

**LABORATOIRE d'INFORMATIQUE THEORIQUE
& APPLICATIONS DE MARSEILLE
L.I.T.A.M.**

Faculté des Sciences Economiques
Université d'Aix-Marseille II

**ISSN
0291-5413**

**INFORMATIQUE
FONDAMENTALE
&
APPLICATIONS**

BULLETIN N° 34

**Comité de
rédaction**

SOMMAIRE

**E. Bianco
R. Cusin
S. Hilala
P. Isoardi
J.M. Knippel
J.P. Lehmann
R. Stutzmann**

P 1 ... Editorial:
Informatique et décrépitude.
E. Bianco

P 5 ... Insertion fractionnée.
E. Bianco

P11 ... Un microprocesseur à jeu
d'instructions réduit, réalisé
en circuits programmables.
Roger Dupuy, Bernard Goosens

P 29 ... Vouzzavédibisar.

Dépositaire

**B.U. Sc. Eco.
Aix-Mars. II**

Mars 1993

Adresse postale:

Faculté des Sciences Economiques
LITAM

14 rue Puvis de Chavannes 13001 Marseille

Tel. 91 13 96 29 - (20, 21)

INFORMATIQUE ET DECREPITUDE.

E. Bianco

Non.

Non un professeur dont la tâche consiste à former des jeunes, à faire avancer la connaissance par une activité de recherche, ne peut pas en même temps devenir un "chef d'entreprise".

Certes l'Université est une entreprise. Mais est-il raisonnable de comparer le fonctionnement de ce genre d'organisme avec une industrie ou un commerce, dont le but est essentiellement de faire du fric et d'engraisser le fisc. Les privilèges de l'argent, les privilèges du pouvoir. Les récentes judiciaires chasses-à-l'homme montrent pour qui veut voir, malgré le brouillard épais dont on entoure les affaires, combien ces deux domaines sont d'ailleurs intriqués.

Question naïve: s'il est vrai que l'éducation coûte cher, combien coûte la non-éducation ?

Pourtant des gens qui se prétendent très sérieux estiment qu'il n'est plus possible de laisser un enseignement se développer indépendamment du monde où devront se placer les futurs travailleurs. On peut interpréter de diverses manières cette façon de voir.

Question naïve: on a trouvé spontanément quarante milliards, avoués c'est-à-dire sans doute au moins le triple, pour défendre les Emirs contre un Saddam, lesquels Emirs sont, comme l'on sait, prêts à signer des deux mains la convention des droits de l'homme (et de la femme), mais impossible de combler le trou de la sécurité sociale, curieux non ?

Il est parfaitement vrai qu'un effort important a été réalisé dans les quarante dernières années pour le développement de l'Université. Dans les années cinquante, les villes un peu importantes avaient par spécialité leur petite Faculté qui suffisait amplement aux quelques dizaines d'étudiants dont la moitié obtenaient des diplômes, les autres se contentaient amplement d'une raison sociale.

Question naïve: avez-vous eu l'occasion d'observer un digne professeur en train de faire le tapin soit au ministère, à Paris, soit auprès des détenteurs de pognon ?

L'ennui, quand on prend ce genre de décision, c'est qu'on déclenche un mécanisme dont il est difficile de rester maître. On comprend parfaitement l'inquiétude du Politicien qui sent, même s'il n'est pas très malin que les implications du processus ont tendance à se multiplier comme un vulgaire déchaînement

nucléaire. Qu'est-ce qu'il fait le Politicien qui s'est placé dans une situation aussi délicate.

- Comment ? il se ronge les doigts ?

- Mais je n'en sais rien Monsieur! Ce que je sais c'est qu'il fait appel au Spécialiste. Le malheur c'est que le spécialiste en l'occurrence, c'est le petit copain qu'on a connu au Rotary, ou bien au cours d'une partie de chasse chez le Comte, c'est-à-dire le mec qui n'a pas grand chose à faire dans son laboratoire. Observez les un peu, c'est surtout visible quand le Président chamboule le ministère et qu'on fait appel à des nouveaux. Vous en voyez rappliquer d'un peu partout des vieilles connaissances, pour "conseiller" l'Eminence. Et qui repartent la queue basse dans leur lointaine province quand l'Eminence a fait la preuve de son incompétence exactement comme ses prédécesseurs.

Question naïve: croyez vous qu'il soit judicieux de dénommer incompétence le fait d'être capable de faire adopter une poll tax qui a valu par ailleurs une magnifique veste électorale aux pires réactionnaires d'outre Manche, ou bien de récupérer au fonds commun (sans fond), une taxe réservée au fonds de solidarité, ou tous autres actes de même nature ?

La sélection naturelle ne perd pour autant jamais ses droits. Il y a quelques dizaines d'années, les gens avaient à cœur de ressembler plutôt à Einstein ou Poincaré ou Pasteur ou n'importe qui de ces gens qui avaient réussi à se forger un ambryon de savoir. De nos jours ce sont plutôt des découvreurs de pognon qui paraissent utiles à la République, aussi les nouveaux candidats encouragés par leurs aînés commencent-ils à ressembler plutôt à Séguela. Quand on sort des imbecillités par exemple sur la mémoire de l'eau ou sur la fusion froide ça vous retombe un jour ou l'autre sur ce qui vous sert de figure. Mais quand on vend du vent, on peut être tranquille, c'est à plus longue portée. Au pire ce sont les successeurs qui héritent.

On avait en fait deux solutions possibles. On démocratisait l'enseignement supérieur et on tentait de maintenir voire d'élever le niveau moyen, ou alors on démocratisait mais en scindant les masses en deux parties, l'une, minuscule pour les élites, et puis le reste. Bien entendu c'est le pire choix qui s'est peu à peu instauré. Facilité en cela par le fait historique qu'en France il existe déjà un enseignement réservé à l'élite qui continue à être puissamment soutenu, alors que l'université qui ne conserve de doré que son nom, réduite à la portion congrue, est peu à peu forcée d'aller se vendre. Vous me rétorquerez sans doute que l'ensemble des Universités représente une force importante, difficile à négliger, qui, unie ne se laisserait pas marcher sur les pieds par un quelconque gouvernement. Certes.

Mais c'est compter sans la mentalité des "nouveaux professeurs". Le brusque gonflement "démocratique" à fait appel à nombre de personnes issues de milieux dont la culture ne facilite pas la réflexion sur la stratégie sociale. Un détail: dans l'enseignement supérieur il n'existe plus de Professeurs, il n'y a plus que des "enseignants-chercheurs". Cette schizoïdie professionnelle, me semble-t-il montre le malaise de ces gens qui ne supportent plus guère une livrée dont la mode peut paraître bien surannée pour leur dynamisme très up to date. Cela explique peut-être la quantité d'entre eux qui tentent de se faire une noblesse, voie détournée, en passant par le suffrage universel.

Question naïve: imaginons que la dignité soit liée à l'idée que l'on se fait de son métier et à la qualité de son travail, comment se fait-il qu'aucun ou presque d'entre ces "nouveaux professeurs" n'ait réagi quand on leur a prouvé la déficience de leur prestation en doublant leur temps de service ?

Question subsidiaire: l'acte cité ci-dessus est attribué à un parti considéré comme populaire de par les élections dont il est issu, les élus des esclaves seraient-ils donc particulièrement doués pour faire marcher des esclaves ?

La fin du vingtième siècle sera-t-elle marquée par la transformation de la noble caste des professeurs en simples moines mendiants condamnés, vêtus de bure, muni du bourdon du pèlerin et la sébille à la main, à frapper à toutes les portes derrière lesquelles leur odorat développé aura perçu de l'argent ? Je ne pense pas que la chose soit si importante qu'il s'en cause longtemps dans les chaumières. D'ailleurs je pense qu'à brève échéance l'Université va devenir l'ultime refuge, une sorte de dernière chance pour les étudiants qui vont peut-être y commencer, voire y achever d'apprendre à lire. Oh, pas la lecture des philosophes, n'exprimons pas d'ambition démesurée, non, simplement la collection Arlequin, et peut-être même P.L. Sulitzer pour l'élite.

L'INSERTION FRACTIONNEE

E. Bianco

Subject Classification Informatics: C3, D3, F1.

Résumé.

En théorie des systèmes informatiques on peut imaginer des structures qui visent à minimiser la complication de tels programmes. Pour cela on est amené à imaginer des concepts qu'on introduit dans des langages de programmations adaptés au problème posé. Mais il est bon, et c'est l'objet de la discussion présentée ici, de mesurer l'impact de la mise en œuvre des notions correspondantes quant on utilise des outils traditionnels.

INSERTION FRACTIONNEE.

Par définition, l'instruction d'insertion fractionnée représente un noeud de communication dans la structure du système. Pour un système élémentaire, entendons par là un système centralisé unique, non séparé en parties multiples gérant chacune l'équivalent d'un ordinateur séparé, l'occurrence de cette instruction est unique puisqu'elle est la matérialisation de la notion d'autojectivité. En effet, pour être parfaitement contrôlable, toute tâche est divisée en quanta de déroulement finis-bornés. Or, pour une unité centrale unique, disons une machine universelle unique, un quantum doit être seul à se dérouler dans son intervalle de temps.

Une structure de système ouvert implique l'utilisation de la notion de sous-système. Or, un sous-système doit insérer une certaine diversité de compilateurs. Si l'on admet que dans le langage on introduit cette nouvelle notion d'insertion fractionnée, et qu'une telle instruction a une place unique, les insertions des compilateurs ne pourront donc se faire qu'au moyen des insertions simples.

Il nous faut pour autant respecter l'autojectivité qui, je le rappelle implique pour chaque quantum un fini borné à la fois en temps et en espace, or, à chaque insertion simple, par construction, la configuration des compilateurs va s'étaler, son contenu disparaissant lors de chaque retour systématique. Il devient donc nécessaire de construire des compilateurs dont la configuration soit à la fois standardisée, réduite à un minimum, et ne comportant que les valeurs temporaires nécessaires au travail dans le quantum. Cela signifie que toutes les variables du compilateur doivent se retrouver, pour demeurer pérennes, dans la configuration de la procédure qui joue le rôle de sous-système. En conséquence, la configuration de ce sous-système contient l'ensemble des variables de tous les compilateurs qu'il intègre.

Il s'agit maintenant de concevoir un moyen d'assurer la jonction compilateur- sous-système lors de l'insertion simple. Je vais faire quelques remarques.

- 1) Le compilateur, de toute évidence va travailler avec des index qui lui permettront, par rapport à l'origine de sa configuration, d'aller puiser dans les files du sous-système.
- 2) Le nombre de variables utilisées par un compilateur est a priori, quelconque. Donc ces variables vont se trouver dans la configuration du sous-système. Les index du compilateur lui permettront d'accéder à ces valeurs de variables.
- 3) Les valeurs des variables ainsi stockées seront soit des valeurs, soit des adresses. Ces dernières, seront forcément des adresses par rapport à l'origine de la configuration du compilateur, bien que stockées en configuration du sous-système. Ceci a des

conséquences sur la constance des positions relatives de ces deux sortes de configuration, d'un quantum à l'autre.

4) L'insertion simple transmet au compilateur, des valeurs d'adresses valables par rapport à l'origine de sa configuration.

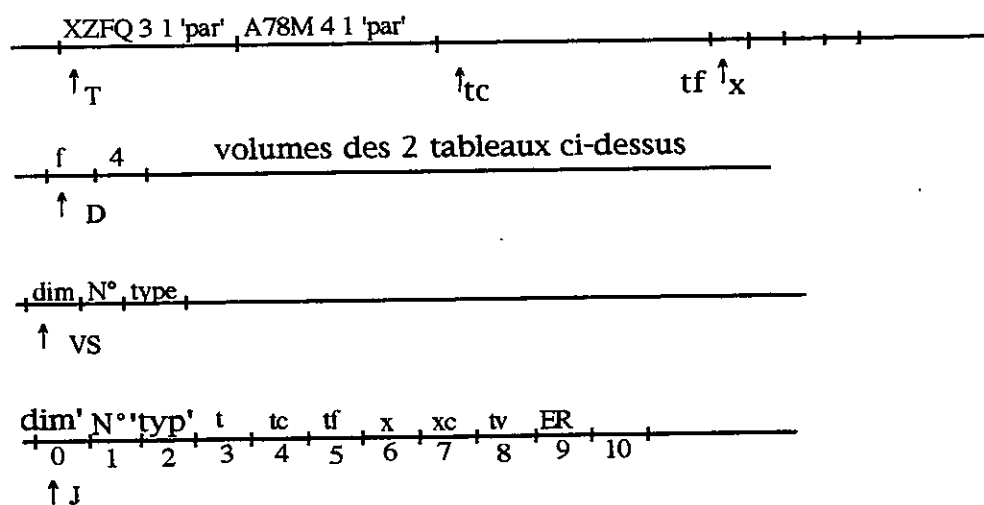
Je présente une solution qui permet au compilateur de se calculer ses jeux d'adresses qu'il replace en configuration du sous-système et auxquels il fait appel à chaque traitement de valeur. La partie de configuration du sous-système, réservée à un compilateur déterminé est, quel que soit le compilateur, subdivisée en un nombre suffisant de parties dont les adresses d'accès sont automatiquement transmises au compilateur par l'instruction d'insertion simple. Le nombre de parties ainsi définies conditionne le nombre de paramètres que fournit l'instruction d'insertion. Et ce nombre doit être le même quel que soit le compilateur.

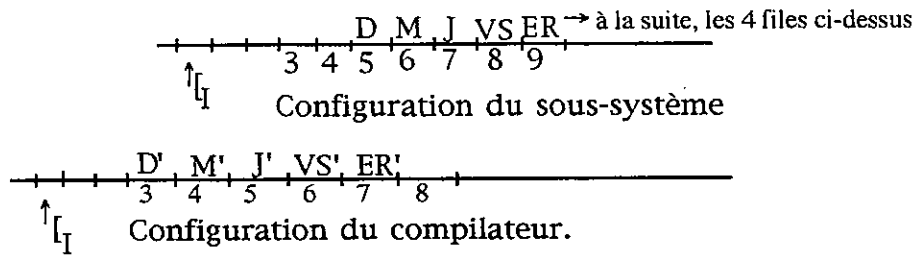
Je subdivise les données en quatre parties:

- 1) Les "n" tableaux alignés à la suite les uns des autres.
- 2) Un tableau constitué de n cases qui contient les volumes des n tableaux précédants.
- 3) Une suite de cases qui correspondent aux variables simples du compilateur.
- 4) Une suite de cases destinées à contenir les index qui vont servir au compilateur.

Je vais illustrer cette technique sur un exemple. Dans le compilateur du LAC j'extraie la procédure qui met en place dans le tableau des variables, une nouvelle variable dont le nom, la dimension, le numéro, le type sont connus. D'abord la procédure vérifie si la variable n'est pas déjà déclarée.

Je désigne par T l'adresse du tableau des variables, par D l'adresse des volumes de chacun des tableaux, ici un seul de volume "f". J'appelle VS l'accès à la liste des variables simples, et J l'accès à la liste des index.





En l'occurrence, si C est le nom du compilateur, l'insérer revient à écrire l'instruction d'insertion simple:

insérer C (5, 6, 7, 8, 9)

auquel cas, les adresses

D, M, J, VS, ER, valables par rapport à l'origine de la configuration du sous-système sont transmises en configuration du compilateur sous les valeurs:

D', M', J', VS', ER', valables par rapport à l'origine de la configuration du compilateur.

La version suivante de l'ajout d'une variable supplémentaire dans la file des variables représente une structure de compilateur. La procédure compilateur, ou fraction de compilateur, quel qu'il soit, dispose de 5 paramètres et de 4 index. Ces variables lui servent à calculer en configuration du sous-système, tous les index dont il a besoin, et, ce coup-ci en nombre quelconque. Dans cet exemple c'est la liste dont l'accès est l'index J.

procédure C

paramètres 5

index

var. loc. 0

début

Commutation contrôlée par l'insertion fractionnée.
L'initialisation ne se déroule qu'à la première insertion.

$l_{5,0} := l_6$

Calcul des index: Ad de dim,

$l_{5,1} := l_6 + 1$

Ad de N°,

$l_{5,2} := l_6 + 2$

Ad de type.

$l_{5,3} := l_4$

calcul de t,

$l_{5,4} := l_4$

de tc

$l_{5,5} := l_4 + l_{3,0}$

de $tf = t + f$

$l_{5,6} := l_4 + l_{3,0}$

de x (le tableau de l'identificateur est après celui des variables.)

$l_{5,7} := l_{5,6}$

de xc,

$l_{5,8} := l_{5,4}$

de tv.

fin initialisation

Commutation qui renvoie à la boucle E1, en cas d'ouverture de cette boucle. Cette commutation n'agit que pendant le déroulement de la boucle.

E1 : si $l_{5,8} = l_{5,3}$ vers F1
 $l_{5,8} := l_{5,8} - 7$
 $l_8 := l_{5,8}$
 $l_9 := l_{5,6}$
 si $l_{8,0} \neq l_{9,0}$ vers E1
 si $l_{8,1} \neq l_{9,1}$ vers E1
 si $l_{8,2} \neq l_{9,2}$ vers E1
 si $l_{8,3} \neq l_{9,3}$ vers E1

Sortie sur l'instruction <u>fin</u> en cas d'ouverture de boucle.

vers ER1

F1 :	si $l_{5,8} = l_{5,5}$ vers ER2	Sortie sur débordement de la file des variables
	$l_8 := l_{5,4}$	Index tc
	$l_9 := l_{5,6}$	x
	$l_{8,0} := l_{9,0}$	Identificateur de la variable
	$l_{8,1} := l_{9,1}$	"
	$l_{8,2} := l_{9,2}$	"
	$l_{8,3} := l_{9,3}$	"

$l_9 := l_{5,0}$
 $l_{8,4} := l_{9,0}$ dim
 $l_9 := l_{5,1}$
 $l_{8,5} := l_{9,0}$ N°
 $l_9 := l_{5,2}$
 $l_{8,6} := l_{9,0}$ type
 $l_{5,4} := l_{5,4} + 7$
 vers FIN

ER1 : $l_{7,0} := 'e1'$ Erreur, cet identificateur est déjà déclaré.
 FIN : fin

On constate que la partie initialisation qui ne doit se dérouler qu'une seule fois, sert à construire la liste des index nécessaires. Ces index, calculés pour être utilisés en configuration du compilateur, sont stockés et récupérés à la demande en configuration du sous-système. Bien entendu, l'évolution des adresses qui se calcule en configuration du compilateur, doit être mémorisée à chaque fois en configuration du sous-système. (Exemple: c'est ce que fait l'instruction qui suit E1.)

CONCLUSION.

La simple solution technique ainsi mise en évidence permet de montrer plusieurs propriétés. D'abord ce sont deux langages de nature différente qui sont mis en jeu. Un langage à structure de procédures indépendantes, c'est la Procédure Formelle, et un

langage à contrôle de temps et d'espace mémoire, c'est le PFS. Dans le cas de la Procédure Formelle, à chaque insertion d'une procédure il se crée une configuration qui ne sera abandonnée qu'à la sortie de l'algorithme de la procédure, quelle qu'en soit la durée du déroulement. Dans le cas du PFS, en tête se déclare le champ fictif qui est valable pour tout le programme. Dans le corps du texte, on peut insérer des sections qui ont le sens de la procédure du point de vue de l'algorithme, mais qui ne créent pas de configuration. Ce ne sont en fait que des changements de nom. Cela permet un contrôle total sur l'occupation de la mémoire utilisée. En conséquence, le problème se pose d'établir des relations entre des compilateurs différents, mais qui appartiennent à un même sous-système. Une file qui serait ainsi commune à un ou à plusieurs compilateurs serait déclarée externe à chacun d'entre eux.

Or, l'organe qui représente le point de contrôle de la circulation de l'information, est précisément l'insertion fractionnée. Cette instruction fait naturellement partie du PFS. Il est intéressant d'étudier ce qui se passe quand on introduit en procédure formelle les trois aspects de l'insertion fractionnée sans rien changer au reste. Il est bien clair, comme exposé plus haut, qu'une difficulté apparaît car on est obligé d'utiliser une insertion simple pour passer du sous-système au compilateur. La gestion de la place est à la charge du système, qui n'a à connaître des différents compilateurs qu'une structure de configuration standard. En effet, le sous-système est inséré par l'insertion fractionnée, les variables et files nécessaires aux différents compilateurs y sont, structurées comme nous l'avons vu, déclarées en tête. Il suffit au système qui dispose de cette information, d'y prévoir la place pour autant de configurations standard que ce qu'il y a de compilateurs dans le sous-système.

En résumé on peut se servir d'une insertion simple pour un compilateur, mais alors ce dernier travaille sur une configuration provisoire dont le nombre figé de paramètres et d'index, lui permet de travailler avec des valeurs d'adresses stockées dans la configuration insérante. Le coût de l'opération se mesure ainsi:

- 1) A la première insertion calcul d'un jeu d'index suffisant pour atteindre aux données, le tout se trouve en configuration insérante,
- 2) pour chaque opération, on calcule dans les index du compilateur qui servent pour obtenir les adresses à partir des valeurs en configuration insérante.
- 3) L'exemple le montre pour l'ouverture de boucle, il est nécessaire que le compilateur soit structuré pour être autojectif.

Il reste à dire deux mots sur l'intérêt d'une telle réflexion. On part de l'idée que le travail de l'ordinateur doit être partagé entre plusieurs tâches indépendantes pendant un même intervalle de

temps, et qu'il faut en même temps surveiller rigoureusement l'occupation de la mémoire centrale, par définition finie-bornée. Ce cas est plus général que ce qu'il pourrait paraître à première vue. En effet même l'utilisateur isolé devant son ordinateur personnel peut avoir à lancer plusieurs longues tâches simultanées: imprimer un texte important pendant qu'un calcul se déroule et qu'on construit un nouvel algorithme, etc. La puissance des plus petits ordinateurs permet désormais tout cela.

Le déroulement de chaque tâche doit donc être subdivisé en quanta. Le quantum devient ainsi une unité sémantique dont seul l'aspect extérieur nous intéresse ici. Pour le maîtriser il faut considérer ses limites dans le temps et dans l'espace mémoire, pour la question soulevée, ce sont les deux paramètres importants. Traditionnellement, ces deux dimensions sont contrôlées de force par le système, qui a donc aussi à charge de prendre toutes les précautions nécessaires, pour conserver les autres aspects sémantiques du quantum. L'idée développée ici part d'un point de vue différent. Le système est destiné à gérer des tâches dont il a tout à ignorer sauf la place qu'elles occupent. Il paraît raisonnable de faire en sorte que chacune des tâches soit alors autostructurée, c'est-à-dire pré-organisée en une suite de quanta finis-bornés. C'est précisément l'application de l'autojectivité, dont il est montré par ailleurs qu'elle est universellement applicable (Cf. PFS).

Cette manière de procéder décharge le système de toute précaution quant à la protection du quantum. Il s'agit alors de montrer ce qui se passe quand on utilise un langage traditionnel de machine, c'est le cas de la procédure formelle.

Or, il est bien connu que chaque élément sémantique, constitué d'une suite quelconque de quantas, est en fait la manifestation de deux sémantiques qui sont, l'une le tribut du problème traité, et l'autre le tribut dû à la structure de l'ordinateur. La deuxième partie peut être répartie de manière différente selon les choix. L'exemple construit plus haut montre un certain coût de programmation conséquence du choix de l'organisation. Le coût est ici explicite dans la mesure où il montre ce qu'il faut ajouter à la main comme calcul en quelque sorte parasite par rapport au traitement utile. Mais cela montre également ce qu'il est possible de faire pour rendre cette surcharge implicite par une prise en compte directement par la machine universelle, à partir du moment où ce que l'on rajoute est systématique.

Un microprocesseur à jeu d'instruction réduit réalisé en circuits programmables.

Roger DUPUY
L.I.A.
Université Paris VI
tour 55-65 pièce 104
4 place Jussieu
75005 Paris

Bernard GOOSSENS
I.B.P. L.I.T.P.
Université Paris VII
tour 55-56 pièce 118
4 place Jussieu
75005 Paris

Résumé: nous présentons l'implantation d'un microprocesseur. 16 bits à jeu d'instruction réduit. Il est réalisé à partir de circuits programmables de la famille des réseaux à logique cellulaire Xilinx. On décrit les quatre unités fonctionnelles qui le compose: ses 8 registres 16 bits, son unité arithmétique, logique et de décalage, son séquenceur câblé et son unité d'interface de bus. On fournit aussi son jeu d'instruction, et le chronogramme de son cycle d'exécution.

1) Introduction

Les L.C.A. (Logic Cell Array) de la famille 3000 de Xilinx se composent de blocs programmables (Configurable Logic Bloc ou C.L.B.) organisés en matrice (8*8, 10*10, 12*12, 14*14 ou 16*20 C.L.B.) et de blocs d'entrées-sorties (Input Output Bloc ou I.O.B.) répartis sur le pourtour du circuit (64, 80, 96, 120 et 144 I.O.B.).

Chaque C.L.B. regroupe deux bascules-D synchrones et un bloc combinatoire. Cinq entrées du C.L.B. servent de variables pour la partie combinatoire. On peut au choix implanter dans un C.L.B. une fonction booléenne à 5 variables ou deux fonctions booléennes de moins de 5 variables (à condition qu'elles n'en aient pas plus de 5 à elles deux). Le ou les résultats peuvent être mémorisés dans les bascules-D, ou exploités directement en sortie du C.L.B. D'autre part, les sorties des bascules peuvent être utilisées dans la partie combinatoire, en remplacement de deux entrées du C.L.B.

Chaque I.O.B. contient deux bascules-D (permettant de verrouiller une entrée ou une sortie du circuit; chaque I.O.B. est bidirectionnel). Les entrées peuvent aussi être directes. Quant aux sorties, on peut les bloquer avec des buffers trois-états.

Les I.O.B. et les C.L.B. peuvent être reliés les uns aux autres par des points de connexion programmables (liaison d'un bloc à son environnement immédiat), des matrices d'aiguillage (liaison à des blocs éloignés) ou des bus internes commandables par des buffers trois-états.

La programmation d'un L.C.A. consiste à fixer pour chaque I.O.B. son sens d'utilisation (entrée, sortie ou bidirectionnelle, directe ou bufferisée), pour chaque C.L.B. la fonction combinatoire qu'il calcule et enfin comment les différents morceaux sont reliés entre eux.

2) Le microprocesseur.

Sur le plan des registres, il regroupe 8 registres 16 bits (r0 à r7) à tout faire (calculs 8 et 16 bits et adressage), dont un (r7) peut servir de pointeur de pile; un registre de conditions (appelé W, 8 bits dont 2 inutilisés: interruptions permises, demi-retenue, dépassement de capacité, nullité, retenue et signe); et un compteur ordinal (PC) de 16 bits.

L'unité arithmétique, logique et de décalage permet d'effectuer des opérations 8 et 16 bits: +, -, \wedge , \vee , \oplus , non, moins unaire, *2, /2, rotation gauche et droite.

Le microprocesseur peut être connecté à une R.A.M. de 64ko (adressage sur 16 bits) et à des périphériques via des ports 8 ou 16 bits (jusqu'à 64k). Il est relié à l'extérieur par trois bus 16 bits: adresse, donnée et contrôle. L'accès à la mémoire se fait par octet, ou par mot de 16 bits (deux octets obligatoirement alignés).

Toutes les instructions s'exécutent en un cycle, y compris les instructions de contrôle et celles d'accès à la mémoire. Deux cycles successifs se recouvrent partiellement grâce à un pipe-line (la phase d'exécution de l'instruction courante se déroule en même temps que se prépare l'adresse de la suivante (PC+2, PC+déplacement)).

3) Le jeu d'instructions.

Les instructions sont codées sur 16 bits. Un programme doit être aligné en mémoire, si bien que toute adresse de code est paire (PC est en réalité un registre de 15 bits).

Les instructions de contrôle permettent d'effectuer des sauts relatifs (VERS d12, SI cond VERS d8) ou absolus (VERS Ri), des appels relatifs (APPEL d15) ou absolus (APPEL Ri) et le retour correspondant (RETOUR).

Pour les registres Ri, on peut en échanger l'octet bas et l'octet haut, effectuer des transferts 8 (partie basse) ou 16 bits, des comparaisons, les empiler (en se servant du registre R7 comme pointeur de pile) et les dépiler (PUSH Ri, POP Ri).

On peut lire et écrire en mémoire (adressage indirect par registre), un ou deux octets (alignés) à la fois ($Ri := M(Rj)$, $Ri_bas := M(Rj)$, $M(Rj) := Ri$, $M(Rj) := Ri_bas$). La lecture d'un mot peut s'accompagner d'une post-incrémentation ou d'une pré-décrémentation du registre d'adresse ($Ri := M(Rj++)$, $Ri := M(--Rj)$, $M(Rj++) := Ri$, $M(--Rj) := Ri$). On peut aussi lire et écrire sur un port (dont le numéro est fourni par un registre: $Ri_bas := P(Rj)$ et $P(Rj) := Ri_bas$).

Pour les opérations, elles ont toujours lieu entre opérandes registres, de 8 (octet bas) ou 16 bits. Les opérations arithmétiques 16 bits opèrent en tenant compte ou pas de la retenue.

Enfin, les registres peuvent recevoir une valeur immédiate sur 8 bits (partie basse; $Ri_bas := k8$). On peut aussi comparer (CMP Ri_bas , k8), opérer ($Ri := Ri + k8$, $Ri_bas := Ri_bas \text{ ET } k8$, idem avec OU et OUX). On peut initialiser W ($W := k8$), le sauver dans Ri ($Ri := W$) et le récupérer ($W := Ri$).

Voici leur mnémotechnique et le code binaire associé (d15, d12, d8 et k8 sont des constantes; c4 est l'un des 16 mots suivants: DR (demi retenue), DC (dépassement de capacité), ZR (nul), CY (retenue), SG (positif), NDR (pas de demi retenue), NDC (pas de dépassement de capacité), NZR (non nul), NCY (pas de retenue), NSG (négatif), IEGN (\leq des entiers naturels), SEGN (\geq dans N), INFZ ($<$ des entiers relatifs), IEGZ (\leq dans Z), SUPZ ($>$ dans Z) ou SEGZ (\geq dans Z); M(adr) est un accès à la mémoire et P(adr) à un port; exts est l'extension signée d'un octet à 16 bits; opl est l'un des 3 mots ET, OU, OUX (ou exclusif); op1 l'une des 8 opérations: - (moins unaire), -CY (idem avec propagation de retenue), DG (décalage à gauche), DD (décalage à droite), DGCY (avec propagation de retenue sur le bit faible), DDCY (avec propagation de retenue sur le bit fort), +CY (propagation de retenue additive), -CY (propagation de retenue soustractive); op2, l'une des 4 opérations + (addition binaire), - (soustraction binaire),

+CY+ (addition avec propagation de retenue), -CY- (soustraction avec propagation de retenue); op3 l'une des 3 opérations: NON, RG (rotation gauche), RD (rotation droite):

mnémonique	fonction	code binaire 16 bits
APPEL d15	R7:= R7-2 M(R7):= PC PC:= PC+(2*d15)	0...d15...
VERS d12	PC:= PC+(2*d12)	1000...d12...
SI c4 VERS d8	si c4 => PC:= PC+(2*exts(d8))	1001...c4...d8...
W:= k8	W:= k8	10100000...k8...
Ri_bas:= k8	Ri<0..7>:= k8	10101...i.....k8...
Ri:= Ri+k8	Ri:= Ri + exts(k8) W<0..4>:= indicateurs	10110...i.....k8...
CMP Ri_bas, k8	Ri<0..7> - k8 W<0..4>:= indicateurs	10111...i.....k8...
Ri_bas:= opl k8	Ri<0..7>:= Ri<0..7> opl k8 W<0..4>:= indicateurs	110...opl...i...k8...
Ri_bas:= Rj_bas	Ri<0..7>:= Rj<0..7>	111000...i...j...10000
Ri_bas:= M(Rj)	Ri<0..7>:= M(Rj)	111000...i...j...10001
Ri_bas:= P(Rj)	Ri<0..7>:= P(Rj)	111000...i...j...10010
M(Rj):= Ri_bas	M(Rj):= Ri<0..7>	111000...i...j...10101
P(Rj):= Ri_bas	P(Rj):= Ri<0..7>	111000...i...j...10110
Ri_bas:= opl Rj_bas	Ri<0..7>:= Ri<0..7> opl Rj<0..7> W<0..4>:= indicateurs	111...opl...i...j...10111
Ri:= Rj	Ri:= Rj	111000...i...j...00000
Ri:= M(Rj)	Ri:= M(Rj)	111000...i...j...00001
Ri:= M(Rj++)	Ri:= M(Rj) Rj:= Rj+2	111000...i...j...00010
Ri:= M(--Rj)	Rj:= Rj-2 Ri:= M(Rj)	111000...i...j...00011
mnémonique	fonction	code binaire 16 bits

M(Rj):= Ri	M(Rj):= Ri	111000...i...j...00101
M(Rj++):= Ri	M(Rj):= Ri Rj:= Rj+2	111000...i...j...00110
M(--Rj):= Ri	Rj:= Rj-2 M(Rj):= Ri	111000...i...j...00111
Ri:= op2 Rj	Ri:= Ri op2 Rj W<0..4>:= indicateurs	111...op2...i...j...00100
CMP Ri, Rj	Ri - Rj W<0..4>:= indicateurs	11100...i...j...01001
Ri:= op1	Ri:= op1 Ri W<0..4>:= indicateurs	11100...i...op1...01010
APPEL Ri	R7:= R7-2 M(R7):= PC PC:= Ri	11100...i...11101100
VERS Ri	PC:= Ri	11100...i...00001101
PUSH Ri	R7:= R7-2 M(R7):= Ri	11100...i...11101110
POP Ri	Ri:= M(R7) R7:= R7+2	11100...i...11101111
Ri_bas:= op3	Ri<0..7>:= op3 Ri<0..7> W<0..4>:= indicateurs	11101...i...0op3...11010
CMP Ri_bas, Rj_bas	Ri<0..7> - Rj<0..7> W<0..4>:= indicateurs	11101...i...j...11001
ECH Ri_bas, Ri_haut	Ri<0..7> \Leftrightarrow Ri<8..15>	11101...i...00011100
W:= Ri-bas	W:= Ri_bas	11101...i...00011110
Ri_bas:= W	Ri_bas:= W	11101...i...00011111
RETOUR	PC:= M(R7) R7:= R7+2	1111111111111000

4) Organisation interne des unités fonctionnelles.

Le microprocesseur est implanté sur 5 circuits Xilinx 3042 (144 C.L.B. et 74 I.O.B. utilisables). Sur le schéma n°1 (les schémas sont en annexe à la fin de l'article), chaque carré représente un 3042. Le premier contient entre autres les registres PC et W. Le second et le troisième sont presque identiques l'un à l'autre et renferment les registres. Le quatrième contient l'U.A.L., des décaleurs combinatoires et la logique de décodage du code opération de l'instruction suivante. Le cinquième, situé sous les quatre autres, est l'unité d'interface de bus.

Ces 5 Xilinx sont reliés entre eux par cinq bus: le bus G-D (il joue deux rôles -voir ci-dessous-, d'où son nom: gauche-donnée), le bus D-A (droit-adresse), le bus R (résultat), le bus I (indicateurs) et le bus C (commandes). Tous sont de largeur 16 bits

sauf I qui est de 5 bits. Quelques signaux isolés supplémentaires apparaissent sur le schéma (carry, rdy, intr, rst et clk: retenue, ready, signal d'interruption, reset et horloge).

Le bus G-D achemine le contenu d'un registre Ri vers l'entrée gauche de l'U.A.L. (opérande gauche) ou vers la mémoire via l'unité d'interface de bus (écriture), ou encore véhicule un mot de la mémoire vers un registre (lecture) ou vers PC (RETOUR).

Le bus D-A sert soit de voie d'adresse en lecture (fetch y compris) ou écriture en mémoire, soit de voie pour l'opérande droit de l'U.A.L.

Le bus R dirige le résultat de l'U.A.L. vers un registre Ri. Il sert aussi ponctuellement dans l'exécution en un cycle de l'instruction APPEL Ri: la sauvegarde de PC en pile utilise les bus G-D (PC) et D-A (R7); le registre Ri est copié dans PC via le bus R, puisque les autres bus sont occupés.

Les indications produites par une opération de l'U.A.L. sont envoyées vers W via le bus I.

5) Le cycle machine.

Deux horloges sont utilisées: H2 et H1 (H2 a une fréquence double de celle de H1). Le cycle machine est égal au cycle de H1. Le schéma n°2 en est un chronogramme simplifié.

Sur le front descendant de H1, la phase d'exécution du cycle précédent se termine; le nouveau cycle commence par le fetch: les registres à modifier (dont PC et peut-être W) sont chargés. PC reçoit, selon le contenu actuel de W (sauts conditionnels), et selon le code de l'instruction qui vient de se terminer, soit PC+2 (pas de rupture de séquence), soit PC+d (d, le déplacement, est une constante figurant dans le code opération actuel), soit le bus G-D (RETOUR), soit le bus R (APPEL Ri). Cette nouvelle valeur de PC est copiée sur le bus D-A.

Le code opération de l'instruction suivante est lu en mémoire et entre dans les divers registres COP (le code est dupliqué dans tous les Xilinx où il est nécessaire) sur le front descendant de H2.

La phase de décodage commence; les commandes sont produites progressivement et certaines sont actives avant même que la phase d'exécution proprement dite ne commence, sur le front montant de H1.

Au niveau haut de H1, se déroulent en parallèle l'exécution de l'instruction courante (ce qui met en jeu les registres, les bus, l'U.A.L. et la RAM en cas de lecture ou d'écriture), et le calcul de l'adresse de l'instruction suivante (PC+2 ou PC+déplacement). Le cycle se termine quand H1 redescend.

On peut constater que lors de l'exécution d'instructions ne faisant pas accès à la mémoire, celle-ci est disponible pendant les trois-quarts du cycle. Ce peut être particulièrement utile dans une machine multiprocesseur pour effectuer des communications inter-processeur par D.M.A.

6) L'unité d'interface de bus.

Le schéma n°3 représente l'intérieur du cinquième Xilinx (unité d'interface de bus). En haut à droite on trouve le bus D-A, qui peut être copié sur le bus d'adresse externe en ouvrant les barrières trois-états. La commande d'ouverture dépend du code

opération de l'instruction en cours d'exécution, mémorisé dans le registre interne COP (accès mémoire ou port et fetch; le rectangle représente un registre de 16 bascules).

Le bus G-D, bidirectionnel à cet endroit, peut recevoir ce qui se trouve sur le bus de donnée externe (ouverture des portes trois-états commandant l'entrée). Pour la sortie, il faut prévoir le cas d'une écriture d'un octet en adresse impaire: l'octet à écrire doit passer de la partie basse du bus G-D (sortie d'un registre Ri_{bas}) à la partie haute du bus de donnée (partie haute=adresse impaire, partie basse=adresse paire). D'où le multiplexeur commandé par le bit 0 de l'adresse et par COP (instruction d'écriture de 8 bits). La sortie sur le bus de donnée est bien entendue contrôlée par des portes trois-états (ouvertes lors d'instructions d'écriture ou d'appel).

Du Xilinx, on devrait utiliser pour ce montage environ 30 C.L.B. (sur 144) et exactement 74 I.O.B. (sur 74).

7) Le séquenceur câblé.

Le schéma n°4 représente l'intérieur du premier Xilinx. On y retrouve les registres PC et W, ainsi qu'une copie du code opération de l'instruction en cours d'exécution (COP).

L'entrée de PC (15 bascules) est choisie parmi les 4 voies suivantes: PC+2 (le circuit désigné +2 est un incrémenteur combinatoire), PC+d (+ désigne un additionneur rapide 16 bits; le déplacement d est fabriqué à partir de COP: d8, d12 ou d15 étendu sur 16 bits selon son signe, et décalé d'un cran à gauche (*2): cette transformation est réalisée par le circuit dénommé EXTS), le bus R et le bus G-D. Le choix PC+d est soumis à une condition établie d'après COP (saut relatif) et selon le contenu de W (saut conditionnel): le circuit SAUT fixe la voie sortant du multiplexeur MUX. La commande sélectionnant la bonne entrée (choixpcin1) provient du Xilinx contenant la logique de décodage (Xilinx n°4 contenant aussi l'U.A.L.). La commande pcin (autorisation d'entrée dans PC) est toujours à l'état haut, sauf quand l'extérieur du microprocesseur en décide autrement (Ready=0: Wait State demandé).

PC est copié sur le bus D-A (fetch) lorsque H1 descend (ouverture des buffers trois-états) et y reste jusqu'à ce que H2 redescende. Il est copié sur le bus G-D en phase d'exécution, lors d'un appel (sauvegarde de l'adresse de retour).

Le registre W (8 bascules) reçoit en entrée sur ses 5 bits de poids faible soit le bus I (opération), soit le bus G-D ($W:= Ri$) soit une partie du COP courant ($W:= k8$). Pour les trois bits de poids fort, le choix se limite à COP et bus G-D. Les commandes win, wout et choixwin2 proviennent du Xilinx n°4.

W peut être copié sur le bus G-D (exécution de l'instruction $Ri:= W$). Le bit de retenue est acheminé jusqu'à l'U.A.L. (carryout pour les opérations avec propagation de retenue).

Le Xilinx n°1 devrait utiliser environ 99 C.L.B. (sur 144) et 66 I.O.B. (sur 74).

8) Les registres.

Le schéma n°5 représente les quatre registres R0-R3. Un second Xilinx est nécessaire pour implanter les quatre autres registres de la machine (R4-R7). Tous deux sont quasiment identiques.

Chaque registre peut recevoir en entrée soit le bus R (en haut à gauche; résultat d'une opération provenant de l'U.A.L.), soit son propre contenu incrémenté ou

décémenté (accès mémoire accompagné d'une modification du registre d'adresse; la voie choisie sort du multiplexeur placé au-dessus de R0). On peut aussi y ranger le contenu du bus G-D, en en permutant éventuellement (c'est le rôle des deux multiplexeurs commandés par le signal swap provenant du Xilinx n°4) l'octet de poids fort et celui de poids faible ($Ri_bas \Leftrightarrow Ri_haut$ ou lecture dans Ri_bas d'un octet de RAM d'adresse impaire).

Les registres peuvent être modifiés en totalité ou seulement leur octet faible ($Ri:=Rj$ ou $Ri_bas:=Rj_bas$): deux signaux clock enable entrent -ech_i et ecl_i-, ech_i agissant sur les C.L.B. associés à l'octet fort de Ri (autorisation d'entrée commandée par h-en) et ecl_i sur ceux de l'octet faible.

Deux registres peuvent être modifiés simultanément (c'est le cas par exemple dans l'instruction $R0:=M(R1++)$; R0 reçoit le contenu du bus G-D et R1 sa propre valeur incrémentée). Le code opération comprend, pour chacun des registres visés, son numéro (bits 8 à 10 pour Ri et 5 à 7 pour Rj, décodés juste sous le registre COP). Les commandes inri et inj autorisent l'entrée dans aucun, l'un ou les deux registres Ri et Rj. Ce dernier ne peut recevoir son nouveau contenu que du bus ++ (choixG-D(j)=0); Pour Ri, l'entrée provient soit du bus G-D ($riinbusG-D=1 \Rightarrow choixG-D(i)=1$) soit du bus R ($riinbusG-D=0$).

Du côté des sorties, les registres peuvent être copiés sur le bus G-D (portes trois-états situées juste sous chacun des registres; écriture en mémoire ou opération binaire) ou sur le bus D-A (deux multiplexeurs ont été ajoutés pour des raisons liées à l'architecture des Xilinx: un C.L.B. ne peut être connecté qu'à un seul bus trois-états; on passe donc les sorties artificiellement dans d'autres C.L.B. et on en profite pour faire une sélection, ce qui diminue le nombre de buffers tri-state utilisés; sur le bus on retrouve le contenu exact du registre choisi ou sa valeur décrementée: accès en pré-décrementation; c'est le rôle du circuit -2 qui est un décrementeur combinatoire; le circuit +2 est un incrémenteur; tous deux peuvent être inhibés par leur commande respective --en/++en). On peut aussi sortir un registre sur le bus R (pour l'instruction APPEL Ri; deux multiplexeurs commandés par le bit 8 de COP servent à sélectionner le registre admis sur le bus). D'autre part, le registre COP bits 0 à 7 peut être envoyé sur la partie basse du bus G-D ($Ri_bas:=k8$).

Pour le Xilinx des registres R4-R7, quatre modifications sont nécessaires (elles figurent en bas à droite du schéma n°5): la validation des décodeurs, réalisée à partir du bit de poids fort du numéro de registre, est inversée: bit 10 et 7 à 1 plutôt que 0; il en est de même pour la sortie sur le bus R (bit 10) et sur le bus D-A (bit 7); enfin, l'entrée dans le registre R7 est validée lors d'un APPEL d15 (COP ne contient pas le code j (7); pour cette raison, inj est mis à 0 et c'est COP-15 qui joue son rôle).

Les commandes swap, h_en, outcopbusG-D, outribusG-D, --en, ++en, outrjbusD-A, riinbusG-D, inri, inj et appelRi proviennent du Xilinx n°4 (U.A.L. et logique de décodage).

Dans ce Xilinx, on emploie 124 C.L.B. et 58 I.O.B.

9) La logique de décodage.

Le schéma n°6 correspond à la partie de décodage du Xilinx n°4 (qui contient en plus l'U.A.L.). On y décode le code opération de l'instruction courante COP par morceaux:

-les bits 13 à 15:	0 à 3: APPEL;
	4: SAUT;
	5, 6: opération avec valeur immédiate;

7: autres instructions
 les sorties du décodeur sont dénommées 15-0 à 15-7;

-les bits 11 et 12: pour le code 5 ci-dessus seulement:
 0: W:= k8;
 1: Ri_bas:= k8;
 2: Ri:= Ri+k8;
 3: CMP Ri_bas, k8;

les sorties du décodeur sont dénommées 12-0 à 12-3;

-les bits 3 et 4: pour le code 7 seulement:
 0: groupe 0 (opérations sur mot);
 1: groupe 1 (autres opérations sur mot);
 2: groupe 2 (opérations sur octet);
 3: groupe 3 (autres opérations sur octet);

les sorties du décodeur sont dénommées 4-0 à 4-3;

-les bits 0 à 2: pour le code 7 seulement:
 8 sous-groupes dans chaque groupe;
 les sorties du décodeur sont dénommées 2-0 à 2-7.

Juste après les décodeurs figurent les fonctions booléennes (représentées par des rectangles) associées aux différentes commandes pilotant les autres Xilinx:

appelri= 4-1 ET 2-4;

win= (15-5 ET ¬12-1) OU 15-6 OU (4-2 ET 2-7)
 OU (4-0 ET 2-4) OU (4-1 ET 2-1) OU (4-1 ET 2-2)
 OU (4-3 ET 2-2) OU (4-3 ET 2-1) OU (4-3 ET 2-6);

wout= 4-3 ET 2-7;

choixwin2= 4-3 ET 2-6;

outcophbusG-D= 15-5 ET 12-1;

outribusG-D= (15-5 ET 12-2) OU (15-5 ET 12-3) OU 15-6
 OU (4-2 ET 2-5) OU (4-2 ET 2-6) OU (4-2 ET 2-7)
 OU (4-0 ET 2-5) OU (4-0 ET 2-6) OU (4-0 ET 2-7)
 OU (4-0 ET 2-4) OU (4-1 ET 2-1) OU (4-1 ET 2-2)
 OU (4-1 ET 2-5) OU (4-1 ET 2-6) OU (4-3 ET 2-2)
 OU (4-3 ET 2-1) OU (4-3 ET 2-4) OU (4-3 ET 2-6);

outrjbusD-A= ¬(15-4 OU 15-5 OU 15-6 OU (4-1 ET 2-2)
 OU (4-1 ET 2-5) OU (4-3 ET 2-2) OU (4-3 ET 2-4)
 OU (4-3 ET 2-6) OU (4-3 ET 2-7));

--en= ¬15 OU (4-0 ET 2-3) OU (4-0 ET 2-7) OU (4-1 ET 2-4)
 OU (4-1 ET 2-6);

++en= (4-0 ET 2-2) OU (4-0 ET 2-6) OU (4-1 ET 2-7)
 OU (4-3 ET 2-0);

swap= (a0 ET (4-2 ET 2-1) OU (4-2 ET 2-2)) OU (4-3 ET 2-4);

h-en= ¬((15-5 ET 12-1) OU 15-6 OU (4-2 ET 2-0)
 OU (4-2 ET 2-1) OU (4-2 ET 2-2) OU (4-2 ET 2-7)

OU (4-3 ET 2-2) OU (4-3 ET 2-7));

choixpcin1= (4-1 ET 2-4) OU (4-1 ET 1-5) OU (4-3 ET 2-0);

inri= (15-5 ET 12-1) OU (15-5 ET 12-2) OU 15-6
 OU (4-2 ET 2-0) OU (4-2 ET 2-1) OU (4-2 ET 2-2)
 OU (4-2 ET 2-7) OU (4-0 ET 2-0) OU (4-0 ET 2-1)
 OU (4-0 ET 2-2) OU (4-0 ET 2-3) OU (4-0 ET 2-4)
 OU (4-1 ET 2-2) OU (4-1 ET 2-7) OU (4-3 ET 2-2)
 OU (4-3 ET 2-4) OU (4-3 ET 2-7);

inrj= (4-0 ET 2-2) OU (4-0 ET 2-3) OU (4-0 ET 2-6)
 OU (4-0 ET 2-7) OU (4-1 ET 2-4) OU (4-1 ET 2-6)
 OU (4-1 ET 2-7) OU (4-3 ET 2-0);

riinbusG-D= (15-5 ET 12-1) OU (4-2 ET 2-0) OU (4-2 ET 2-1)
 OU (4-2 ET 2-2) OU (4-0 ET 2-0) OU (4-0 ET 2-1)
 OU (4-0 ET 2-2) OU (4-0 ET 2-3) OU (4-1 ET 2-7)
 OU (4-3 ET 2-4) OU (4-3 ET 2-7);

10) L'U.A.L.

C'est évidemment sur elle que repose l'efficacité du microprocesseur. Il est indispensable d'en réaliser une version rapide, avec, pour les opérations arithmétiques, un calcul anticipé des retenues. Cependant, les schémas classiques d'un 74/381 et d'un 74/382 (UAL et carry look-ahead) ne sont implantables directement dans un Xilinx qu'au prix d'un trop grand nombre de C.L.B. (les fonctions booléennes internes calculées dans ces circuits qui ont plus de 4 variables seront réalisées par deux C.L.B. en série, ce qui augmente la complexité et diminue l'efficacité). La bonne façon consiste à reprendre les équations et à faire apparaître le maximum de fonctions de 4 ou 5 variables.

Rappelons que l'U.A.L. doit pouvoir effectuer les opérations suivantes: addition, soustraction, et, ou, ou exclusif et complémentation (les décalages sont effectués par un décaleur combinatoire séparé). Elle doit bien entendu produire les indications nécessaires, à savoir: demi-retendue, dépassement de capacité, résultat nul, retenue, signe.

Un C.L.B. unique se prête tout à fait à l'implantation d'un additionneur/soustracteur ordinaire (un bit: $a+b+re$ ou $a-b-re$ selon op (0 pour + et 1 pour -)):

$$s = a \oplus b \oplus re \quad (1 \text{ fonction de 3 variables})$$

$$rs = ((a \oplus op) \vee b) \cdot re \vee (a \oplus op) \cdot b \quad (1 \text{ fonction de 3 variables})$$

La partie combinatoire d'un C.L.B. a 5 entrées, dénommées .a, .b, .c, .d et .e, et deux sorties appelées .f et .g. Pour notre additionneur/soustracteur d'un bit, on les fixe comme suit: .a: a, .b: b, .c: re et .d: op. Les sorties sont .f: s et .g: rs.

Pour un additionneur/soustracteur 16 bits, nous allons calculer certaines retenues par avance. Calculons à partir de chaque couple $(a_i \oplus op, b_i)$ ($0 \leq i < 16$) les termes:

$$\begin{aligned}
 P_i &= (a_i \oplus_{op}) \vee b_i && \text{(propagation)} \\
 G_i &= (a_i \oplus_{op}) \wedge b_i && \text{(génération)} \\
 M_i &= (a_i \oplus b_i) && \text{(somme modulo 2)}
 \end{aligned}$$

Voici comment nous obtenons s_0, s_1 et r_1 :

$$\begin{aligned}
 s_0 &= M_0 \oplus re \\
 s_1 &= M_1 \oplus r_0 = M_1 \oplus (P_0.re \vee G_0) \\
 r_1 &= P_1.r_0 \vee G_1 = P_1.P_0.re \vee P_1.G_0 \vee G_1
 \end{aligned}$$

De même pour s_2, s_3 et r_3 à partir de $M_2, r_1, M_3, P_2, G_2, P_3$ et G_3 . Il faut 8 C.L.B pour fournir les différents P_i , 8 autres pour les G_i , encore 8 pour les M_i et 4 pour les s_j et r_j . La sortie r_3 est disponible après stabilisation de trois étages de C.L.B. (M_i, P_i, G_i puis r_1 puis r_3).

Pour anticiper encore le calcul des autres retenues, on calcule:

$$\begin{aligned}
 N_i &= a_i \oplus_{op} \\
 Q_i &= (\neg N_i \neg N_{i-1} b_i b_{i-1}) \vee (\neg N_i N_{i-1} b_i \neg b_{i-1}) \vee (N_i \neg N_{i-1} \neg b_i b_{i-1}) \\
 H_i &= (\neg N_i \neg N_{i-1} b_i) \vee (N_{i-1} b_i b_{i-1})
 \end{aligned}$$

Q_i est le terme de propagation de la retenue r_{i-2} du rang $i-1$ au rang $i+1$, et H_i celui de génération pour les deux rangs $i-1$ et i . On ne calcule que $Q_5, Q_7, Q_9, Q_{11}, Q_{13}$ et Q_{15} ainsi que les H_i de même rang. D'où:

$$\begin{aligned}
 S_i &= M_i \oplus r_{i-1} \\
 S_{i+1} &= M_{i+1} \oplus r_i = M_{i+1} \oplus (P_i.r_{i-1} \vee G_i) \\
 r_{i+1} &= Q_{i+1}.Q_{i-1}.r_{i-3} \vee Q_{i+1}.H_{i-1} \vee H_{i+1} \quad (i+1=7, 11 \text{ ou } 15) \\
 r_{i+1} &= Q_{i+1}.r_{i-1} \vee H_{i+1} \quad (i+1=5, 9 \text{ ou } 13)
 \end{aligned}$$

R_{15} est obtenu après passage dans 6 C.L.B. successifs (3 jusqu'à r_3 , puis 1 pour r_7 , 1 pour r_{11} et un dernier pour r_{15}). Quant aux sorties S_{14} et S_{15} , elles sont stables après que 7 C.L.B. aient été traversés.

Pour les indicateurs, on fournit la demi-retendue (r_3), la retenue (r_{15}) et le signe (S_{15}) sans rien rajouter. Le dépassement de capacité est produit par:

$$DC = r_{15} \oplus r_{14} = r_{15} \oplus (P_{14}.r_{13} \vee G_{14}) \quad (7 \text{ C.L.B. à traverser})$$

Le plus délicat est de produire l'indication de nullité dans le même temps que le résultat:

$$Z = \neg(S_0 \vee \dots \vee S_{15})$$

Dès que S_0 et S_1 sont disponibles (2 étages de C.L.B.), on produit $z_1 = \neg S_0. \neg S_1$. Puis, $z_5 = z_1. \neg S_2. \neg S_3. \neg S_4. \neg S_5$, ensuite $z_9 = z_5. \neg S_6. \neg S_7. \neg S_8. \neg S_9$. Ce dernier est obtenu après une traversée de 6 C.L.B.

On calcule $\neg(S_{10} \vee S_{11})$ et $\neg(S_{12} \vee S_{13})$ directement à partir de :

$$\neg(S_{10} \vee S_{11}) = \neg((M_{10} \oplus r_9) \vee (M_{11} \oplus (P_{10} \cdot r_9 \vee G_{10})))$$

$$\neg(S_{12} \vee S_{13}) = \neg((M_{12} \oplus r_{11}) \vee (M_{13} \oplus (P_{12} \cdot r_{11} \vee G_{12})))$$

De même pour $\neg S_{14}$ et $\neg S_{15}$:

$$\neg S_{14} = M_{14} \oplus (Q_{13} \cdot r_{11} \vee H_{13})$$

$$\neg S_{15} = M_{14} \oplus (\alpha \cdot r_{11} \vee \beta)$$

Avec $\alpha = P_{14} \cdot Q_{13}$ et $\beta = P_{14} \cdot H_{13} \vee G_{14}$

Toutes ces valeurs sont obtenues en 6 temps. Il ne reste plus qu'à les assembler pour obtenir Z en 7 temps:

$$Z = z_9 \cdot \neg(S_{10} \vee S_{11}) \cdot \neg(S_{12} \vee S_{13}) \cdot \neg S_{14} \cdot \neg S_{15}$$

Le schéma n°7 représente tous ces calculs. Chaque carré figure un C.L.B. (4 ou 5 entrées et 1 ou 2 sorties). Le tout est une partie du Xilinx n°4.

Pour produire les résultats des autres opérations il suffit de récupérer les sorties des calculs de P_i ($a_i \vee b_i$ en fixant $op=0$), G_i ($a_i \wedge b_i$), M_i ($a_i \oplus b_i$) et N_i ($\neg a_i$ en fixant $op=1$). Tous ces résultats sont dirigés vers des buffers trois-états reliés à un bus commun. Il en est de même des indicateurs.

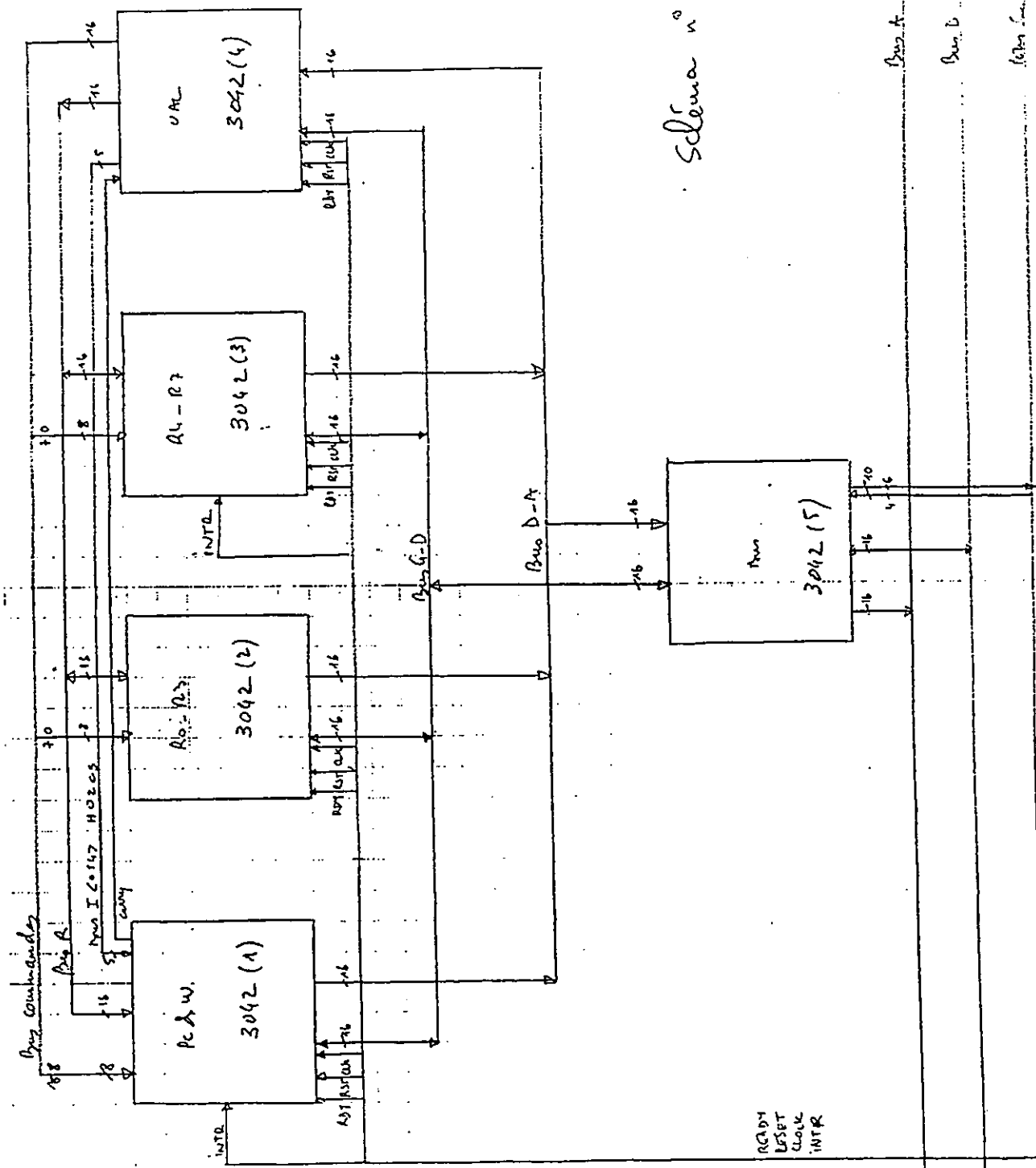
Sur le schéma n°8, on voit les différents morceaux de l'U.A.L.: calcul des P_i , G_i , M_i , N_i , Q_i et H_i , R_i , S_i . Un circuit de décalage combinatoire a été ajouté (dont le schéma interne n'est pas précisé mais il ne présente aucune difficulté particulière). Le bus R est alimenté par le résultat de l'opération décodée de COP. Le bus I reçoit les indications produites.

Au total ce circuit utilise 84 C.L.B. (64 pour l'U.A.L., 15 pour le décaleur et 5 pour les indicateurs) auxquels il faut ajouter, dans le même Xilinx, ceux utilisés par la logique de décodage (un peu plus de 40).

11) Conclusion.

Le microprocesseur que nous venons de décrire devrait être réalisé sur une carte prototype à installer dans un IBM-PC. Pour le tester, il suffit d'ajouter à la carte une RAM statique rapide (30ns de temps d'accès), un port série et un port parallèle pour communiquer avec le PC. Celui-ci doit charger les circuits Xilinx (via le port série), la RAM d'un programme initial (via le port parallèle) et effectuer un reset du système. Un interrupteur serait également souhaitable pour pouvoir arrêter le microprocesseur (Ready mis à 0; le PC lit les résultats dans la RAM par l'intermédiaire du port parallèle).

Du point de vue de la rapidité, le temps de cycle pourrait être voisin de 150ns (3042 à 100Mhz; avec une instruction par cycle, cela donne une puissance de 6.7MIPS; rappelons qu'un 80386 à 20Mhz n'atteint que 5MIPS): l'U.A.L. fournit son résultat après 7 étages de C.L.B., ce qui fait 49ns; cela laisse $150/2-49=26$ ns pour acheminer le résultat jusqu'au registre.



Schema n° 1

Bus A

Bus B

Bus C

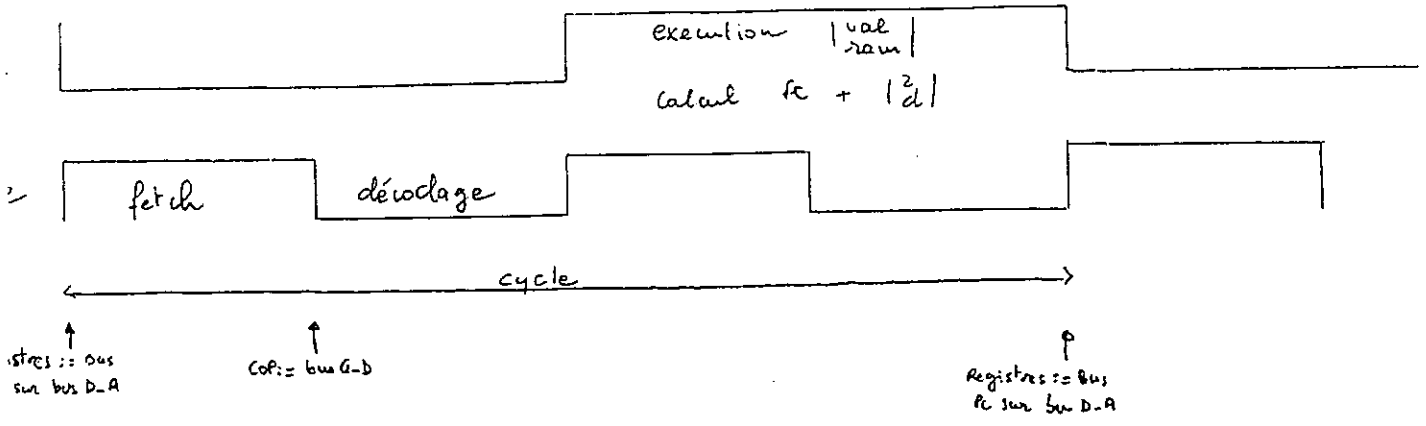


schéma n° 2

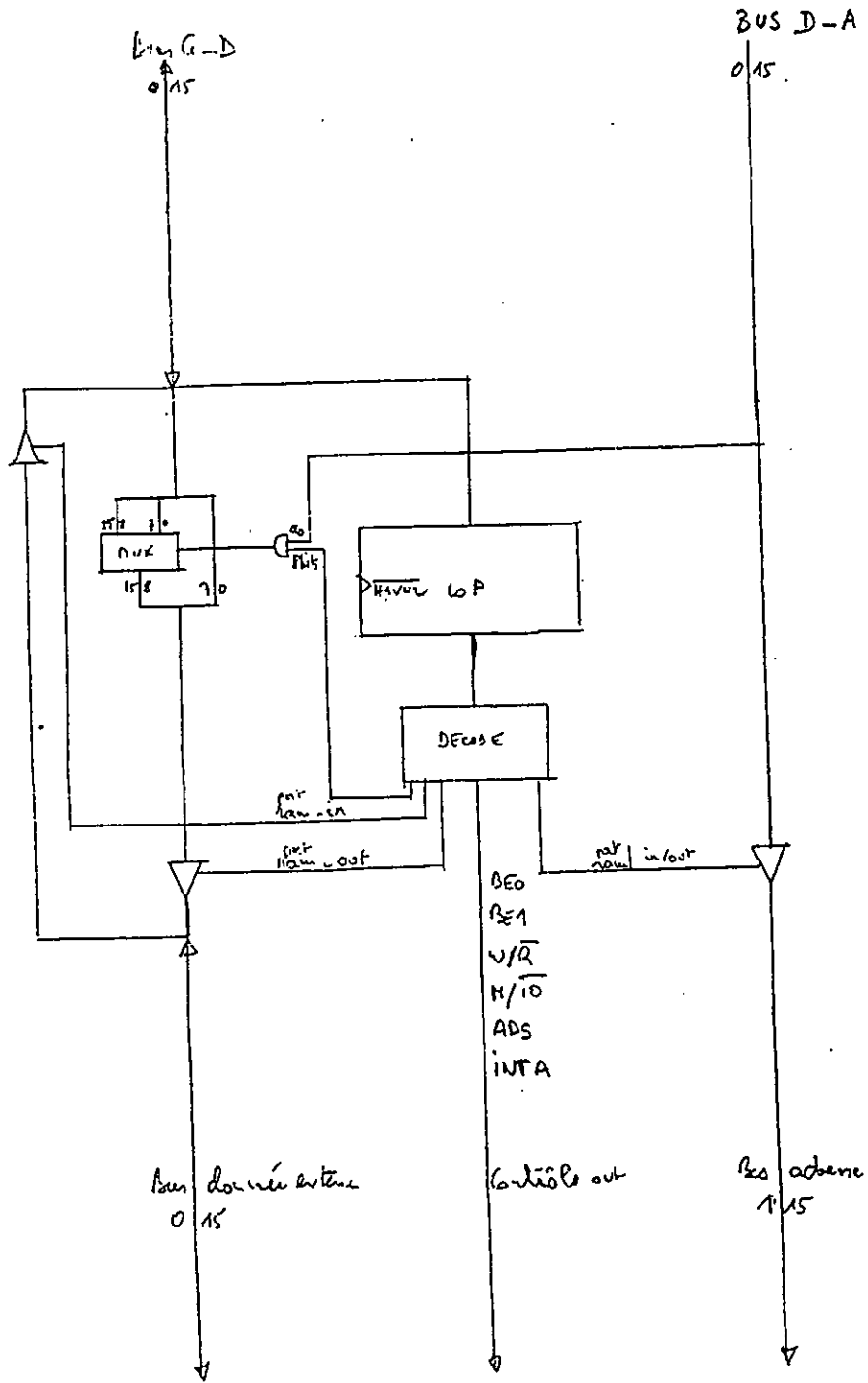
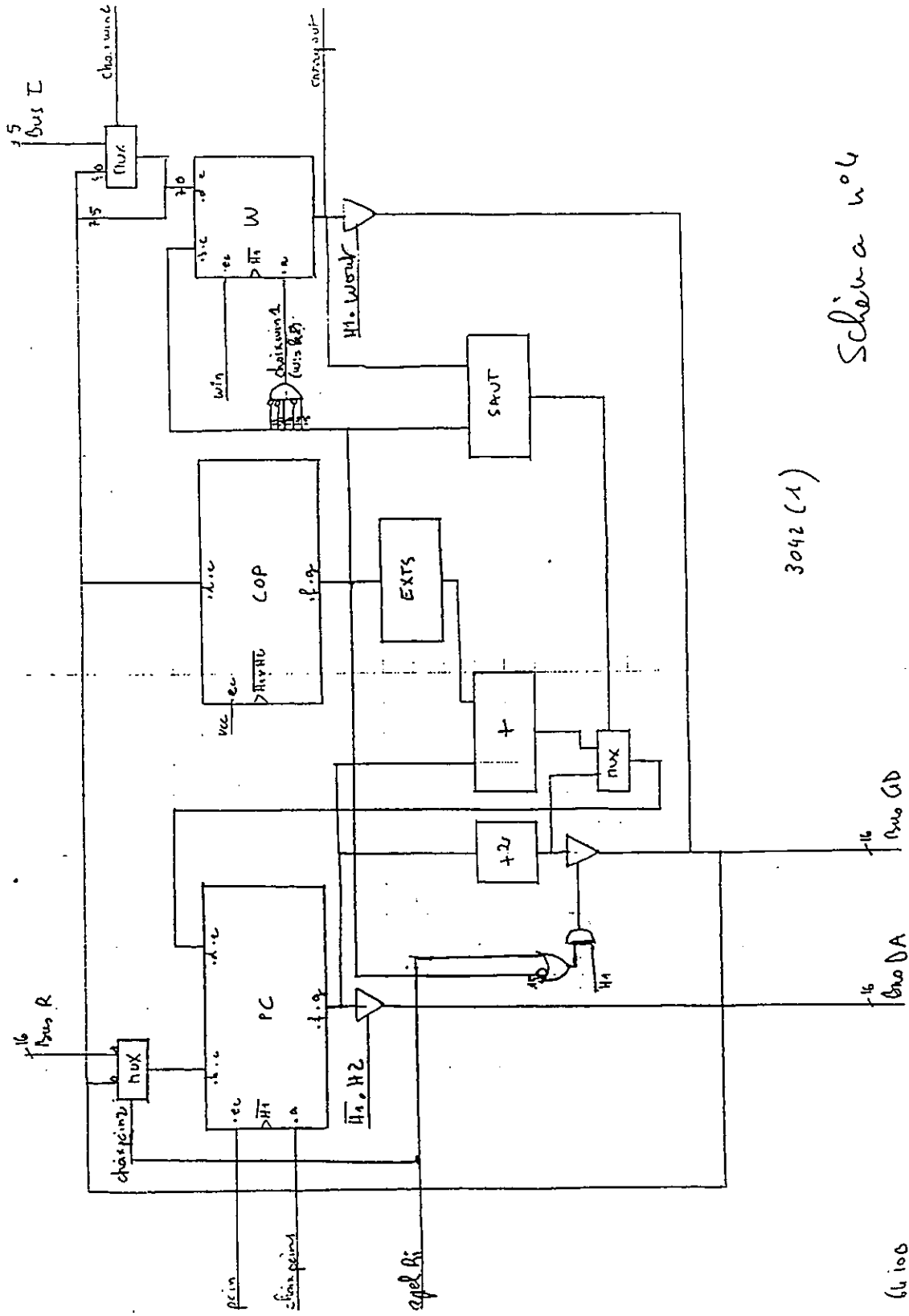


schéma n° 3



3042 (1)

Schéma n°6

64 100
94 008

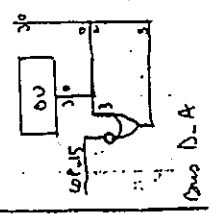
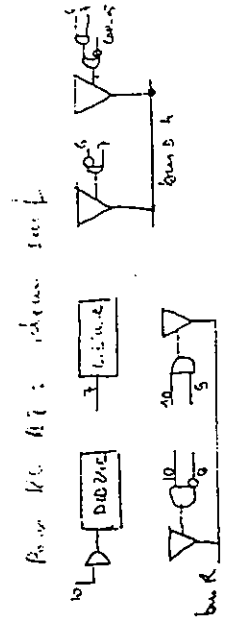
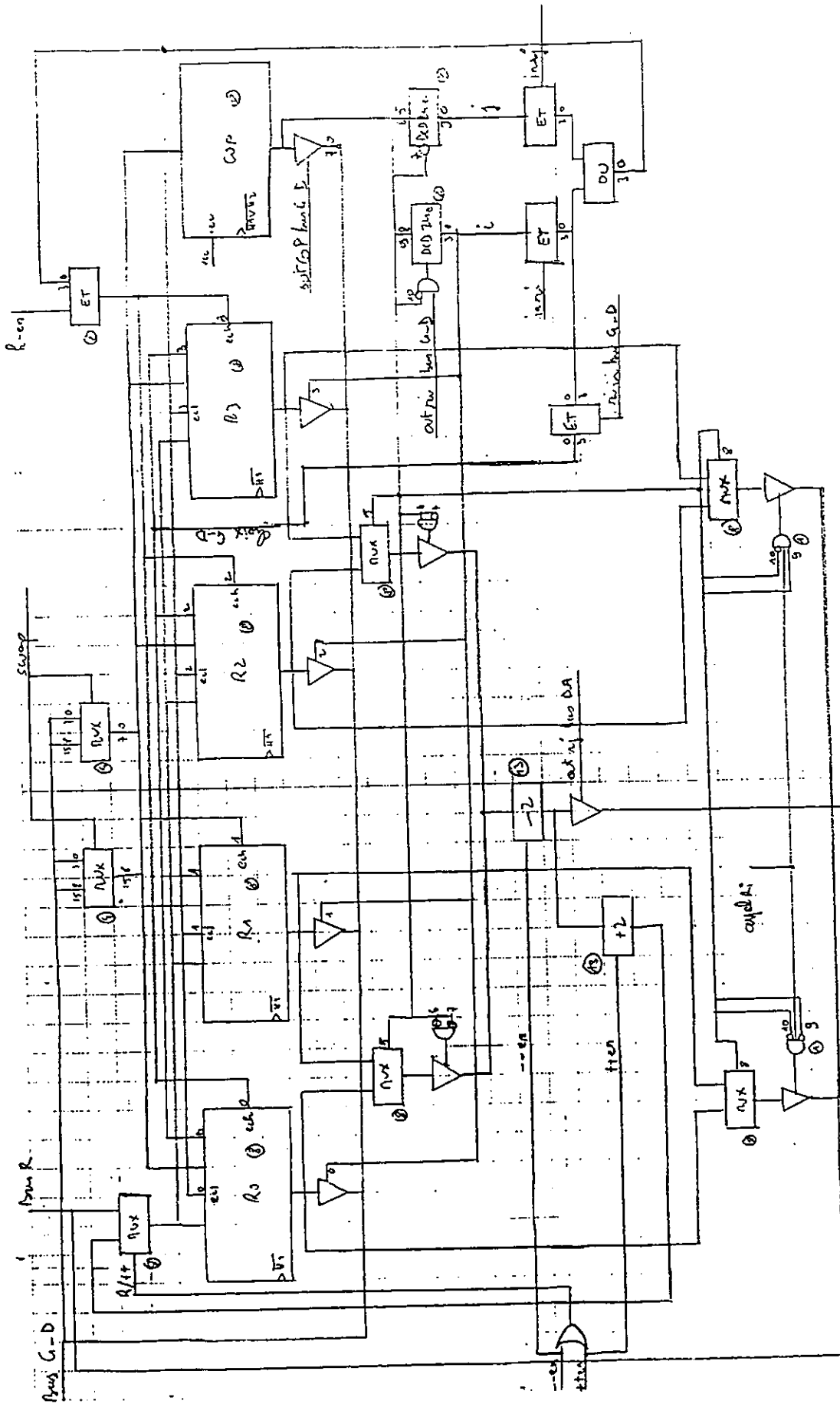
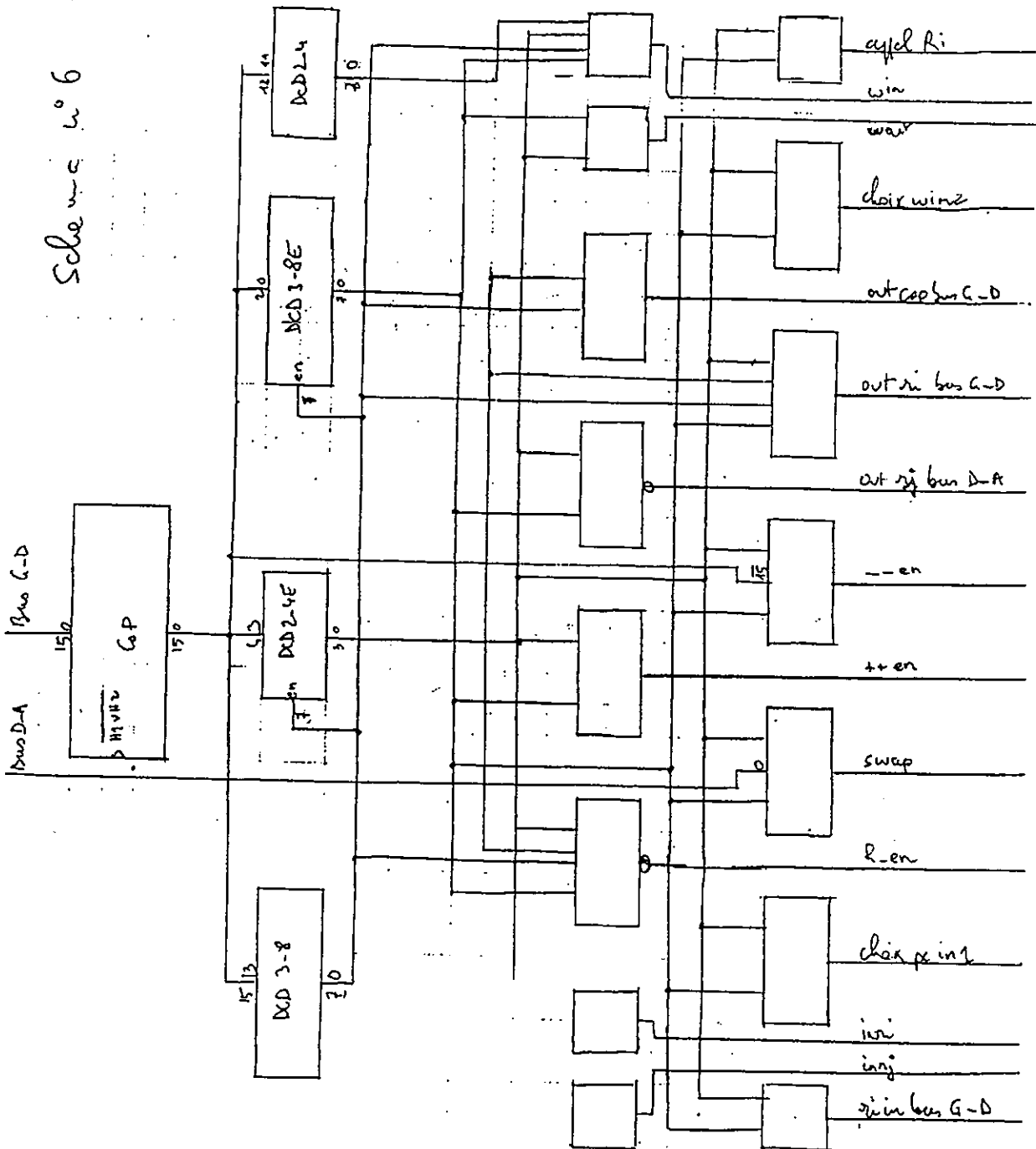


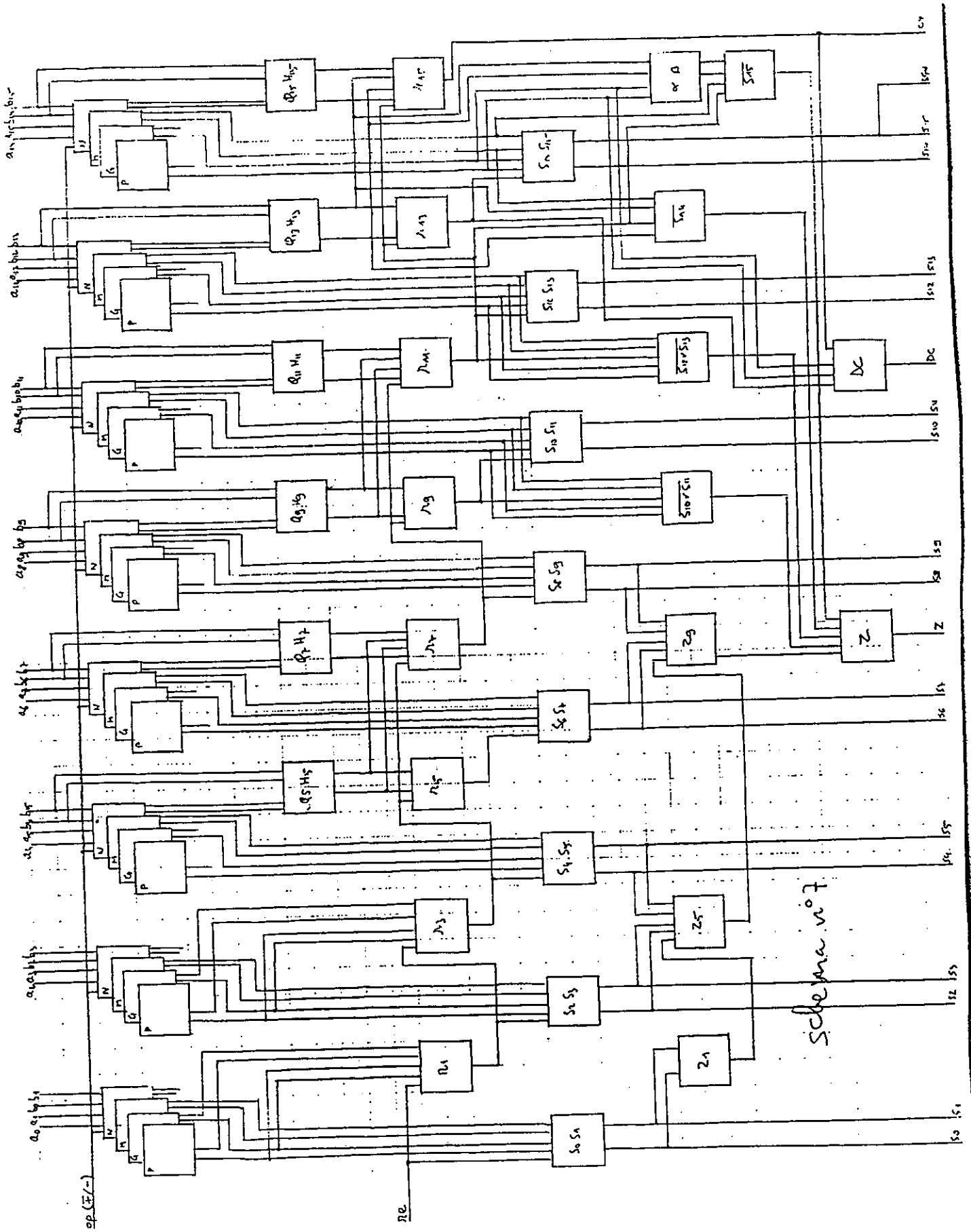
schéma n° 5

124 000
57 100

Schemes n° 6



63 64



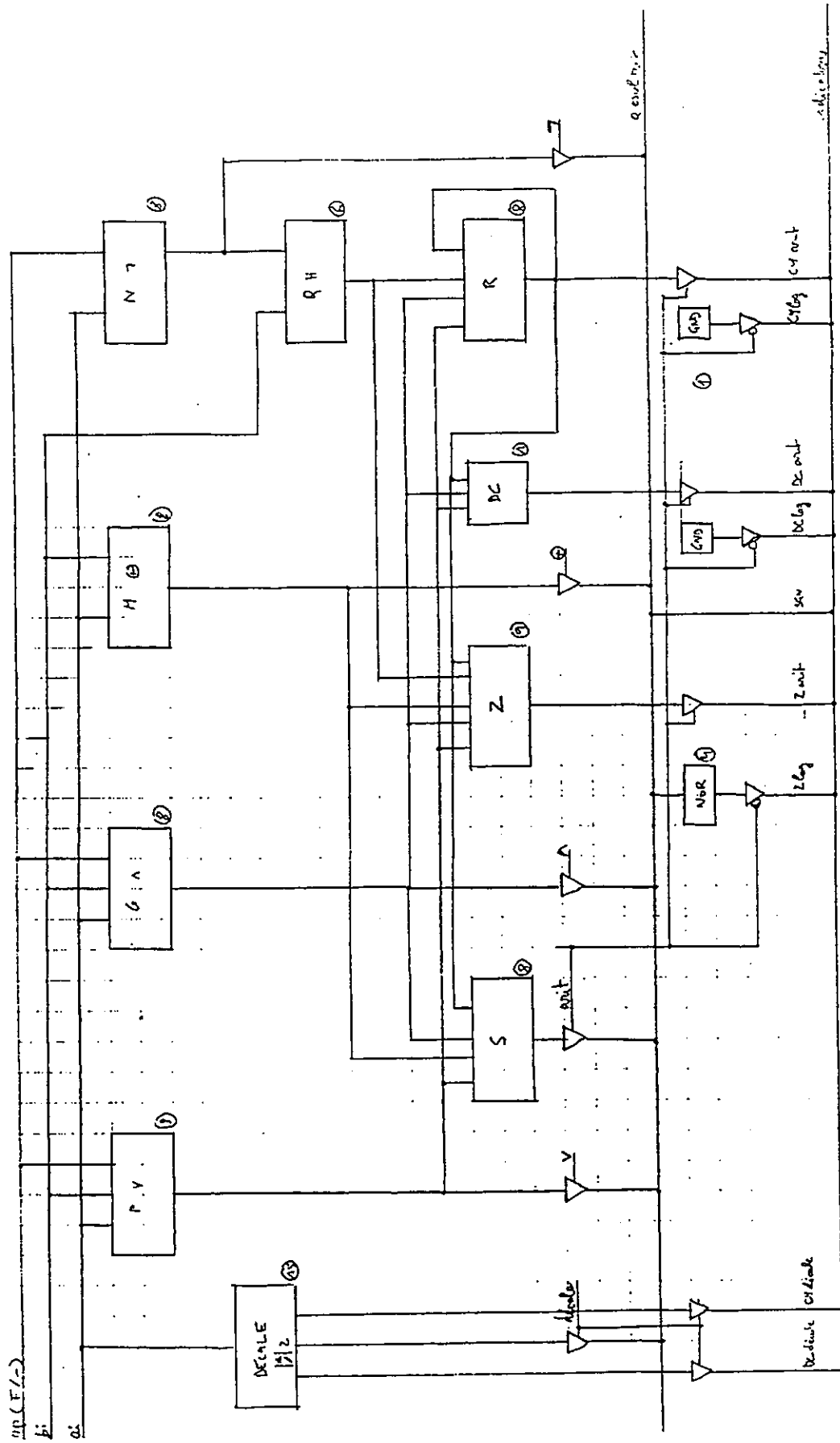


schéma n° 8

VOUZZAUEDIBISAR

E. Bianco

L'Honneur et la Nation.

Quel plus beau fleuron peut-on mettre en avant dans une nation comme la notre pour illustrer l'honneur, sinon l'Armée. Oui, je sais, cet organe viril de la fierté du Peuple est passablement décrié, mais je le dirai comme je le pense, les meilleurs détracteurs sont à mon avis des gens à la conscience peu claire et qui n'ont pas le sens de l'éthique.

J'ai eu le grand honneur, vers la fin des années cinquante, et comme nombre d'autres jeunes français, de participer à l'opération de tentative, vaine d'ailleurs, de maintien de la plus grande France. De Dunkerque à Tamanrasset, comme on disait alors. J'ai pu ainsi voyager dans des paysages magnifiques et qui touchaient parfois au fantastique, à la frontière du troisième pays du nord de l'Afrique. J'ai longé la barrière électrifiée qui eut, entre autres renommées, celle de compléter la richesse d'un homme qui avait commencé sa fortune dans une opération tout aussi louable. Sans doute a-t-on maintenant un peu oublié le Mur de la forteresse Europe bâti en bel et bon béton armé, du temps où la collaboration défendait des valeurs extrêmes.

J'ai ainsi connu les Nementcha. C'est une chaîne de montagnes, parallèle à la Méditerranée, sillonnée de nombreux oueds, dont le célèbre Oued Hallail, dont la plupart coulent vers le sud pour maintenir humides les chotts, ces lacs de boue noire, qui brillent comme l'aluminium. Qu'on imagine une terre rouge, un paysage ruiniforme entaillé de profondes failles au fond desquelles ruisselaient quelques traces d'eau au milieu de magnifiques massifs de lauriers et de grenadiers. Sur les sommets un piqueté vert de gris marquait le domaine de l'alpha, cette grosse touffe d'herbe dure comme du cuir. Et puis les rochers tombent de manière abrupte dans le sable, l'immense désert commence là. Les soirs calmes, la transparence de l'air donne l'impression que les montagnes sont là, juste sous la main, dans le sud lointain une barre violette sépare le jaune cru du sable du bleu dur du ciel, et bientôt une lune d'arc électrique éclate sur un fond bleu-noir d'acier trempé, au long d'un crépuscule sans fin.

Ce calme brutal cache des violences inouïes. Je n'étais pas là-bas pour admirer les gigantesques nuages verticaux bordés de gris-jaune, qui roulaient d'énormes volutes menaçantes et envahissaient d'un coup le paysage dans le rugissement d'un vent de sable. C'est un autre déchaînement qui forçait nos présences. Dans la lutte qui nous occupait alors il y avait des points privilégiés et le 'bec de canard' était l'un de ces points. Sur la carte, un peu en dessous de la mer on observait dans la forme de la frontière, une sorte d'excroissance, qui pouvait rappeler effectivement une sorte de bec de canard. Question d'imagination. Et si rien ni personne n'est plus civil qu'un militaire comme aimait à le dire un commandant que j'ai bien connu, l'imagination est également une qualité essentielle de cet estimable corps.

Le bec de canard était le lieu privilégié d'importantes actions militaires. En effet, de l'autre côté de la frontière, tout près, dans une zone quasi-désertique, des combattants avaient installé un camp d'entraînement pour venir en aide aux forces de l'intérieur que nous combattions. Et de temps en temps un petit commando bien entraîné forçait la frontière pour rejoindre les leurs. Ils devaient franchir le barrage électrifié, ce qui les obligeait à déployer mille astuces. Quelquefois des missions d'intervention directe étaient lancées à partir du bec de canard. Une sorte d'état-major inter-armes était installé plus ou moins à demeure sur une éminence qui permettait de diriger les opérations.

Il faut dire aussi que nous utilisions du matériel de récupération de l'armée américaine. Et notamment le T6 et le B26. Le premier était un petit avion

d'entraînement réservé alors pour l'observation et l'intervention légère, mitraillage au sol et largage de petites bombes. On avait bien essayé de le charger davantage, mais lors d'opérations menées dans des paysages tourmentés au milieu d'une aérologie particulièrement turbulente, nombre de ces appareils s'étaient perdus corps et biens à la suite de décrochages intempestifs. Le second était un glorieux bombardier en piqué de la dernière Grande. Bimoteur puissant aux ailes courtes et hautes on sentait en le voyant la belle et dangereuse machine de guerre. Malheureusement bien vieille et bien usée par sa gloire passée, au point que nombre d'entre elles ayant disparu mystérieusement lors d'opérations bénignes de simple reconnaissance, on s'était douté que les pilotes émoustillés par la nervosité de la bête avaient tenté de faire revivre quelques instant le passé fulgurant par quelque fatale voltige. Bref le B26 avait été interdit de piqué.

Ce jour-là une intervention directe avait été prévue car les Renseignements avaient détecté à l'extérieur une activité particulièrement redoutable. Un groupe de trois B26 devait aller jeter quelques bombes afin de faire montre de notre force aérienne. Or chacun sait que le renseignement travaille sur le terrain, les état-majors sur des cartes, et les pilotes se fient à leur compas et leur badin. Après que les avions déjà en l'air aient reçu les ultimes consignes, ils devaient passer à la verticale du camp de l'état-major afin de recevoir cap et distance, comme une sorte de pointage dans un tir d'artillerie à longue portée. Il faut savoir aussi que le pilote avait sous la main sept verrouillages de sécurité à débloquent avant de pouvoir larguer ses bombes par simple pression sur un bouton situé sur le manche à balai. Les consignes du déblocage des soutes à bombes sont toujours très strictes.

Aussi, est-ce joyeusement que les têtes galonnées virent approcher les trois points noirs à faible altitude dans un grondement qui se déchaînait rapidement. Maladresse, court-circuit dans les vieilles commandes de l'un des bombardiers, brefs ses soutes s'ouvrirent à la verticale du poste de commandement...

J'étais bien plus au sud, quand le capitaine Ray Chauffour me raconta cet accident. J'étais alors membre de l'Armée de l'Air, et j'assurai la relation entre notre poste de commandement et l'état-major du RIMA stationné dans la région. Le capitaine Chauffour en était le chef du troisième bureau. Comme pour tout officier du renseignement, ce n'était pas son vrai nom, nécessité de la fonction oblige. Ce brave homme, petit, tout rond, jovial, se frappait sur les cuisses de joie pendant qu'il me mettait en boîte: « ... Ah, vous l'Armée de l'Air, bravo ! voilà enfin un peu de place pour de l'avancement! ...»

Il est vrai que le chargement complet du B26 était tombé en plein sur le poste de commandement dans un feu d'artifice extraordinaire de galons et d'étoiles de toutes couleurs. Jamais les gens d'en face n'auraient pu rêver d'un tel succès.