1 %	0.4 %
MLP_2	6.6 %	0.5 %
MLP_3	13.2 %	1.0 %

TAB. 6.1 : Sensibilité aux upsets injectés dans les RAM des circuits testés

codés selon un format réel spécial sur 8 bits (signe et valeur absolue), tandis que le L-Neuro utilise un format réel 8 bits codé en binaire complément à 2.

6.3 Résultats obtenus pour le réseau OCR

L’OCR a été chargé avec des fichiers de configurations et des bitmaps “corrompus”, obtenus à partir des fichiers corrects par changement d’un ou plusieurs bits.

L’expérience a été effectuée sur deux zones de la mémoire interne : la première contenant les stimuli (zone de stockage de la bitmap du caractère à reconnaître), la deuxième contenant le programme neuronal [Assoum95b]. Cette dernière zone est réservée au stockage de la configuration du réseau de neurones, des séquences d’instructions des processeurs neuronaux et des valeurs échantillonnées de la fonction d’activation.

Perturbation de la zone bitmap

Les caractères à reconnaître sont présentés sous forme d’une matrice de 24×24 pixels codés sur 8 bits (correspondant à 255 niveaux de gris). L’expérience consiste à choisir aléatoirement un pixel, inverser le bit le plus significatif (bit 7) ce qui correspond à la perturbation pire cas, puis lancer l’OCR avec comme entrée la bitmap ainsi perturbée. Cette opération a été répétée, en cumulant les fautes, avec mémorisation des bits perturbés, jusqu’à ce que la bitmap introduite ne soit plus reconnue par l’OCR. La totalité de l’alphabet ainsi que les 10 chiffres décimaux ont été testés et ont montré un bon comportement face à ce type de fautes avec environ 150 fautes cumulées tolérées sur les MSB. Pour étudier l’influence de la suite aléatoire de perturbations sur le nombre de perturbations tolérées, nous avons répété plusieurs fois les simulations sur les mêmes caractères. Les histogrammes de la figure 6.3 montrent le nombre de perturbations tolérées pour 9 suites aléatoires différentes et pour certains des caractères traités.

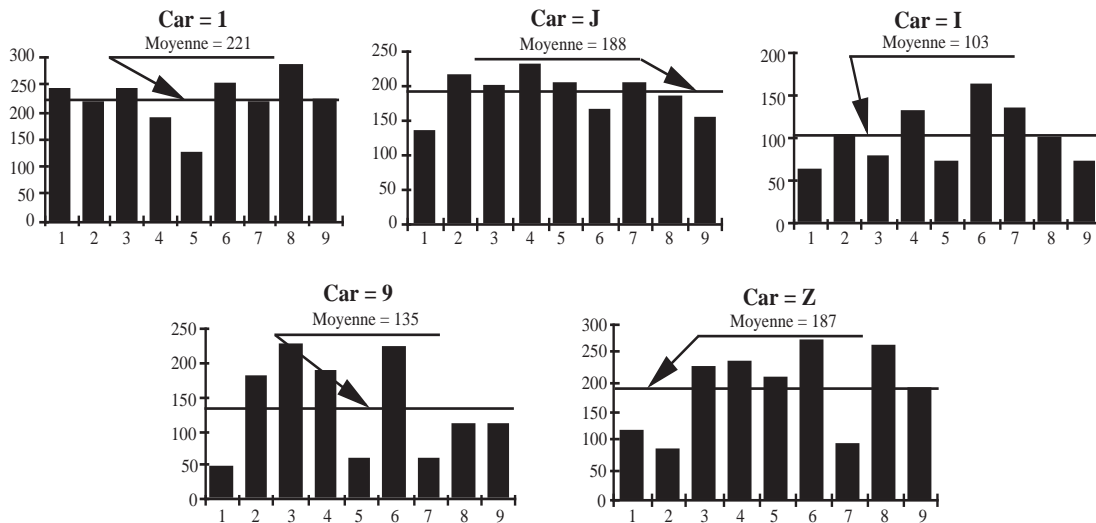


FIG. 6.3 : Injection de fautes dans la zone bitmap

L'OCR présente une très bonne tolérance face aux fautes dans la zone de stockage du caractère à reconnaître, même si le nombre d'erreurs tolérées est assez variable selon le caractère à reconnaître et la séquence aléatoire utilisée. Tous caractères confondus, la moyenne d'erreurs cumulées que le réseau peut tolérer est 167, le minimum étant 103 pour le caractère "I", le maximum étant 221 pour le caractère "1".

Cette robustesse est due d'une part, à la grande taille du réseau de neurones implanté, qui implique une redondance élevée de l'information. D'autre part, le fond dans les bitmaps des caractères à reconnaître occupe une surface importante ce qui fait que les fautes qui y surviennent ont une faible incidence sur la performance du réseau. En effet, seulement un cinquième environ des pixels sont significatifs pour une bitmap donnée, le reste étant réservé au fond (FIG. 6.4).

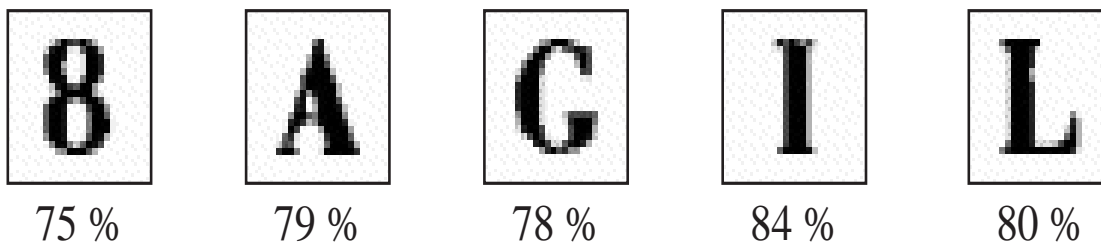


FIG. 6.4 : Pourcentage de la partie fond pour les bitmaps de quelques caractères

Perturbation de la zone programme

Dans cette simulation un seul bit de la zone programme est inversé à chaque itération avant de lancer le processus de reconnaissance sur l'OCR. Cette opération a

été répétée, pour chacune des bitmaps à reconnaître, pour tous les bits de la zone mémoire de chacun des processeurs P_i ($i \in \{1, 2, 3, 4\}$). Cette zone contient les poids synaptiques du réseau de neurones implanté, les échantillons de la fonction d'activation ainsi que les programmes des 4 processeurs. Notons que la zone mémoire associée au processeur P_5 est externe au circuit et n'est donc pas altérée dans le cadre de ces expériences (bien qu'elle puisse être sensible aux upsets). Plusieurs caractères ont été testés, les résultats obtenus sont donnés sur la figure 6.5.

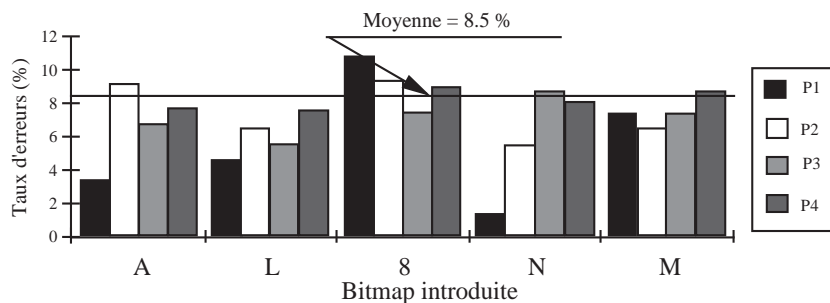


FIG. 6.5 : Injection des fautes dans la zone programme entière

La sensibilité aux fautes dans la zone programme est donnée en termes de taux d'erreur c'est à dire le pourcentage des bits de la mémoire dont l'inversion provoque des erreurs de reconnaissance. La figure 6.5 montre que, pour les 5 caractères montrés, le réseau implanté sur l'OCR tolère des erreurs sur plus de 90% des bits de la zone programme. Une analyse des résultats obtenus [Assoum95d], en fonction de la zone perturbée dans la zone programme, montre que les perturbations survenant sur la zone de stockage de la fonction d'activation n'ont aucun effet, tandis que celles qui affectent la zone de stockage des poids ont une faible incidence sur le taux de reconnaissance (inférieure à 1%). La contribution principale à la dégradation du taux de reconnaissance est due aux perturbations affectant la zone mémoire où sont stockées les instructions des processeurs. Ceci est illustré sur la figure 6.6.

Les chiffres précédents sont cependant à prendre avec précaution, vu le manque d'information sur le contenu exact des différentes zones perturbées qui peut ne pas être significatif pour l'application implantée.

Il est donc clair que les upsets, même multiples, sur la zone de la mémoire destinée au stockage de la bitmap du caractère à reconnaître, sont totalement tolérés. Les upsets sur la zone programme peuvent donner lieu à des erreurs, mais seulement pour une partie des bits de cette zone. Ces résultats permettraient d'envisager ce type de circuit comme candidat pour des équipements destinés à fonctionner dans

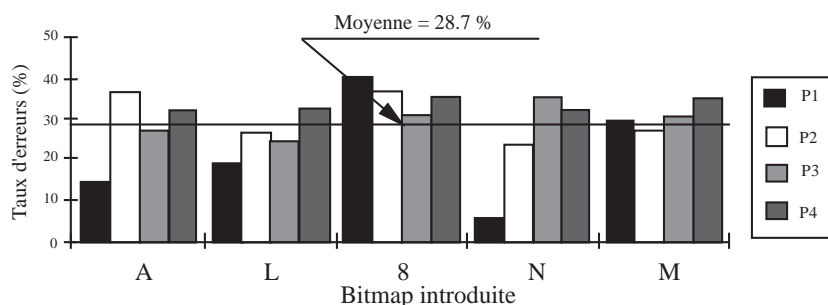


FIG. 6.6 : Injection des fautes dans la zone du stockage des instructions

l'espace, malgré l'espace mémoire interne important (plus de 2 koctets) que l'OCR nécessite pour implanter le réseau de neurones dédié à l'application (ceci sous-entend que l'application envisagée soit implantable sur le circuit).

6.4 Résultats obtenus pour le réseau EBF

Des programmes correspondants au réseau EBF (détection des sifflements) ont été développés respectivement pour le L-Neuro 1.0, le DSP96002 et le T225, les résultats obtenus par les essais d'injection de SEU sont détaillés dans la suite.

6.4.1 Cas du L-Neuro 1.0

Le réseau de détection de sifflements a été implanté sur la carte fille construite autour du circuit L-Neuro 1.0 de Philips et contrôlée par le testeur à travers la carte maître 68000. Tout l'espace de la mémoire RAM interne utilisé par le réseau TDNN implanté a été balayé (79 parmi 1024 octets correspondant à une implantation à poids partagés codés sur 8 bits) [Assoum96a]. La figure 6.7 montre le mapping d'upsets correspondant à deux motifs d'entrée (choisis aléatoirement dans la base de données disponible), l'un d'entre eux correspondant à un sifflement.

6.4.1.1 Discussion

Compte tenu du fait que le réseau considéré utilise environ 8% de la mémoire interne du L-Neuro 1.0 pour stocker les poids synaptiques, les résultats de la figure 6.7 montrent que moins de 24.0% des bits utilisés sont sensibles aux upsets dans le cas d'un motif d'entrée correspondant à un sifflement (161 bits). Ceci veut dire que, seul 2.0% des upsets dans la RAM interne contribueront à la section efficace de

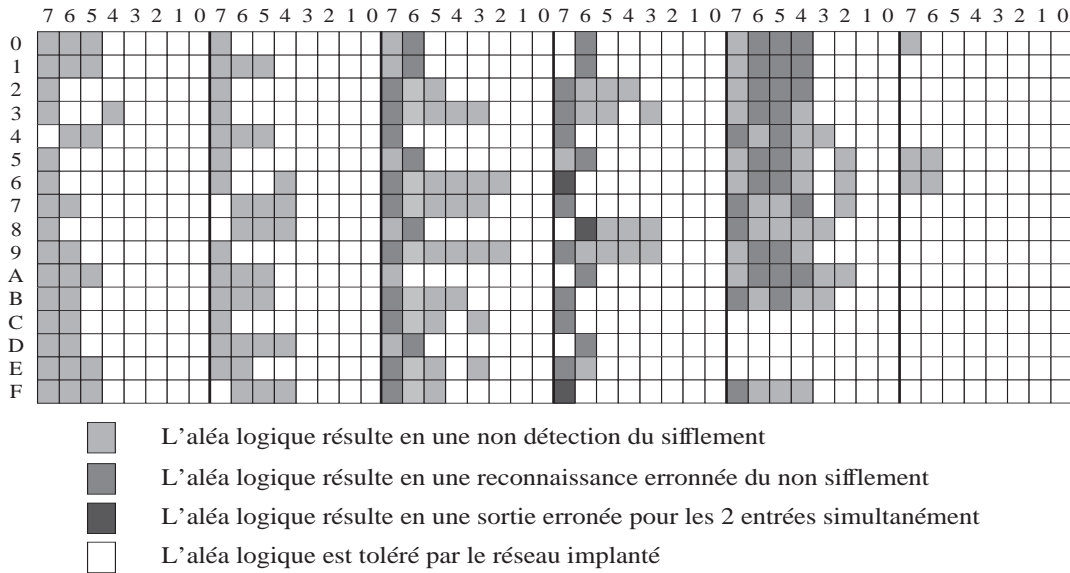


FIG. 6.7 : Mapping d'upsets pour deux motifs d'entrée différents du problème de détection de sifflements

l'application. Dans le cas où le stimuli d'entrée ne correspond pas à un sifflement les chiffres sont différents : seuls 60 bits sont sensibles aux upsets (9.0% de la zone RAM utilisée par l'application) ce qui correspond à 0.8% de la RAM entière.

A partir de cette expérience d'injection de fautes et de la section efficace d'upsets obtenue à l'aide d'un test statique, la section efficace de l'application peut être estimée en utilisant l'équation 6.1 par l'intermédiaire d'un coefficient K qui dépend du nombre de bits de la RAM interne sensibles aux upsets et de la taille de la RAM entière (eq. 6.2). Ceci a été rendu possible puisque le test statique a montré que tous les upsets détectés ont eu lieu sur la mémoire synaptique.

$$\sigma_{appli} = \frac{\sigma_{statique}}{K} \quad (6.1)$$

$$K = \frac{\text{Taille de la RAM entière}}{\text{Nombre de bits sensibles de l'application}} \quad (6.2)$$

Il est évident que la valeur de K dépend du stimuli d'entrée. Pour pouvoir estimer la section efficace d'une application donnée à partir d'un test statique, K devrait être évalué comme la moyenne obtenue lorsque l'expérience précédente est réalisée sur un ensemble représentatif de stimuli. C'est la raison pour laquelle le processus d'injection a été répété sur le même sous-ensemble de 100 patterns utilisé lors des simulations logicielles. Les résultats obtenus sont illustrés sur la figure 6.8 en termes de sensibilité moyenne pour chacun des bits des poids. Ils sont très proches des résultats de simulation logicielle (FIG. 5.7) pour le même format utilisé (8 bits). Une différence

inférieure à 1% peut être remarquée et est due à la différence de codage des entrées (4 bits pour l'implantation L-Neuro et 8 bits pour la simulation).

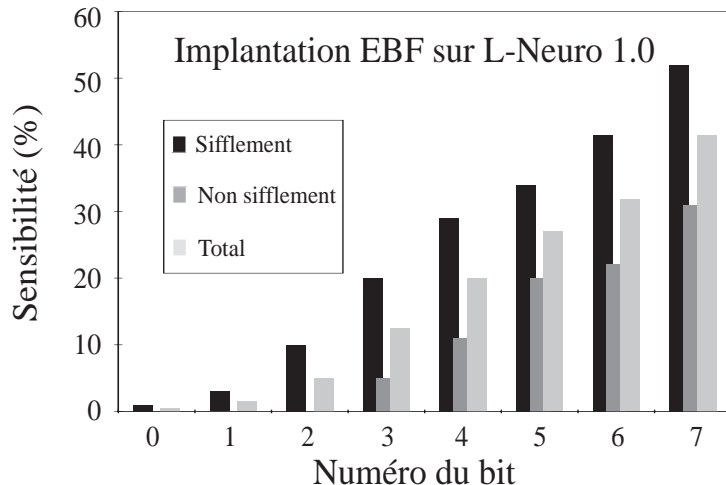


FIG. 6.8 : Sensibilité aux upsets des différents bits à l'intérieur d'un octet

Une analyse détaillée des résultats permet d'estimer la valeur du facteur K , à partir du nombre moyen de bits sensibles pour l'ensemble des entrées. Cette estimation donne $K = 73.1$ ($K=54.6$ et $K=109.2$ pour les sifflements et les non-sifflements respectivement) pour un seuil de sortie $Th_{out} = 0.75$. Ceci veut dire que la section efficace de l'application sera environ 73 fois plus petite que celle obtenue par un test statique.

Pour ce qui est des upsets sur les motifs d'entrée, un test d'injection aléatoire de SEU a été effectué en cumulant les fautes sur les patterns d'entrées. Les résultats obtenus montrent qu'une moyenne de 4 fautes cumulées sur les bits MSB est tolérée par le réseau implanté avant qu'il donne un résultat erroné.

6.4.2 Cas du DSP96002

Comme nous nous attendions, les résultats de l'injection d'upsets sur la zone de stockage de données (stimuli d'entrée, poids et seuils) étaient les mêmes que ceux des simulations logicielles pour le codage en format 32 bits (FIG. 5.5 et 5.7). Ceci est évident vu que ce format (le standard IEEE 754-1985) est le même pour le DSP et pour le processeur de la station sur laquelle les simulations ont été lancées (processeur SPARC). Une légère différence non montrée (moins de 0.5%) peut être distinguée et est due à l'évaluation approximative de la fonction d'activation des neurones dans le cas de l'émulation DSP (développement limité). Les résultats montrés sur les figures

5.5 et 5.7 sont donc également valables pour l’injection des fautes sur les données. En ce qui concerne la fonction d’activation, les coefficients de l’approximation polynomiale correspondante ont été perturbés bit à bit. Les résultats obtenus montrent que seuls 3.1% des bits sont sensibles (9 bits parmi 288).

6.4.2.1 Injection de fautes dans la zone du programme

D’après ce qui précède, les résultats obtenus par injection matérielle et par simulations logicielles des fautes sur la zone de stockages de données sont pratiquement les mêmes et indépendants de la machine d’exécution. Les résultats obtenus sont donc valables pour d’autres processeurs pourvu que le même format de codage soit utilisé. Cependant, ce n’est pas le cas pour la zone programme. En effet, dans ce cas, le code stocké dépend du processeur exécutant le programme, de la nature et de la sensibilité de ses instructions. Ceci veut dire que les résultats obtenus pour le DSP69002 ne seront pas forcément les mêmes pour un autre processeur (DSP ou autre) et qu’une étude indépendante devra être effectuée pour chaque processeur testé.

Dans le cas étudié (DSP96002), une expérience d’injection de fautes a été effectuée sur la zone de stockage du programme neuronal avant de lancer son exécution pour un stimuli d’entrée donné. Tous les bits du programme ont été perturbés, indépendamment de l’information stockée (opcodes, données immédiates, déplacements, etc.).

D’autre part, afin de tester l’influence de l’utilisation des sous-programmes sur la sensibilité de l’application émulée, 2 versions du programme neuronal ont été développées. Dans la première version (*Programme₁*), le calcul de la fonction d’activation se fait par un sous-programme, tandis que dans la deuxième version (*Programme₂*), il est fait en rajoutant le code correspondant, autant de fois qu’il faut, dans le programme. Le tableau 6.2 donne, pour les 2 versions testées, la taille du programme en nombre de bits et la sensibilité correspondante (pourcentage des bits dont la perturbation se manifeste par des erreurs au niveau de la sortie).

	Longueur du programme (bits)	Sensibilité aux upsets (%)
<i>Programme₁</i>	3328	58.0
<i>Programme₂</i>	4864	50.0

TAB. 6.2 : Sensibilité aux upsets de la zone programme

Le tableau 6.2 montre que près de la moitié des upsets survenant sur la zone programme seront tolérés. En plus, ces résultats montrent que, bien que l'utilisation des sous-programmes diminue la taille du programme, elle augmente sa sensibilité aux upsets. En effet, une réduction de 31.2% de la longueur du programme (en utilisant des sous-programmes) augmentera sa sensibilité de 8.0%. Cependant, il faudrait noter que ce pourcentage s'obtient pour une zone cible plus petite et pourrait donc être compensé par une plus faible occurrence d'upsets. De même, ce résultat est à prendre avec précaution puisque le réseau étudié est un cas particulier et que le comportement pourra être différent pour d'autres applications et d'autres circuits.

6.4.3 Cas du T225

6.4.3.1 Injection de fautes dans la zone des données

Les zones concernées par cette expérience sont celles où sont stockées les stimuli d'entrée, les poids synaptiques, les seuils et la table d'interpolation (fonction d'activation échantillonnée). Les tests ont été effectués sur un sous-ensemble choisi aléatoirement dans la base de patterns d'entrée disponible et différent de celui utilisé précédemment. Pour la perturbation des stimuli d'entrée les deux versions à poids partagés et non partagés ont été testées. Les résultats obtenus sont résumés sur le tableau 6.3 en termes de nombre de bits sensibles et pourcentage d'occupation de la zone de stockage correspondante.

	Poids & seuils (poids partagés)	Poids & seuils (poids non partagés)	Table d'interpolation
Sifflements	110 (17.4%)	383 (8.0%)	7 (0.1%)
Non Sifflements	48 (7.6%)	125 (2.6%)	6 (0.1%)

TAB. 6.3 : *Sensibilité aux upsets de la zone données*

6.4.3.2 Injection de fautes dans la zone du programme

Ce sont les instructions de calcul des états des différents neurones et séquences de contrôle associées qui sont perturbées dans ce cas. Le tableau 6.4 donne les sensibilités (en nombre de bits et pourcentage d'occupation correspondant) de la zone programme et de l'application entière. Cette sensibilité a été calculée à partir du nombre total de bits sensibles de l'application (zones données et programme).

	Zone programme	Application entière
Sifflements	1195 (45.7%)	1261 (9.9%)
Non Sifflements	1052 (40.2%)	1096 (8.5%)

TAB. 6.4 : *Sensibilité aux upsets de la zone programme et de l'application entière*

6.4.3.3 Discussion

Les résultats obtenus suite à l'expérience d'injection montrent une bonne tolérance face aux fautes de type SEU (TAB. 6.3 et 6.4). Pour l'injection des fautes dans la zone de données, ces résultats sont légèrement différents de ceux des simulations logicielles pour le même format de codage (8bits) (figures 5.5 à 5.8). Ceci est dû essentiellement à la manière dont la fonction d'activation est évaluée et qui n'est pas la même dans les 2 cas (fonction continue pour les simulations et table d'interpolation pour l'injection). D'autre part, la contribution principale à la sensibilité de l'application est due essentiellement à la zone de stockage du programme. En effet, les résultats donnés dans le tableau 6.4 montrent qu'environ 95% des bits sensibles appartiennent à la zone programme malgré qu'elle ne constitue que 20% de la zone totale de stockage de l'information nécessaire au fonctionnement de l'application. De même les résultats obtenus pour l'injection des fautes dans la zone programme du T225 sont comparables à ceux obtenus pour le DSP96002 (TAB. 6.2). Cependant, le T225 est moins sensible que le DSP96002. Ceci pourrait être dû à la nature du jeu d'instructions réduit du T225 qui fait qu'une partie non négligeable du code soit réservée aux opérandes des différentes instructions plutôt qu'aux opcodes. Néanmoins, la confirmation (ou l'infirmité) de cette supposition nécessite que des essais similaires soient réalisés pour d'autres processeurs CISC et RISC.

Conclusion

Ce chapitre a été consacré à la présentation des résultats d'injection matérielle de fautes dans les zones de mémorisation correspondant aux différents circuits testés. L'approche a été appliquée pour les différents réseaux étudiés. Les résultats de l'injection de fautes dans la zone de stockage de données (stimuli d'entrée et poids) étaient similaires à ceux obtenus lors des simulations logicielles. Ces dernières peuvent, par conséquent, suffire pour évaluer la sensibilité d'un réseau donné face aux upsets survenant sur les données. Cependant, la

technique d'injection reste indispensable pour évaluer la sensibilité de la zone programme. Elle a mis en évidence, selon le cas, une tolérance qui varie entre 30 et 50%. Néanmoins, les résultats obtenus par simulations et par injection ne sont pas suffisants à cause des zones sensibles inaccessibles à l'utilisateur, ce qui nécessite l'application d'un test aux ions lourds et fera l'objectif du chapitre suivant.

Chapitre 7

Résultats des tests aux ions lourds

Résumé du contenu du chapitre

Ce chapitre est consacré à la présentation des résultats obtenus par des tests aux ions lourds effectués sur les différents circuits testés. Le but est de mettre en évidence la tolérance aux fautes de type upset des implantations et des émulations matérielles des réseaux de neurones. Deux stratégies de test ont été adoptées : test statique et test dynamique. Une technique d'extrapolation de la section efficace d'une application à partir d'un test statique a été validée.

7.1 Introduction

A la différence des tests aux ions lourds généralement effectués sur des processeurs classiques (architectures de von Neumann), dont le but est la détermination de la section efficace complète d'upset, notre objectif est la mise en évidence de la tolérance aux upsets des implantations et des émulations matérielles des réseaux de neurones. Ceci sera réalisé en comparant leur section efficace avec celle d'une application classique. Les tests effectués comblent la lacune des simulations logicielles et de l'injection matérielle des fautes décrites dans les chapitres précédents car ils permettent d'étudier les effets des upsets aussi sur les zones de mémorisation inaccessibles du circuit sous test.

D'autre part, si la propriété de tolérance intrinsèque aux fautes des réseaux de neurones est valable pour le cas des upsets, alors nous pourrions estimer par des tests aux ions lourds que, pour les circuits implantant (émulant) les réseaux testés, ceci peut rendre encore plus éloignés les résultats obtenus avec les deux types de

stratégie statique et dynamique (diminution de la valeur de la section efficace). En effet, la formule de Petersen (eq. 7.1) montre que le taux d'erreurs N (en orbite géosynchrone) est proportionnel à la section efficace de saturation σ_s du circuit sous test et inversement proportionnel à son LET seuil L_s .

$$N = 5 \times 10^{-10} \frac{\sigma_s}{L_s} \quad (7.1)$$

En plus, les tests aux ions lourds permettent d'extrapoler la sensibilité aux upsets de l'application finale à partir d'une séquence donnée d'upsets détectés lors d'un test statique et du mapping des bits sensibles résultant d'une injection de fautes. Ceci conduit à une diminution du coût des tests d'un circuit donné tout en permettant d'utiliser la même séquence d'upsets (test statique) pour l'extrapolation de la section efficace de plusieurs applications (test dynamique).

7.2 Résultats obtenus pour l'UTAK

Des tests au Californium ont été effectués pour l'UTAK [Assoum95c]. Les sections efficaces des SEU et SEL obtenues à partir du test statique de la zone sensible sont respectivement de 4×10^{-4} et $2.8 \times 10^{-4} \text{ cm}^2/\text{composant}$. L'analyse du mapping des SEU survenant durant ce test montre que tous les upsets se sont produits dans la RAM interne.

Les tests dynamiques ont été effectués avec les 3 versions de perceptrons multicouches MLP_i . Dans le tableau 7.1 sont données les sections efficaces obtenues aussi bien à partir du test dynamique qu'en faisant une extrapolation à partir du mapping des cellules mémoires sensibles aux upsets (FIG. 6.1) et de la trace des upsets obtenus durant le test statique (89 upsets). Les déviations entre les valeurs mesurées et celles estimées par extrapolation (TAB. 7.1) sont essentiellement dues au fait que les séquences d'upsets utilisées dans les 2 cas ainsi que les durées des programmes de test (statique et dynamique) ne sont pas les mêmes.

Ces résultats n'ont pas mis en évidence des différences significatives entre les sections efficaces dynamiques des ANN émulsés, la plus grande étant celle de l'ANN avec plus de couches cachées. Ce qui veut dire que la redondance, qui est en général un facteur "positif" pour les systèmes classiques de traitement de données [Lala85], ne l'est pas forcément pour un système de type réseau de neurones.

D'autre part, vu sa sensibilité élevée aux latchups (fréquence des SEL 10 fois plus grande que celle des SEU), des tests en accélérateur n'ont pas été jugés nécessaires.

σ_{SEU} UTAK	MLP_1	MLP_2	MLP_3
Mesurée	2.25×10^{-5}	3.96×10^{-5}	7.6×10^{-5}
Extrapolée	1.78×10^{-5}	8.9×10^{-6}	2.23×10^{-5}

TAB. 7.1 : *Sections efficaces dynamiques mesurées et extrapolées ($cm^2/composant$)*

7.3 Résultats obtenus pour le L-Neuro 1.0

Les tests préliminaires, réalisés avec le Cf252, ont montré que le L-Neuro possède une section efficace statique d'upsets de $1,1 \times 10^{-3} cm^2/composant$. Comme pour l'UTAK, l'analyse du mapping d'upsets montre qu'ils sont tous survenus sur des bits de la RAM interne.

Des tests plus complets ont été réalisés en utilisant l'accélérateur Tandem Van de Graaff de l'IPN. Les sections efficaces de SEU et de SEL obtenues à partir des test statiques, pour les particules utilisées, sont montrées dans le tableau 7.2. Ces résultats montrent un bon comportement du L-Neuro face aux radiations : la section efficace statique de saturation est environ $3.5 \times 10^{-3} cm^2/composant$ pour un LET de l'ordre de $36 Mev/mg/cm^2$.

Ion	LET $Mev/mg/cm^2$	σ_{SEU}	σ_{SEL}
19 F (55°)	7.3	1.4×10^{-4}	1.6×10^{-5}
35 Cl	12.7	3.3×10^{-4}	2.2×10^{-5}
79 Br	36.0	3.5×10^{-3}	1.6×10^{-4}

TAB. 7.2 : *Sections efficaces statiques SEU et SEL du L-Neuro ($cm^2/composant$)*

Des résultats des tests dynamiques ont été obtenus aussi bien pour les réseaux MLP_i que pour le réseau EBF et sont décrits dans ce qui suit.

7.3.1 Cas des réseaux MLP_i

Les particules utilisées pour les tests sont de type ions de Chlore – 35Cl (LET 12.7 $Mev/mg/cm^2$). Les résultats obtenus [Assoum95c] sont similaires à ceux obtenus pour l'UTAK : l'augmentation du nombre de couches cachées n'est pas forcément un facteur positif pour les systèmes de type réseau de neurones (TAB. 7.3).

Aucun des upsets survenant durant le test statique ne correspond à une cellule mémoire sensible pour les tests dynamiques, c'est pour cela que les valeurs des sections

σ_{SEU} L-Neuro	MLP_1	MLP_2	MLP_3
Mesurée	2.6×10^{-6}	2×10^{-6}	5×10^{-6}
Extrapolée	$< 4.1 \times 10^{-6}$	$< 4.1 \times 10^{-6}$	$< 4.1 \times 10^{-6}$

TAB. 7.3 : Sections efficaces dynamiques mesurées et extrapolées ($cm^2/composant$) obtenues pour le circuit L-Neuro avec un faisceau de Chlore

efficaces extrapolées données dans le tableau 7.3 sont des valeurs majorées.

Les sections efficaces statiques et dynamiques obtenues pour les différentes particules utilisées sont résumées dans la figure 7.1. Les sections efficaces dynamiques sont moyennées sur les résultats obtenus durant l'extrapolation pour les trois MLP_i . La diminution observée pour le LET de $12,7 Mev/mg/cm^2$ (Chlore) peut être expliquée par la différence entre les séquences de SEU utilisées pour faire l'extrapolation.

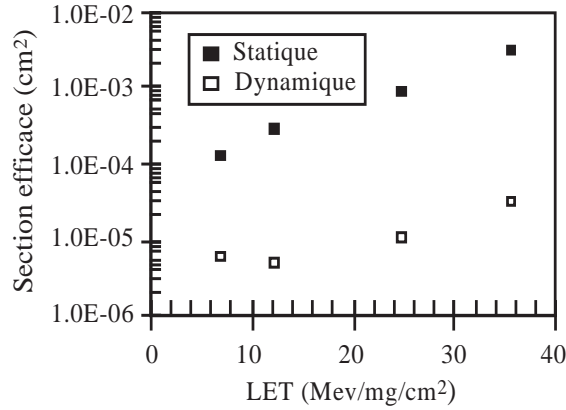


FIG. 7.1 : Sections efficaces statiques et dynamiques extrapolées du L-Neuro

7.3.2 Cas du réseau EBF

Dans ce test, les particules utilisées sont de type ions de Fluor – 19F (LET $4.2 Mev/mg/cm^2$) tiltés¹ à 55° (aucun événement n'a pu être détecté pour une incidence normale). Différentes bitmaps d'entrée ont été utilisées pour stimuler le réseau EBF durant l'exposition du L-Neuro 1.0 au faisceau de Fluor tilté [Assoum96a]. Les sections efficaces moyennes obtenues lorsque le circuit L-Neuro 1.0 évaluait des bitmaps d'entrée arbitrairement sélectionnées, sont données dans les équations 7.2 et 7.3, pour

1. Un faisceau est dit *tilté* d'un angle θ quand l'angle d'incidence par rapport à la normale vaut θ .

chacun des types d'entrée traités (sifflements et non-sifflements).

$$\sigma_{SEU}(F \text{ à } 55^\circ) = 1.2 \times 10^{-5} \text{ cm}^2 / \text{composant} \text{ (Sifflements)} \quad (7.2)$$

$$\sigma_{SEU}(F \text{ à } 55^\circ) < 1.0 \times 10^{-5} \text{ cm}^2 / \text{composant} \text{ (Non - sifflements)} \quad (7.3)$$

La section efficace extrapolée à partir de la section efficace statique et le coefficient K moyenné sur tous les motifs d'entrées ($K_{moyenne} = 71.3$) est donnée dans l'équation 7.4.

$$\sigma_{appl}(F \text{ à } 55^\circ) = \frac{\sigma_{stat}(F \text{ à } 55^\circ)}{K_{moyenne}} = 2.0 \times 10^{-6} \text{ cm}^2 / \text{composant} \quad (7.4)$$

Cette valeur de la section efficace estimée est presque 8 fois inférieure à la section efficace mesurée (eq. 7.2) durant les tests au Fluor. La différence est due à l'utilisation du facteur $K_{moyenne}$. En effet, si on utilise le facteur K correspondant aux entrées traitées ($K = 21$ et 127 pour le sifflement et le non-sifflement respectivement) la valeur de la section efficace extrapolée devient 6.7×10^{-6} et 1.1×10^{-6} respectivement ce qui est en corrélation avec les valeurs mesurées.

Dans le tableau 7.4 sont montrés les emplacements des erreurs (numéro d'octet) et les numéros des bits correspondants aux SEU obtenus lors du test statique (354 SEU), sur le mapping des SEU correspondant à la zone mémoire utilisée par le réseau de neurones traité lors des tests accélérateur. La section efficace dynamique qui ressort de cette démarche peut être calculée à partir du nombre des SEU affectant un bit sensible et de la fluence de particules utilisée lors du test statique 3. On obtient : $\sigma_{dyn} = 2.4 \times 10^{-5} \text{ cm}^2 / \text{composant}$ pour les sifflements et $\sigma_{dyn} = 5.3 \times 10^{-6} \text{ cm}^2 / \text{composant}$ pour les non-sifflements. De cette extrapolation ressort que pour les réseaux de neurones artificiels, la méthode exposée permet d'obtenir des estimations correctes des sections efficaces des différentes applications sans réaliser d'essais accélérateur.

7.4 Résultats obtenus pour l'OCR

Les essais réalisés au Californium - 252Cf ont mis en évidence une fréquence élevée de latchups, empêchant le recueil de résultats sur les upsets. Dans un but de déterminer les seuils d'upset et de latchup, des essais ont été réalisés à l'accélérateur avec différents faisceaux de particules ayant des LET inférieurs au LET effectif du 252Cf. Des faisceaux de Chlore (12.7 Mev/mg/cm^2), Fluor (4.2 Mev/mg/cm^2) et Carbone (1.7 Mev/mg/cm^2) ont été utilisés. Des latchups ont été obtenus avec les faisceaux du Chlore 35Cl et du Fluor 19F, révélant, pour le circuit OCR, un seuil de latchup

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	3			1												2
1	7	0		6, 3	4		3					0				
2	5	6, 0				0			5, 3			0			7	
3			2	6	2	3				4		0	0			
4					4, 0									NU	NU	NU
5		1		4, 3				NU	NU	NU	NU	NU	NU	NU	NU	NU

TAB. 7.4 : Mapping des SEU dans la zone utilisée dérivé de la séquence du test statique
(NU: Non Utilisé dans l'implantation de l'application)

inférieur à 4.2 Mev/mg/cm^2 . Les essais réalisés avec un faisceau de Carbone 12C perpendiculaire à la puce et tilté à 48° n'ont pas permis la détection d'événements (ni upset ni latchup).

De ces résultats nous pouvons conclure que les seuils de latchup et d'upset de l'OCR sont compris entre 2,54 et $4,2 \text{ Mev/mg/cm}^2$. De plus, ces seuils étant suffisamment proches l'un de l'autre, ils empêchent l'obtention de données sur la section efficace du réseau de neurones implanté.

7.5 Résultats obtenus pour le DSP96002

Afin de compléter les expériences d'injection d'upsets effectuées sur le DSP et décrites dans le chapitre précédent, des tests aux ions lourds ont été réalisés avec un accélérateur de particules au moment où le processeur exécute le programme correspond au réseau EBF. Différents types de particules ont été prévus pour le test. Cependant la sensibilité élevée aux latchups manifestée par le 96002 nous a empêché d'avoir des résultats complets dessus. En effet, avec la version du circuit actuellement disponible, nous étions incapables de tirer des résultats significatifs sur les upsets survenant durant le test surtout que leur fréquence d'occurrence était très inférieure à celle des latchups. C'est la raison pour laquelle le DSP96002 a été exclu, à l'instar de l'OCR, des utilisations dans des applications spatiales à moins que des versions, immunes aux latchups, soient disponibles.

7.6 Résultats obtenus pour le T225

Contrairement au DSP96002, le transputeur T225 a montré une immunité aux latches [Assoum95a]. Ceci a permis d'effectuer des tests aux ions lourds à l'aide d'un accélérateur de particules. Différents types d'ions ont été utilisés pour irradier le T225 pendant qu'il émulait le réseau EBF (Chlore, Fluor et Brome) [Assoum96b]. Toute l'information en relation avec le programme d'émulation (données et code) a été stockée en mémoire externe après avoir désactivé la RAM interne du T225. Dans le tableau 7.5, sont montrées les sections efficaces obtenues pour chacune des particules testées. Afin de pouvoir juger la qualité des résultats obtenus, les sections efficaces d'un programme de calcul classique (produit matriciel) occupant une taille mémoire équivalente [Assoum95a] exécuté par le T225 sont également montrées, pour les mêmes particules.

Ion utilisé	σ_{EBF} avant/après filtrage $cm^2/composant$	Pourcentage de filtrage	$\sigma_{Matrice}$ $cm^2/composant$
Brome	$3.9 \times 10^{-4}/1.3 \times 10^{-4}$	66.7	1.3×10^{-3}
Chlore	$4.9 \times 10^{-5}/2.7 \times 10^{-5}$	44.6	2.1×10^{-4}
Fluor	$2.7 \times 10^{-6}/1.5 \times 10^{-6}$	42.7	3.0×10^{-5}

TAB. 7.5 : *Section efficace dynamique mesurée (EBF vs. produit matriciel)*

Le tableau 7.5 illustre bien la tolérance aux fautes de l'application EBF émulée par le T225. En effet les valeurs des sections efficaces obtenues sont inférieures à celles obtenues pour des programmes de calcul non neuronaux (environ un ordre de grandeur). Ceci est dû à la nature non-linéaire et bornée de la fonction d'activation utilisée (tangente hyperbolique) qui fait que, même en présence d'upsets les sorties des neurones ne peuvent pas dépasser l'intervalle $[-1, +1]$. D'autre part, durant le test aux ions lourds, les upsets sont détectés en comparant les états des différents neurones du réseau à des valeurs de référence ; par conséquent, toutes les différences dues à des SEU contribueront à la section efficace "brute" de l'application (référéncée "avant filtrage" dans le tableau). Parmi ces erreurs, seront éliminées celles pour lesquelles la réponse finale du réseau reste dans la marge de départ ($[Th_{out}, 1]$ pour les sifflements et $[-1, Th_{out}[$ pour les non-sifflements) et la section efficace correspondante sera référéncée "après filtrage". Le tableau 7.5 met en évidence un facteur d'ordre 2 entre les sections efficaces avant et après filtrage et ce pour différents ions testés. En

d'autres termes, environ 50% des upsets détectés sont tolérés et n'ont donc pas d'effet sur la réponse finale du réseau.

7.7 Discussion

Les résultats des tests aux ions lourds qui ont été décrits dans ce chapitre ont montré, pour les différents circuits et réseaux testés, une tolérance moyenne aux upsets de 50% ce qui est cohérent avec les résultats obtenus lors de l'injection matérielle de fautes dans la zone programme.

Cependant, la corrélation entre les tests aux ions lourds et les résultats de simulation et d'injection n'est pas toujours évidente et dépend du circuit testé et de la nature du stockage des données et/ou du programme neuronal. En effet, une partie plus ou moins importante de la zone mémoire où sont stockés les paramètres du réseau n'est pas directement exposée au faisceau.

D'autre part, afin de pouvoir comprendre les résultats donnés dans le tableau 6.4, rappelons que, durant le test aux ions lourds, la RAM interne du T225 était désactivée. Par conséquent, tous les upsets détectés sont ceux qui ont eu lieu durant le traitement dynamique au sein du T225 (recherche d'instructions, traitement de données, etc.). En effet, les erreurs survenant sur les ressources internes du T225 autres que la RAM (registres, bascules, etc.) ne peuvent pas être simulées et ne sont détectées que durant un test réel aux ions lourds.

Une comparaison a été effectuée entre les résultats obtenus pour des approches classiques de traitement de l'information et des approches neuronales qu'il s'agisse d'une implantation ou d'une émulation. Ceci sous-entend le cas d'un test dynamique ayant les mêmes conditions de test (même type de particules et même angle d'incidence). Cela peut être résumé dans ce qui suit sous forme du gain de tolérance aux fautes apporté par l'approche neuronale et les tableaux (ou équations) à partir desquels ce gain a été déduit.

1. Comparaison classique/neuronal ($Matrice/EBF_{L-Neuro}$): cas des implantations.
Le gain apporté par l'approche neuronale est d'environ 65 (TAB. 7.5 et 7.3).
2. Comparaison classique/neuronal ($Matrice/EBF_{T225}$): cas des émulations.
Le gain apporté par l'approche neuronale varie entre 7 et 20 selon la particule choisie (TAB. 7.5).

De même, le gain apporté par une implantation (circuit dédié) sur une émulation (circuit à usage général) est d'environ 10.

Les chiffres donnés ci-dessus montrent l'avantage de l'utilisation des réseaux de neurones matériels par rapport à celle des approches classiques. Ceci se manifeste, pour une application s'exécutant en milieu radiatif, par une diminution du taux d'erreurs d'un facteur égal au gain apporté (7 dans le pire des cas). Rappelons que cette performance est obtenue sans aucun durcissement, ni au niveau du processus de fabrication, ni au niveau de la conception.

Conclusion

Dans ce chapitre ont été décrits les tests aux ions lourds réalisés sur les différents circuits utilisés au cours de la thèse (dédiés et à usage général). Les résultats obtenus ont montré une bonne tolérance aux upsets pour différents réseaux testés ce qui est cohérent avec les résultats des simulations logicielles et une injection matérielle de fautes. En plus, l'approche réseaux de neurones a fait preuve d'un gain déduit au niveau de la tolérance aux fautes de l'application testée en comparaison avec une approche classique de traitement de l'information. Ceci se manifeste, pour une application s'exécutant en milieu radiatif, par une diminution du taux d'erreurs d'un facteur égal au gain apporté. Les réseaux de neurones semblent donc bien adaptés aux applications spatiales.

Conclusions et perspectives

Dans cette thèse nous avons étudié les conséquences d'un des phénomènes induits par l'environnement spatial sur les applications de type réseaux de neurones artificiels implantés sous forme matérielle, le phénomène dit "d'upset". Ce dernier est provoqué par l'impact d'une particule lourde dans une zone sensible du circuit sous test et se traduit par le basculement intempestif d'un point mémoire.

Les conséquences d'un tel phénomène sur une application neuronale implantée (ou émulée) par des circuits et embarquée à bord d'un satellite peuvent selon les cas, être graves surtout si les perturbations résultantes touchent des zones contenant une information critique pour l'application embarquée. Ceci montre la nécessité d'évaluer la sensibilité des circuits utilisés face à des fautes de type upset.

Plusieurs circuits ont été choisis pour réaliser les tests et leur sensibilité aux upsets a été évaluée. Il s'agit de trois circuits digitaux dédiés aux réseaux de neurones et de deux processeurs à usage général (un DSP et un transputer). Une carte de test a été développée autour de chacun des circuits cible.

Trois différents types de tests ont été réalisés :

- simulations logicielles de fautes sur ordinateur ;
- injection matérielle de fautes sur les cartes de test ;
- tests aux ions lourds : évaluation du fonctionnement d'un circuit pendant qu'il est exposé à un faisceau de particules.

Pour les tests aux ions lourds, deux types de séquence de test ont été développés pour chacun des circuits testés. Il s'agit du test statique et du test dynamique. Le premier permet d'évaluer la sensibilité des différents éléments de mémorisation du

circuit accessibles à l'utilisateur, tandis que le deuxième sert à déterminer la sensibilité de l'application exécutée.

Le contrôle des circuits cible (application de la séquence de test, récupération des résultats) a été effectué avec le testeur FUTE16 par l'intermédiaire d'une carte d'interface construite autour d'un microprocesseur maître et développée dans le cadre de cette thèse.

Plusieurs réseaux ont été testés, certains parmi eux correspondent à des "cas d'école" (évaluation de fonctions booléennes), d'autres à des problèmes plus complexes (reconnaissance de caractères et détection des sifflements à bord d'engins spatiaux).

Des techniques de simulations logicielles et d'injection matérielle de fautes de type upset ont été mises en œuvre, pour différentes applications afin de déterminer le nombre de bits sensibles et par suite le taux total de sensibilité pour l'application testée.

Les résultats des simulations montrent une bonne tolérance des réseaux de neurones face aux fautes de type upset sur les zones de stockage de données avec une sensibilité moyenne de 10% et 5% pour les poids et les entrées respectivement. Ces chiffres sont du même ordre de grandeur que ceux obtenus par injection matérielle de fautes sur les zones de données.

Cependant, dans le cas où le réseau de neurones testé est émulé par un circuit programmable, sa sensibilité peut être due, non seulement à celles de ses données (stimuli et poids), mais aussi à celle du programme effectuant les calculs correspondant à son émulation. C'est pourquoi, l'expérience d'injection des fautes a été aussi effectuée sur la zone de stockage du code de programme neuronal. Les résultats obtenus ont montré une sensibilité moyenne de 45%. Par conséquent, la sensibilité totale de l'application dépend principalement de celle du programme. Une alternative permettant d'augmenter la tolérance aux fautes serait donc de stocker le programme dans une ROM.

D'autre part, les tests aux ions lourds réalisés à l'aide d'une source radioactive de Californium et d'un accélérateur de particules ont conduit à l'exclusion de deux

des circuits testés à cause de leur grande sensibilité face aux latchups. Cependant, pour les autres circuits, les tests accélérateurs ont été possibles. Les résultats obtenus montrent que les sections efficaces des applications neuronales sont inférieures d'un ordre de grandeur à celles obtenues pour des programmes de test classique, aussi bien pour des implantations sur des circuits dédiés que pour des émulations par des processeurs standards.

Enfin, une nouvelle technique pour évaluer la section efficace a été utilisée avec succès et consiste en l'extrapolation de la section efficace dynamique d'une application implantée sur un circuit donné à partir des traces d'upsets (mapping) obtenues lors d'un test statique. Il s'agit, en d'autres termes, de montrer l'influence des upsets détectés durant le test statique d'un circuit donné sur l'application (test dynamique) exécutée par le même circuit.

En conclusion, l'étude qui a été faite au cours de cette thèse montre l'avantage des réseaux de neurones matériels, qu'ils soient implantés sur des circuits dédiés ou émulés par des processeurs à usage général, par rapport aux systèmes classiques de traitement de l'information. Cet avantage est donné en termes de pourcentage de sensibilité face à des erreurs de type upsets avec la possibilité d'utilisation de composants non durcis.

Le travail effectué a été conclu par deux cartes neuronales acceptées pour faire partie du projet MPTB (Microelectronics and Photonic Test Bed) des laboratoires NRL (Naval Research Laboratories) et consiste à embarquer des cartes électroniques dans un satellite scientifique dont le lancement est prévu fin 1997. L'expérience bâtie autour de ces cartes aura pour objectif l'obtention de données en vol concernant la sensibilité face aux SEU, de réseaux de neurones artificiels implantés aussi bien sur des circuits dédiés que sur des circuits à usage général. Les résultats obtenus permettront de juger de la possibilité d'utilisation de cette technique novatrice de traitement de l'information dans des projets spatiaux futurs.

Quant aux perspectives de travail futur, nous pouvons citer l'application des réseaux de neurones pour la reconnaissance de textures en provenance des mesures effectuées à bord d'un satellite (images SPOT) et l'étude de la même application par plusieurs approches (classiques et neuronales). Une autre perspective consiste à étu-

dier l'effet, sur les réseaux de neurones, des fautes du type upsets introduites durant l'apprentissage ainsi qu'à explorer la possibilité de réaliser l'apprentissage des réseaux dans un milieu radiatif.

Annexe A

Circuits implantant des réseaux de neurones

Cet annexe décrit brièvement les principaux circuits existants pour les différents types d'implantations des réseaux de neurones artificiels. Une bonne partie de la bibliographie a été recueillie par Internet directement de la page Web des différentes compagnies de développement logiciel et matériel. C'est pour cela qu'on a jugé utile de mettre les adresses Web correspondantes.

A.1 Implantations digitales

A.1.1 Architectures en tranches

Le circuit NBS MD-1220 (Neural Bit Slice) de Micro Devices¹. C'est un produit VLSI CMOS sur lequel peuvent être implantés 8 neurones sous forme d'unités de traitement (Processing Elements – PE) travaillant en parallèle. Les entrées sont binaires tandis que les poids sont codés sur 16 bits. Les produits intermédiaires *poids* \times *entrée* sont stockés dans des accumulateurs, et une fonction seuil est appliquée à la fin du calcul pour obtenir la sortie du neurone courant. La précision de l'accumulateur est de 16 bits et la fréquence de fonctionnement est de 20 MHz. La performance peut atteindre 8.9 MCPS pour un réseau de 64 neurones.

Le circuit NLX420 de NeuraLogix²: Il s'agit d'un processeur neuronal digital en

1. <http://www.amd.com/>

2. Le nouveau nom est Adaptive Logic, <http://www.adaptivelogic.com/>

tranches ayant 16 PE comprenant chacune un accumulateur de 32 bits. L'architecture en tranches permet la construction facile des configurations multicircuits facilitant la construction des réseaux de grande taille. Utilisant le multiplexage temporel, un seul circuit NLX420, configuré pour des entrées 16 bits, peut implanter un maximum de 1048576 neurones, ayant chacun 64 Kentrées. Les poids sont stockés dans une mémoire externe. Les activations des neurones sont déterminées par des approximations fonctionnelles continues chargées par l'utilisateur. Le calcul de la somme pondérée au sein de chaque PE (16 multiplications synaptiques) se fait en un seul cycle d'horloge. Les poids peuvent être codés sur 8 ou 16 bits tandis que les entrées sont chargées sous forme de mots de 16 bits et peuvent être interprétées de plusieurs manières : 16×1 , 4×4 , 2×8 ou 1×16 bits. Des contre-réactions internes sont possibles afin de permettre les traitements multicouches au sein d'un même circuit. Une architecture monocircuit permet d'avoir une performance moyenne de 250 MCPS (la valeur pic étant de l'ordre de 300 MCPS).

Le circuit L-Neuro 1.0 (Learning Neurochip) de Philips³ fut le premier produit neuronal commercialisé par Philips. Il comprend 16 unités de traitement (PE) ainsi que des registres de 16 bits divisés en deux bancs de 16 sous-registres de 8 bits. Les 16 bits peuvent être traités par une PE comme étant des états de neurones codés sur 16×1 , 4×4 , 2×8 ou 1×16 bits. Par exemple, un seul circuit peut implanter un réseau de 256 neurones de 1 bit. Une mémoire interne de 1 Ko permet le stockage de 1024 poids synaptiques de 8 bits durant la phase de rappel et 512 poids de 16 bits durant la phase d'apprentissage. Les fonctions d'activation peuvent être implantées, sous forme d'une table de recherche (lookup table), extérieure au circuit. Plusieurs circuits peuvent être cascades, ce qui augmente la capacité du calcul neuronal (taille du réseau implanté, vitesse du traitement et précision des données). Les sorties des 16 PE sont lues d'une façon série, par l'intermédiaire d'un système hôte externe. Le L-Neuro 1.0 est facilement interfaçable à des processeurs hôtes du type Transputeur grâce à son bus compatible avec le leur. Du point de vue performance, un seul L-Neuro permet d'atteindre 100 MCPS et 26 MCPS pour les modes 1 et 8 bits respectivement. Le circuit peut être programmé pour réaliser la phase d'apprentissage et permet d'atteindre une performance de 160 MCUPS et 32 MCUPS pour les modes 1 et 8 bits respectivement.

3. <http://www.philips.com/>

A.1.2 Architectures multiprocesseurs

Circuits SIMD

Le **N64000** d'Inova⁴ [Hammerstrom90] est composé de 80 unités de traitement (PE) parmi lesquelles seulement 64 sont utilisées, les 16 PE supplémentaires étant réservées aux traitements d'erreurs dans le but d'améliorer le fonctionnement du circuit. Chaque PE contient un multiplieur entier 9×16 bits et un accumulateur de 32 bits auxquels s'ajoutent une mémoire cache de 4 Ko permettant le stockage des poids (8 ou 16 bits) et 32 registres. Les processeurs exécutent tous la même instruction en parallèle (SIMD). Enfin, l'utilisation du contrôle commun et des bus de données permet de cascader plusieurs circuits pour implanter des réseaux à grande taille. Les performances moyennes atteintes durant les phases d'apprentissage et de rappel sont d'environ 220 MCUPS et 871 MCPS respectivement.

Le **100-NAP (Neurocomputer Array Processor)** de HNC (Hecht-Nielson Computers⁵) [Means91] est composé de 100 PE fonctionnant en mode SIMD. Chacune d'entre elles peut exécuter des opérations arithmétiques 32 bits à point flottant (contrairement au N64000 qui le fait en arithmétique entière). Un bus d'adresse de 17 bits permet à chaque PE d'accéder 512 Ko de mémoire externe. Un seul NAP peut exécuter 160 MFLOPS et plusieurs circuits peuvent être mis en cascade.

Le **circuit CNAPS-1064 (Connected Network for Adaptive ProcessorS)** d'AS (Adaptive Solutions⁶) [Hammerstrom90]. Il s'agit d'un processeur digital parallèle composé de 64 sous-processeurs SIMD. Chaque sous-processeur possède sa propre mémoire locale de 4 Ko ainsi qu'une unité arithmétique à point fixe permettant d'effectuer des opérations arithmétiques entières sur 1, 8 ou 16 bits. Un ou plusieurs neurones peuvent être implantés sur un seul sous-processeur et plusieurs circuits peuvent être utilisés ensemble.

Le **circuit L-Neuro 2.3 (Learning Neurochip)** de Philips. Il s'agit de la seconde génération de circuits neuroniques fabriqués par Philips [Deville95]. S'appuyant sur l'expérience de L-Neuro 1.0, le L-Neuro 2.3 a marqué le début d'une nouvelle géné-

4. <http://www.inovacorp.com/>

5. <http://www.hnc.com/>

6. <http://www.asi.com/>

ration du développement matériel des réseaux de neurones surtout dans le domaine de pluralité des fonctions. Il est cascadable comme son prédécesseur. Il comprend 12 DSP internes, de 100 Mops chacun, pouvant fonctionner soit en mode parallèle (SIMD - Single Instruction Multiple Data), soit en mode pipeline (MISD - Multiple Instruction Single Data). Ce dernier peut se faire sur 10 niveaux, ce qui permet au L-Neuro 2.3 d'effectuer 36 opérations arithmétiques par cycle d'horloge (ce qui est l'équivalent de 144 instructions RISC). Les poids et les états de sortie des neurones sont codés sur 16 bits par défaut. Chacun des 12 processeurs contient 128 registres de 16 bits pour le stockage des poids et des états, un multiplieur $16 \rightarrow 32$, et une UAL de 32 bits. Un micro-codage du circuit peut être effectué par l'utilisateur ce qui permet d'adapter le circuit pour une application donnée. La fréquence de fonctionnement de 60 MHz permet au L-Neuro 2.3 de calculer la somme pondérée de 12 entrées de 16 bits toutes en 17 ns. Ceci correspond à une performance de 720 MCPS, soit 27 fois celle du L-Neuro 1.0 pour le mode 8 bits. Durant la phase d'apprentissage, la mise à jour des 12 poids se fait en parallèle en 34 ms. En plus de ces grandes capacités de calcul neuronal, le L-Neuro 2.3 peut traiter les applications de traitement du signal, d'images ainsi que dans le domaine de la logique floue (en utilisant sa capacité d'extraire les valeurs minimale et maximales dans les vecteurs de 12 éléments).

Circuits systoliques

Le circuit MA-16 de Siemens AG⁷ [Beichter91, Ramacher92]. Ce circuit offre la possibilité d'effectuer d'une manière rapide des calculs matriciels (multiplication, soustraction et addition) entre matrices $4 \times 4 \times 16$ bits. Le multiplieur et les accumulateurs ont une précision de 48 bits. Les poids sont stockés hors circuit ainsi que la fonction d'activation des neurones qui doit être implantée sous forme des tables de recherche (lookup table). Un contrôleur externe est nécessaire pour la synchronisation avec les dispositifs d'E/S. Plusieurs circuits peuvent être cascades. En ce qui concerne la performance, un seul MA-16 peut offrir jusqu'à 800 MCPS pour une fréquence de fonctionnement de 50 MHz.

7. <http://www.siemens.de/>

A.1.3 Autres implantations digitales

Le circuit d'AAC (Accurate Automation Corporation⁸). Il s'agit d'un processeur parallèle MIMD (Multiple Instructions Multiple Data) qui peut implanter jusqu'à 8192 neurones. Une mémoire externe (hors circuit) est utilisée pour stocker les poids synaptiques du réseau implanté. Les neurones peuvent être totalement interconnectés si l'on dispose d'une mémoire suffisante. Plusieurs sous-réseaux peuvent aussi être implantés ensemble. Le circuit utilise une arithmétique complément à 2 d'une longueur de 16 bits. Il est programmable et permet d'implanter tous les algorithmes d'apprentissage existants. La performance d'un seul circuit peut atteindre 140 MCPS.

Le circuit NISP (Neural Instruction Set Processor) – MT19003 de MEC (Microcircuit Engineering Corporation⁹) [MT19003]. C'est un processeur RISC très simple ayant un jeu de 7 instructions optimisées pour l'implantation des réseaux multicouches. Les entrées, poids synaptiques et sorties sont codés sur 13 bits. Le circuit possède un multiplieur de 16 bits et un accumulateur de 35 bits. Les poids sont stockés hors circuit. Quatre fonctions d'activation sont intégrées dans le circuit sous forme de tables. La performance du NISP est de 40 MCPUS pour un fonctionnement nominal à 40 MHz.

Les circuits d'ICT (Innovative Computing Technologies¹⁰). Ce sont des circuits qui permettent d'atteindre une performance de l'ordre de 6 GCPS tout en offrant la possibilité d'apprentissage intégré. Parmi ces circuits nous distinguons les HIP (Hybrid Image processor) conçus pour être utilisés dans le traitement temps réel d'images : détection du contour, compression vidéo et analyse de mouvements avec toutes les applications qui peuvent s'en servir (robotique, contrôle de circulation, systèmes de sécurité, etc.). D'autres circuits ont été développés dans le domaine du traitement de signal (séparation des sources en temps réel) ainsi que dans le domaine du contrôle adaptatif flou dans le but de compenser l'incertitude des capteurs et actionneurs tout en réalisant un contrôle intelligent et robuste.

Le circuit RSC-164 de SC (Sensory circuits¹¹). Il s'agit d'un circuit de reconnais-

8. <http://www.accurate-automation.com/>

9. <http://www.microcircuit.com/>

10. <http://www.webcom.com/~ictech/>

11. <http://www.sensoryc.com/>

sance de parole conçu pour être utilisé dans les applications électroniques "grand public". Il combine un processeur 8 bits avec un algorithme neuronique pour une reconnaissance de parole haute qualité. Il permet aussi l'enregistrement et le "play-back" de voix, la synthèse de musique et de parole ainsi que le contrôle des systèmes. Le circuit est fabriqué avec la technologie CMOS, il intègre une RAM, une ROM, 16 lignes d'E/S à usage général, des convertisseurs A/D et D/A et un processeur dédié 4 Mips. Le RSC-164 utilise un réseau de neurones pré-appris pour la reconnaissance des paroles tandis que la synthèse de paroles haute qualité est effectuée en utilisant une compression "time-domain" qui améliore le codage analogique-digital classique ADPCM (Analog to Digital Pulse Coded Modulation). Un filtrage digital intégré améliore la précision de reconnaissance par un prétraitement des signaux d'entrée. Le RSC-164 comprend une interface de mémoire externe lui permettant de se connecter avec des dispositifs mémoires pour l'enregistrement/play-back audio, et l'extension des longueurs des messages pour la synthèse de parole.

Le circuit pRAM-256 (Probabilistic RAM) d'UCLi Ltd [Clarkson93]. Il s'agit d'un processeur versatile de réseaux de neurones qui comporte un module d'apprentissage intégré. Il offre la flexibilité d'une solution logicielle avec la vitesse du matériel. Les connexions entre les neurones sont reconfigurables, ce qui permet de modifier l'architecture du réseau au moment désiré. Un seul circuit comprend 256 modules p-Ram qui peuvent être utilisés ensembles ou d'une façon indépendante. Durant la phase d'apprentissage, une passe complète des 256 pRAM constituants dure moins de 250 ns pour un fonctionnement à la fréquence maximale (33 MHz). Vu le grand nombre de pRAM pouvant être implanté sur le processeur pRAM-256, un réseau de neurones typique peut être construit en utilisant un seul module pRAM. Plusieurs modules pRAM peuvent fonctionner en parallèle ce qui permet de construire des réseaux plus complexes. Afin de pouvoir utiliser le pRAM-256 dans les systèmes actuels (et éventuellement futurs), des interfaces avec les systèmes utilisant les bus VME ou EISA ont été définies.

A.2 Implantations analogiques

Le circuit ETANN (Electrically Trainable Analog Neural Network) d'Intel¹² [Holler89]. Déposé par Intel sous le numéro TM80170NX, l'ETANN [80170NX91] est le premier circuit analogique commercial de son genre. Il peut implanter jusqu'à 64 neurones et 10240 connexions. Sa conception flexible (contre-réactions internes et division de l'ensemble des 10240 poids en 2 bancs de 64×80 chacun) permet 2 configurations de réseaux de neurones feed-forwards : (1) réseaux à 3 couches de 64 noeuds chacune et (2) réseaux à 2 couches ayant 128 noeuds d'entrée et 64 noeuds de sortie. Les poids sont directement chargés dans le circuit, l'apprentissage devant être effectué hors circuit. Le stockage des poids, d'une manière non volatile, se fait grâce à l'utilisation de la technologie à portes flottantes programmables (Programmable Floating Gates), une sorte d'EEPROM analogiques ayant une taille de 6 bits. Quant aux multiplications, elles sont effectuées par 4 bits en utilisant des multiplieurs de Gilbert. La fonction d'activation des neurones est la fonction sigmoïde (cf. eq. 2.9). Du point de vue performance, le temps de propagation, pour un réseau à deux couches est de 8 μs ce qui correspond à une vitesse de 2 GCPS. Plusieurs applications qui ont utilisé ETANN avec succès, ont montré sa fiabilité. Cependant, comme a été cité précédemment, il faut bien contrôler la stabilité des niveaux des tensions utilisées.

Le circuit LSANC (Large Scale Analog Neural Computer) de l'université de Pennsylvania¹³ [Der Spiegel94]. Il s'agit d'un circuit analogique implantant un réseau de neurones de grande taille. Ce dernier peut supporter en total jusqu'à 3072 entrées ; il comprend 1024 neurones ayant chacun 94 entrées. La programmabilité de son architecture et de ses composants en fait un outil de développement et de recherche pour de nombreuses applications. En plus, les constantes du temps du réseau sont programmables ce qui le rend capable d'effectuer des calculs en temps réel et en temps comprimé tout en représentant le temps par une variable de calcul. Ce circuit a été utilisé avec succès dans de nombreuses applications telles que la décomposition des signaux acoustiques dans la reconnaissance de parole, la reconnaissance de formes des signaux sonar et l'estimation de mouvement. Il a été également utilisé comme outil de développement pour les circuits neuronaux spécifiques (ASIC).

12. <http://www.intel.com/>

13. <http://www.upenn.edu/>

A.3 Implantations hybrides

Le circuit ANNA (Artificial Neural Network ALU) d'AT&T¹⁴ [Boser92]. C'est un circuit *digital à l'extérieur, analogique à l'intérieur*. Il peut être utilisé pour une large gamme de réseaux de neurones mais est optimisé pour les réseaux localement connectés, à poids partagés ainsi que les réseaux à neurones retardés dans le temps (Time Delay Neural Networks TDNN). Les poids synaptiques sont calculés hors circuit et y sont représentés sous forme de tensions. L'interface avec le circuit est purement digitale et se fait grâce à 2 CAD (convertisseurs analogique/digital) intégrés qui transforment les poids synaptiques, codés sur 6 bits, en leur équivalent en tension. L'ANNA peut implanter jusqu'à 8 neurones et 4096 connexions. Les opérations de multiplication y sont effectuées grâce à 512 multiplieurs organisés en 8 groupes de 64 synapses chacun. La performance moyenne d'un seul circuit ANNA est de l'ordre de 2 GCPS (la performance pique étant de 5 GCPS).

Le circuit CLNN-32 de Bellcore (Bell Communications Research¹⁵). Ce circuit utilise une architecture complètement récurrente. Il peut implanter l'algorithme d'apprentissage de Boltzmann [Alspector91] et supporter jusqu'à 32 neurones. Les entrées et les sorties sont analogiques, ainsi que les traitements internes, tandis que les poids sont digitaux. L'apprentissage peut se faire très rapidement et l'on peut atteindre une performance de 100 MCUPS. Deux circuits peuvent être combinés par l'intermédiaire d'un circuit implantant uniquement des connexions (le CLNS64), ce qui permet d'obtenir un réseau complètement connecté à 64 neurones.

Le circuit Epsilon (Edinburgh Pulse Stream Implementation of a Learning Oriented Network) [Murray94a]. Il s'agit d'un circuit neuronal pouvant implanter jusqu'à 30 neurones et 3600 poids synaptiques tout en permettant de construire des réseaux plus grands en cascasant plusieurs circuits. L'Epsilon peut être utilisé comme un accélérateur pour des ordinateurs classiques aussi bien que comme un processeur autonome. Les poids synaptiques sont implantés grâce à un circuit transconductance multiplieur utilisant les transistors MOS dans la région de fonctionnement linéaire. Les courants de sortie générés sont proportionnels au produit des 2 tensions d'entrée dont l'une est fixe et l'autre correspond à l'état du neurone à multiplier par le poids.

14. <http://www.att.com/>

15. <http://www.bellcore.com/>

Quant aux implantations des neurones, 2 catégories peuvent être distinguées : (1) l'implantation synchrone qui utilise la modulation impulsionnelle de phase et qui est spécialement conçue pour les applications du type tacitement d'image et de parole et (2) l'implantation asynchrone utilisée dans les réseaux récurrents avec la modulation impulsionnelle de fréquence. Côté performance un seul circuit Epsilon peut atteindre 360 MCPS.

Le circuit NeuroClassifier de MRI (Mesa Research Institute¹⁶) [Masa94, N. Class96].

Il peut implanter, d'une façon analogique, 70 entrées, 6 unités cachées et une seule sortie, les poids étant codés en digital sur 5 bits. La sortie finale est obtenue sous forme de somme pondérée (sans passage par la fonction d'activation), de manière à ce que plusieurs circuits puissent être cascades dans le but d'augmenter le nombre d'unités cachées. La performance atteinte est de l'ordre de 20 GCPS pour une implantation monocircuit.

A.4 Implantations optoélectroniques

Le circuit EOLC (Electro-Optical Learning Chip) de Mitsubishi¹⁷ intègre le processus d'apprentissage grâce à des connexions électriquement programmables. Pour cela, des photodiode à sensibilité variable (VSPD) ont été utilisées. Ceci permet, en effet, de changer la sensibilité de la photo-détection en changeant la tension appliquée.

16. <http://www.el.utwente.nl/mesa/>

17. <http://www.merl.com/>

Annexe B

Cartes accélératrices des réseaux de neurones

Dans cet annexe sont décrites brièvement les principales cartes accélératrices commercialisées utilisées pour le développement et l'exécution des réseaux de neurones. De telles cartes peuvent être construites aussi bien autour de circuits standards à usage général (microprocesseurs, transputeurs, DSP) qu'autour de circuits dédiés aux calculs neuronaux. Cette dernière catégorie comprend les circuits neuronaux à usage général et les circuits ASIC conçus pour des applications spécifiques. Des cartes hybrides combinant les 2 types de circuits peuvent également exister. Comme pour l'annexe A, une bonne partie de la bibliographie a été recueillie directement de la page Web des différentes compagnies de développement logiciel et matériel par Internet (les adresses Web correspondantes sont données sous forme de notes de bas de pages).

B.1 Accélérateurs construits autour des circuits standards

Dans ce qui suit, sont cités les accélérateurs commerciaux les plus répandus ; bien que certains d'entre eux existent en versions anciennes, nous nous contenterons de décrire brièvement les dernières versions commercialisées (des nouvelles versions peuvent être en cours).

L'accélérateur professionnel BrainMaker de California Scientific Software¹ est

1. <http://www.calsci.com/>

un système complet de développement des réseaux de neurones comprenant 2 cartes DSP (bus PC/ISA) construites autour de 4 processeurs TI-TMS320C25 fonctionnant en parallèle et interfacées grâce à un outil de développement logiciel accompagnant. La carte peut atteindre une vitesse de 500 MFLOPS, ce qui correspond à une performance de 40 MCPS. De nouvelles cartes ont été construites autour des circuits CNAPS d'Adaptive Solutions dans le but d'amélioration de performances.

L'accélérateur ANZA plus de HNC² consiste en une carte coprocesseur construite autour du microprocesseur MC68020 et du coprocesseur arithmétique MC68881 de Motorola³ et dont l'interface avec l'utilisateur se fait par une librairie de sous-programmes [Trealeaven89]. La performance moyenne de la carte est de 6 MCPS et 1.5 MCUPS pour les phase de rappel et d'apprentissage respectivement.

L'accélérateur Neuro Turbo (MMBB) de NIT (Nagoya Institute of Technology⁴) [Arif93] est une carte à usage général construite autour des DSP 96002 de Motorola et pouvant atteindre une performance de 16.5 MCUPS.

L'accélérateur Balboa de HNC est un système de développement des réseaux de neurones comprenant une partie logicielle (ExploreNet), et une partie matérielle consistant en une carte (VME ou PC/AT) construite autour du circuit RISC i860 d'Intel. Ce système peut supporter 20 architectures différentes des réseaux de neurones et peut atteindre une performance moyenne de 25 MCPS et 9 MCUPS pour un réseau BP (retro-propagation).

L'accélérateur NETSIM de TI (Texas Instruments⁵). C'est un système composé de plusieurs cartes qui émulent des réseaux de neurones. Ces cartes sont connectées en un tableau tridimensionnel et sont commandées par l'intermédiaire d'un système hôte (PC) [Garth87]. Une carte NETSIM est construite autour d'un microprocesseur 80188 et peut émuler jusqu'à 256 neurones ayant chacun jusqu'à 256 connexions. Un maximum de quinze cartes NETSIM peut être adressé par dimension. La performance totale atteinte est de 450 MCPS et 90 MCUPS.

L'accélérateur MUSIC (MUltiprocessor System with Intelligent Communication) [Muller95] est un système à usage général qui peut être utilisé dans le développement des réseaux de neurones. Il est construit autour de 66 processeurs (63 DSP 96002 de Motorola et 3 T805 d'Inmos) ainsi qu'un espace de stockage mémoire

2. <http://www.hnc.com/>

3. <http://www.motorola.com/>

4. <http://www.nitech.ac.jp/index-e.html>

5. <http://www.ti.com/>

statique et dynamique d'environ 180 Moctets. L'interface avec l'utilisateur se fait par l'intermédiaire d'un ordinateur hôte (Station Sun, PC ou Macintosh) et d'un environnement de développement logiciel (C et assembleur).sa performance moyenne est de 1.4 GFLOPS ce qui correspond à 817 MCPS et 330 MCUPS.

La carte accélératrice NT6000 de Neural Technologies Limited⁶ est construite autour du DSP TI-TMS320C20 et peut supporter des réseaux ayant jusqu'à 2700 neurones avec une performance de 2 MCPS. Les traitements internes y sont digitaux tandis que les E/S sont analogiques et des convertisseurs A/D et D/A permettent la mise en forme entre signaux.

L'accélérateurs XP50 de NeuroDynamX⁷ consiste en une carte PC/ISA construite autour du processeur RISC i860 d'Intel fonctionnant à 25 MHz et permettant une performance de l'ordre de 45 MCPS tout en offrant un espace mémoire de stockage de 64 Moctets.

La carte NeuroSimulator de Vision Harvest⁸ est un accélérateur neuronal utilisant le processeur RISC i860 d'Intel. Le développement des réseaux de neurones y est effectué grâce à l'outil logiciel NeuroVision qui permet d'atteindre une performance de 30 MCPS.

La carte RAP (Ring Array Processor) d'ICSI (International Computer Science Institute⁹) [Morgan92]. Il s'agit d'un accélérateur neuronal puissant utilisé, depuis 1990, dans le domaine du développement des algorithmes connexionistes pour la reconnaissance de parole. Il peut supporter entre 4 et 40 processeurs de signaux TI-TMS320C30 connectés via un anneau de circuits PGA (Programmable Gate Array) du type Xilinx. La partie logicielle comprend un interpréteur de commandes, ainsi qu'une librairie sous-programmes de calculs matriciels. La performance atteinte par une seule carte est de 57 MCPS et 13.2 MCUPS.

D'autres systèmes construits autour des processeurs standards existent mais sont moins connus, parmi lesquels nous pouvons citer le système QED de Mosaic¹⁰ qui est un système à usage général qui peut être configuré pour le développement des réseaux de neurones. Un autre système est l'accélérateur NeuralBoard de Ward Systems¹¹ pouvant faire exécuter simultanément plusieurs réseaux de neurones tout en offrant

6. <http://www.neuralt.com/>

7. <http://www.teleport.com/~cognizer/COMPANY/42088936.HTM>

8. <http://www.teleport.com/~cognizer/COMPANY/42133510.HTM>

9. <http://www.icsi.com/>

10. <http://www.mosaic-industries.com/>

11. <http://www.wardsystems.com/>

la possibilité d'installer plusieurs cartes (jusqu'à 10) sur une même machine.

B.2 Accélérateurs construits autour des circuits neuronaux

Les accélérateurs de Rapid Imaging¹². Il s'agit de 2 cartes à usage général construite autour du circuit ETANN d'Intel : la carte PCX 0491E1-ISA et la carte VME Ultima respectivement. Une environnement spécifique permet d'effectuer l'apprentissage avant que le réseau de neurones considéré soit implanté sur les circuits ETANN.

L'accélérateur 1000 NeuroEngine de Telebyte¹³. C'est une carte qui joue le rôle d'un coprocesseur pour PC en lui apportant la vitesse des traitements neuronaux. Elle est construite autour du processeur MIMD développé par AAC¹⁴, et permet d'atteindre une performance de 140 MCPS pour une carte à un seul circuit. Une carte similaire a été développée pour les systèmes fonctionnant sur des bus du type VME.

L'accélérateur SYNAPSE-1 (Synthesis of Neural Algorithms on a Parallel Systolic Engine) de Siemens AG¹⁵ [Beichter93]. Il s'agit d'un système complet matériel/logiciel construit autour de 8 circuits MA-16. C'est un neuro-ordinateur qui peut être programmé pour apprendre des réseaux avec la plupart des algorithmes existants. La performance moyenne est de l'ordre de 800 MCPS et peut atteindre des valeurs piques de 3.2 GCPS pour une fréquence de fonctionnement de 25 MHz.

Le système CNAPS d'AS (Adaptive Solution¹⁶) [Hammerstrom90] est un environnement complet pour le développement logiciel et matériel des réseaux de neurones. Réalisé autour d'une carte compatible VME, il offre un espace de stockage de 16 Mo et peut être configuré pour implanter jusqu'à 4 circuits CNAPS-1064 (cf. 2.8.3.1.2). La programmation de ces derniers en vue d'y implanter des réseaux de neurones se fait grâce à un outil de programmation puissant, le CNAPS tools. Ce dernier comprend un compilateur et un débogueur C (CNAPS-C), une bibliothèque de libraries standards (QuickLib) qui peuvent être appelées par CNAPS-C, des algorithmes d'apprentissage

12. <http://www.nm.org/~landform/>

13. <http://www.telebyteusa.com/>

14. <http://www.accurate-automation.com/>

15. <http://www.siemens.de/>

16. <http://www.asi.com/>

pré-codés (BuildNet) et un débogeur de langage assembleur. Une version PC/ISA, identique à la précédente, a également été développée.

Du point de vue performance, on peut atteindre un nombre d'opérations multiplication/accumulation par seconde de 1.28 G pour un seul circuit et de 10.24 G pour 4 circuits interconnectés. Ceci correspond, pour la BP, à 1.16 GCPS et 293 MCUPS pour le premier cas et à 5.80 GCPS et 1.95 GCUPS dans le deuxième.

La carte accélératrice SNAP (SIMD Numerical Array Processor) de HNC¹⁷ [SNAP93] est construite autour du circuit 100-NAP (cf. 2.8.3.1.2) et permet un flux de données pouvant atteindre 4 Go/sec. Une configuration standard à deux cartes (32 processeurs 100-NAP) peut offrir une performance de 500 MCPS (réseaux multicouches) et 128 MCUPS (BP).

L'accélérateur MY-NEUPOWER de Hitachi¹⁸ a été développé autour des circuits neuronaux spécifiques. Il peut supporter différents algorithmes d'apprentissage : BP, Hopfield et LVQ (Learning Vector Quantization) [Sato93] et peut implanter jusqu'à 512 neurones physiques. Les réseaux plus grands peuvent être simulés en attribuant les neurones à la mémoire globale du système, on peut alors implanter jusqu'à 4096 neurones virtuels. La performance atteinte est de l'ordre 1.26 GCUPS et 12.8 GCPS.

L'accélérateur de Philips¹⁹ [Mauduit91]. Il s'agit d'une machine prototype à 4 cartes et 16 processeurs du type L-Neuro 1.0 qui peut implanter des réseaux ayant plus de 2000 neurones. La performance atteinte est 19 MCPS et 4.2 MCUPS.

L'accélérateur iNNTS d'Intel²⁰ est un environnement de développement PC comprenant une carte construite autour du circuit ETANN ainsi qu'un module d'apprentissage extérieur et la partie logicielle correspondante à l'émulation et l'interface. Deux versions existent avec un et 8 circuits respectivement. L'apprentissage des réseaux doit se faire avant leur implantation puisque l'ETANN ne le permet pas. Du point de vue performance, le temps de propagation, pour un réseau à deux couches est de 8 μs ce qui correspond à une vitesse de 2 GCPS.

17. <http://www.hnc.com/>

18. <http://www.hitachi-mc.co.jp/>

19. <http://www.philips.com/>

20. <http://www.intel.com/>

Annexe C

Classement des applications neuronales par domaines d'application

Les domaines d'applications actuelles et potentielles des réseaux de neurones sont d'une grande diversité. Ceci est dû principalement à la diversité des secteurs d'activités touchés par cette approche de traitement de l'information. Principalement, nous pouvons distinguer les domaines d'applications suivants :

1. La reconnaissance des formes ;
2. Le traitement du signal ;
3. Le contrôle ;
4. Le traitement d'images ;
5. Le traitement de la parole ;
6. La prévision et la modélisation ;
7. L'optimisation.

A ces domaines s'ajoutent les aspects recherche des réseaux de neurones qui s'attachent à la modélisation du système nerveux et des phénomènes cognitifs.

D'autres applications neuronales existent et ne font pas forcément partie des domaines cités précédemment. Des documentations en ligne peuvent être trouvées en [Archive96].

Dans ce qui suit, nous allons essayer de donner, pour ces différents domaines, quelques exemples d'applications. Ces derniers seront choisis de sorte à éviter les redondances avec les applications citées dans le paragraphe 2.9.3.

C.1 Reconnaissance de formes

C'est un domaine privilégié d'application pour les réseaux de neurones et c'est lui qui a marqué leur début. Le terme de reconnaissance est un terme général qui désigne les processus traitant des données pour en extraire des informations afin de parvenir à leur classification. Une partie non négligeable des applications neuronales actuelles appartient à cette catégorie, parmi lesquelles nous pouvons citer :

- reconnaissance de caractères Latins [Jacquet94], Thaï [Devel95], Hindi [Verma95], Chinois [Ohhira95], Coréens [Jung95], Japonais [Hammerston93], Kanji [Kimura94] et Russes [Tierney94];
- reconnaissance de visages [Augusteijn93, Eide96, Zhang96];
- classification d'images [Nath94] et de documents [Le94];
- diagnostic des maladies [Shen95];
- diagnostic des pannes dans les installations électriques [Nogami94], nucléaires [Bartal94], mécaniques [Lucking94, Neuredge 94] ainsi que dans les systèmes de télécommunications [Fu95].

C.2 Traitement du signal

Le traitement des signaux stochastiques (filtrage, classification) provenant des systèmes non linéaires est une tâche difficile avec les méthodes classiques de traitement. En effet, la résolution de ce type de problèmes implique la capacité de modéliser d'une façon adéquate, le système inconnu afin d'estimer sa sortie, ce qui n'est possible que pour des cas simples. La capacité d'apprentissage des réseaux de neurones leur permet d'extraire les propriétés statistiques des systèmes non linéaires, ce qui les rend aptes à traiter ce type de problèmes. Parmi les applications publiées dans ce domaine nous pouvons citer :

- filtrage et suppression de bruits : amélioration du rapport signal/bruit [Vanhulle95].

- analyse de signaux [Malferrari90, Conde94];
- diagnostic de pannes [Keyvan93] et de maladies [Dorffner94];
- démodulation multi-utilisateurs [Mitra93];
- classification des cibles radars dans les avions [Chakrabarti95];
- classification des bruits [Legitimus90];
- compression [Schmidhuber96] et fusion [Wolff93] des données.

C.3 Contrôle adaptatif

Le contrôle adaptatif est une tâche critique dans les systèmes à contrôle automatique dont les paramètres et/ou la structure d'objets changent avec le temps (systèmes non stationnaires), rendant ainsi difficile la modélisation de l'application correspondante. La nature non linéaire des réseaux de neurones ainsi que leur capacité d'apprentissage à base d'exemples du problème traité leur permettent de résoudre, avec une facilité relative, ce type de problèmes. Nous distinguons des applications telles que :

- détermination du rapport air/fuel dans la chaîne de contrôle d'un moteur à combustion [Linde95];
- gestion adaptative d'un réseau de communication [Hiramatsu94];
- gestion de la tension artérielle chez les patients sérieusement affectés [Xu95];
- contrôle des réacteurs chimiques [Panetsos93];
- contrôle de température [Khalid95];
- contrôle intelligent de la circulation [Park94].

C.4 Traitement d'images

Les applications actuelles et potentielles des réseaux de neurones dans ce domaine sont très variées. Parmi les travaux publiés nous pouvons citer :

- détection et suppression de bruit dans les images [Kong95a];

- compression d'images statiques et dynamiques [Burel93, Liu93];
- identification d'empreintes digitales [Kamijo93];
- détection de contours des images [Iwata95];
- segmentation des images médicales [Dhawan93, Peter95];
- reconnaissance de tumeurs dans les images mammographiques [Feltham94];
- restauration aveugle d'images [Liu93].

C.5 Traitement de la parole

Depuis la fin des années 80, les réseaux de neurones sont utilisés avec succès dans les applications traitant la parole, en particulier dans la classification des phonèmes et des mots et dans la reconnaissance automatique de la parole. Plusieurs modèles de réseaux ont été utilisés : les perceptrons multicouches (MLP) [Hackbarth89], les réseaux LVQ (Learning Vector Quantization) [Mac Dermott89], les réseaux TDNN (Time Delay Neural Network) [Waibel88] et les réseaux RBF (Radial Basis Function) [Elman88]. Parmi les applications trouvées dans la littérature, nous pouvons citer :

- identification d'interlocuteur [He95];
- reconnaissance de paroles [Wang95, Ruan95];
- conversion texte/phonème [Lu95];
- catégorisation syntaxique des mots [Garner95].

C.6 Modélisation

Parmi les travaux publiés dans ce domaine, nous pouvons citer :

- modélisation de processus chimiques [Tsapsinos93, Christo95] et biochimiques [Solaiman94];
- modélisation de la reproduction de la parole [Valafar95];
- modélisation du système cardiovasculaire [Killer95];
- modélisation mécanique [Mazumdar95].

C.7 Prédiction

La prédiction constitue une tâche complexe dont la résolution n'est pas toujours évidente avec les systèmes classiques de traitement de l'information vu la nature chaotique et imprévisible de l'information traitée et la grande quantité de données à manipuler. Les réseaux de neurones ont été utilisés avec succès dans les applications de prédiction telles que :

- prédiction de cours financiers [Green95];
- prédiction de la consommation [Brown94] et des salaires [Falas94];
- prédiction de survie dans les soins traumatiques [Mac Conigal94];
- prévisions météorologiques [Shinozawa94];
- prédiction du flux de circulation sur les autoroutes [Ulbricht94].

C.8 Optimisation

Le problème le plus fréquemment traité dans la littérature correspondant à ce domaine est le problème du voyageur de commerce (Travelling Salesman Problem – TSP). Il s'agit d'un problème didactique dont la résolution consiste à trouver le plus court chemin reliant N villes en passant une fois et une seule par une ville donnée, connaissant les coordonnées de chaque ville et les distances intervilles. Cependant, les réseaux de neurones ont été utilisés dans les applications de ce domaine telles que l'affectation, la planification et l'allocation des ressources. Parmi les travaux trouvés, nous pouvons citer :

- optimisation de processus chimiques [Safavi95];
- optimisation des périodes de radio-diffusions des satellites [Ansari95];
- contrôle optimal des fourneaux à arcs électriques (consommation électrique, usage des électrodes, etc.) [N. A. Corp.];
- optimisation du système électoral [Saito95].

C.9 Robotique

La robotique est parmi les applications actuelles et futures prometteuses, notamment dans le domaine de l'industrie (contrôle et asservissement) et de navigation des robots mobiles (éviter d'obstacles, localisation de positions et planification de chemins). Vu la nature imprévisible de l'environnement de fonctionnement de telles applications et le besoin d'adaptation avec cet environnement, les réseaux de neurones constituent un bon choix d'implantation. Parmi les travaux publiés dans ce domaine nous pouvons mentionner :

- contrôle adaptatif d'actionneurs de robots [Li95] ;
- asservissement visuel d'un robot pour la coordination bras-œil [Kuhn95] ;
- reconstruction d'environnement pour la navigation d'un robot dans un milieu sévère [Bohn94] ;
- positionnement du bras d'un robot [Der Smagt95, Lewis96] ;
- contrôle d'un robot mobile [Ito95].
- contrôle sensori-moteur des robots autonomes [Tarassenko90] ;

Annexe D

Schémas de la carte 68000

FIG. D.1 : Buffers d'adresses

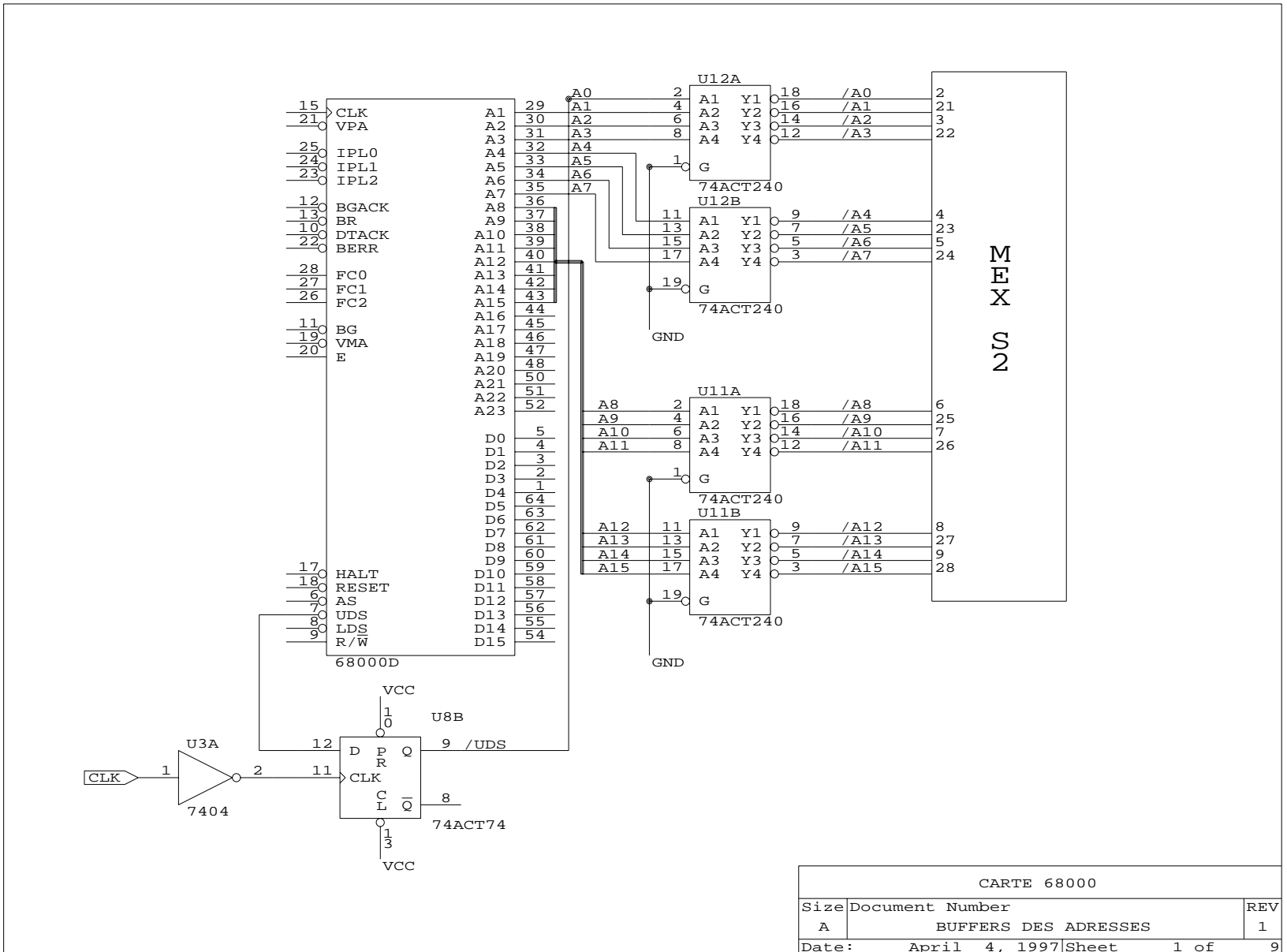
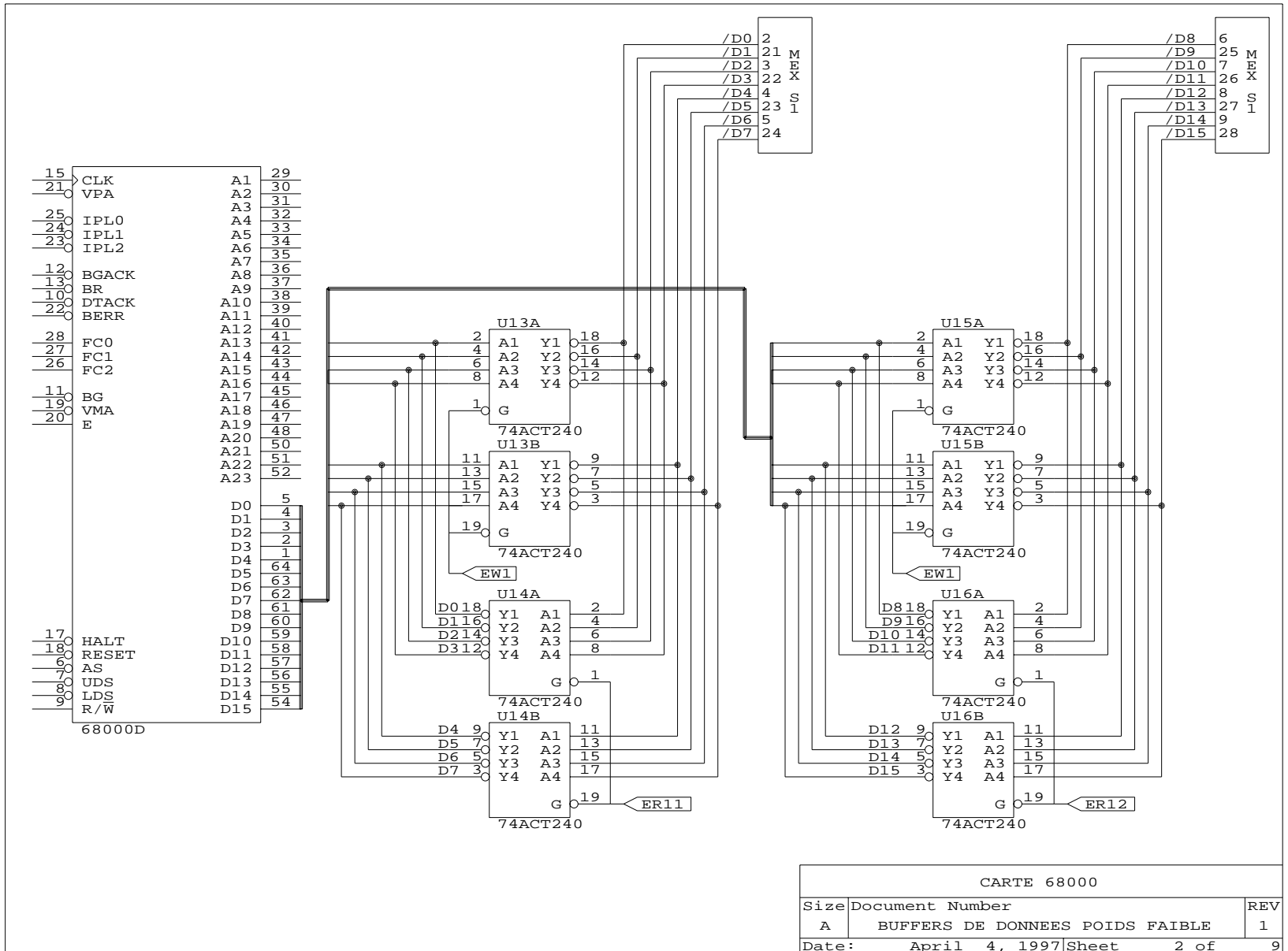
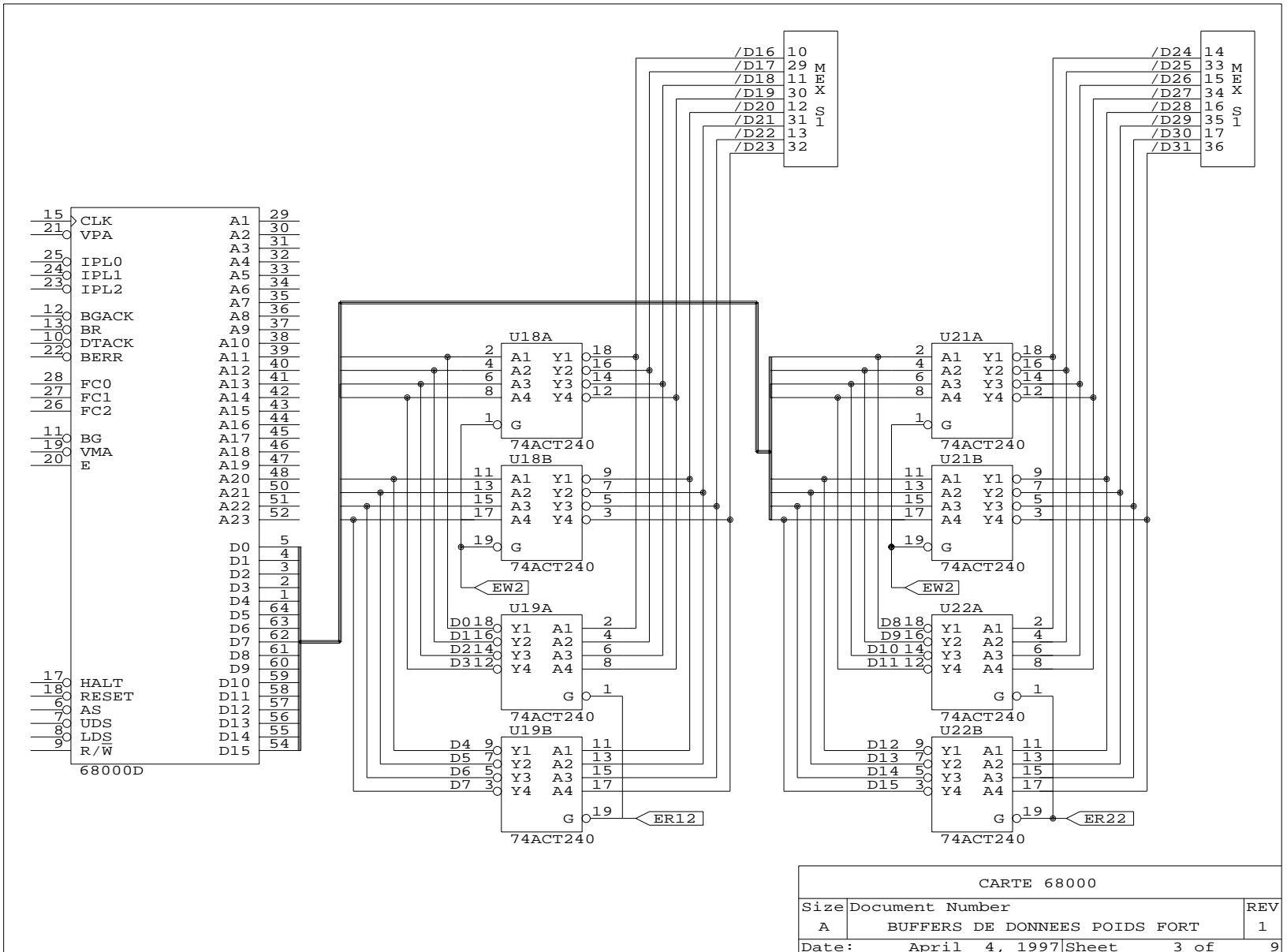


FIG. D.2 : Buffers de données poids faible



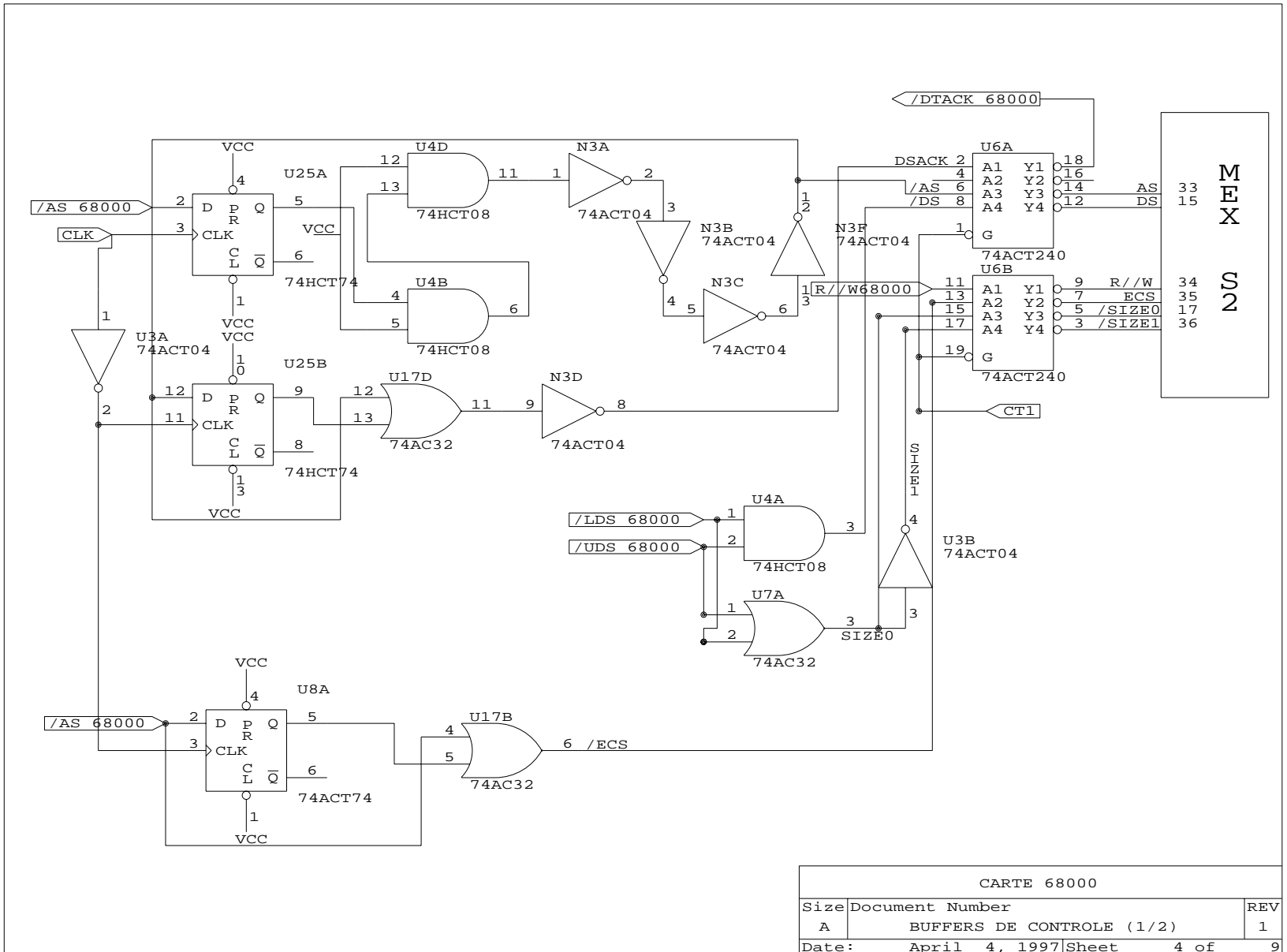
CARTE 68000		
Size	Document Number	REV
A	BUFFERS DE DONNEES POIDS FAIBLE	1
Date:	April 4, 1997	Sheet 2 of 9

FIG. D.3 : Buffers de données poids fort



CARTE 68000		
Size	Document Number	REV
A	BUFFERS DE DONNEES POIDS FORT	1
Date:	April 4, 1997	Sheet 3 of 9

FIG. D.4 : Buffers de contrôle (1/2)



CARTE 68000		
Size	Document Number	REV
A	BUFFERS DE CONTROLE (1/2)	1
Date:	April 4, 1997	Sheet 4 of 9

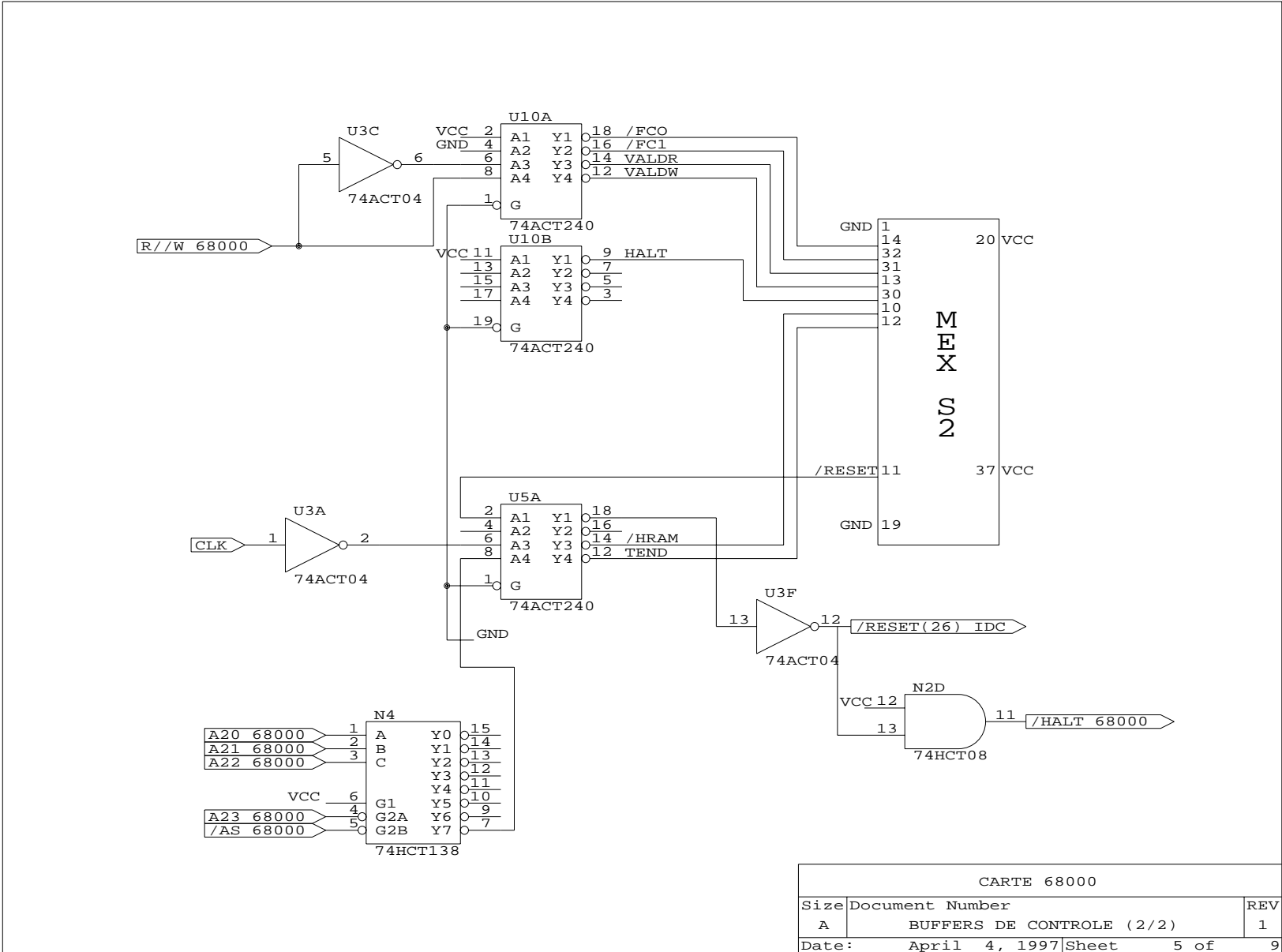


FIG. D.5 : Buffers de contrôle (2/2)

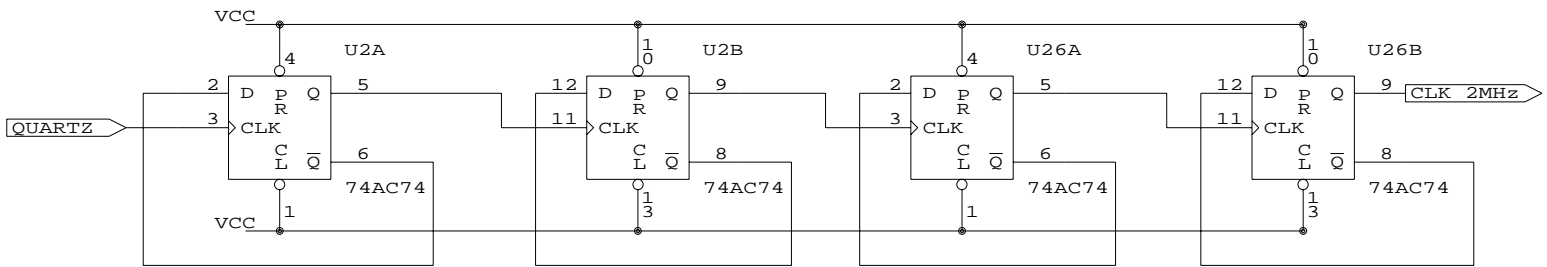
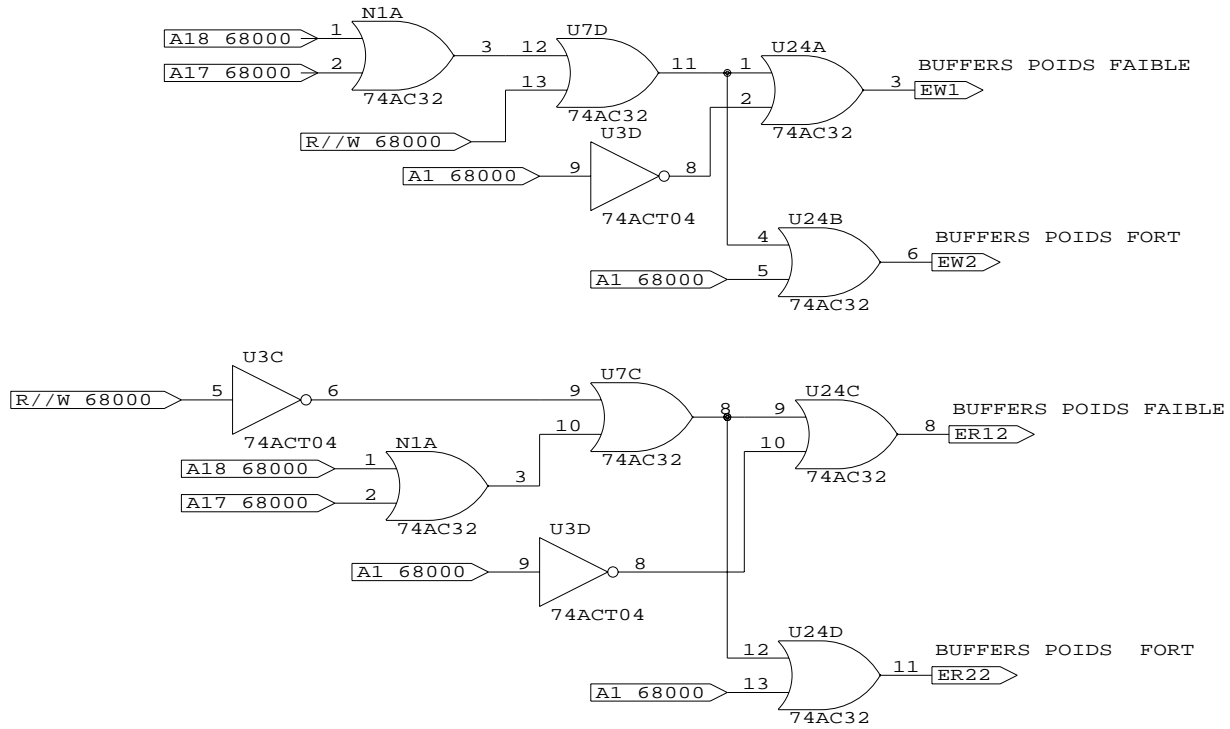


FIG. D.6 : Génération de l'horloge

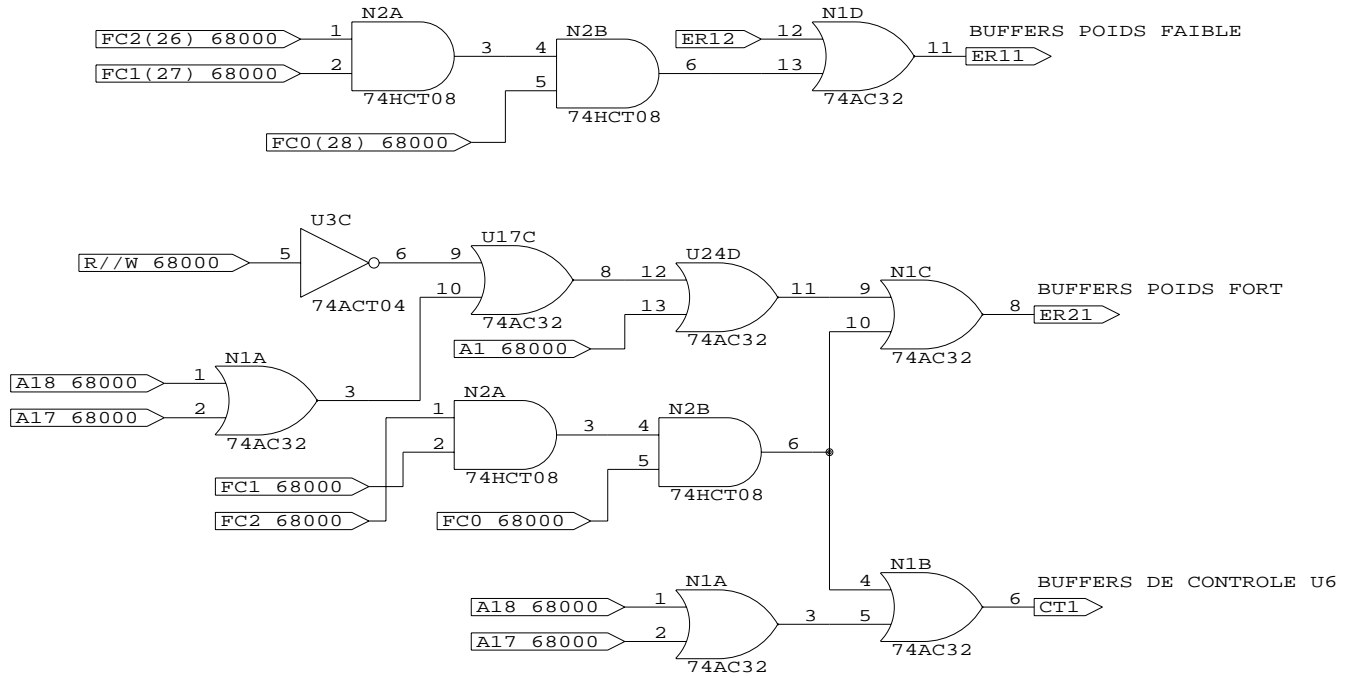
CARTE 68000		
Size	Document Number	REV
A	GENERATION DE L'HORLOGE	1
Date:	April 4, 1997	Sheet 6 of 9

FIG. D.7 : Génération des signaux de contrôle (1/2)



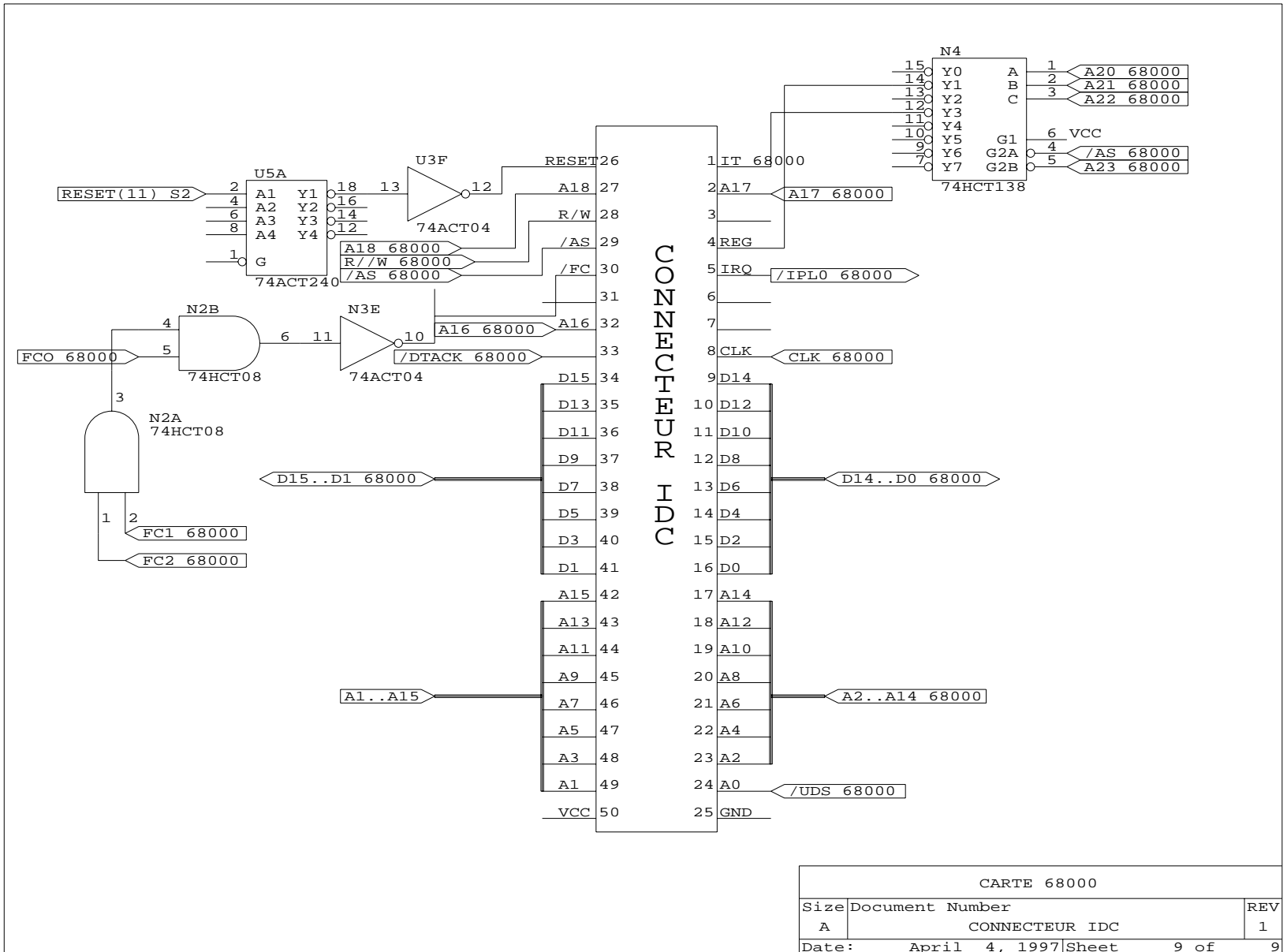
CARTE 68000		
Size	Document Number	REV
A	SIGNAUX DE CONTROLE DE BUFFERS (1/2)	1
Date:	April 4, 1997	Sheet 7 of 9

FIG. D.8 : Génération des signaux de contrôle (2/2)



CARTE 68000		
Size	Document Number	REV
A	SIGNAUX DE CONTROLE DE BUFFERS (2/2)	1
Date:	April 4, 1997	Sheet 8 of 9

FIG. D.9 : Connecteur IDC (interface carte mère – carte fille)



CARTE 68000		
Size	Document Number	REV
A	CONNECTEUR IDC	1
Date:	April 4, 1997	Sheet 9 of 9

Annexe E

Schémas de la carte UTAK

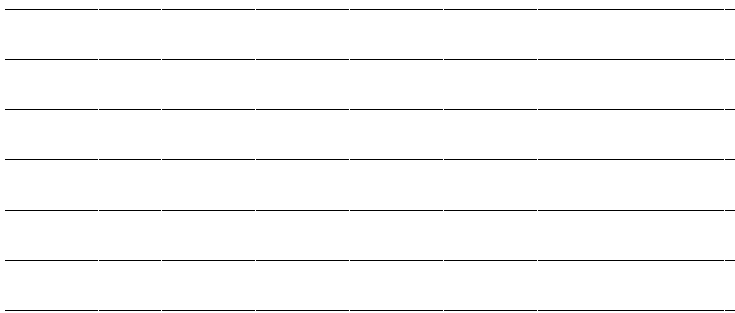
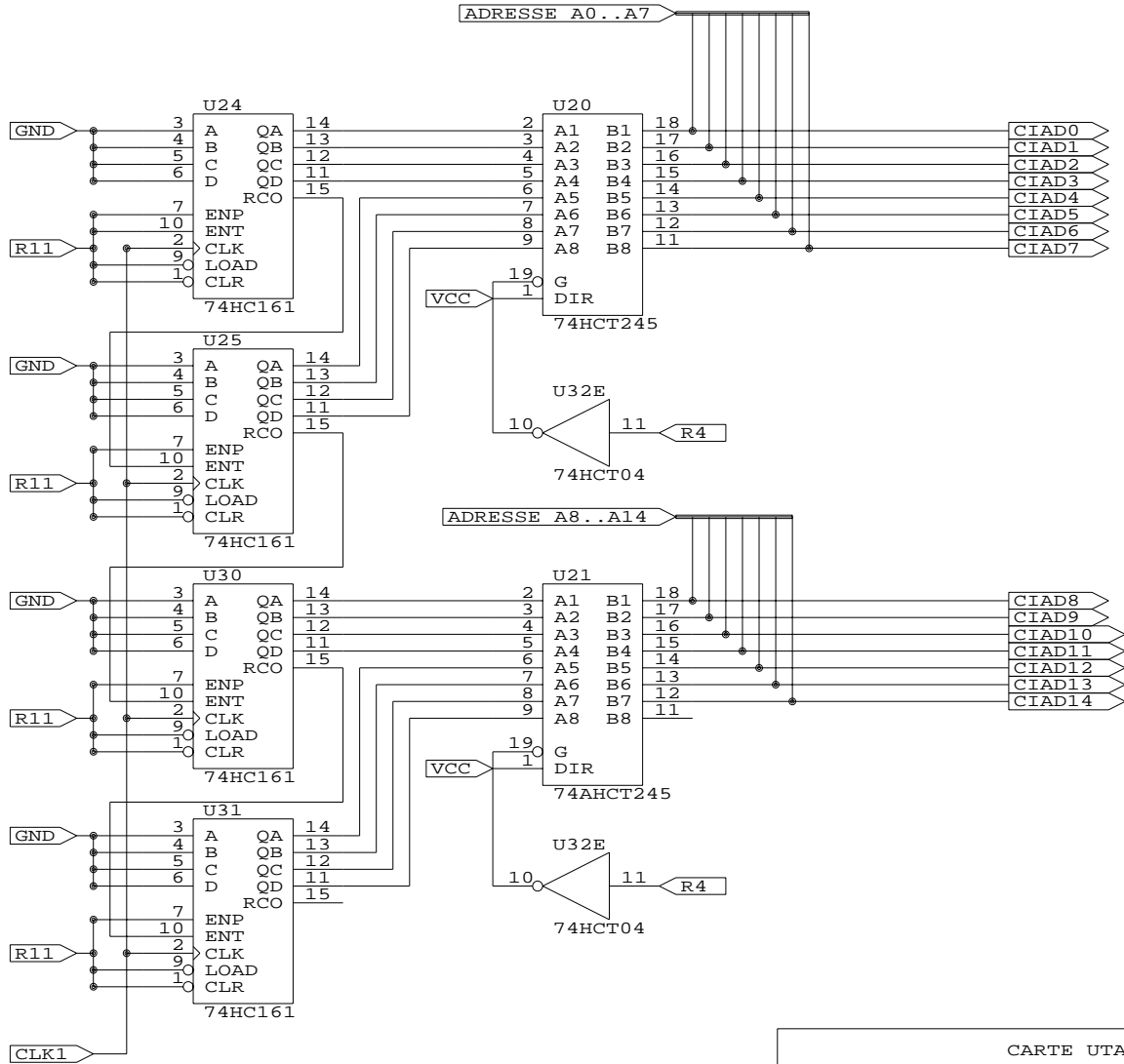
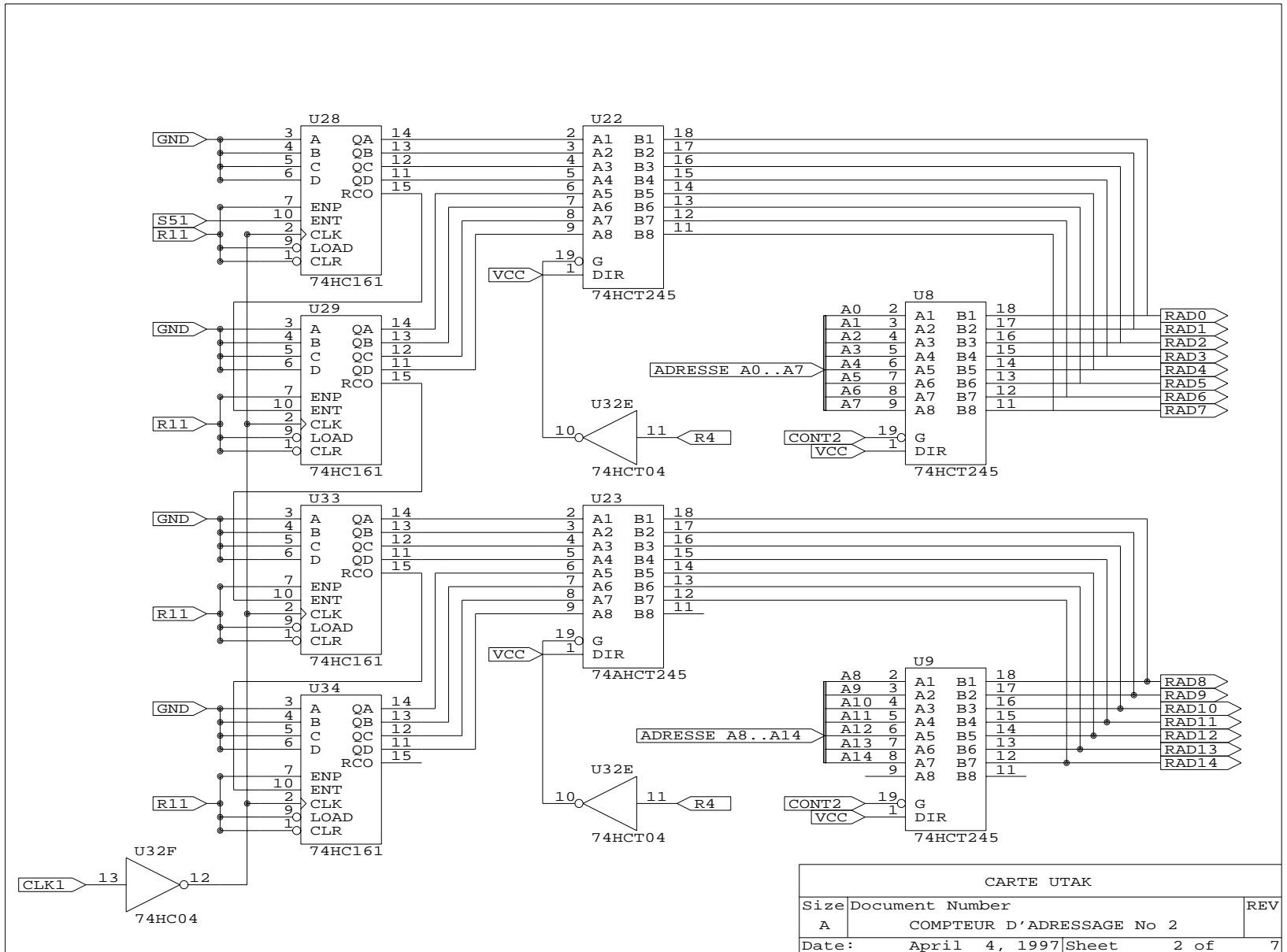


FIG. E.1 : *Compteur d'adressage N° 1*



CARTE UTAK		
Size	Document Number	REV
A	COMPTEUR D'ADRESSAGE No 1	
Date:	April 4, 1997	Sheet 1 of 7

FIG. E.2 : Compteur d'adressage N° 2



CARTE UTAK		
Size	Document Number	REV
A	COMPTEUR D'ADRESSAGE No 2	
Date:	April 4, 1997	Sheet 2 of 7

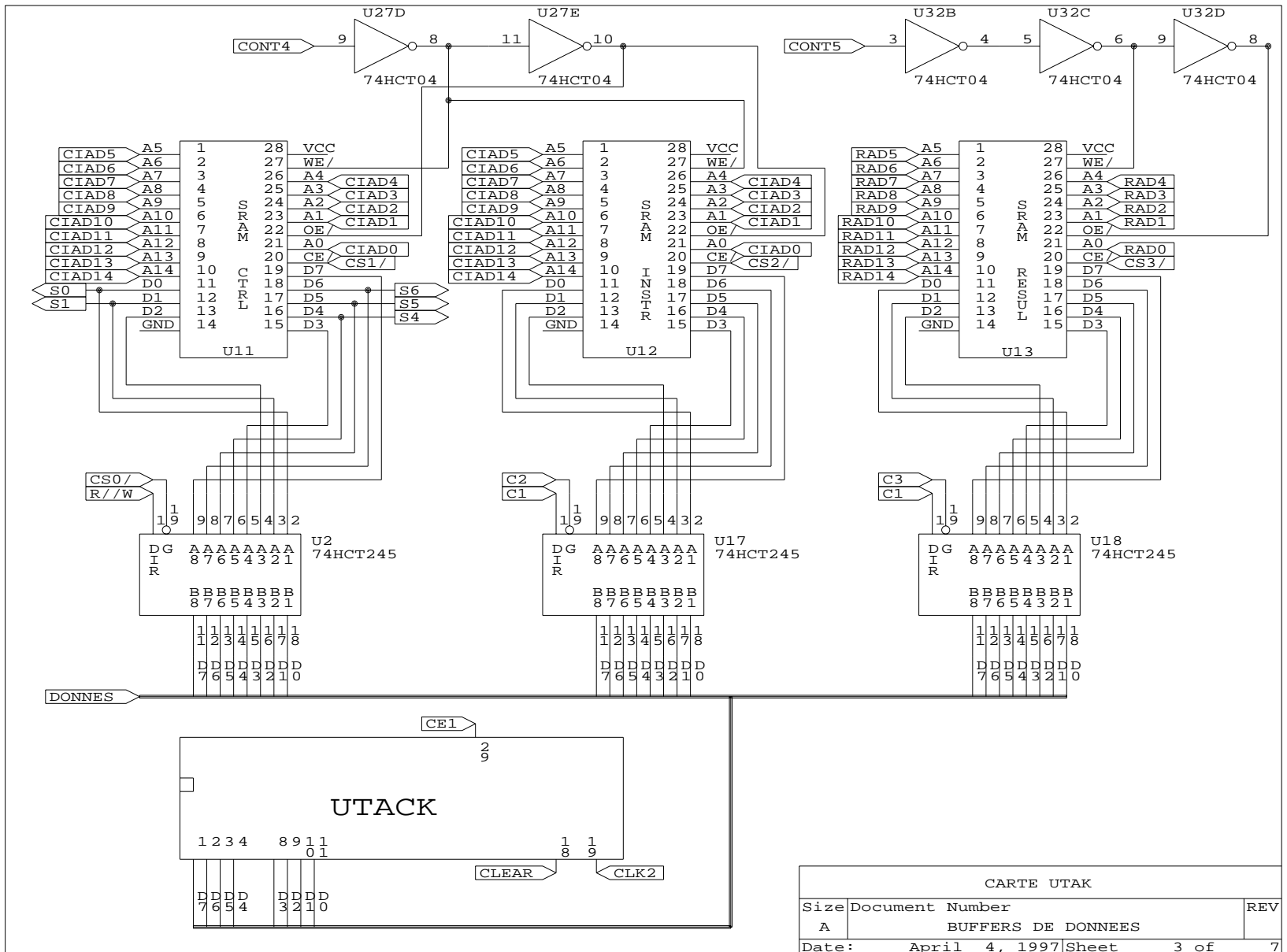


FIG. E.3 : Buffers de données

CARTE UTAK		
Size	Document Number	REV
A	BUFFERS DE DONNEES	
Date:	April 4, 1997	Sheet 3 of 7

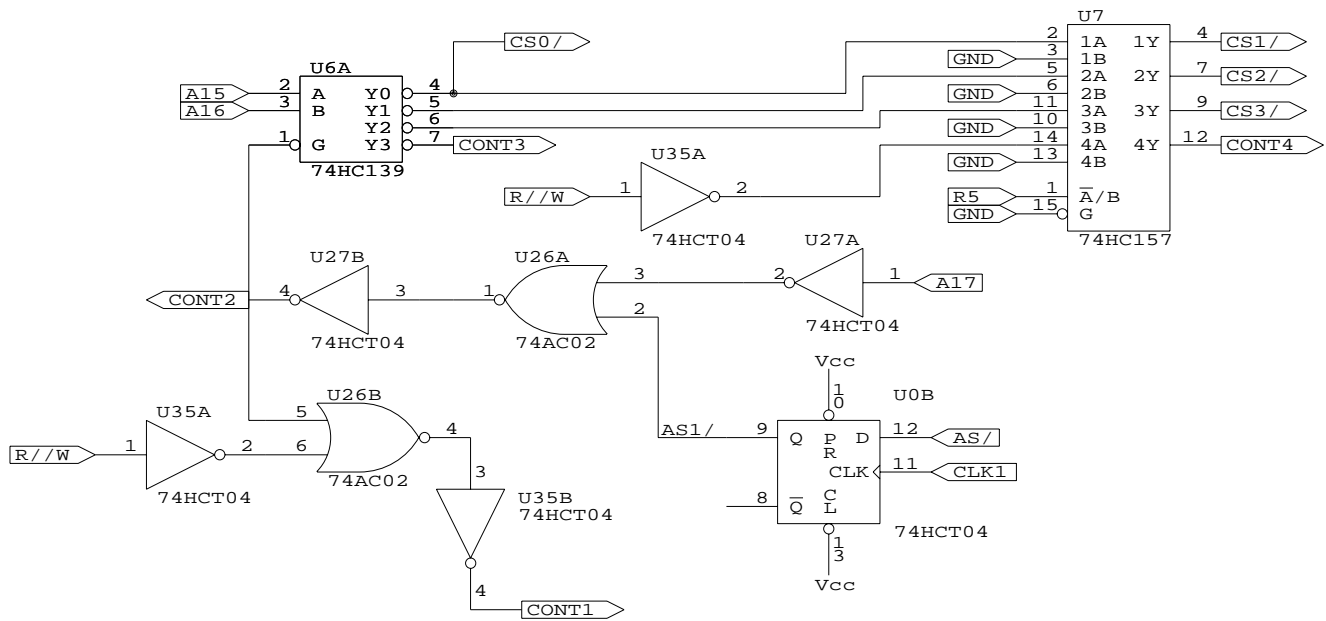


FIG. E.4 : Génération des signaux de contrôle (1/2)

CARTE UTAK		
Size	Document Number	REV
A	SIGNAUX DE CONTROLE (1/2)	
Date:	April 4, 1997	Sheet 4 of 7

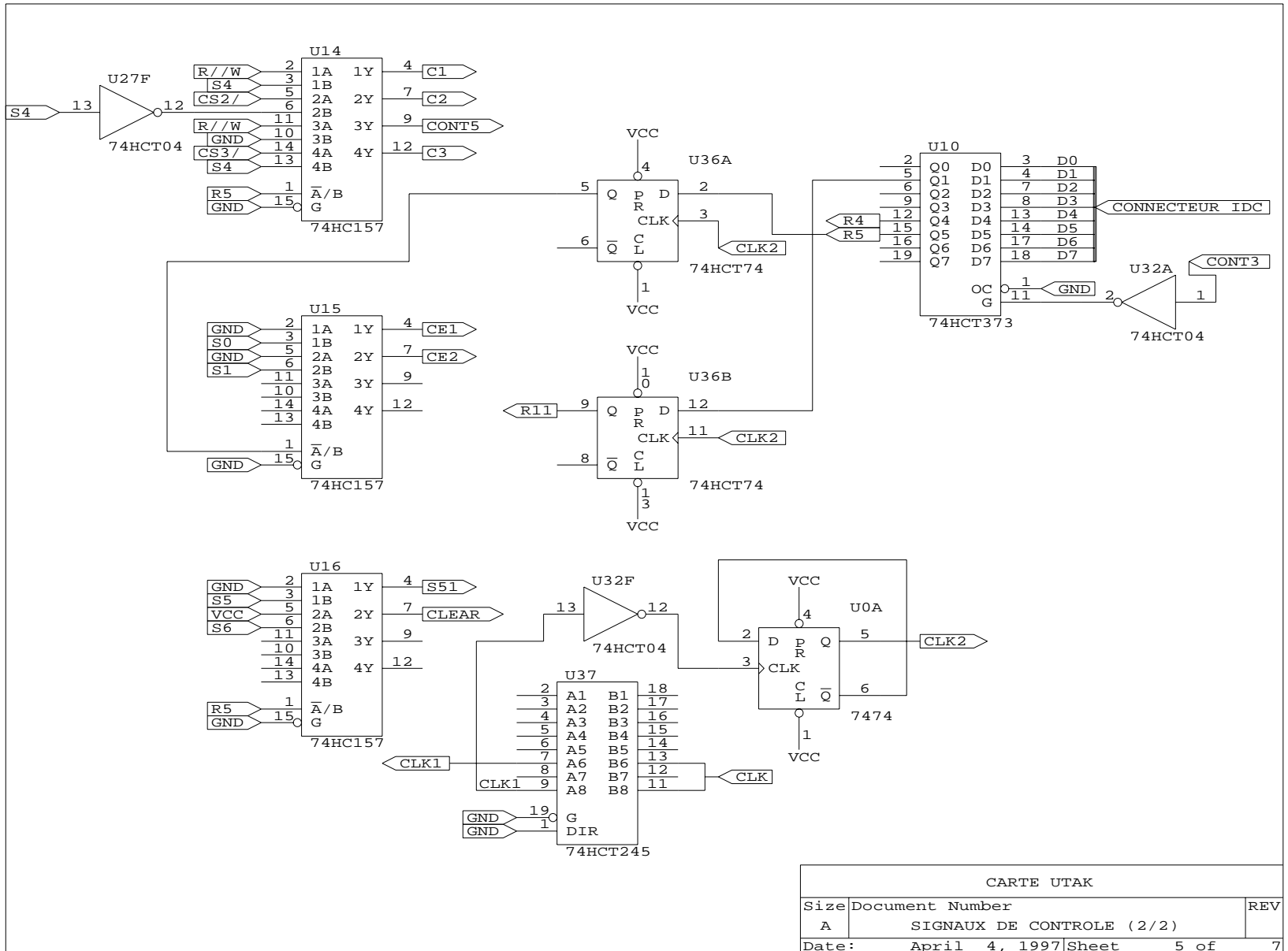
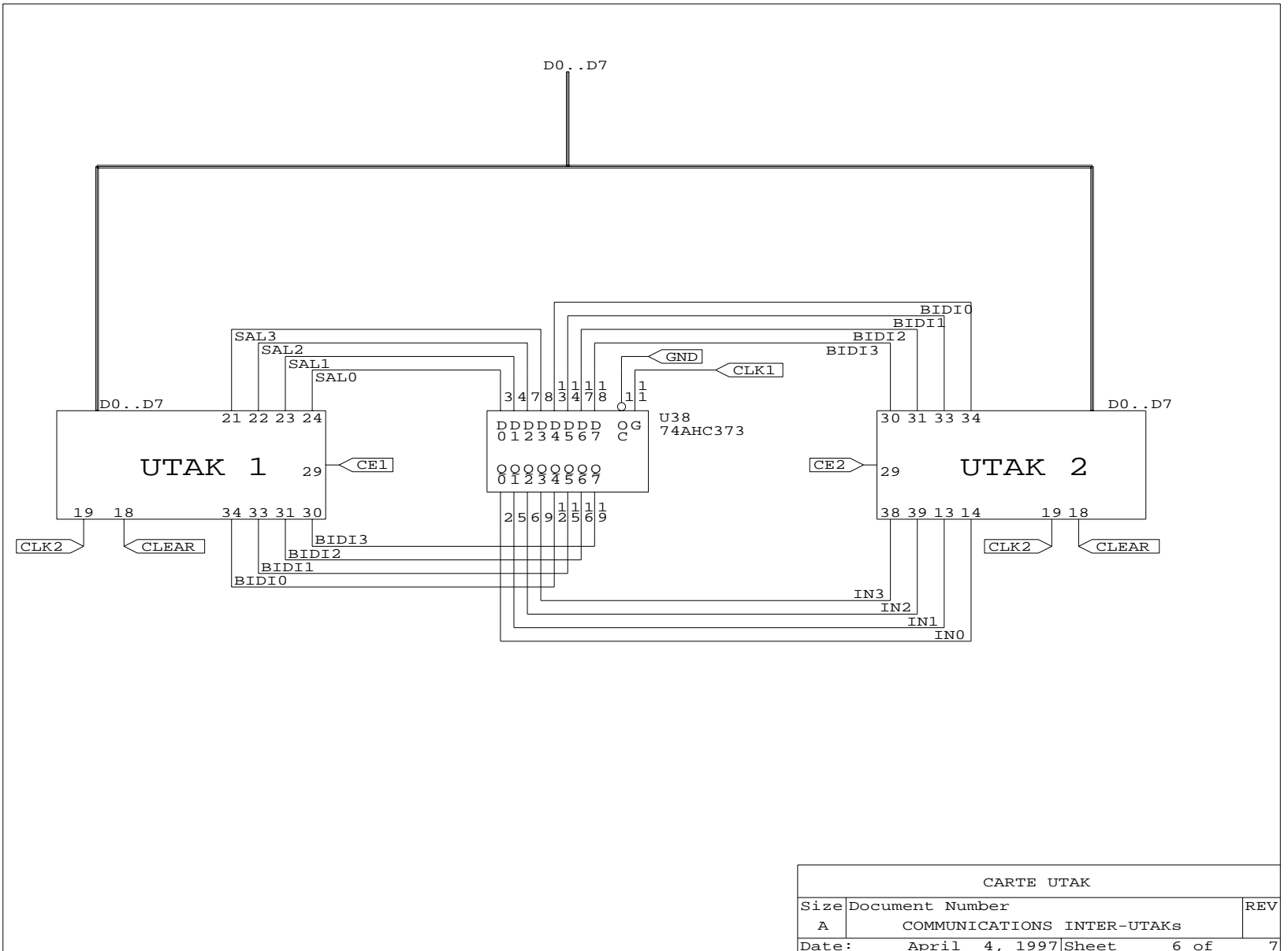


FIG. E.5 : Génération des signaux de contrôle (2/2)

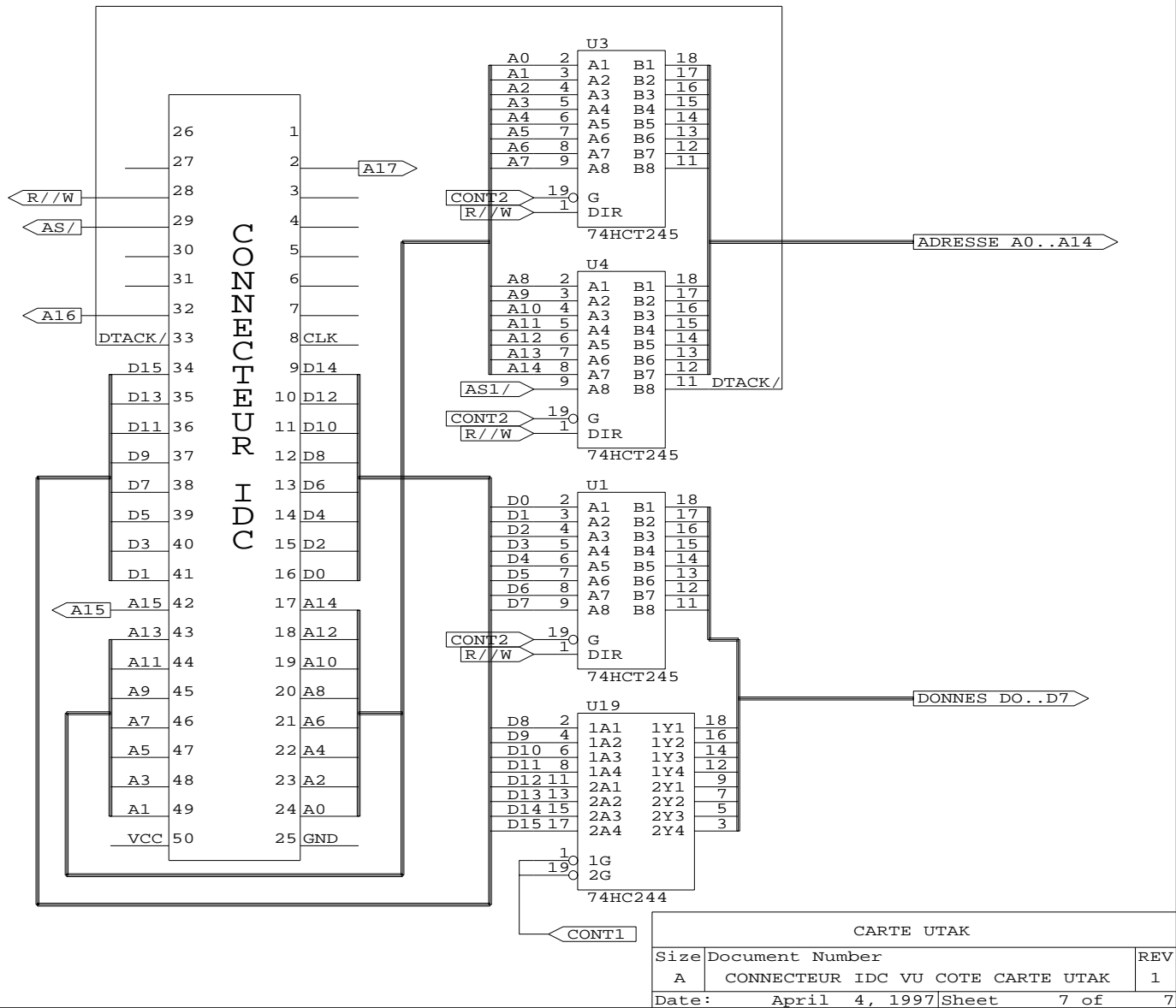
CARTE UTAK		
Size	Document Number	REV
A	SIGNAUX DE CONTROLE (2/2)	
Date:	April 4, 1997	Sheet 5 of 7

FIG. E.6 : Communication inter-UTAK



CARTE UTAK		
Size	Document Number	REV
A	COMMUNICATIONS INTER-UTAKs	
Date:	April 4, 1997	Sheet 6 of 7

FIG. E.7 : Connecteur IDC vu côté carte UTAK



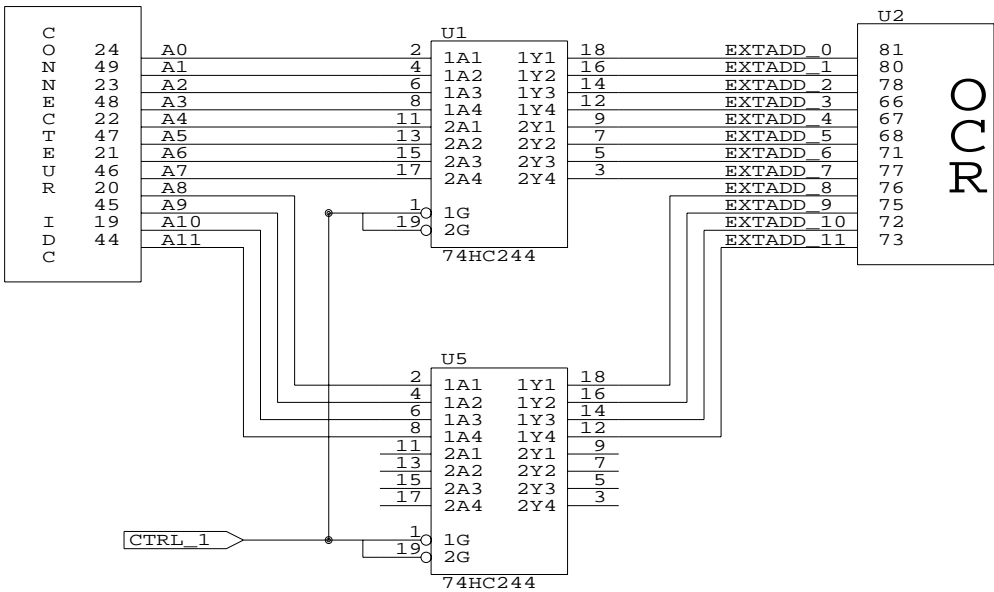


FIG. F.1 : Buffers d'adresses 68000 - RAMOCR

CARTE OCR		
Size	Document Number	REV
A	BUFFERS D'ADRESSES 68000 / RAMOCR	1
Date:	April 4, 1997	Sheet 1 of 12

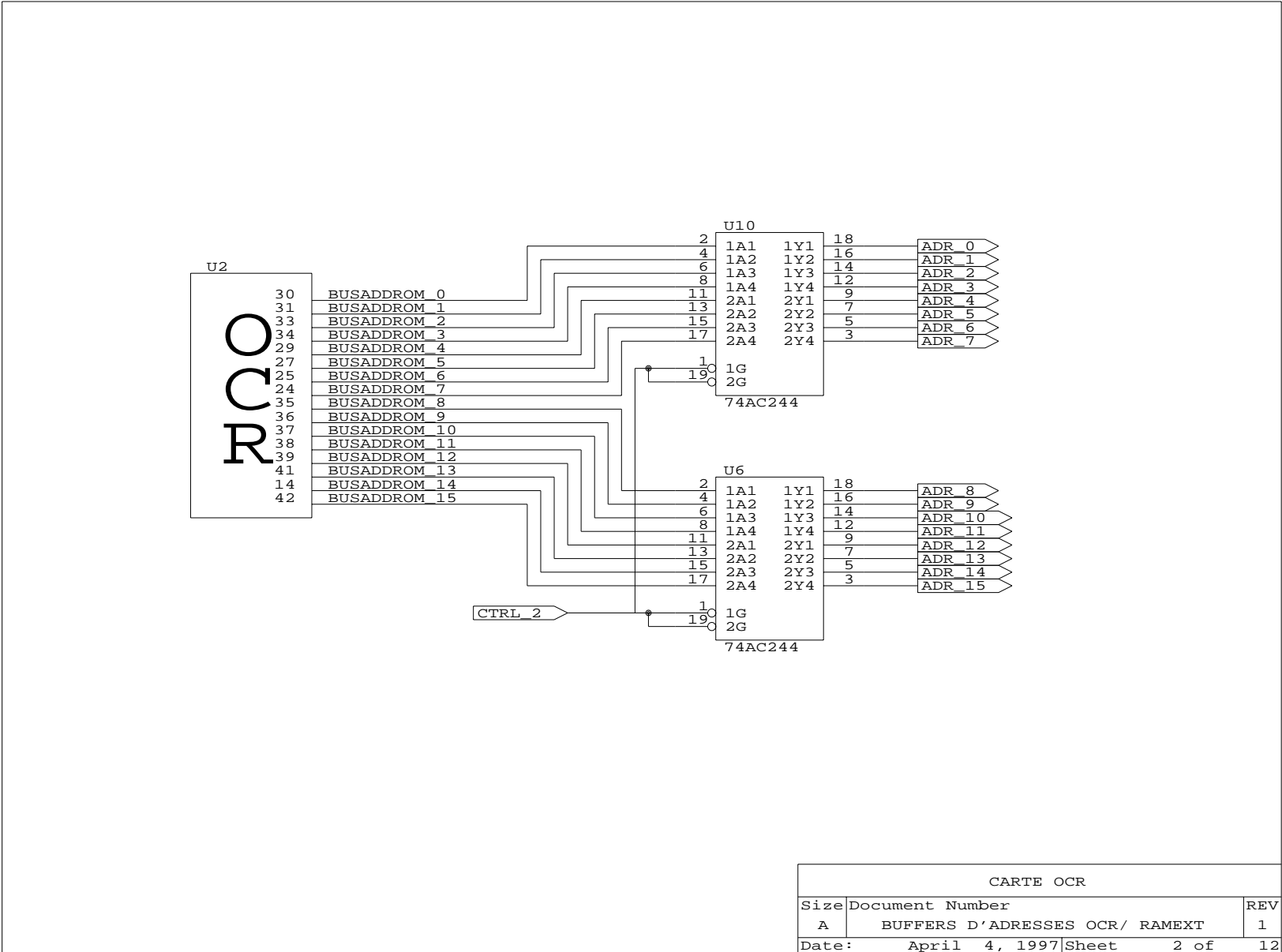


FIG. F.2 : Buffers d'adresses OCR - RAM externes

CARTE OCR		
Size	Document Number	REV
A	BUFFERS D'ADRESSES OCR/ RAMEXT	1
Date:	April 4, 1997	Sheet 2 of 12

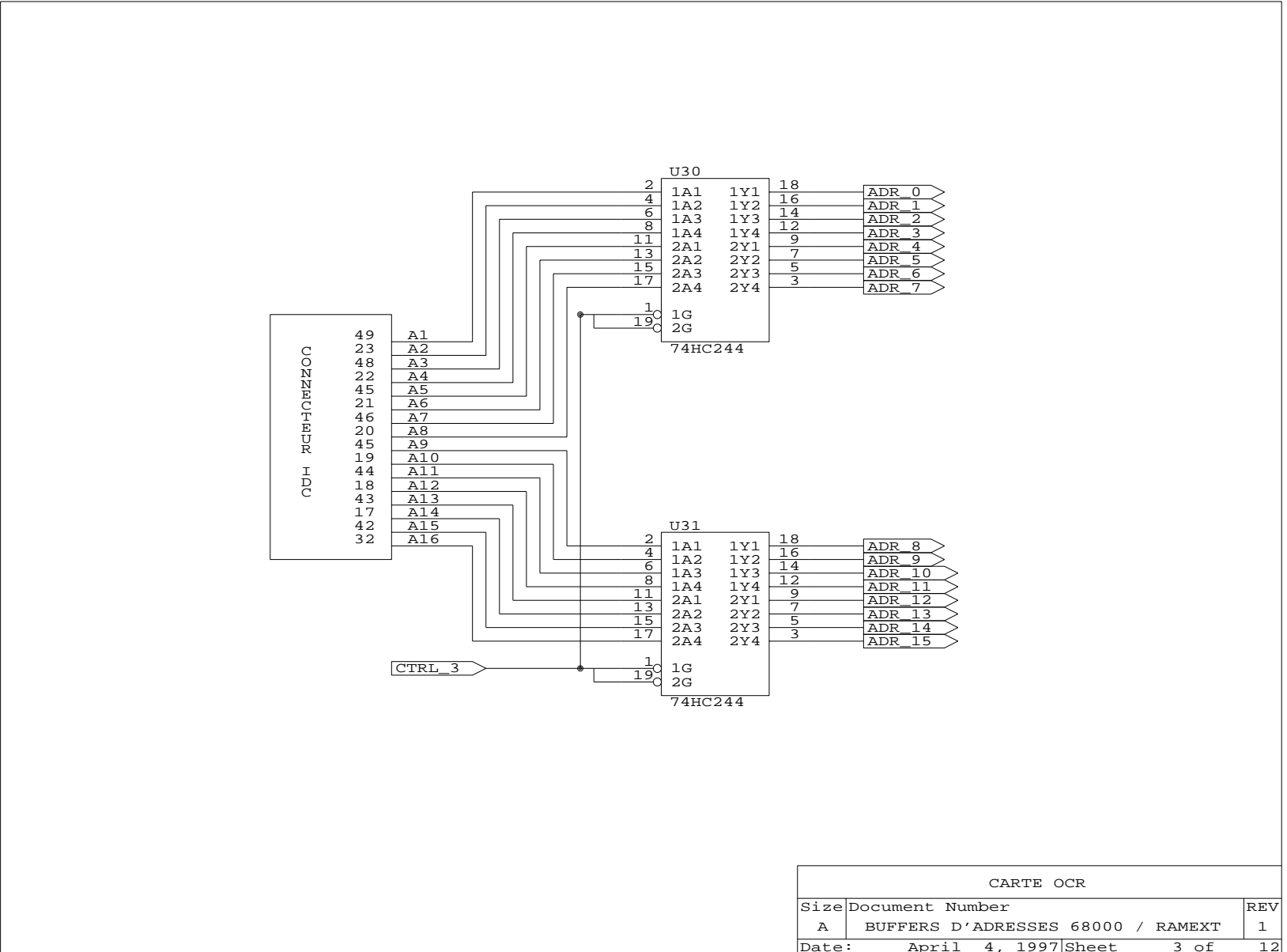
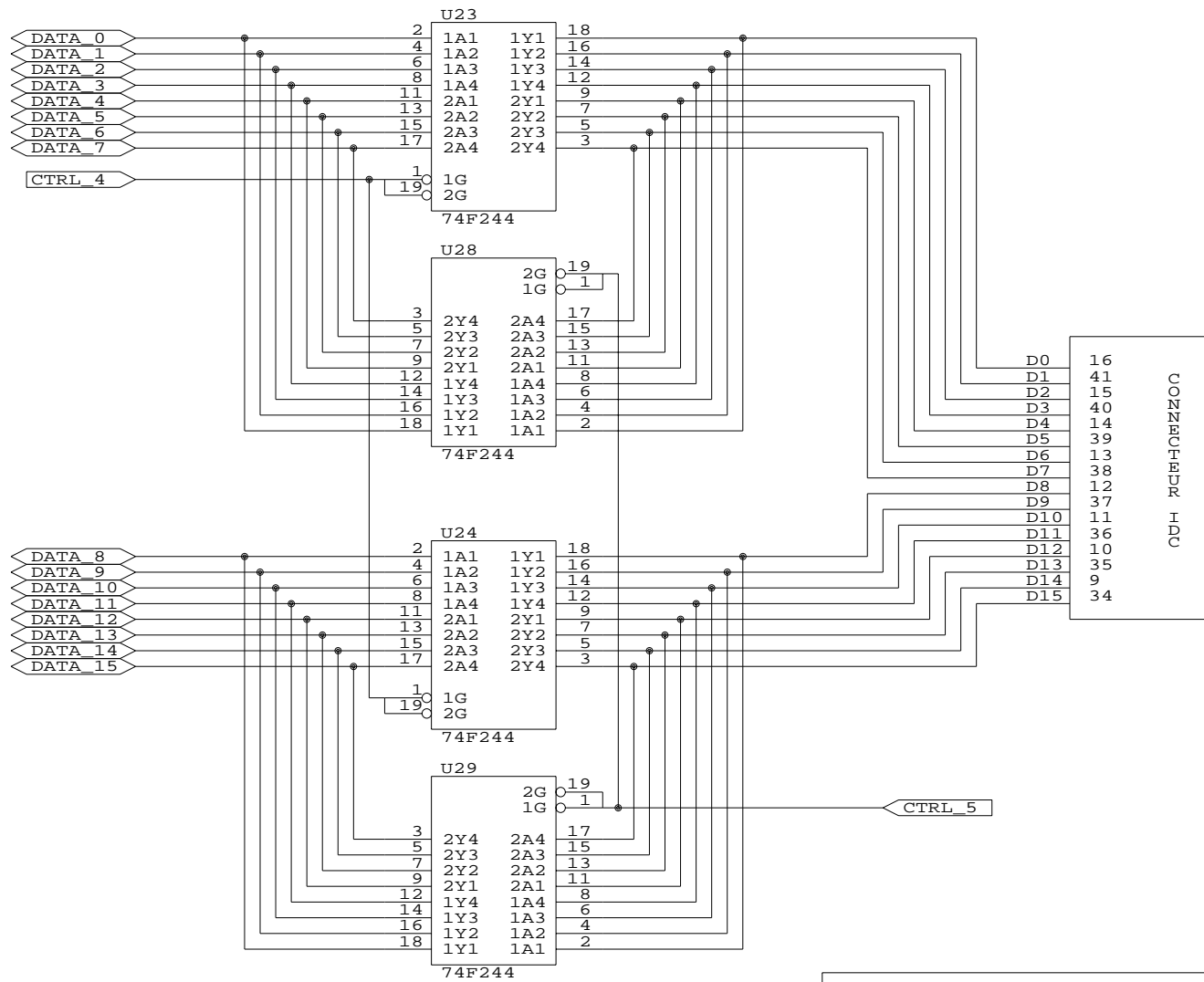


FIG. F.3 : Buffers d'adresses 68000 - RAM externes

CARTE OCR		
Size	Document Number	REV
A	BUFFERS D'ADRESSES 68000 / RAMEXT	1
Date:	April 4, 1997	Sheet 3 of 12

FIG. F.4 : Buffers de données 68000 – RAM externes



CARTE OCR		
Size	Document Number	REV
A	BUFFERS DE DONNEES 68000 / RAMEXT	1
Date:	April 4, 1997	Sheet 4 of 12

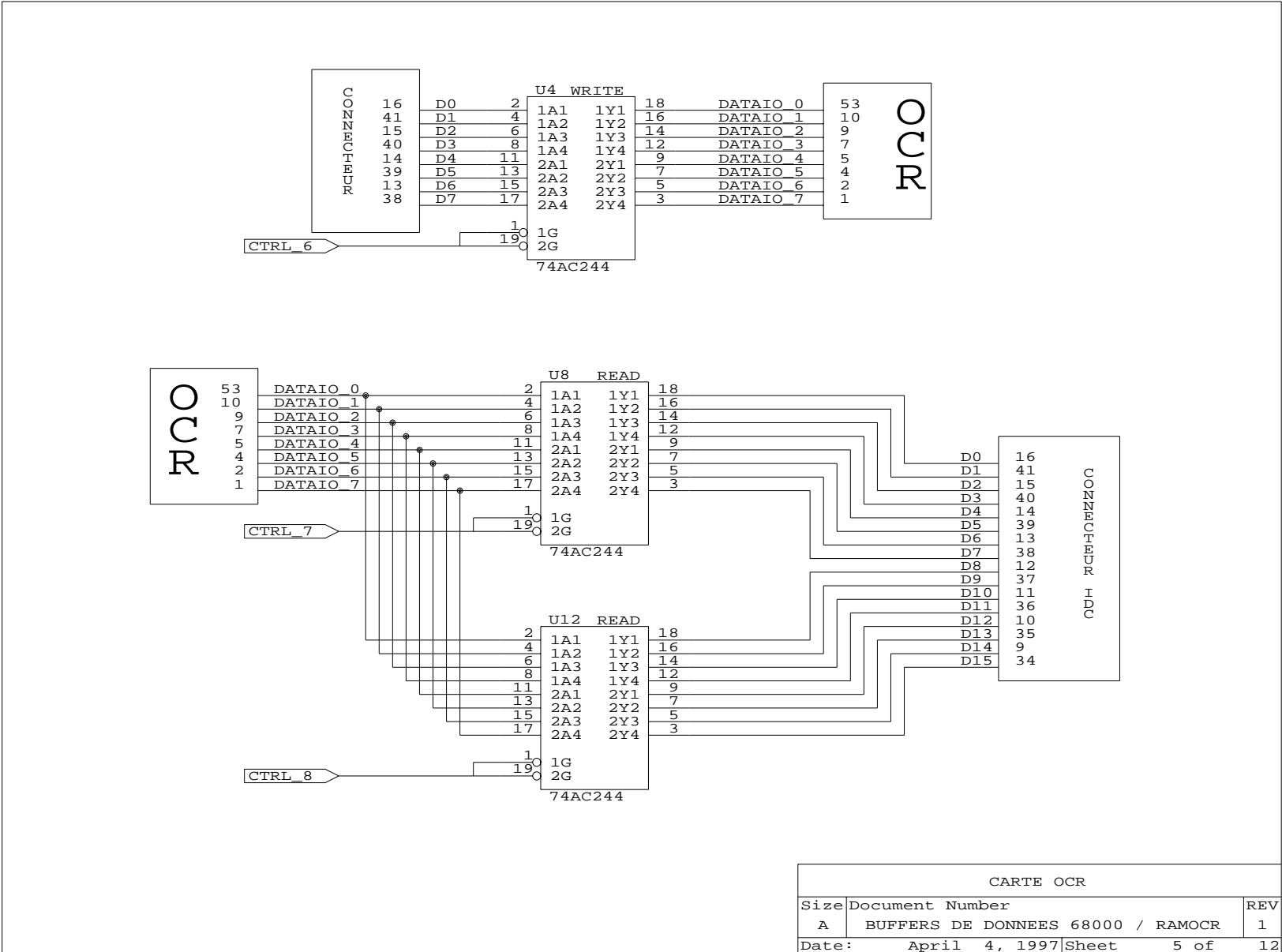
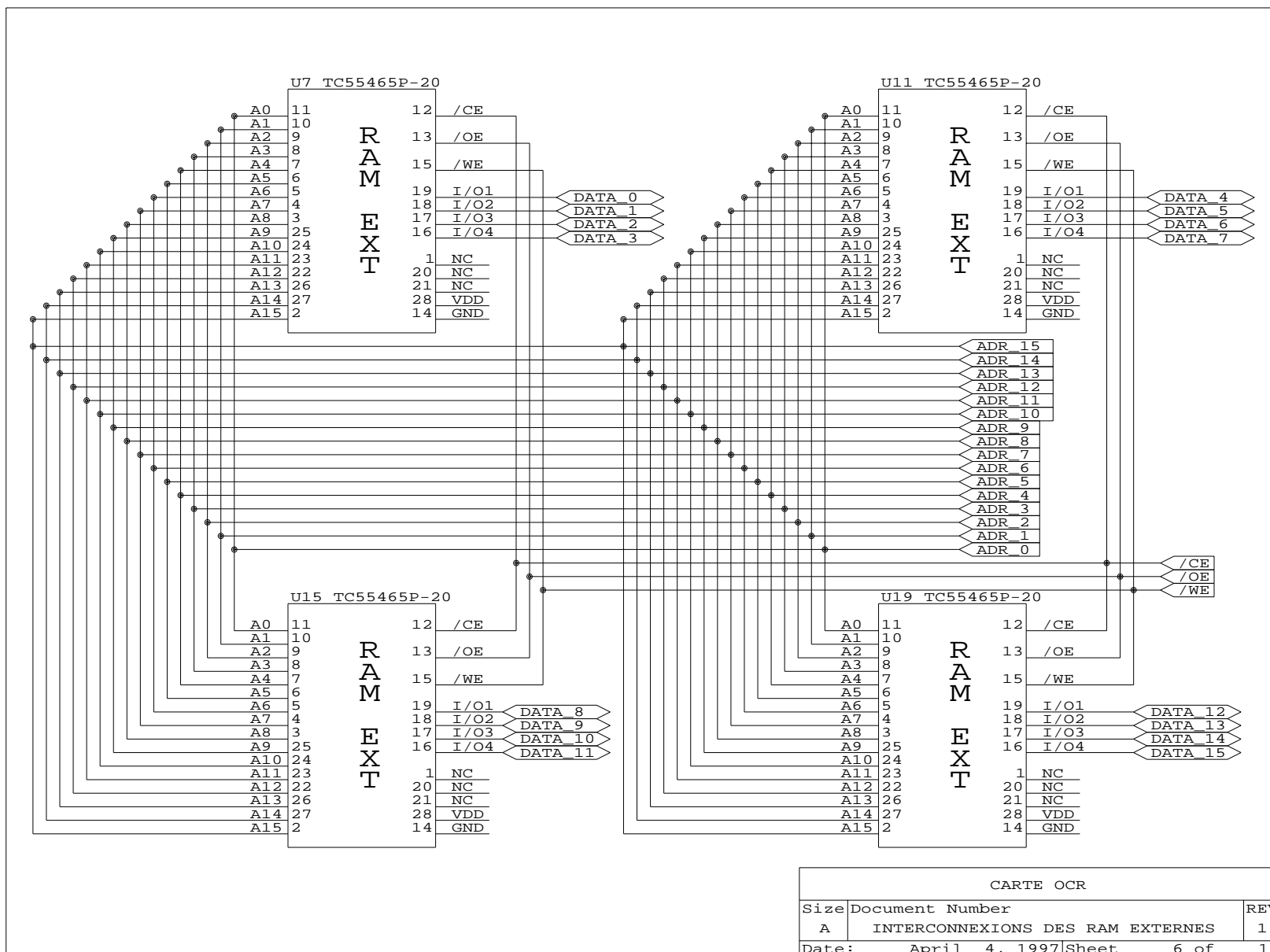


FIG. F.5 : Buffers de données 68000 – RAMOCR

CARTE OCR		
Size	Document Number	REV
A	BUFFERS DE DONNEES 68000 / RAMOCR	1
Date:	April 4, 1997	Sheet 5 of 12

FIG. F.6 : Interconnexions des RAM externes



CARTE OCR		
Size	Document Number	REV
A	INTERCONNEXIONS DES RAM EXTERNES	1
Date:	April 4, 1997	Sheet 6 of 12

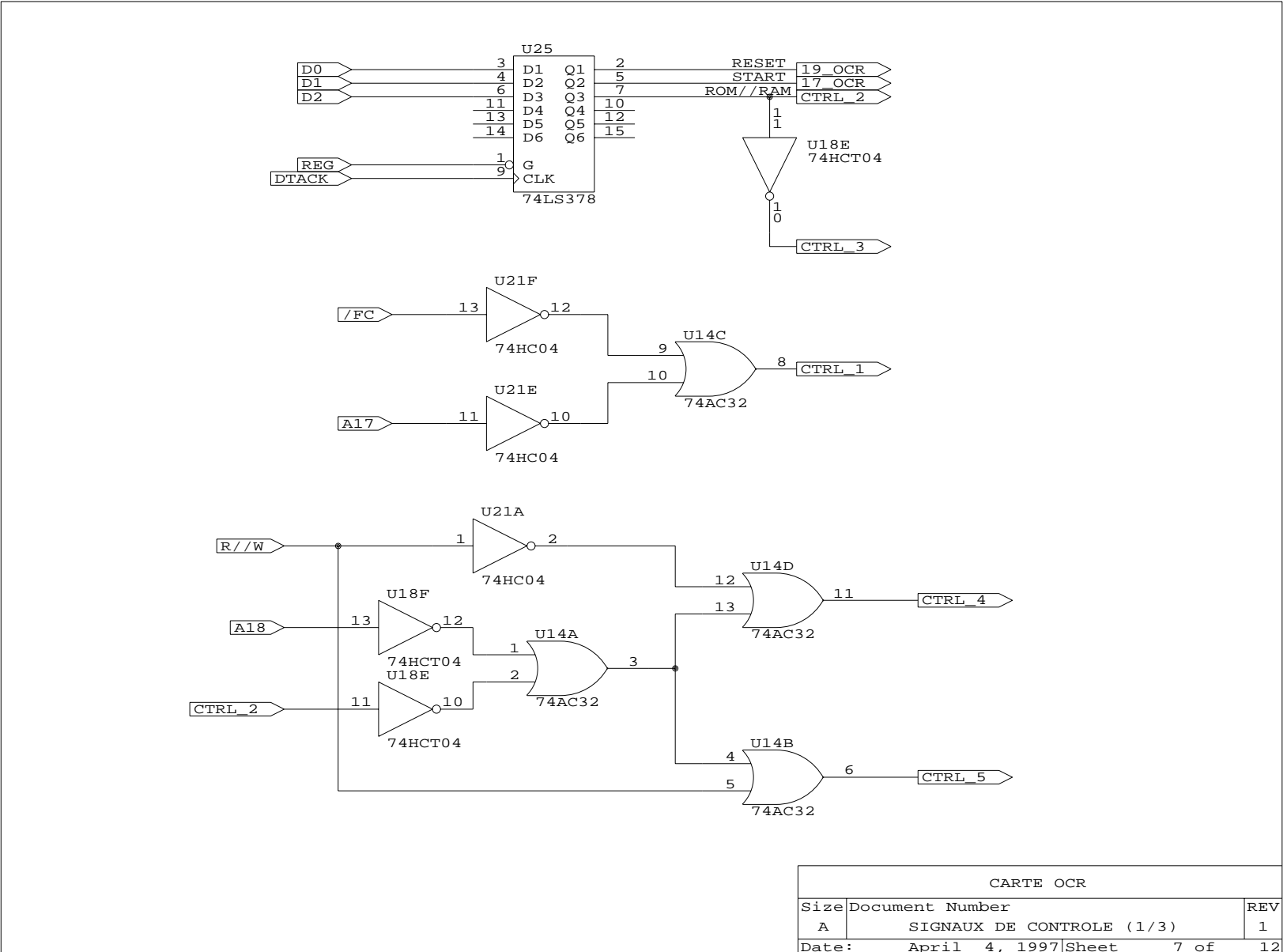
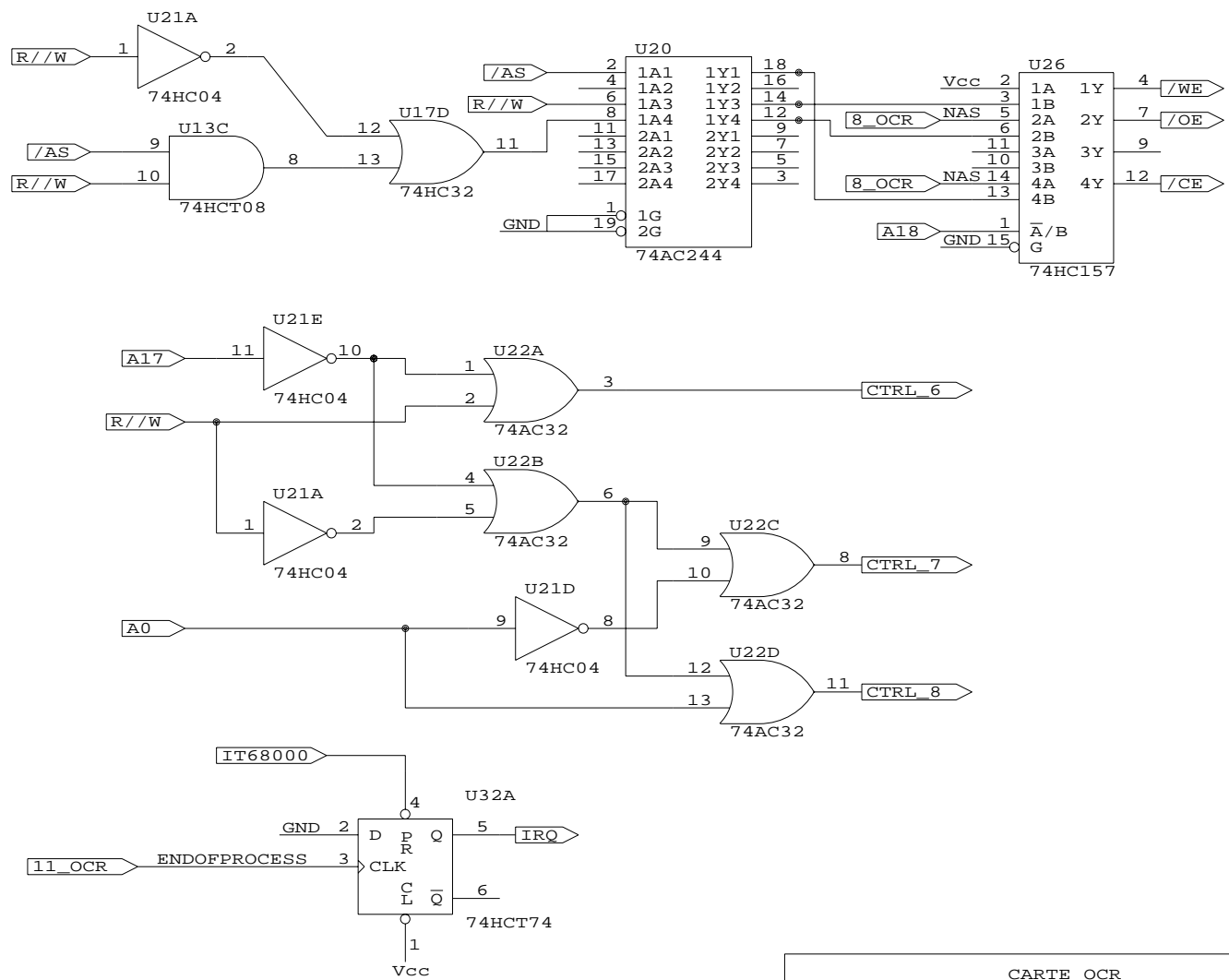


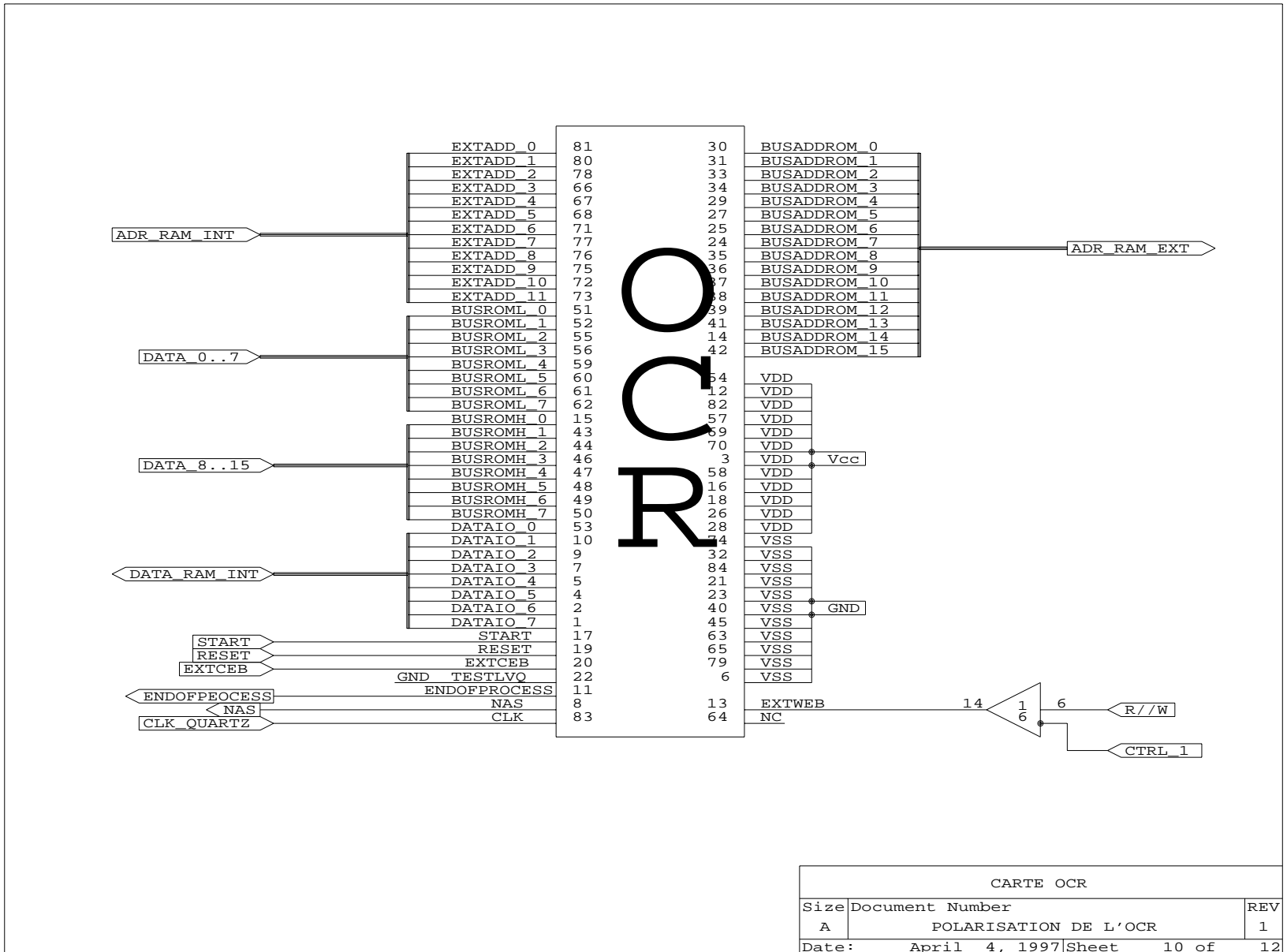
FIG. F.7 : Génération des signaux de contrôle (1/3)

FIG. F.8 : Génération des signaux de contrôle (2/3)



CARTE OCR		
Size	Document Number	REV
A	SIGNAUX DE CONTROLE (2/3)	2
Date:	April 4, 1997	Sheet 8 of 12

FIG. F.10 : Polarisation du circuit OCR



CARTE OCR		
Size	Document Number	REV
A	POLARISATION DE L'OCR	1
Date:	April 4, 1997	Sheet 10 of 12

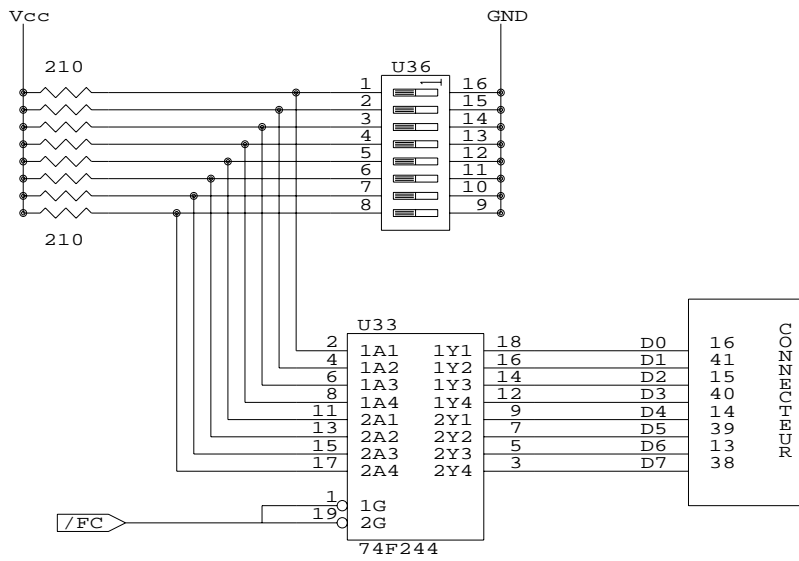


FIG. F.11 : Génération de l'interruption

CARTE OCR		
Size	Document Number	REV
A	GENERATION DE L'INTERRUPTION	1
Date:	April 4, 1997	Sheet 11 of 12

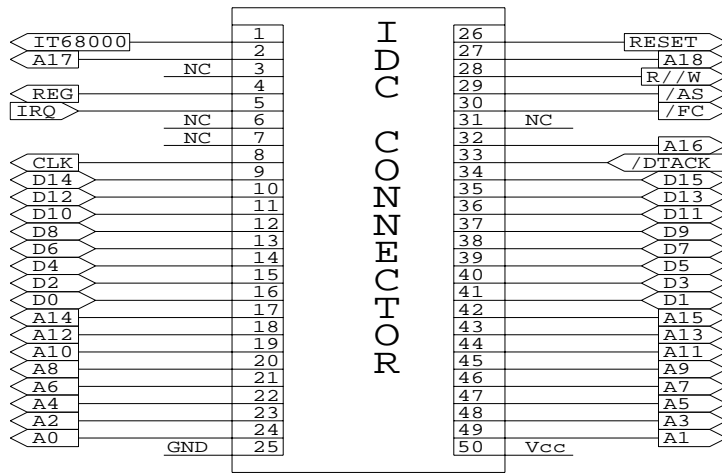


FIG. F.12 : Connecteur IDC vu côté carte OCR

CARTE OCR		
Size	Document Number	REV
A	CONNECTEUR IDC VU COTE CARTE OCR	1
Date:	April 4, 1997	Sheet 12 of 12

Annexe G

Schémas de la carte T225 – L-Neuro 1.0

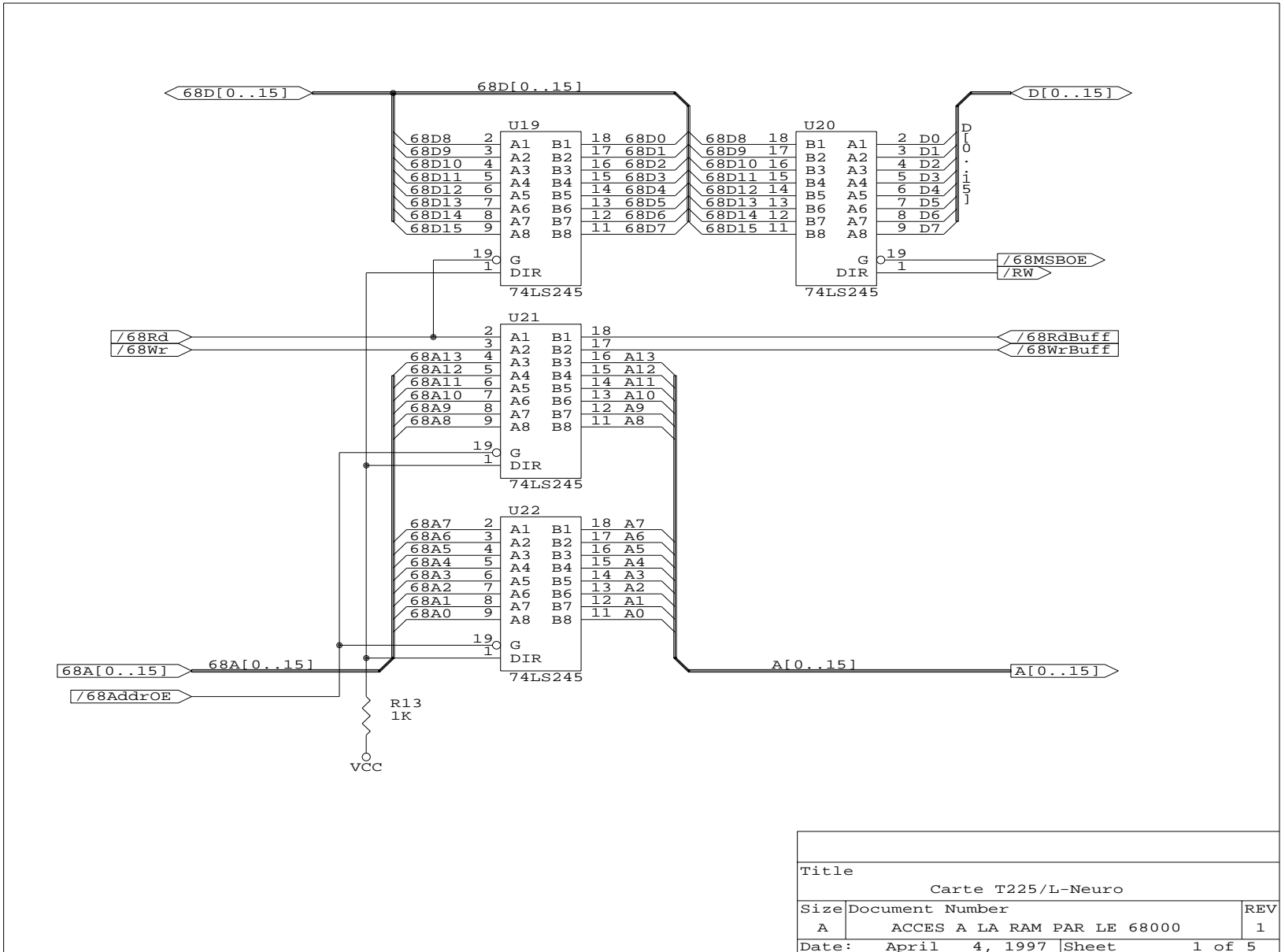
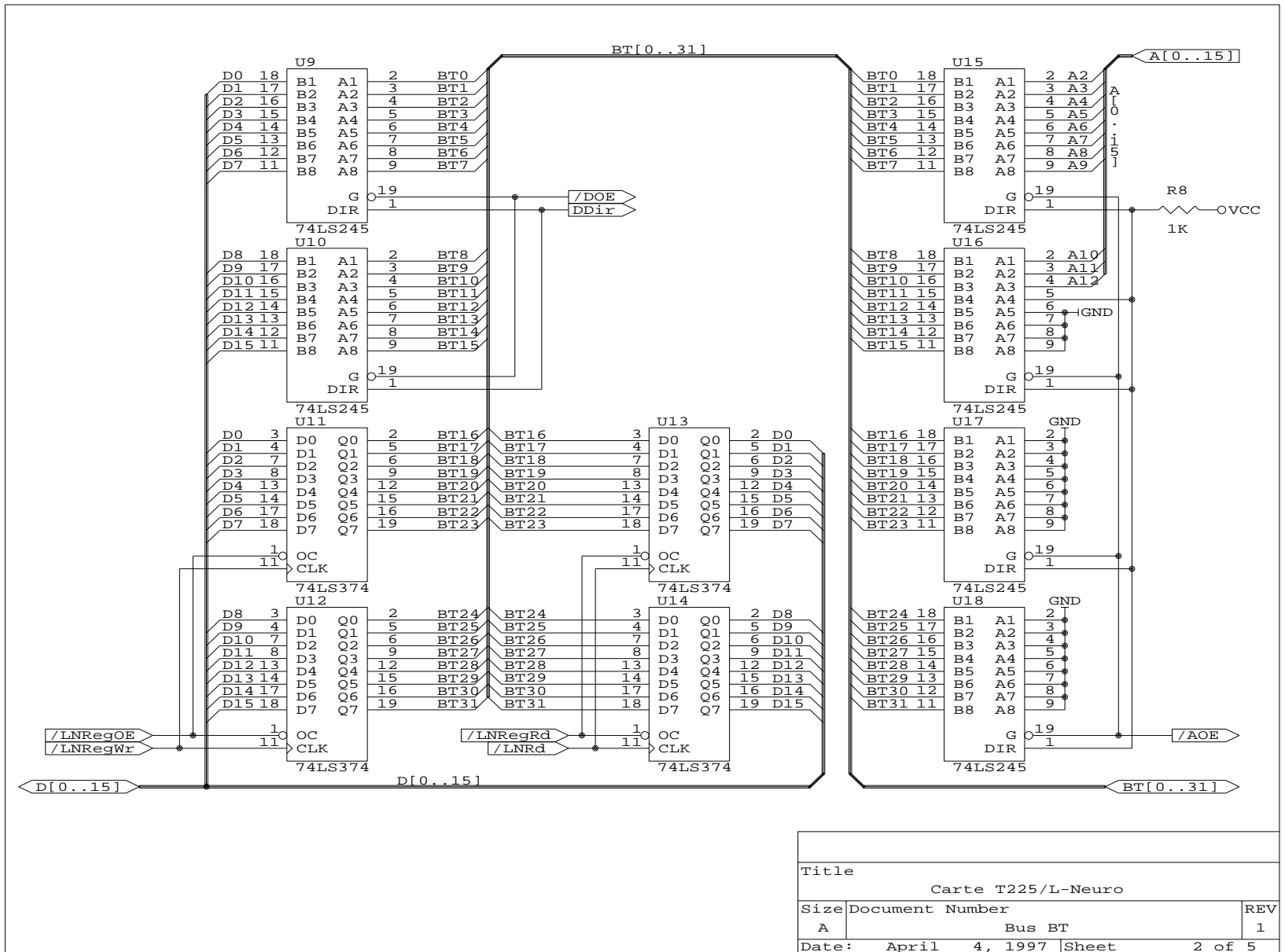


FIG. G.1 : Interface 68000 - RAM externes

Title		
Carte T225/L-Neuro		
Size	Document Number	REV
A	ACCES A LA RAM PAR LE 68000	1
Date:	April 4, 1997	Sheet 1 of 5

FIG. G.2 : Interface T225 - L-Neuro 1.0



Title		
Carte T225/L-Neuro		
Size	Document Number	REV
A	Bus BT	1
Date:	April 4, 1997	Sheet 2 of 5

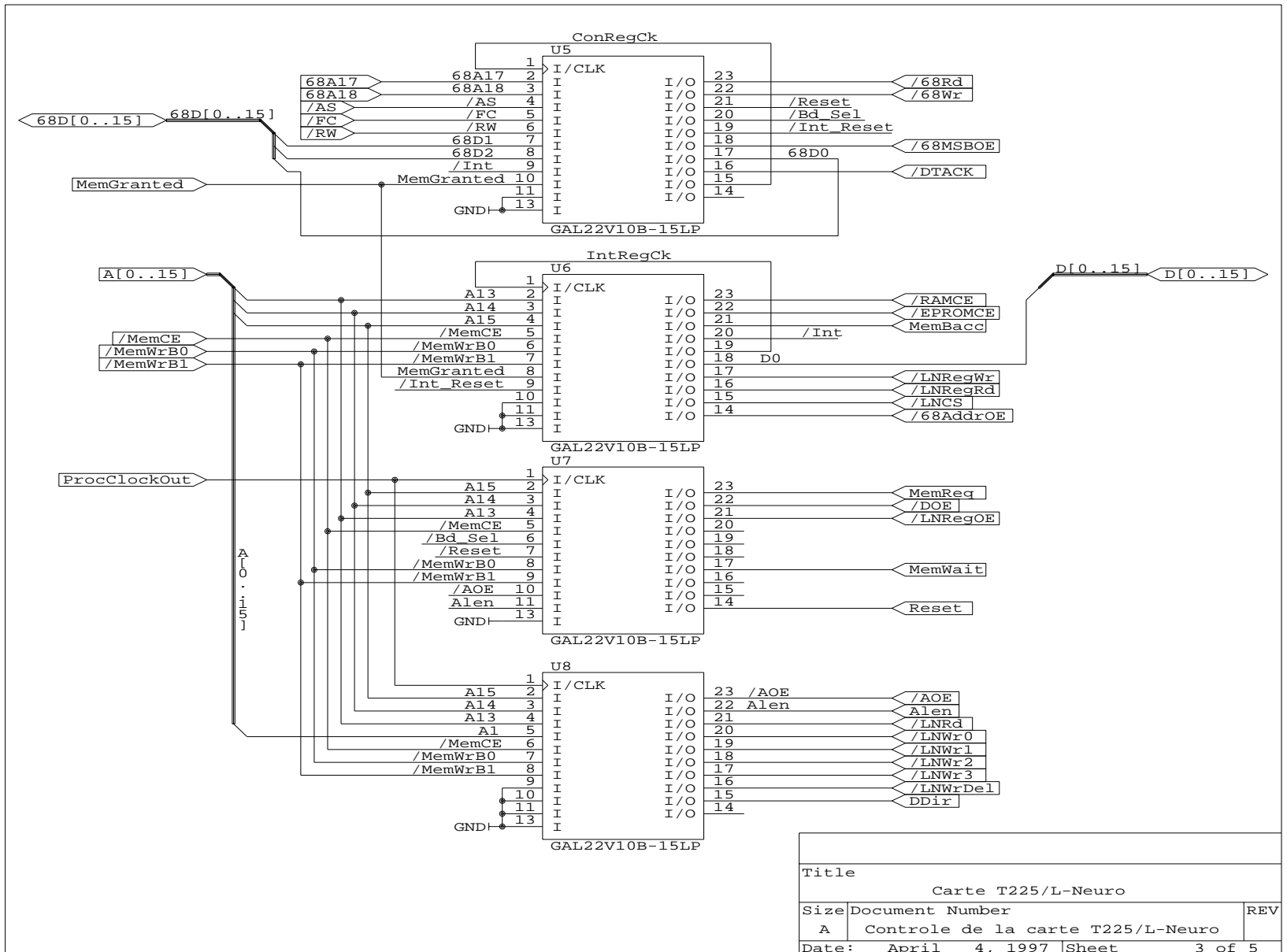


FIG. G.3 : Génération des signaux de contrôle

Title		
Carte T225/L-Neuro		
Size	Document Number	REV
A	Contrôle de la carte T225/L-Neuro	
Date:	April 4, 1997	Sheet 3 of 5

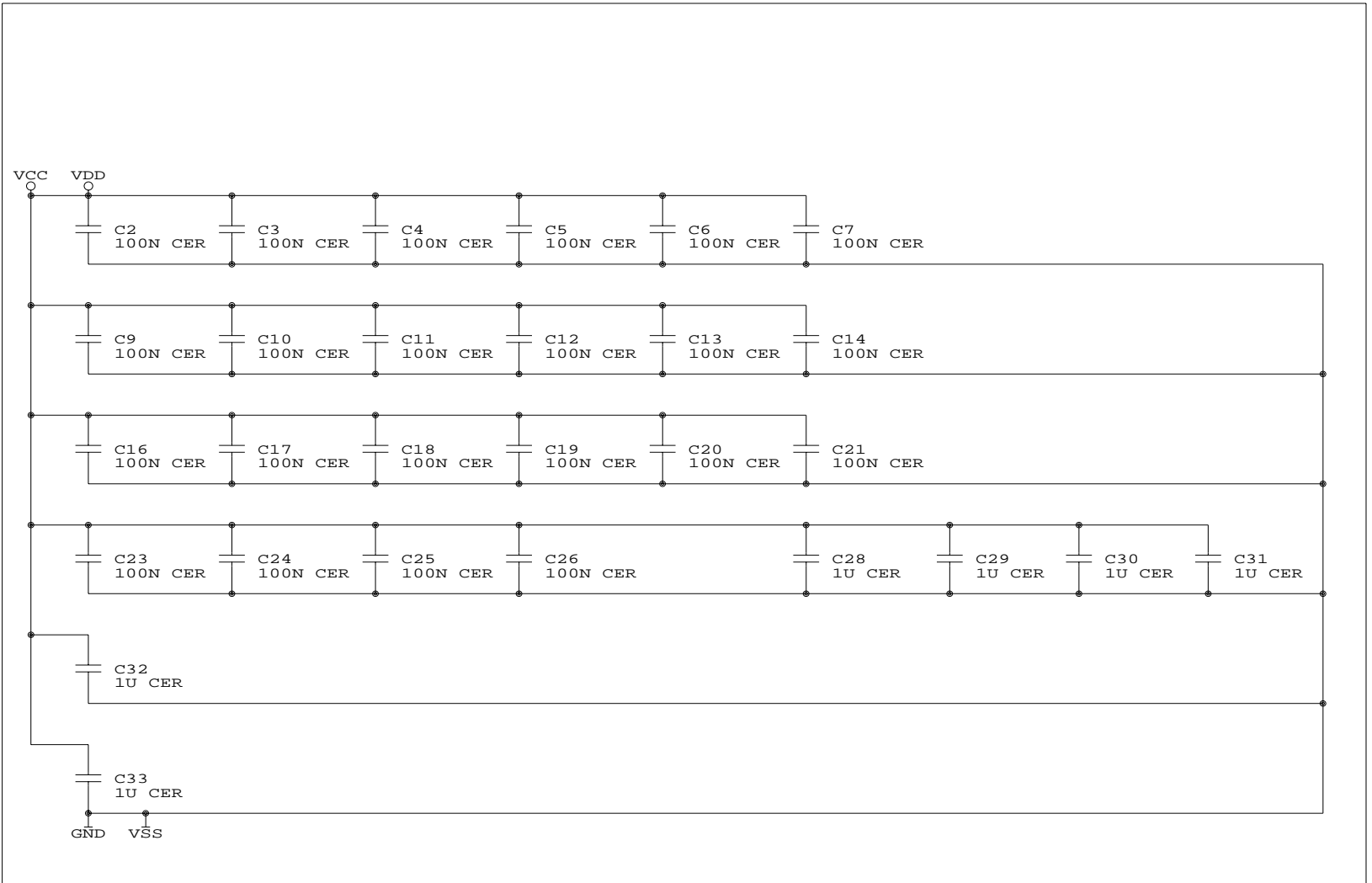
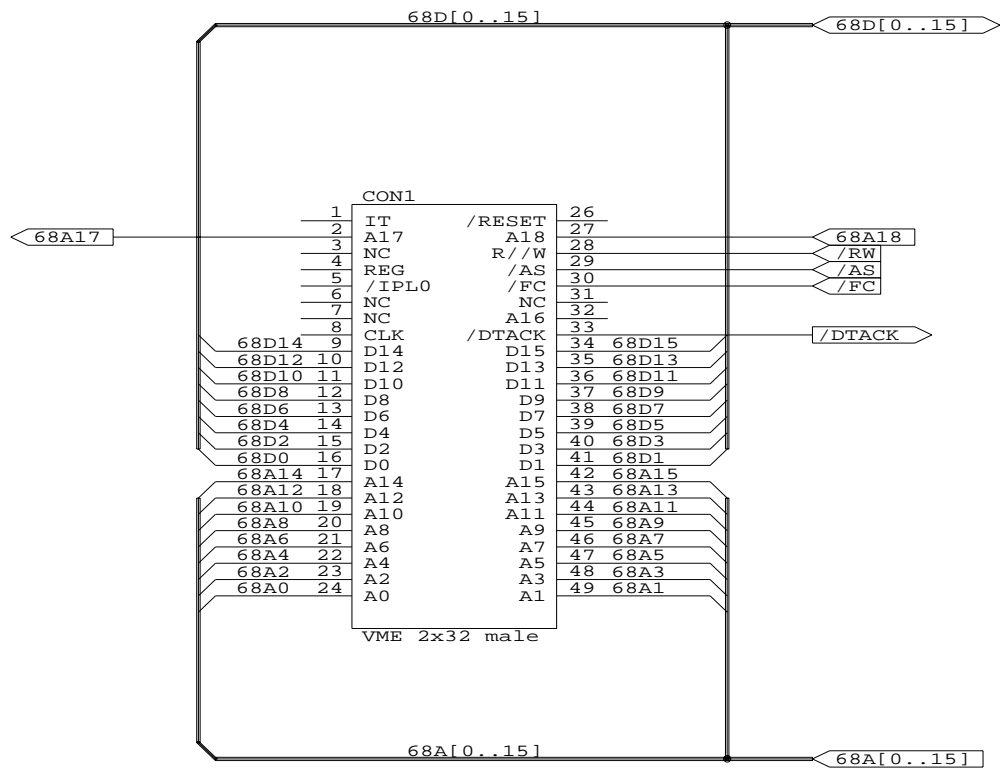


FIG. G.4 : Capacités de découplage de l'alimentation

Title		
Carte T225/L-Neuro		
Size	Document Number	REV
A	Decouplage de l'alimentation	
Date:	April 4, 1997	Sheet 4 of 5

FIG. G.5 : Connecteur IDC vu côté carte T225 - L-Neuro 1.0



Title		
Carte T225/L-Neuro		
Size	Document Number	REV
A	Connecteur avec la carte 68000	
Date:	April 4, 1997	Sheet 5 of 5

Annexe H

Glossaire des abréviations utilisées

ANN (*Artificial Neural Networks*) : Réseaux de neurones artificiels, des systèmes parallèles et adaptatifs de traitement de l'information inspirés par le fonctionnement du cerveau

BP (*Back Propagation*) : Algorithme d'apprentissage des réseaux de neurones basé sur la retropropagation de l'erreur

CAM (*Content Addressable Memory*) : Mémoire adressable par le contenu

CISC (*Complex Instruction Set Computer*) : Processeur à jeu d'instructions complexe

CMOS (*Complementary Metal Oxide Silicium*) : Technologie de fabrication de circuits intégrés utilisant des transistors CMOS

CPPS (*Connection Primitives Per Second*) : Unité de mesure de la performance d'un ANN tenant compte de la précision des données

CPS (*Connection Per Second*) : Unité de mesure de la performance d'un ANN durant la phase de rappel

CPSPW (*Connection Per Second Per Weight*) : Unité de mesure de la performance d'un ANN durant la phase de rappel, moyennée sur le nombre total de poids du réseau

CST (*Composant Sous Test*) : Composant cible sur lequel sont réalisées les différentes expériences

CUPS (*Connection Update Per Second*) : Unité de mesure de la performance d'un ANN durant la phase d'apprentissage

CUPSPW (*Connection Update Per Second Per Weight*) : Unité de mesure de la performance d'un ANN durant la phase d'apprentissage, moyennée sur le nombre total de poids du réseau

DSP (*Digital Signal Processor*) : Processeur de signaux

EBF (*Extrêmement Basses Fréquences*) : Fréquences inférieures à 1.5 *KHz*

LET (*Linear Energie Transfer*) : Pouvoir d'arrêt d'une particule (perte d'énergie par unité de longueur parcourue)

LSB (*Least Significant Bit*) : Bit du poids le plus faible dans un nombre binaire

LVQ (*Learning Vector Quantization*) : Algorithme d'apprentissage des réseaux de neurones

MLP (*Multi Layer Perceptron*) : Réseau de neurones organisé en couches

MSB (*Most Significant Bit*) : Bit du poids le plus fort dans un nombre binaire

OCR (*Optical Character Recognition*) : Reconnaissance de caractères

RISC (*Reduced Instruction Set Computer*) : Processeur à jeu d'instructions réduit

SEU (*Single Event Upset*) : Phénomène d'aléa logique (upset) pouvant affecter les composants électroniques soumis aux radiations

SIMD (*Single Instruction stream Multiple Data stream*) : Architecture d'ordinateur

dans laquelle les unités de traitements exécutent de façon parallèle la même instruction sur des données différentes

TBF (*Très Basses Fréquences*) : Fréquences situées entre 1 et 40 KHz

TDN (*Time Delay Neurone*) : Neurone à retard permettant de suivre l'évolution temporelle du signal d'entrée

TDNN (*Time Delay Neural Network*) : Réseau de neurones constitué de TDN

Table des matières

1	Effets de l'environnement spatial sur les circuits intégrés	11
1.1	Environnement spatial et électronique embarquée	11
1.2	Effets du rayonnement sur la matière	12
1.2.1	Effets de la dose cumulée	13
1.2.2	Effets singuliers induits par les ions lourds	13
1.2.3	Pouvoir d'arrêt, notion de LET	13
1.2.4	Le phénomène SEU	14
1.2.5	Notion de LET seuil et de section efficace	15
1.3	Moyens d'actions face aux effets des particules chargées	16
1.3.1	Moyens de prévention	17
1.3.2	Moyens de prévision	18
1.4	Mise en œuvre de test aux ions lourds	18
1.4.1	Moyens de simulation des SEU	19
1.4.2	Système de test	19
1.4.2.1	Fonctions principales	19
1.4.2.2	Architecture adoptée	19
1.4.3	Test d'upset	20
1.4.3.1	Test statique	20
1.4.3.2	Test dynamique	20
1.4.3.3	Entre le test statique et le test dynamique?	21
1.5	Environnement de test et approches adoptées	21
2	Réseaux de neurones : principes, implantations et applications	23
2.1	Introduction	23
2.2	Bref historique	24
2.3	Modèle de base du neurone : <i>le neurone formel</i>	25

2.4	Architectures d'un réseau de neurones	26
2.4.1	Les réseaux proactifs	27
2.4.1.1	Réseaux proactifs monocouches	28
2.4.1.2	Réseaux proactifs multicouches	28
2.4.2	Réseaux récurrents	30
2.5	Apprentissage d'un réseau de neurones	31
2.6	Modes de fonctionnement d'un ANN	32
2.7	Exemple d'un réseau de neurones	33
2.8	Implantations des réseaux de neurones	34
2.8.1	Réseaux de neurones : du logiciel au matériel	34
2.8.2	Caractéristiques des réseaux de neurones matériels	38
2.8.2.1	Spécifications et critères de performance	38
2.8.2.2	Principales considérations de conception	39
2.8.3	Réseaux de neurones en VLSI	40
2.8.3.1	Implantations digitales	41
2.8.3.1.1	Architectures en tranches	41
2.8.3.1.2	Architectures multiprocesseurs	42
2.8.3.1.3	Autres implantations digitales	42
2.8.3.2	Implantations analogiques	42
2.8.3.3	Implantations hybrides	43
2.8.3.4	Implantations optoélectroniques	45
2.8.4	Cartes accélératrices de réseaux de neurones	46
2.8.4.1	Les systèmes utilisant les processeurs standards	47
2.8.4.2	Les systèmes utilisant les processeurs dédiés	47
2.9	Applications des réseaux de neurones	47
2.9.1	Des raisons d'utiliser les réseaux de neurones artificiels	48
2.9.1.1	Le parallélisme	48
2.9.1.2	La capacité d'adaptation	48
2.9.1.3	La capacité de généralisation	49
2.9.1.4	La mémoire distribuée	49
2.9.1.5	La facilité de construction	50
2.9.2	Des caractéristiques d'une bonne application	50
2.9.3	Revue des différentes applications	51
2.9.3.1	Secteur industriel	52

2.9.3.2	Secteur financier	53
2.9.3.3	Secteur des télécommunications	53
2.9.3.4	Secteurs maritime et océanique	54
2.9.3.5	Secteur médical	54
2.9.3.6	Secteur de l'environnement	55
2.9.3.7	Secteur spatial	55
3	Tolérance aux fautes des réseaux de neurones artificiels : état de l'art	57
3.1	Introduction	57
3.2	Terminologie de fiabilité	59
3.3	Domaines d'étude de la tolérance aux fautes	59
3.4	Considérations sur les modèles de fautes	60
3.5	Exemple d'illustration de la tolérance aux fautes des ANN	61
3.6	Facteurs liés à la tolérance aux fautes des ANN	62
3.6.1	La redondance	63
3.6.2	Nature distribuée de l'information et des traitements	63
3.6.3	Format de données	63
3.6.4	Présence des rétroactions	64
3.6.5	Notion d'éléments de traitement	64
3.7	Etat de l'art de tolérance aux fautes des ANN	64
3.7.1	Tolérance durant la phase de rappel	64
3.7.1.1	Tolérance aux fautes survenant sur les stimuli d'entrée	65
3.7.1.2	Tolérance aux fautes survenant sur les poids synaptiques	65
3.7.2	Tolérance durant la phase d'apprentissage	65
3.7.2.1	Réapprentissage	66
3.7.2.2	Apprentissage avec restriction de poids	66
3.7.2.3	Apprentissage avec injection de fautes artificielles	67
3.7.2.4	Procédure de rajout/suppression	68
3.7.2.4.1	Elimination des nœuds non importants	68
3.7.2.4.2	Réapprentissage du réseau réduit	68
3.7.2.4.3	Rajout de nœuds redondants	69
3.7.2.5	Comparaison entre quelques techniques d'amélioration	69
3.8	Discussion	69

4	Méthodologies de test, circuits et réseaux de neurones choisis	71
4.1	Stratégies de test	71
4.1.1	Simulations logicielles d'upsets	71
4.1.2	Injection matérielle d'upsets	72
4.1.3	Test aux ions lourds	72
4.2	Extrapolation de la section efficace dynamique	73
4.3	Schéma bloc d'un réseau de neurones matériel	75
4.4	Mise en œuvre de tests des ANN matériels	76
4.5	Circuits cibles et zones sensibles correspondantes	78
4.5.1	Contexte	78
4.5.2	Le circuit UTAK	78
4.5.3	Le circuit L-Neuro 1.0	79
4.5.4	Le circuit OCR	79
4.5.5	Le processeur de signaux DSP96002	80
4.5.6	Le transputer T225	81
4.5.7	Zones sensibles aux SEU des circuits cibles	81
4.6	Choix des réseaux de neurones à tester	81
4.6.1	Réseau de classification de motifs	81
4.6.2	Réseaux d'évaluations des fonctions booléennes	82
4.6.3	Réseau de reconnaissance de caractères	83
4.6.4	Réseau de détection de sifflements	83
4.6.4.1	Description du problème	83
4.6.4.2	Choix de l'approche neuronale pour la détection des sifflements	85
4.6.4.3	Description du réseau EBF	86
4.7	Programmes de test	88
5	Résultats des simulations logicielles d'upsets	91
5.1	Introduction	91
5.2	Résultats obtenus pour le perceptron	92
5.2.1	Effets des perturbations sur les motifs d'entrée	92
5.2.2	Perturbations des poids synaptiques	93
5.3	Résultats obtenus pour le réseau EBF	95
5.3.1	Résultats obtenus pour la perturbation des stimuli d'entrée	95
5.3.2	Résultats obtenus pour la perturbation des poids	96

6	Résultats de l'injection matérielle d'upsets	101
6.1	Introduction	101
6.2	Résultats obtenus pour les réseaux MLP_i	101
6.3	Résultats obtenus pour le réseau OCR	103
6.4	Résultats obtenus pour le réseau EBF	106
6.4.1	Cas du L-Neuro 1.0	106
6.4.1.1	Discussion	106
6.4.2	Cas du DSP96002	108
6.4.2.1	Injection de fautes dans la zone du programme	109
6.4.3	Cas du T225	110
6.4.3.1	Injection de fautes dans la zone des données	110
6.4.3.2	Injection de fautes dans la zone du programme	110
6.4.3.3	Discussion	111
7	Résultats des tests aux ions lourds	113
7.1	Introduction	113
7.2	Résultats obtenus pour l'UTAK	114
7.3	Résultats obtenus pour le L-Neuro 1.0	115
7.3.1	Cas des réseaux MLP_i	115
7.3.2	Cas du réseau EBF	116
7.4	Résultats obtenus pour l'OCR	117
7.5	Résultats obtenus pour le DSP96002	118
7.6	Résultats obtenus pour le T225	119
7.7	Discussion	120
A	Circuits implantant des réseaux de neurones	127
A.1	Implantations digitales	127
A.1.1	Architectures en tranches	127
A.1.2	Architectures multiprocesseurs	129
A.1.3	Autres implantations digitales	131
A.2	Implantations analogiques	133
A.3	Implantations hybrides	134
A.4	Implantations optoélectroniques	135

B	Cartes accélératrices des réseaux de neurones	137
B.1	Accélérateurs construits autour des circuits standards	137
B.2	Accélérateurs construits autour des circuits neuronaux	140
C	Classement des applications neuronales par domaines d'application	143
C.1	Reconnaissance de formes	144
C.2	Traitement du signal	144
C.3	Contrôle adaptatif	145
C.4	Traitement d'images	145
C.5	Traitement de la parole	146
C.6	Modélisation	146
C.7	Prévision	147
C.8	Optimisation	147
C.9	Robotique	148
D	Schémas de la carte 68000	149
E	Schémas de la carte UTAK	159
F	Schémas de la carte OCR	167
G	Schémas de la carte T225 – L-Neuro 1.0	181
H	Glossaire des abréviations utilisées	187

Table des figures

1.1	Le phénomène d'upset - cas d'un bistable CMOS	14
1.2	Forme typique de la courbe section efficace vs. LET ($\sigma = f(LET)$)	16
2.1	Fonctions d'activation d'un neurone	26
2.2	Modèle général d'un neurone	27
2.3	Réseau proactif monocouche (perceptron)	28
2.4	Réseau proactif complètement connecté avec une seule couche cachée	29
2.5	Réseau récurrent avec neurones cachés	30
2.6	Réseau d'évaluation du OU exclusif de deux variables	33
2.7	Principe d'un réseau de neurones optoélectronique	45
4.1	Mapping des bits sensibles de l'application testée obtenu suite à un test par injection matérielle de fautes	74
4.2	Carte d'une séquence d'upset obtenue suite à un test statique	74
4.3	Schéma bloc d'une implantation/émulation neuronale à une seule mémoire	75
4.4	Schéma bloc d'une implantation/émulation neuronale à plusieurs mémoires	76
4.5	Interface FUTE16/circuits neuronaux	77
4.6	Fonctions booléennes évaluées	82
4.7	Réseaux MLP_i correspondant aux fonctions booléennes	83
4.8	Réseau de neurones utilisé pour la reconnaissance de caractères	84
4.9	Réseau de neurones TDNN utilisé pour la détection de sifflements	87
4.10	Exemples de stimuli d'entrée du réseau et sorties associées	87
5.1	Taux de reconnaissance ρ vs. le nombre b de bits perturbés dans les motifs d'entrée	93

5.2	Taux de reconnaissance ρ vs. le nombre b de bits perturbés dans les poids synaptiques	94
5.3	Taux de reconnaissance moyen $\bar{\rho}$ par faute vs. la taille d'apprentissage α	94
5.4	Nombre moyen de bits d'entrée sensibles aux upsets pour différentes valeurs du seuil de sortie	96
5.5	Nombre de bits de poids sensibles par motif d'entrée (réseau à poids partagés)	97
5.6	Nombre de bits de poids sensibles par motif d'entrée (réseau à poids non partagés)	97
5.7	Sensibilité aux upsets en fonction du bit perturbé (réseau à poids partagés)	98
5.8	Sensibilité aux upsets en fonction du bit perturbé (réseau à poids non partagés)	98
6.1	Mapping des cellules sensibles aux upsets de l'UTAK pour un réseau à deux couches cachées (seuls 67 parmi les 1 Kbits sont sensibles aux upsets)	102
6.2	Mapping des cellules sensibles aux upsets du L-Neuro pour un réseau à deux couches cachées (34 bits sont sensibles parmi les 8 Kbits de la mémoire synaptique)	102
6.3	Injection de fautes dans la zone bitmap	104
6.4	Pourcentage de la partie fond pour les bitmaps de quelques caractères	104
6.5	Injection des fautes dans la zone programme entière	105
6.6	Injection des fautes dans la zone du stockage des instructions	106
6.7	Mapping d'upsets pour deux motifs d'entrée différents du problème de détection de sifflements	107
6.8	Sensibilité aux upsets des différents bits à l'intérieur d'un octet	108
7.1	Sections efficaces statiques et dynamiques extrapolées du L-Neuro	116
D.1	Buffers d'adresses	150
D.2	Buffers de données poids faible	151
D.3	Buffers de données poids fort	152
D.4	Buffers de contrôle (1/2)	153
D.5	Buffers de contrôle (2/2)	154
D.6	Génération de l'horloge	155

D.7	Génération des signaux de contrôle (1/2)	156
D.8	Génération des signaux de contrôle (2/2)	157
D.9	Connecteur IDC (interface carte mère – carte fille)	158
E.1	Compteur d’adressage N° 1	160
E.2	Compteur d’adressage N° 2	161
E.3	Buffers de données	162
E.4	Génération des signaux de contrôle (1/2)	163
E.5	Génération des signaux de contrôle (2/2)	164
E.6	Communication inter-UTAK	165
E.7	Connecteur IDC vu côté carte UTAK	166
F.1	Buffers d’adresses 68000 – RAMOCR	168
F.2	Buffers d’adresses OCR – RAM externes	169
F.3	Buffers d’adresses 68000 – RAM externes	170
F.4	Buffers de données 68000 – RAM externes	171
F.5	Buffers de données 68000 – RAMOCR	172
F.6	Interconnexions des RAM externes	173
F.7	Génération des signaux de contrôle (1/3)	174
F.8	Génération des signaux de contrôle (2/3)	175
F.9	Génération des signaux de contrôle (3/3)	176
F.10	Polarisation du circuit OCR	177
F.11	Génération de l’interruption	178
F.12	Connecteur IDC vu côté carte OCR	179
G.1	Interface 68000 – RAM externes	182
G.2	Interface T225 – L-Neuro 1.0	183
G.3	Génération des signaux de contrôle	184
G.4	Capacités de découplage de l’alimentation	185
G.5	Connecteur IDC vu côté carte T225 – L-Neuro 1.0	186

Liste des tableaux

2.1	Valeurs des poids et seuils du réseau XOR	33
2.2	Valeurs des sorties des neurones pour différentes combinaisons d'entrée	34
2.3	Types des réseaux pour différentes applications	52
3.1	Valeurs des sorties des neurones pour des entrées perturbées	62
3.2	Comparaison des sensibilités moyenne et pire cas pour différents algorithmes d'apprentissage, cas des erreurs de type mise à "0" et à "1" .	69
4.1	Zones sensibles aux upsets des différents circuits cibles	81
4.2	Performances atteintes par les différentes approches testées	86
4.3	Liste des programmes de test des différents circuits cibles	90
4.4	Expériences réalisées pour les différents réseaux choisis	90
5.1	Sensibilité moyenne aux upsets des réseaux testés pour les formats 8/32 bits	98
6.1	Sensibilité aux upsets injectés dans les RAM des circuits testés	103
6.2	Sensibilité aux upsets de la zone programme	109
6.3	Sensibilité aux upsets de la zone données	110
6.4	Sensibilité aux upsets de la zone programme et de l'application entière	111
7.1	Sections efficaces dynamiques mesurées et extrapolées ($cm^2/composant$)	115
7.2	Sections efficaces statiques SEU et SEL du L-Neuro ($cm^2/composant$)	115
7.3	Sections efficaces dynamiques mesurées et extrapolées ($cm^2/composant$) obtenues pour le circuit L-Neuro avec un faisceau de Chlore	116
7.4	Mapping des SEU dans la zone utilisée dérivé de la séquence du test statique (NU : Non Utilisé dans l'implantation de l'application)	118
7.5	Section efficace dynamique mesurée (EBF vs. produit matriciel)	119

Bibliographie

- [80170NX91] 80170NX ETANN Data Sheet, Intel, 1991.
- [Ackley85] Ackley (D.H.), Hinton (G.E.) et Sejnowski (T.J.). – A Learning Algorithm for Boltzmann Machines. *Cognitive Science*, vol. 9, 1985, pp. 147–169.
- [Alspector91] Alspector (J.), Jayakumar (T.) et Luna (S.). – Experimental Evaluation of Learning in a Neural Microsystem. *Proc. of Advances in Neural Information Processing Systems Conference*. pp. 871–878. – San Mateo, Ca., USA, 1991.
- [Angéniol90] Angéniol (B.). – Industrial Applications of Neural Networks. *Neural Networks: Biological Computers or Electronic Brains*. pp. 65–70. – Springer Verlag, 1990.
- [Ansari95] Ansari (N.), Hou (E.S.H.) et Yu (Y.). – A New Method to Optimize the Satellite Schedules Using the Mean Field Annealing of a Hopfield Neural Network. *IEEE Transactions on Neural Networks*, vol. 6, n° 2, 1995, pp. 470–483.
- [Ansari96] Ansari (N.), Arulambalam (A.) et Balasekar (S.). – Traffic Management of a Satellite Communication Network Using Stochastic Optimization. *IEEE Transactions on Neural Networks*, vol. 7, n° 3, 1996, pp. 732–744.
- [Archive96] Neural Networks Archive, Applications Subject Area, <http://www.lpac.ac.uk/SEL-HPC/Articles/GeneratedHtml/neural.appl.html>, 1996.
- [Arif93] Arif (A.F.), Kuno (S.), Iwata (A.) et Yoshida (Y.). – A Neural Network Accelerator Using Matrix Memory with Broad-

cast Bus. *Proc. of the IJCNN'93*, pp. 3050–3053. – Nagoya, Japon, 1993.

- [Assoum95a] Assoum (A.), Bezerra (F.), Velazco (R.) et Benezech (D.). – SEU and Latchup Results on Transputers. *Proc. of RA-DECS'95*, pp. 340–345. – Arcachon, France, 1995.
- [Assoum95b] Assoum (A.), Torres (J.M.), Radi (N.E.), Velazco (R.), Gordon (M.B.) et Ecoffet (R.). – Robustness against Single Event Upsets of Digital Implementations of Neural Networks. *Proc. of the ICANN'95, Industrial Conference, Session 9, VLSI and Dedicated Hardware*. – Paris, France, 1995.
- [Assoum95c] Assoum (A.), Velazco (R.), Radi (N.E.), Ecoffet (R.) et Botey (X.). – SEU Fault Tolerance in Artificial Neural Networks. *IEEE Transactions on Nuclear Science*, vol. 42, n° 6, 1995, pp. 1856–1862.
- [Assoum95d] Assoum (A.), Velazco (R.), Torres (J.M.) et Gordon (M.B.). – Robustness of Hardware Implementations of Artificial Neural Networks : A Case Studied. *First IEEE on-line Testing Workshop*, pp. 83–88. – Nice, France, 1995.
- [Assoum96a] Assoum (A.), Radi (N.E.), Velazco (R.), Elie (F.) et Ecoffet (R.). – Robustness against SEU of an Artificial Neural Network Space Application. *IEEE Transactions on Nuclear Science*, vol. 43, n° 3, 1996, pp. 973–978.
- [Assoum96b] Assoum (A.), Velazco (R.), Cheynet (Ph.), Olmos (M.) et Ecoffet (R.). – SEU Experiments on an Artificial Neural Network Implemented by Means of Digital Processors. *IEEE Transactions on Nuclear Science*, vol. 43, n° 6, 1996, pp. 2889–2896.
- [Assoum97] Assoum (A.), Velazco (R.) et Ziade (H.). – Transient Errors Simulations on Neural Networks : A Case Studied. *Sou-*

mis à International Conference on Computer Simulation (ICCS'97). – Beyrouth, Liban, Sep. 1997.

- [Augusteijn93] Augusteijn (M.F.) et Skufca (T.L.). – Identification of Human Faces Through Texture-Based Feature Recognition and Neural Network Technology. *Proc. of the IEEE ICNN'93*, pp. 392–398. – San Francisco, Ca., USA, 1993.
- [Barber95] Barber (J.C.). – Neural Nets in Finance: An Overview. *Risks and Rewards, the Newsletter of the Investment*, n° 23, Juin 1995.
- [Barillot96] Barillot (C.) et Calvel (Ph.). – Review of Commercial Spacecraft Anomalies and Single-Event Effect Occurrences. *IEEE Transactions on Nuclear Science*, vol. 43, n° 2, 1996, pp. 453–460.
- [Bartal94] Bartal (Y.), Lin (J.) et Uhring (R.E.). – Nuclear Power Plants Transient Diagnostics Using LVQ or Some Networks Don't Know That They Don't Know. *Proc. of the IEEE ICNN'94*, pp. 3744–3749. – Orlando, Fa., USA, 1994.
- [Bedworth88] Bedworth (M.D.) et Lowe (D.). – Fault Tolerance in Multi-Layer Perceptrons: a Preliminary Study. *RSRE: Pattern Recognition and Machine Intelligence Division*. – Juil. 1988.
- [Beichter91] Beichter (J.), Bruels (N.), Meister (E.), Ramacher (U.) et Klar (H.). – Design of General-Purpose Neural Signal Processor. *International Conference on Microelectronics for Neural Networks*, pp. 311–315. – Munich, Germany, 1991.
- [Beichter93] Beichter (J.), Bruels (N.), Sicheneder (E.), Ramacher (U.) et Klar (H.). – Design of a General Purpose Neural Signal Processor. *International Journal on Neurocomputing*, vol. 5, n° 1, 1993, pp. 17–23.
- [Bessot93] Bessot (D.). – *Radiation Hardening Technics Facing Total Dose, SEU and SEL in the Space Environment*. – Rapport

technique n° B465, Oxon, UK, Harwell Laboratory, Oct. 1993.

- [Binder75] Binder (D.), Smith (E.C.) et Holman (A.B.). – Satellites Anomalies from Galactic Cosmic Rays. *IEEE Transactions on Nuclear Science*, vol. 22, n° 6, Déc. 1975, pp. 2675–2680.
- [Binh95] Binh (L.N.) et Chong (H.C.). – A Neural Network Contention Controller for Packet Switching Networks. *IEEE Transactions on Neural Networks*, vol. 6, n° 6, 1995, pp. 1402–1410.
- [Bohn94] Bohn (S.) et Thornton (E.). – Environment Reconstruction for Robot Navigation. *Proc. of the SPIE'94, International Society for Optical Engineering*. – Avr. 1994.
- [Bolt91] Bolt (G.). – *Investigating Fault Tolerance in Artificial Neural Networks*. – Rapport technique n° YCS 154, Heslington, York, UK, University of York, Department of Computer Science, 1991.
- [Boser92] Boser (B.E.), Sackinger (E.), Bromely (J.), LeCun (Y.) et Jackel (L.D.). – Hardware Requirements for Neural Networks Pattern Classifiers. *IEEE Micro*, vol. 12, n° 1, 1992, pp. 32–40.
- [Bourrieau88] Bourrieau (J.). – Irradiations – Doses Cumulées – Ions Lourds. *Cours de Technologie Spatiale*. CNES, p. 597. – Mar. 1988.
- [Bourrieau91] Bourrieau (J.). – L'environnement Spatial : Flux, Dose, Blindage, Effets des Ions Lourds. *Tutorial short course*. IEEE – RADECS. – Sep. 1991.
- [Brown94] Brown (R.), Kharouf (P.), Feng (X.), Piessens (L.) et Nestor (D.). – Development of Feed-Forward Network Models to Predict Gas Consumption. *Proc. of the IEEE ICNN'94*, pp. 3693–3696. – Orlando, Fa., USA, 1994.

- [Brucker84] Brucker (G.J.), Smeltzer (R.), Kolasinski (W.A.) et Koga (R.). – Soft Error Dependence on Feature Size. *IEEE Transaction on Nuclear Science*, vol. 31, n° 6, Déc. 1984, pp. 1562–1564.
- [Burel93] Burel (G.) et Catros (J.Y.). – Image Compression Using Topological Maps and MLP. *Proc. of the IEEE ICNN'93*, pp. 727–731. – San Francisco, Ca., USA, 1993.
- [Capdevielle84] Capdevielle (J.N.). – Les Rayons Cosmiques. *Collection: Que Sais-Je?* – Paris, Presses Universitaires de France, 1984.
- [Castillo91] Castillo (F.), Cabestany (J.) et Moreno (J.M.). – An Integrated Circuit for Artificial Neural Networks. *Artificial Neural Networks*, éd. par Prieto (A.). pp. 328–332. – Springer Verlag, 1991.
- [CERN96] European Laboratory for Particle Physics (ancien Centre Européen pour la Recherche Nucléaire), <http://www1.cern.ch/NeuralNets/nnwInHepHard.html>, 1996.
- [Chakrabarti95] Chakrabarti (S.), Bindal (N.) et Theagarajan (K.). – Robust Radar Target Classifier Using Artificial Neural Networks. *IEEE Transactions on Neural Networks*, vol. 6, n° 3, 1995, pp. 760–766.
- [Chenette84] Chenette (D. L.) et Dietrich (W.F.). – The Solar Flare Ion in Environment for Single Event Upsets – A Summary of Observations over the Last Solar Cycles 1973 – 1983. *IEEE Transaction on Nuclear Science*, vol. 31, n° 6, Déc. 1984, pp. 1217–1222.
- [Chiu93a] Chiu (C.T.), Mehrotra (K.), Mohan (C.K.) et Ranka (S.). – Fault Tolerance of Feed-Forward Neural Networks with Hardware Limitations. *Proc. of the International Symposium on Artificial Neural Networks*, pp. E31–E40. – Hsinchu, Taiwan, Déc. 1993.

- [Chiu93b] Chiu (C.T.), Mehrotra (K.), Mohan (C.K.) et Ranka (S.).
– Robustness of Feedforward Neural Networks. *Proc. of the IEEE International Conference of Neural Networks (ICNN'93)*, pp. 783–788. – San Fransisco, Ca., USA, 1993.
- [Chiu94] Chiu (C.T.), Mehrotra (K.), Mohan (C.K.) et Ranka (S.).
– Training Technics to Obtain Fault Tolerant Neural Networks. *Proc. of the International Symposium on Fault Tolerant Computing (FTCS-24)*, pp. 360–369. – Austin, TX, USA, 1994.
- [Choong94] Choong (P.) et De Silva (C.). – Breast Cancer Prognosis Using the Emn Architecture. *Proc. of the IEEE ICNN'94*, pp. 3516–3520. – Orlando, Fa., USA, 1994.
- [Choong96] Choong (P.L.), DeSilva (C.J.S.), Dawkins (H.J.S.) et Sterrett (G.F.). – Entropy Maximization Networks: An Application to Breast Cancer Prognosis. *IEEE Transactions on Neural Networks*, vol. 7, n° 3, 1996, pp. 568–577.
- [Christo95] Christo (F.C.), Masri (A.R.), Nebot (E.M.) et Turanyi (T.).
– Utilising Artificial Neural Network and Repro-Modelling in Turbulent Combustion. *Proc. of the IEEE ICNN'95*, pp. 911–916. – Perth, Australie, 1995.
- [Clarkson93] Clarkson (T.G.), Ng (C.K.) et Guan (Y.). – The pRam : An Adaptive VLSI Chip. *IEEE Transactions on Neural Networks*, vol. 4, n° 3, 1993, pp. 408–412.
- [CNES93] CNES–DTI/PT. – *Journée Thématique: Les Réseaux de Neurones Formels et Leurs Utilisations au Domaine Spatial*. – Paris, France, Jan. 1993.
- [Conde94] Conde (T.). – Automatic Neural Detection of Anomalies Electrocardiogram (ECG) Signals. *Proc. of the IEEE ICNN'94*, pp. 3552–3558. – Orlando, Fa., USA, 1994.
- [Cunningham84] Cunningham (S.S.). – Cosmic Rays, Single Event Upsets and Things that Go Bump in the Night. *The Annual Rocky*

Mountain Guidance and Control Conference, p. 6. – Keystone, Fév. 1984.

- [Damarla89] Damarla (T.R.) et Bhagat (P.K.). – Fault Tolerance of Neural Networks. *Proc. of Energy and Information Technology in the Southeast Conference (IEEE Southeastcon)*, pp. 328–331. – Columbia, SC, USA, Avr. 1989.
- [Da Silva95] DaSilva (L.E.B.), Torres (G.L.), Saturno (E.C.), DaSilva (A.P.A.) et Do (X.D.). – Simulation of a Neural Net Controller for Motor Drives. *Proc. of the IEEE ICNN'95*, pp. 205–209. – Perth, Australie, 1995.
- [Davallo90] Davalo (E.) et Naim (P.). – *Des Réseaux de Neurones*. – Paris, France, Eyrolles, 1990.
- [Denoeux95] Denoeux (T.) et Rizand (P.). – Analysis of Radar Images for Rainfall Forecasting Using Neural Networks. *Neural Computing and Applications*, vol. 3, n° 1, 1995, pp. 50–61.
- [Der Smagt95] Van Der Smagt (P.), Groen (F.C.A.) et Krose (B.J.A.). – A Monocular Robot Arm Can Be Neurally Positioned. *Proc. of the International Conference on Intelligence Autonomous Systems (IAS'95)*, pp. 123–130. – Karlsruhe, Germany, 1995.
- [Der Spiegel94] Van Der Spiegel (J.), Donham (C.), Etienne-Cummings (R.), Fernando (S.), Mueller (P.) et Blackman (D.). – Large Scale Neural Computer with Programmable Architecture and Programmable Time Constants for Temporal Analysis. *Proc. of the IEEE ICNN'94*, pp. 1830–1835. – Orlando, Fa., USA, 1994.
- [Devel95] DeVel (O.), Wangsuya (S.) et Coomans (D.). – On Thai Character Recognition. *Proc. of the IEEE ICNN'95*, pp. 2095–2098. – Perth, Australie, 1995.
- [Deville95] Deville (Y.). – Digital VLSI Neural Networks : from Versatile Neural Processors to Application-Specific Chips. *Proc.*

of the ICANN'95, Industrial Conference, Session 9, VLSI and Dedicated Hardware. – Paris, France, 1995.

- [Dhawan93] Dhawan (A.P.) et Arata (L.). – Segmentation of Medical Images Through Competitive Learning. *Proc. of the IEEE ICNN'93*, pp. 1277–1282. – San Francisco, Ca., USA, 1993.
- [Diehl83] Diehl (S.E.), Vinson (J.E.), Shafer (B.D.) et Mnich (T.M.). – Consideration for Single Event Upset Immune VLSI Logic. *IEEE Transactions on Nuclear Science*, vol. 30, n° 6, Déc. 1983, pp. 4501–4507.
- [Dorffner94] Dorffner (G.), Leitgeb (E.) et Koller (H.). – Towards Improving Exercise ECG for Detecting Ischemic Heart Disease with Recurrent and Feed-Forward Neural Nets. *Proc. of Neural Networks for Signal Processing Workshop*, pp. 499–508. – New York, NY, USA, 1994.
- [Eide96] Eide (A.), Jahren (C.), Jorgensen (S.), Lindblad (T.), Lindsey (C.S.) et Osterud (K.). – Eye Identification for Face Recognition with Neural Networks. *Proc. of Applications and Science of Artificial Neural Networks, SPIE'1996, Symposium on Aerospace/Defense Sensing and Control and Dual Use Photonics*, pp. 333–338. – Avr. 1996.
- [Elman88] Elman (J.L.). – *Finding Structure in Time.* – Rapport technique, CRL Tech, UCSD, 1988.
- [Emmerson91] Emmerson (M.D.), Damper (R.I.), Hey (A.J.G.) et Upstill (C.). – Fault Tolerance and Redundancy of Neural Nets for the Classification of Acoustic Data. *Proc. of the IEEE ICASSP'91*, pp. 1053–1056. – 1991.
- [Emmerson92] Emmerson (M.D.) et Damper (R.I.). – Relations between Fault Tolerance and Internal Representations for Multi-Layer Perceptrons. *Proc. of the IEEE ICASSP'92*, pp. 281–284. – 1992.

- [Estrémé93] Estrémé (F.), Velazco (R.), Karoui (S.) et Benezech (D.). – SEU and Latchup Results for SPARC Processors. *IEEE Radiation Effects Data Workshop*, pp. 13–19. – 1993.
- [Falas94] Falas (T.), Charitou (A.) et Charalambous (C.). – The Application of Artificial Neural Networks in the Prediction of Earnings. *Proc. of the IEEE ICNN'94*, pp. 3629–3633. – Orlando, Fa., USA, 1994.
- [Falcon 96] La Société Falcon Asset Managment, 1996, <http://www.fam.com/>.
- [Feltham94] Feltham (R.) et Xing (G.). – Pyramidal Neural Network for Mammogram Tumour Pattern Recognition. *Proc. of the IEEE ICNN'94*, pp. 3546–3551. – Orlando, Fa., USA, 1994.
- [Flynn66] Flynn (M.J.). – Very High-Speed Computing Systems. *Proceedings of the IEEE*, pp. 1901–1909. – 1966.
- [Freisleben95] Freisleben (B.) et Ripper (K.). – Economic Forecasting Using Neural Networks. *Proc. of the IEEE ICNN'95*, pp. 833–838. – Perth, Australie, 1995.
- [Fu95] Fu (H.C.), Tung (W.L.) et Shen (L.J.). – A Neural Network for the Automatic Diagnosis of the Telephone Switching Systems. *Proc. of the IEEE ICNN'95*, pp. 482–487. – Perth, Australie, 1995.
- [Gagné90] Gagné (G.) et Savaria (Y.). – Optimisation of Resistively Hardened Latches. *IEEE Transactions on Nuclear Science*, vol. 37, n° 1, 1990, pp. 7–14.
- [Garner95] Garner (N.), Howard (D.), Tyrrel (A.) et Breen (A.). – Neural Network for Synatxic Categorisation of Words. *Proc. of the IEEE ICNN'95*, pp. 2863–2866. – Perth, Australie, 1995.
- [Garth87] Garth (S.C.J.). – A Chipset for High-Speed Simulation of Neural Networks. *Proc. of the IEEE ICNN'87*, pp. 443–452. – 1987.

- [Glesner94] Glesner (M.) et Pochmuller (W.). – *Neurocomputers: An Overview of Neural Networks in VLSI*. – London, Chapman & Hall, 1994.
- [Goh94] Goh (A.). – A Backpropagation Approach for Predicting Seismic Liquefaction Potentiel in Soils. *Proc. of the IEEE ICNN'94*, pp. 3322–3325. – Orlando, Fa., USA, 1994.
- [Gordon93] Gordon (M.B.) et Berchier (D.). – Minimeror: A perceptron learning rule that finds the optimal weights. *Proc. of the ESANN'93*, éd. par Verleysen (M.). – Bruxelles, Belgique, 1993.
- [Graf93] Graf (H.P.), Sackinger (E.) et Jackel (L.D.). – Recent Developments of Electronic Neural Nets in North America. *Journal of VLSI Signal Processing*, vol. 1, n° 5, 1993, pp. 19–31.
- [Green95] Green (H.G.) et Pearson (M.A.). – Artificial Intelligence in Financial Markets. *Proc. of the ICNN'95*, pp. 839–844. – Perth, Australia, 1995.
- [Gussenhoven93] Gussenhoven (M.S.) et Mullen (E.G.). – Space Radiation Effects Program: An Overview. *IEEE Transactions on Nuclear Science*, vol. 40, n° 2, 1993, pp. 221–227.
- [Habib95] Habib (I.W.). – Neurocomputing in High-Speed Networks. *IEEE Communications Magazine*, vol. 33, n° 10, Oct. 1995, pp. 38–40.
- [Hackbarth89] Hackbarth (H.) et Krause (A.). – Scaly Artificial Neural Network for Speaker-Independent Recognition of Isolated Words. *Proc. of the IEEE ICASSP'89*, pp. 21–24. – 1989.
- [Haitao95] Haitao (T.) et Simula (O.). – Neural Aadaptation for Optimal Traffic Shaping in Telephone Systems. *Proc. of the IEEE ICNN'95*, pp. 1561–1565. – Perth, Australie, 1995.

- [Hamilton92] Hamilton (A.), Murray (A.F.), Baxter (D.J.), Churcher (S.), Reekie (H.M.) et Tarassenko (L.). – Integrated Pulse Stream Neural Networks: Results, Issues and Pointers. *IEEE Transactions on Neural Networks*, vol. 3, n° 3, 1992, pp. 385–393.
- [Hammerston93] Hammerston (D.). – Neural Networks at Work. *IEEE Spectrum*, Juin 1993, pp. 26–32.
- [Hammerstrom90] Hammerstrom (D.) et Means (E.). – System Design for a Second Generation Neurocomputer. *Proc. of the IJCNN'90*, pp. 80–83. – Washington DC, USA, 1990.
- [Hammerstrom90] Hammerstrom (D.). – A VLSI Architecture for High-Performance, Low-Cost, On-chip Learning. *Proc. of the IJCNN'90*, pp. 537–544. – San Diego, Ca., USA, Juin 1990.
- [Hancock88] Hancock (P.). – *Data Representation in Neural Nets: an Empirical Study*. – Morgan Kaufmann, 1988.
- [Hayakawa89] Hayakawa (M.). – Satellite Observation of Low-Altitude VLF Radio Noises and their Association with Thunderstorms. *Journal of Geomagnetism*, vol. 43, 1989, p. 573.
- [Haykin94] Haykin (S.). – *Neural Networks: a Comprehensive Foundation*. – Macmillan College Publishing Company, 1994.
- [He95] He (J.), Liu (L.) et Palm (G.). – Speaker Identification Using LVQ-SLP Networks. *Proc. of the IEEE ICNN'95*, pp. 2052–2055. – Perth, Australie, 1995.
- [Healy92] Healy (A.J.). – A Neural Network Approach to Failure Diagnostics for Underwater Vehicles. *Proc. of the AUV'92*, pp. 131–134. – Washington DC, USA, 1992.
- [Heemskerk95] Heemskerk (J.N.H.). – Overview of Neural Hardware. <ftp://ftp.mrc-apu.cam.ac.uk/pub/nn/murre/neurhard.ps>. – 1995.

- [Hiramatsu94] Hiramatsu (A.). – ATM Call Admission Control Using a Neural Network Trained with a Virtual Output Buffer Method. *Proc. of the IEEE ICNN'94*, pp. 3611–3616. – Orlando, Fa., USA, 1994.
- [Holler89] Holler (M.), Tam (S.), Castro (H.) et Benson (R.). – An Electrically Trainable Artificial Neural Network (ETANN) with 10240 Floating Gate Synapses. *Proc of the IJCNN'89*, pp. 191–196. – Washington DC, USA, 1989.
- [Holler91] Holler (M.). – VLSI Implementation of Learning and Memory Systems: A Review. *Advances in Neural Information Processing Systems*. – San Mateo, Ca., USA, 1991.
- [Holmstrom92] Holmstrom (L.) et Koistinen (P.). – Using Additive Noise in Backpropagation Training. *IEEE Transactions on Neural Networks*, vol. 3, n° 1, 1992, pp. 24–38.
- [Hopfield82] Hopfield (J.J.). – Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. of the National Academy of Sciences of the USA*, pp. 2554–2558. – 1982.
- [Holmes85] Holmes Siedle (A.)– Calibration and Flight Testing of a Low Field PMOS Dosimeter. *IEEE Transaction on Nuclear Science*, vol. 32, n° 6, Déc. 1985, pp. 4425–4429.
- [Ichiyanagi95] Ichiyanagi (K.), Goto (Y.), Mizuno (K.), Yokomizu (Y.) et Matsumara (T.). – An Artificial Neural Network to Predict River Flow Rate into a Dam for a Hydro-Power Plant. *Proc. of the IEEE ICNN'95*, pp. 2679–2682. – Perth, Australie, 1995.
- [Inmos92] The Inmos Transputer Databook, SGS-THOMSON Microelectronics, Troisième édition, 1992.
- [Ito95] Ito (H.), Hagiwara (T.) et Furuya (T.). – Intelligent Mobile Robot. *Proc. of the IEEE ICNN'95*, pp. 2699–2702. – Perth, Australie, 1995.

- [Iwata95] Iwata (H.), Agui (T.) et Nagahashi (H.). – Boundary Detection of Color Images Using Neural Networks. *Proc. of the IEEE ICNN'95*, pp. 1426–1431. – Perth, Australie, 1995.
- [Jacquet94] Jacquet (D.) et Saucier (G.). – Design of a Digital Neural Chip : Application to Optical Character Recognition by Neural Network. *Proc. of the EURO-ASIC'94*, pp. 256–260. – Paris, France, 1994.
- [Janusz95] Janusz (B.) et Riedmiller (M.). – Self-Learning Neural Control of a Mobile Robot. *Proc. of the IEEE ICNN'95*, pp. 2358–2363. – Perth, Australie, 1995.
- [Jung95] Jung (K.C.), Kim (S.K.) et Kim (K.J.). – Recognition-Based Segmentation of on-Line Cursive Korean Character. *Proc. of the IEEE ICNN'95*, pp. 3101–3106. – Perth, Australie, 1995.
- [Kah95] Kah (A.H.), San (Q.Y.), Guan (S.C.), Kiat (W.C.) et Koh (Y.C.). – Smart Air Conditionning System Using Multilayer Perceptron Neural Network with a Modular Approach. *Proc. of the IEEE ICNN'95*, pp. 2314–2319. – Perth, Australie, 1995.
- [Kamijo93] Kamijo (M.). – Classifying Fingerprint Images Using Neural Networks : Deriving the Classification State. *Proc. of the IEEE ICNN'93*, pp. 1932–1937. – San Francisco, Ca., USA, 1993.
- [Kerezsi95] Kerezsi (B.) et Howard (I.). – Vibration Fault Detection of Large Turbogenerators Using Neural Networks. *Proc. of the IEEE ICNN'95*, pp. 121–126. – Perth, Australie, 1995.
- [Kerlirzin93] Kerlirzin (P.) et Vallet (F.). – Robustness in Multilayer Perceptrons. *Neural Computation*, vol. 5, n° 3, 1993, pp. 473–482.
- [Keyvan93] Keyvan (S.), Durg (A.) et Rabelo (L.C.). – Evaluation of the Performance of Various Artificial Neural Networks to the

- Signal Faults Diagnosis in Nuclear Reactor Systems. *Proc. of the IEEE ICNN'93*, pp. 1719–1723. – San Francisco, Ca., USA, 1993.
- [Khalid95] Khalid (M.), Omatu (S.) et Yusof (R.). – Temperature Regulation with Neural Networks and Alternative Control Schemes. *IEEE Transactions on Neural Networks*, vol. 6, n° 3, 1995, pp. 572–582.
- [Killer95] Killer (P.E.), Kangas (L.J.), Hashem (S.), Kouzes (R.T.) et Allen (P.A.). – A Novel Approach to Modelling and Diagnosing the Cardiovascular System. *Proc. of the World Conference of Neural Networks (WCNN'95)*, pp. 767–770. – Mahwah, NJ, USA, 1995.
- [Kilpatrick95] Kilpatrick (D.) et Williams (R.). – Unsupervised Classification of Antarctic Satellite Imagery Using Kohonen's Self-organizing Feature Map. *Proc. of the IEEE ICNN'95*, pp. 32–36. – Perth, Australie, 1995.
- [Kimura94] Kimura (Y.). – A New Scheme which Incrementally Generates Neural Networks for Distorted Handprinted Kanji Pattern Recognition. *Proc. of the IEEE ICNN'94*, pp. 3852–3855. – Orlando, Fa., USA, 1994.
- [Klimasauskas89] Klimasauskas (C.). – Neural Nets and Noise Filtering. *Dr Dobb's Journal*, Jan. 1989, pp. 32–48.
- [Koga96] Koga (R.). – Single-Event Effect Ground Test Issues. *IEEE Transactions on Nuclear Science*, vol. 43, n° 2, 1996, pp. 661–670.
- [Kohonen88] Kohonen (T.). – *Self-Organization and Association Memory*. – Heidelberg, Springer Verlag, 1988.
- [Kohonen90] Kohonen (T.). – Unsupervised Learning Algorithms. *Neural Networks: Biological Computers or Electronic Brains*. pp. 29–36. – Springer Verlag, 1990.

- [Kolasinski81] Kolasinski (W. A.), Koga (R.), Blake (J.B.) et Diehl (S.E.). – Soft Errors Susceptibility of CMOS RAMs : Dependence Upon Power Supply Voltage. *IEEE Transaction on Nuclear Science*, vol. 28, n° 6, Déc. 1981, pp. 4013–4016.
- [Kong95a] Kong (H.) et Guan (L.). – Detection and Removal of Impulse Noise by a Neural Network Guided Adaptive Median Filter. *Proc. of the IEEE ICNN'95*, pp. 845–849. – Perth, Australie, 1995.
- [Kong95b] Kong (J.H.L.) et Martin (G.P.M.D.). – A Backpropagation Neural Network for Sales Forecasting. *Proc. of the IEEE ICNN'95*, pp. 1007–1011. – Perth, Australie, 1995.
- [Kuhn95] Kuhn (D.), Buessler (J.L.) et Urban (J.P.). – Neural Approach to Visual Servoing for Robot Hand Eye Coordination. *Proc. of the IEEE ICNN'95*, pp. 2364–2369. – Perth, Australie, 1995.
- [Kung80] Kung (H.T.) et Leiserson (C.E.). – Systolic Arrays for VLSI. *Introduction to VLSI Systems*. – Addison-Wesley, 1980.
- [Kwiatkowska95] Kwiatkowska (E.) et Torsun (I.S.). – Hybrid Neural Network System for Cloud Classification from Satellite Images. *Proc. of the IEEE ICNN'95*, pp. 1907–1912. – Perth, Australie, 1995.
- [Lala85] Lala (P.K.). – *Fault Tolerant and Fault Testable Hardware Design*. – NJ, USA, Prentice Hall, 1985.
- [Le94] Le (D.X.), Thoma (G.R.) et Wechsler (H.). – Document Classification Using Connectionist Models. *Proc. of the IEEE ICNN'94*, pp. 3009–3014. – Orlando, Fa., USA, 1994.
- [Legitimimus90] Legitimimus (D.) et Schwab (L.). – Natural Underwater Sounds Identification by the Use of Neural Networks and Linear Techniques. *Proc. of the International Neural Networks Conference (INNC'90)*, pp. 123–126. – Paris, France, 1990.

- [Lerner94] Lerner (B.), Guterman (H.), Dinstein (I.) et Romen (Y.). – Feature Selection and Chromosome Classification Using a Multilayer Perceptron Neural Network. *Proc. of the IEEE ICNN'94*, pp. 3540–3545. – Orlando, Fa., USA, 1994.
- [Lewis96] Lewis (F.L.), Yesildirek (A.) et Liu (K.). – Multi-layer Neural-Net Robot Controller with Guaranteed Tracking Performance. *IEEE Transactions on Neural Networks*, vol. 7, n° 2, 1996, pp. 388–399.
- [Li95] Li (Q.), Lim (C.M.), Poo (A.N.) et Ang (M.). – Neuro-Based Adaptive Internal Model Control for Robot Manipulators. *Proc. of the IEEE ICNN'95*, pp. 2353–2357. – Perth, Australie, 1995.
- [Linde95] Linde (A.), Taveniku (M.) et Svensson (B.). – Using Neural Networks for Air/Fuel Estimation in Two-Stroke Combustion Engines. *Proc. of the International Conference on Engineering Applications of Neural Networks (EANN'95)*, pp. 327–334. – Helsinki, Finland, 1995.
- [Lindsey94] Lindsey (C.S.) et Lindblad (T.). – Review of Hardware Neural Networks : A User's Perspective. *Proc. of the third Workshop on Neural Networks : From Biology to High Energy Physics*. – Marciana Mariana, Isola d'Elba, Italie, Sep. 1994.
- [Lindsey95] Lindsey (C.S.) et Lindblad (T.). – Survey of Neural Networks Hardware. *Proc. of Applications and Science of Artificial Neural Networks Conference*, pp. 1194–1205. – Orlando, Fla., USA, 1995.
- [Liu93] Liu (H.J.) et Sun (Y.). – Blind Bilevel Image Restoration Using Hopfield Neural Networks. *Proc. of the IEEE ICNN'93*, pp. 1656–1661. – San Francisco, Ca., USA, 1993.
- [Liu93] Liu (Y.). – Image Compression Using Stochastic Neural Networks. *Proc. of the IEEE ICNN'93*, pp. 1558–1563. – San Francisco, Ca., USA, 1993.

- [Lu95] Lu (Y.L.), Mak (M.W.) et Siu (W.C.). – Application of a Fast Real Time Recurrent Learning Algorithm to Text-to-Phoneme Conversion. *Proc. of the IEEE ICNN'95*, pp. 2853–2857. – Perth, Australie, 1995.
- [Lucking94] Lucking (W.), Darnell (M.) et Chesmore (E.). – Acoustic Condition Monitoring of a Mechanical Gearbox Using Artificial Neural Networks. *Proc. of the IEEE ICNN'94*, pp. 3307–3311. – Orlando, Fa., USA, 1994.
- [Mac Dermott89] Mac Dermott (E.) et Katagiri (S.). – Shift-Invariant, Multi-Category Phoneme Recognition Using Kohonen LVQ2. *Proc. of the IEEE ICASSP'89*, pp. 81–84. – 1989.
- [Mac Conigal94] Mac Gonigal (M.). – A New Technic for Survival Prediction in Trauma Care Using a Neural Network. *Proc. of the IEEE ICNN'94*, pp. 3495–3498. – Orlando, Fa., USA, 1994.
- [Malferrari90] Malferrari (L.), Serra (R.) et Valastro (G.). – An Application of Neural Networks to Oil Well Drilling. *Proc. of the International Neural Network Conference (INNC'90)*, pp. 127–130. – Paris, France, 1990.
- [Mallach75] Mallach (E.G.). – Emulator Architecture. *Computer*, Août 1975, pp. 24–32.
- [Masa94] Masa (P.), Hoen (K.) et Wallinga (H.). – A High-Speed Analog Neural Processor. *IEEE Micro*, vol. 14, n° 3, Juin 1994, pp. 40–50.
- [Matsuoka94] Matsuoka (K.) et Kawamoto (M.). – A Neural Net for Blind Separation of Nonstationary Signal Sources. *Proc. of the IEEE ICNN'94*, pp. 221–227. – Orlando, Fa., USA, 1994.
- [Matsuoka92] Matsuoka (K.). – Noise Injection into Inputs in Backpropagation Learning. *IEEE Transactions on Systems Man and Cybernetics*, vol. 22, n° 3, 1992, pp. 436–440.

- [Mauduit91] Mauduit (N.), Duranton (M.), Gobert (J.) et Sirat (J.A.). – Building up Neuromimetic Machines with L-Neuro 1.0. *Proc. of the IJCNN'91*, pp. 602–607. – Singapore, 1991.
- [Mauduit92] Mauduit (N.), Duranton (M.) et Gobert (J.). – L-Neuro 1.0 : A Piece of Hardware LEGO for Building Neural Networks Systems. *IEEE Transactions on Neural Networks*, vol. 1, n° 3, Mai 1992, pp. 414–422.
- [May79] May (T.C.) et Woods (M.H.). – Alpha-Particle Induced Soft Errors in Dynamic Memories. *IEEE Transactions on Electron Device*, vol. 26, n° 1, Jan. 1979, pp. 2–9.
- [Mazumdar95] Mazumdar (S.K.) et Lim (C.C.). – The Application of Neural Networks to Anti-Skid Break System Design. *Proc. of the IEEE ICNN'95*, pp. 2409–2414. – Perth, Australie, 1995.
- [Mc Lean84] Mc Lean (F. B.). – A Framework of Understanding Radiation-Induced Interface in SiO₂ MOS Structures. *IEEE Transaction on Nuclear Science*, vol. 31, n° 6, Déc. 1984, pp. 1651–1657.
- [Means91] Means (R.W.) et Lisenbee (L.). – Extensible Linear Floating Point SIMD Neurocomputer Array Processor. *Proc. of the IJCNN'91*, pp. 587–592. – Washington DC, USA, Juil. 1991.
- [Minière96] Minière (X.), Pincon (J.L.) et Lefeuvre (F.). – A Neural Network Approach to the Classification of Electron and Proton Whistler. *Journal of Atmospheric and Terrestrial Physics*, vol. 58, n° 7, 1996, pp. 911–924.
- [Minière94] Minière (X.). – *Approche Réseaux de Neurones pour la Classification d'Emissions Structurées de Type Sifflements*. – Paris, France, Thèse de PhD, Université Paris 7, 1994.
- [Mirsepassi95] Mirsepassi (A.), Cathers (B.) et Dharmappa (H.B.). – Application of Artificial Neural Networks to the Real Time

- Opertaion of Water Treatment Plant. *Proc. of the IEEE ICNN'95*, pp. 516–521. – Perth, Australie, 1995.
- [Mitra93] Mitra (U.) et Poor (H.V.). – Neural Network Techniques for Multi-User Demodulation. *Proc. of the IEEE ICNN'93*, pp. 1538–1543. – San Francisco, Ca., USA, 1993.
- [Moore88] Moore (W.R.). – Conventional Fault-Tolerance and Neural Computers. *Neural Computers*. pp. 29–37. – Springer Verlag, 1988.
- [Morgan92] Morgan (N.), Beck (J.), , Kohn (P.), Bilmes (J.), Allman (E.) et Beer (J.). – The Ring Array Processor : A Multiprocessing Peripheral for Connectionest Applications. *Journal of Parallel and Distributed Computing*, n° 14, 1992, pp. 248–259.
- [Motorola91] DSP96002 IEEE Floating-Point Dual-Port Processor User's Manual, Motorola, Juil. 1991.
- [MT19003] MT19003 Data Sheet, Micro Circuit Engineering, 1993.
- [Muller95] Muller (U.A.), Gunzinger (A.) et Guggenbuhl (W.). – Fast Neural Net Simulation with a DSP Processor Array. *IEEE Transactions on Neural Networks*, vol. 6, n° 1, 1995, pp. 203–213.
- [Murray91] Murray (A.F.), Del Corso (D.) et Tarassenko (L.). – Pulse Stream VLSI Neural Networks Mixing Analog and Digital Techniques. *IEEE Transactions on Neural Networks*, vol. 2, n° 2, 1991, pp. 193–204.
- [Murray94a] Murray (A.F.), Churcher (S.), Hamilton (A.), Holmes (A.J.), Jackson (G.B.), Reekie (H.M.) et Woodburn (R.J.). – Pulse Stream VLSI Neural Networks. *IEEE Micro Journal*, vol. 14, n° 3, 1994, pp. 29–39.
- [Murray94b] Murray (A.F.) et Edwards (P.J.). – Enhanced MLP Performance and Fault Tolerance Resulting from Synaptic Weight

- Noise During Learning. *IEEE Transactions on Neural Networks*, vol. 5, n° 5, 1994, pp. 792–802.
- [Murshed95] Murshed (N.A.), Bortolozzi (F.) et Sabourin (R.). – Off-Line Signature Verification Using Fuzzy Artmap Neural Network. *Proc. of the IEEE ICNN'95*, pp. 2179–2184. – Perth, Australie, 1995.
- [N. A. Corp.] La Société Neural Applications Corporation, The Intelligent Arc Furnance Controller, <http://www.neural.com/IAFController/IAFController.html>, 1996.
- [Nath94] Nath (S.), Wincheski (B.), Fulton (J.P.) et Namkung (M.). – Classification of Magneto-Optic Images Using Neural Networks. *Proc. of ASNT Conference*, pp. 125–127. – New Orleans, La., USA, Mar. 1994.
- [Nekovei95] Nekovei (R.) et Sun (Y.). – Back-Propagation Network and its Configuration for Blood Vessel Detection in Angiograms. *IEEE Transactions on Neural Networks*, vol. 6, n° 1, 1995, pp. 64–72.
- [Neocleous95] Neocleous (C.C.) et Schizas (C.N.). – Artificial Neural Network in Marine Propeller Design. *Proc. of the IEEE ICNN'95*, pp. 1098–1102. – Perth, Australie, 1995.
- [Neuredge 94] Testing Tyres to the Limit – High Speed Fault Detection, Neural Edge, Issue 6, 1994.
- [N. Class96] <http://www.ice.el.utwente.nl/Finished/Neuro/NeuroClassifier.htm>, 1996.
- [Nogami94] Nogami (T.), Yokoi (Y.), Ichiba (H.) et Atsumi (Y.). – Gas Discrimination Method for Detection Transformer Faults by Neural Network. *Proc. of the IEEE ICNN'94*, pp. 3800–3805. – Orlando, Fa., USA, 1994.
- [Nordstrom95] Nordstrom (E.), Carlstrom (J.), Gallmo (O.) et Asplund (L.). – Neural Networks for Adaptive Traffic Control in

- ATM Networks. *IEEE Communications Magazine*, vol. 33, n° 10, Oct. 1995, pp. 43–49.
- [Normand96] Normand (E.). – Single-Event Effects in Avionics. *IEEE Transactions on Nuclear Science*, vol. 43, n° 2, 1996, pp. 461–474.
- [Ntl 95] La Société Neural Technology Limited, Neural Computing in Finance and Insurance, <http://www.neuralt.com/>, 1995.
- [Ochiai95] Ochiai (K.), Suzuki (H.), Shinozawa (K.), Sonehara (N.) et Fujii (M.). – Snowfall and Rainfall Forecasting from Weather Radar Images with Artificial Neural Networks. *Proc. of the IEEE ICNN'95*, pp. 1182–1187. – Perth, Australie, 1995.
- [Ogawara95] Ogawara (Y.). – Feedback-Error-Learning Neural Network for the Automatic Maneuvering System of a Ship. *Proc. of the ICNN'95*, pp. 225–230. – Perth, Australia, 1995.
- [Ohhira95] Ohhira (T.), Pecharanin (N.), Taghshi (A.), Ijima (N.), Akima (Y) et Sone (M.). – Chinese Character Recognition by the auto Recognition system. *Proc. of the IEEE ICNN'95*, pp. 2222–2225. – Perth, Australie, 1995.
- [Panetsos93] Panetsos (F.), Garcia (A.) et Zaldival (J.M.). – Using Neural Networks in the Control of Chemical Reactors. *Proc. of the IEEE ICNN'93*, pp. 1759–1763. – San Francisco, Ca., USA, 1993.
- [Park94] Park (S.) et Yang (J.). – A hierarchical Neural Network approach to intelligent traffic Control. *Proc. of the IEEE ICNN'94*, pp. 3358–3362. – Orlando, Fa., USA, 1994.
- [Park95] Park (Y.K.) et Lee (G.). – Applications of Neural Networks in High-Speed Communications Networks. *IEEE Communications Magazine*, vol. 33, n° 10, Oct. 1995, pp. 68–74.

- [Parrot93] Parrot (M.), Achache (J.), Berthelie (J.J.), Blanc (E.), Deschamps (A.), Lefevre (F.), Menvielle (M.), Planet (J.L.), Tarits (P.) et Villain (J.P.). – High-Frequency Seismo-Electromagnetic Effects. *Phys. Earth Plan. Int.*, vol. 77, 1993, pp. 65–83.
- [Parten90] Parten (C.R.), Rap (R.M.) et Thomas (C.). – Neurocontrol Applied to Telerobotics for the Space Shuttle. *Proc. of the International Neural Network Conference (INNC'90)*, pp. 229–236. – Paris, France, 1990.
- [Patterson85] Patterson (D.A.). – Reduced Instruction Set Computers. *Communications of the ACM*, vol. 28, n° 1, 1985, pp. 8–21.
- [Peter95] Peter (J.), Muller (T.) et Freyer (R.). – Optimized Constraint Satisfaction Neural Network for Medical Image Segmentation. *Proc. of the IEEE ICNN'95*, pp. 2592–2595. – Perth, Australie, 1995.
- [Petersen82] Petersen (E. L.), Shapiro (P.), Adams (J.H.) et Burke (E.A.). – Calculation of Cosmic Rays Induced Soft Upsets and Scaling in VLSI Devices. *IEEE Transaction on Nuclear Science*, vol. 29, n° 6, Déc. 1982, pp. 2055–2063.
- [Petersen83a] Petersen (E. L.). – Single Event Upsets in Space: Basic Concepts. *Tutorial Short Course. IEEE NSREC'83*. – Juil. 1983.
- [Petersen83b] Petersen (E. L.), Langworthy (J.B.) et Diehl (S.E.). – Suggested Single Event Upset Figure of Merit. *IEEE Transaction on Nuclear Science*, vol. 30, n° 6, Déc. 1983, pp. 4533–4539.
- [Phatak95] Phatak (D.S.) et Koren (I.). – Complete and Partial Fault Tolerance of Feedforward Neural Nets. *IEEE Transactions on Neural Networks*, vol. 6, n° 2, 1995, pp. 446–456.

- [Pickel83] Pickel (J. C.). – Single Event Upsets Mechanisms and Predictions. *Tutorial Short Course*. IEEE NSREC'83. – Juil. 1983.
- [Plamondon94] Plamondon (R.). – Automatic Signature Verification. *Special Issue of Int'l Journal of Pattern Recognition and Artificial Intelligence*, Juin 1994.
- [Pottier94] Pottier (I.) et Burel (G.). – Identification and Authentication of Handwritten Signatures with a Connexionist Approach. *Proc. of the IEEE ICNN'94*, pp. 2948–2951. – Orlando, Fa., USA, 1994.
- [Ramacher92] Ramacher (U.). – SYNAPSE – A Neurocomputer That Synthesizes Neural Algorithms on a Parallel Systolic Engine. *Journal of Parallel and Distributed Computing*, vol. 1, n° 14, 1992, pp. 306–318.
- [Roehl95] Roehl (N.M.) et Teles De Azevedo (H.L.). – Fuzzy Artificial Neural Network Approach for Incipient Fault Detection and Isolation in Rotating Machines. *Proc. of the IEEE ICNN'95*, pp. 538–542. – Perth, Australie, 1995.
- [Rosenblatt58] Rosenblatt (F.). – The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, n° 65, 1958, pp. 386–408.
- [Rosenblatt62] Rosenblatt (F.). – *Principles of Neurodynamics*. – Washington DC, Spartan Books, 1962.
- [Ruan95] Ruan (H.) et Sankar (R.). – Applying Neural Network to Robust Keyword Spotting in Speech Recognition. *Proc. of the IEEE ICNN'95*, pp. 2882–2886. – Perth, Australie, 1995.
- [Rumelhart86a] Rumelhart (D.E.), Hinton (G.E.) et Williams (R.J.). – Learning Representations by Back-Propagation Errors. *Nature*, n° 323, 1986, pp. 533–536.

- [Rumelhart86b] Rumelhart (D.E.), Mc Clelland (J.L.) et Group (PDP Research). – *Parallel Distributed Processing: Foundations, vol. 1.* – Cambridge, UK, MIT Press, 1986.
- [Safavi95] Safavi (A.A.) et Romagnoli (J.A.). – Application of Wave-Nets to Modelling and Optimisation of a Multidimensional Chemical Process. *Proc. of the IEEE ICNN'95*, pp. 1724–1728. – Perth, Australie, 1995.
- [Saito95] Saito (T.) et Takefuji (Y.). – Neural Computing Approach to Japanese Electoral System. *Proc. of the IEEE ICNN'95*, pp. 2202–2207. – Perth, Australie, 1995.
- [Sato93] Sato (Y.), Shibata (K.), Asai (M.), Ohki (M.) et Sugie (M.). – Development of a High-Performance General Purpose Neurocomputer Composed of 512 Digital Neurons. *Proc. of the IJCNN'93*, pp. 1967–1970. – Nagoya, Japon, 1993.
- [Schmidhuber96] Schmidhuber (J.) et Heil (S.). – Sequential Neural Text Compression. *IEEE Transactions on Neural Networks*, vol. 7, n° 1, 1996, pp. 142–146.
- [Séquin90] Séquin (C.H.) et Clay (R.D.). – Fault Tolerance in Artificial Neural Networks. *Proc. of the IJCNN'90*, pp. 703–708. – San Diego, Ca., USA, Juin 1990.
- [Serebryakova92] Serebryakova (O.N.), Bilichenko (S.V.), Chmyrev (V.M.), Parot (M.), Rauch (J.L.), Lefeuvre (F.) et Pokotelov (O.A.). – Electromagnetic ELF Radiation from Earthquake Regions as Observed by Low-Altitude Satellites. *Geophys. Res. Lett.*, vol. 19, 1992, pp. 91–94.
- [Setiono94] Setiono (R.) et Lu (G.). – Image Compression Using a Feed-Forward Neural Network. *Proc. of the IEEE ICNN'94*, pp. 4176–4179. – Orlando, Fa., USA, 1994.
- [Shaeffer91] Shaeffer (D.L.), Kimbrough (J.R.), Denton (S.M.), Kashmitter (J.L.), Wilburn (J.W.), Davis (R.W.), Colella (N.J.)

- et Holtkamp (D.B.). – High Energy Proton SEU Test Results for the Commercially Available MIPS R3000 Microprocessors and R3010 Floating Point Unit. *IEEE Transactions on Nuclear Science*, vol. 38, n° 6, 1991, pp. 1421–1428.
- [Shen95] Shen (Z.), Clarke (M.), Jones (R.) et Alberti (T.). – A New Neural Network Structure for Detection of Coronary Heart Disease. *Neural Computing and Applications*, vol. 3, n° 3, 1995, pp. 172–178.
- [Shinozawa94] Shinozawa (K.), Fujii (M.) et Sonehara (N.). – A Weather Radar Image Prediction Method in Local Parallel Computing. *Proc. of the IEEE ICNN'94*, pp. 4210–4215. – Orlando, Fa., USA, 1994.
- [SIENA96] Simulation Initiative for European Neural Applications. <http://www.mbfys.kun.nl/snn/siena/>, 1996.
- [Siewiorek84] Siewiorek (D.P.). – Architecture of Fault-Tolerant System. *IEEE Transactions on Computers*, Oct. 1984, p. 9.
- [Sim95] Sim (A.), Parvin (B.) et Keagy (P.). – Invariant Representation and Hierarchical Network for Inspection of Nuts from X-Ray Images. *Proc. of the IEEE ICNN'95*, pp. 738–743. – Perth, Australie, 1995.
- [SNAP93] High-Performance Parallel Computing: SIMD Numerical Array Processor Data Sheet, 1993.
- [Solaiman94] Solaiman (B.) et Picart (D.). – Neural Networks Modelling of Biochemical Reactions. *Proc. of the IEEE ICNN'94*, pp. 3348–3351. – Orlando, Fa., USA, 1994.
- [Solla90] Solla (S.A.). – Supervised Learning and Generalization. *Neural Networks: Biological Computers or Electronic Brains*. pp. 21–28. – Springer Verlag, 1990.
- [Speidel92] Speidel (S.L.). – Neural Adaptive Sensory Processing for Undersea Sonar. *IEEE Journal of Oceanic Engineering*, vol. 17, n° 4, 1992, pp. 341–350.

- [Stevenson90] Stevenson (M.), Winter (R.) et Widrow (B.). – Sensitivity of Feedforward Neural Networks to Weights Errors. *IEEE Transactions on Neural Networks*, vol. 1, n° 1, 1990, pp. 71–80.
- [Tanaka89] Tanaka (H.). – Study of a High Reliable System against Electric Noise and Element Failures. *Proc. of the International Symposium on Noise and Clutter Rejection in Radars and Imaging Sensors*, pp. 415–420. – 1989.
- [Tarassenko90] Tarassenko (L.), Brownlow (M.J.) et Murray (A.F.). – VLSI Neural Networks for Autonomous Robot Navigation. *Proc. of the International Neural Network Conference (INNC'90)*, pp. 213–216. – Paris, France, 1990.
- [Tarassenko96] Tarassenko (L.), Whitehouse (R.), Gasparini (G.) et Harris (L.). – Neural Network Prediction of Relapse in Breast Cancer Patients. *Neural Computing and Applications*, vol. 4, n° 2, 1996, pp. 105–113.
- [Thiesing95] Thiesing (F.M.), Middelberg (U.) et Vornberger (O.). – Short Term Prediction of Sales in Supermarkets. *Proc. of the IEEE ICNN'95*, pp. 1028–1031. – Perth, Australie, 1995.
- [Tierney94] Tierney (J.) et Revell (N.). – Printed Cyrillic Character Recognition. *Proc. of the IEEE ICNN'94*, pp. 3856–3861. – Orlando, Fa., USA, 1994.
- [Trealeaven89] Trealeaven (P.C.). – Neurocomputers. *International Journal of Neurocomputing*, vol. 1, 1989, pp. 4–31.
- [Tsaptsinos93] Tsaptsinos (D.) et Leigh (J.R.). – Modelling of a Fermentation Process Using Multilayer Perceptrons : Epoch vs. Pattern Learning, Sigmoid vs. Linear Transfer Function. *Journal of Microcomputer Applications*, n° 16, Avr. 1993, pp. 125–136.

- [Tso95] Tso (S.K.) et Lin (N.L.). – Adaptive Neural Network Controller for Robot Manipulator Systems. *Proc. of the IEEE ICNN'95*, pp. 2320–2325. – Perth, Australie, 1995.
- [Ulbricht94] Ulbricht (C.). – Multi-Recurrent Networks for Traffic Forecasting. *Proc. of the American National Conference on Artificial Intelligence (AAAI'94)*, pp. 883–888. – Seattle, Washington, USA, 1994.
- [Valafar95] Valafar (F.), Valafar (H.), Ersoy (O.K.) et Schwartz (R.G.). – Comparative Studies of two Neural Network Architectures for Modeling Human Speech Production. *Proc. of the IEEE ICNN'95*, pp. 2056–2061. – Perth, Australie, 1995.
- [Vanhulle95] VanHulle (M.M.). – Learning Rate Adaptation Achieved in Unsupervised Competitive Learning: An Application to Noise Cancelling. *Proc. of the IEEE ICNN'95*, pp. 860–864. – Perth, Australie, 1995.
- [Vankeulan94] VanKeulan (E.), Colak (S.), Withagen (H.) et Hegt (H.). – Neural Network Hardware Performance Criteria. *Proc. of the IEEE Conference on Neural Networks*, pp. 1885–1888. – 1994.
- [Velazco92] Velazco (R.), Karoui (S.) et Martinet (B.). – A Low Cost Functional Test System: The FUTE16 Tester. *Proc. of the International Conference of Microelectronics (ICM'92)*, pp. 5.1.1.1–5.1.1.4. – Monastir, Tunisie, 1992.
- [Verma95] Verma (B.K.). – Handwritten Hindi Character Recognition Using multilayer Perceptron and Radial Basis Function Neural Networks. *Proc. of the IEEE ICNN'95*, pp. 2111–2115. – Perth, Australie, 1995.
- [Waibel88] Waibel (A.), Hanazawa (T.), Hinton (G.), Shikano (K.) et Lang (K.). – Phoneme Recognition: Neural Networks vs. Hidden Markov Model. *Proc. of the IEEE ICASSP'88*, pp. 107–110. – 1988.

- [Waibel89] Waibel (A.), Hanazawa (T.), Hinton (G.), Shikano (K.) et Lang (K.). – Phoneme Recognition Using Time Delay Neural Networks. *IEEE Transactions on Acoustic Speech Signal Processing*, vol. 37, n° 3, 1989, pp. 328–339.
- [Waldemark95a] Waldemark (J.), Dovner (P.O.) et Karlsson (J.). – Hybrid Neural Network Pattern Recognition System for Satellite Measurements. *Proc. of the IEEE ICNN'95*, pp. 195–199. – Perth, Australie, 1995.
- [Waldemark95b] Waldemark (J.) et Norqvist (P.). – Multiple Neural Network Filtering for In-Flight Calibration of Satellite Measurements. *Proc. of the IEEE ICNN'95*, pp. 507–510. – Perth, Australie, 1995.
- [Wang95] Wang (J.H.) et Lin (J.H.). – Adaptive Spatiotemporal Neural Network for Speech Recognition. *Proc. of the IEEE ICNN'95*, pp. 2876–2881. – Perth, Australie, 1995.
- [Watt95] Watt (R.C.), Sisemore (C.), Kanemoto (A.) et Mylrea (K.). – Artificial Neural Networks Facilitate Bispectral Analysis of Electroencephalographic Data. *Proc. of the IEEE ICNN'95*, pp. 2596–2599. – Perth, Australie, 1995.
- [Widrow85] Widrow (B.) et Stearns (S.D.). – *Adaptive Signal Processing*. – NJ, USA, Prentice-Hall, 1985.
- [Widrow88] Widrow (B.) et Winter (R.G.). – Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition. *IEEE Computer*, Mar. 1988, pp. 25–39.
- [Wolff93] Wolff (G.J.). – Sensory Fusion: Integrating Visual and Auditory Information for Recognizing Speech. *Proc. of the IEEE ICNN'93*, pp. 672–677. – San Francisco, Ca., USA, 1993.
- [Xu95] Xu (Z.M.), Packer (J.S.) et Cade (J.F.). – A Hybrid Fuzzy-Neural Control System for Management of Mean Ar-

- terial Pressure of Seriously Ill Patients. *Proc. of the IEEE ICNN'95*, pp. 2281–2286. – Perth, Australie, 1995.
- [Xue90] Xue (Q.), Hu (Y.H.) et Milenkovic (P.). – Analysis of the Hidden Units of the Multi-Layer Perceptron and its Application to Acoustic-to-Articulatory Mapping. *Proc. of the IEEE ICASSP'90*, pp. 869–872. – Albuquerque, NM, USA, 1990.
- [Yamamoto95] Yamamoto (I.). – Application of Neural Network to Marine Vehicle. *Proc. of the ICNN'95*, pp. 220–224. – Perth, Australia, 1995.
- [Yong95] Yong (C.K.). – Intact Egg Freshness Quality Inspection Using Neural Networks. *Proc. of the IEEE ICNN'95*, pp. 1080–1085. – Perth, Australie, 1995.
- [Yu95] Yu (X.), Dent (D.) et Osborn (C.). – The Selection of Weights Precision for Ballistocardiography Classification. *Proc. of the IEEE ICNN'95*, pp. 2485–2489. – Perth, Australie, 1995.
- [Zaradoshti95] Zardoshti Kermani (M.) et Afshordi (A.). – Classification of Chromosomes Using Higher-Order Neural Networks. *Proc. of the IEEE ICNN'95*, pp. 2587–2591. – Perth, Australie, 1995.
- [Zhang96] Zhang (M.) et Fulcher (J.). – Face Recognition Using Artificial Neural Networks Group-Based Adaptive Tolerance (GAT) Trees. *IEEE Transactions on Neural Networks*, vol. 7, n° 3, 1996, pp. 555–567.
- [Zheng94] Zheng (B.), Qian (W.) et Clarke (L.). – Multistage Neural Network for Pattern Recognition in Mammogram Screening. *Proc. of the IEEE ICNN'94*, pp. 3437–3447. – Orlando, Fa., USA, 1994.