

THÈSE

85/230

présentée à

L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

pour obtenir le titre de

DOCTEUR INGENIEUR

INFORMATIQUE

par

Dante Augusto COUTO BARONE



SABi



UFRGS

05225158

CONCEPTION D'UN CIRCUIT INTEGRE ARBITRE DE BUS
DE COMMUNICATION MULTIPROTOCOLES : ABC M.

soutenue le 7 novembre 1984 devant la commission d'examen :

Monsieur

J. MOSSIERE

Président

Messieurs

G. MAZARE

G. MICHEL

M. NG. X. DANG

Examineurs

U F R G S
BIBLIOTECA
CPD/PCCC

Personnalités habilitées à diriger des travaux de recherche
(Décision du Conseil Scientifique)

E.N.S.E.E.G.

ALLIBERT	Colette	DIARD	Jean Paul	NGUYEN TRUONG	Bernadette
BERNARD	Claude	EUSTATHOPOULOS	Nicolas	RAVAINE	Denis
BONNET	Roland	FOSTER	Panayotis	SAINFORT	(CENG)
CAILLET	Marcel	GALERIE	Alain	SARRAZIN	Pierre
CHATILLON	Catherine	HAMMOU	Abdelkader	SIMON	Jean Paul
CHATILLON	Christian	MALMEJAC	Yves (CENG)	TOUZAIN	Philippe
COULON	Michel	MARTIN GARIN	Régina	URBAIN	Georges (Laboratoire des ultra-refractaires ODEILLO).

E.N.S.E.R.G.

BARIBAUD	Michel	CHEHIKIAN	Alain	HERAULT	Jeanny
BOREL	Joseph	DOLMAZON	Jean Marc	MONLLOR	Christian
CHOVET	Alain				

E.N.S.I.E.G.

BORNARD	Guy	KOFMAN	Walter	MAZUER	Jean
DESCHIZEAUX	Pierre	LEJEUNE	Gérard	PERARD	Jacques
GLANGEAUD	François			REINISCH	Raymond

E.N.S.H.G.

ALEMANY	Antoine	MICHEL	Jean Marie	ROWE	Alain
BOIS	Daniel	OBLÉD	Charles	VAUCLIN	Michel
DARVE	Félix			WACK	Bernard

E.N.S.I.M.A.G.

BERT	Didier	COURTOIS	Bernard	FONLUPT	Jean
CALMET	Jacques	DELLA DORA	Jean	SIFAKIS	Joseph
COURTIN	Jacques				

U.E.R.M.C.P.P.

CHARUEL Robert

C.E.N.G.

CADET	Jean	JOUVE	Hubert (LETI)	PERROUD	Paul
COEURE	Philippe (LETI)	NICOLAU	Yvan (LETI)	PEUZIN	Jean Claude (LETI)
DELHAYE	Jean Marc (STT)	NIFENECKER	Hervé	TAIEB	Maurice
DUPUY	Michel (LETI)			VINCENDON	Marc

Laboratoires extérieurs :

C.N.E.T.

DEMOULIN	Eric	GERBER	Roland	MERCKEL	Gérard
DEVINE	R.A.B.			PAULEAU	Yves

I.N.S.A. Lyon

GAUBERT C.

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT-ETIENNE

Directeur : Monsieur M. MERMET
Directeur des Etudes et de la formation : Monsieur J. LEVASSEUR
Directeur des recherches : Monsieur J. LEVY
Secrétaire Général : Mademoiselle M. CLERGUE

Professeurs de 1ère Catégorie

COINDE	Alexandre	Gestion
GOUX	Claude	Métallurgie
LEVY	Jacques	Métallurgie
LOWYS	Jean-Pierre	Physique
MATHON	Albert	Gestion
RIEU	Jean	Mécanique - Résistance des matériaux
SOUSTELLE	Michel	Chimie
FORMERY	Philippe	Mathématiques Appliquées

Professeurs de 2ème catégorie

HABIB	Michel	Informatique
PERRIN	Michel	Géologie
VERCHERY	Georges	Matériaux
TOUCHARD	Bernard	Physique Industrielle

Directeur de recherche

LESBATS	Pierre	Métallurgie
---------	--------	-------------

Maîtres de recherche

BISCONDI	Michel	Métallurgie
DAVOINE	Philippe	Géologie
FOURDEUX	Angeline	Métallurgie
KOBYLANSKI	André	Métallurgie
LALAUZE	René	Chimie
LANCELOT	Francis	Chimie
LE COZE	Jean	Métallurgie
THEVENOT	François	Chimie
TRAN MINH	Canh	Chimie

Personnalités habilitées à diriger des travaux de recherche

DRIVER	Julian	Métallurgie
GUILHOT	Bernard	Chimie
THOMAS	Gérard	Chimie

Professeur à l'UER de Sciences de Saint-Etienne

VERGNAUD	Jean-Maurice	Chimie des Matériaux & chimie industrielle
----------	--------------	--

à Vera

AVANT - PROPOS

Le travail présenté dans cette thèse a été réalisé au laboratoire de Génie Informatique de l'Imag, dans le cadre de la collaboration du groupe Parallélisme, Communication et Circuits Intégrés avec le Centre National d'Etudes des Télécommunications, Centre Norbert-Segard de Grenoble.

Il a été effectué en liaison avec le département Architectures des Micro-Systèmes, division Conception des Circuits Intégrés du CNET CNS.

Je tiens à remercier,

Monsieur le Professeur Jacques MOSSIERE, Directeur du Laboratoire de Génie Informatique, de me faire l'honneur d'accepter de présider le jury de cette thèse,

Monsieur le Professeur Guy MAZARE, Professeur à l'Institut National Polytechnique de Grenoble, de m'avoir accepté au sein de son Groupe Parallélisme, Communication et Circuits Intégrés et d'avoir été un Directeur de Recherches toujours très sensible aux problèmes rencontrés,

Monsieur Gérard MICHEL, Chef du Département d'Architecture de Micro Systèmes du CNET/CNS, pour m'avoir accepté dans son Département pour y développer mes travaux, ainsi que pour les remarques et conseils qu'il m'a prodigués tout au long de ce travail,

Monsieur Michel NGUYEN HUAN DANG, Maître Assistant à l'Université de Grenoble II, pour sa collaboration constante et sa grande contribution à l'amélioration de cette thèse par ses remarques constructives,

Je voudrais également remercier :

Messieurs D. OLIVE, M. REMY et D. ROUQUIER, ingénieurs au CNET qui m'ont beaucoup aidé dans l'implantation de l'ABC M.

Que soient aussi remerciés :

Les collègues des Départements Architecture de Microsystèmes (AMS), Méthodes de Conception de Circuits (MCC) et Recherche en Conception Assistée (RCA) du CNET, et les membres de l'équipe Parallélisme, Communication et Circuits Intégrés (PCCI) présents au CNET, qui m'ont toujours apporté aide et collaboration,

ainsi que mes interlocuteurs de la Société APSIS de Meylan,

Madame Clotilde CHALAND à qui revient tout le mérite de la préparation matérielle de ce document.

RESUME :

L'étude de différents bus de communication parallèle à usage multi-microprocesseur (bus SM 90, MULTIBUS, DME), ainsi que des techniques d'arbitrage associées, a conduit à s'intéresser à la compatibilité de l'arbitre de bus intégré ABC 90 de la SM 90 (dont les fonctionnalités sont les plus puissantes) avec les autres types de bus (MULTIBUS, DME).

La première étape de l'étude se traduit par la proposition d'utilisation de l'ABC 90 comme organe d'allocation de bus dans différentes configurations d'architectures, et ce par adjonction d'éléments discrets.

La seconde étape consiste à proposer un circuit intégré d'arbitre de bus multiprotocole en partant des spécifications de l'ABC 90 et en y intégrant les résultats obtenus dans la proposition précédente.

La validation de ces deux propositions a été obtenue par simulation.

MOTS - CLES :

Bus parallèle. Architecture multi-micro-processeur. Techniques d'arbitrage. Interface. Compatibilité. Validation par simulation. Circuit arbitre de bus ULSI.

ABSTRACT :

The existence of so many parallel communication multi-micro-processor buses (buses of the SM 90, MULTIBUS & UME structures) and their different arbiter techniques led us to study the compatibility of the integrated bus arbiter ABC 90 of the SM 90 (which presents the widest range of functions) with other types of buses (MULTIBUS and UME).

The first part of the study involved the feasibility of using the ABC 90 circuit as bus arbiter in different architectural configurations ; this has been realized by the addition of discrete components.

The second step consisted in the design of an integrated multi - protocol communication arbiter, as an extension of the ABC 90's specifications and based on the results obtained in the first part of the study.

The validation of both proposals was carried out by simulation.

KEY - WORDS :

Parallel bus. Multi-microprocessor architecture. Arbiter techniques. Interface. Compatibility. Validation by simulation. ULSI bus arbiter circuit.

RESUMO :

O estudo de diferentes "bus" de comunicação paralela utilizados em arquiteturas multi-microprocessadores ("bus" das estruturas SM 90, MULTIBUS e VME), assim que suas técnicas de arbitragem respectivas, nos permitiram de conduzir nosso trabalho sobre o estudo de compatibilidade do circuito integrado arbitro de bus ABC 90 da estrutura SM 90 (cujas funções são as mais potentes) com os outros tipos de "bus" (MULTIBUS e VME).

A primeira etapa de nosso estudo se traduz pela proposição de utilização do circuito ABC 90 como órgão de alocação de "bus" em diferentes configurações arquiteturais multi-microprocessadores através da introdução de componentes discretos.

A segunda etapa consiste na proposição de um circuito integrado arbitro de "bus" multi-protocolos partindo das especificações do circuito ABC 90 e dos resultados obtidos pela primeira proposição.

A validação das duas proposições sugeridas por este trabalho foi obtida através de simulações.

PALAVRAS CHAVE :

"Bus" de comunicação paralela. Arquitetura multi-microprocessadores. Técnica de arbitragem. Interface. "Compatibilidade". Validação por simulação. Circuito arbitro de bus VLSI.

TABLE DES MATIERES

Introduction	001
1. Les bus de communication	005
1. Introduction	007
2. Bus SM 90	011
2.1. Structure du bus	011
2.2. Connexion d'un module de traitement au bus	015
2.3. Technique d'arbitrage	016
2.4. Arbitre de bus ABC 90	016
3. Bus MULTIBUS	019
3.1. Structure du bus	019
3.2. Connexion d'une unité de traitement au bus	021
3.3. Technique d'arbitrage	023
3.4. Arbitre de bus 8289	025
4. Bus UME	028
4.1. Structure du bus	028
4.2. Connexion d'une unité de traitement au bus	031
4.3. Technique d'arbitrage	032
4.4. Arbitre de bus MC68452	035
5. Méthodes d'arbitrage	037
6. Fonctions d'un arbitre de bus	040
7. Conclusion	041

2. Etude de la compatibilité entre le circuit arbitre de bus ABC 90 et les bus de communication MULTIBUS et UME : approche discrète	043
1. Introduction	045
2. Comparaison entre arbitres	046
3. Etude de compatibilité MULTIBUS-ABC 90	047
3.1. Proposition d'un ensemble de fils cohérents	047
3.2. Proposition pour rendre possible l'utilisation de l'ABC 90 dans une structure MULTIBUS	049
3.3. Architectures compatibles proposées	054
3.3.1. Architecture centralisée	054
3.3.2. Architecture décentralisée	058
4. Etude de compatibilité UME-ABC 90	063
4.1. Proposition d'un ensemble de fils cohérents	063
4.2. Proposition pour rendre possible l'utilisation de l'ABC 90 dans une structure UME	064
4.3. Architectures compatibles proposées	064
4.3.1. Architecture centralisée	064
4.3.2. Architecture décentralisée	067
5. Validation par simulation des architectures proposées	069
5.1. Approches de validation	069
5.2. Validation logique	070
5.3. Caractéristiques des simulations	071
5.4. Analyse des résultats	072
6. Conclusion	075

3. Proposition d'un circuit arbitre de bus de communication multi-protocoles : ABC M	077
1. Introduction	079
2. Proposition d'un circuit arbitre de bus multi-protocoles : ABC M	080
2.1. Architectures dans lesquelles on peut placer l'ABC M	080
2.2. Avantages de l'utilisation de l'ABC M par rapport à l'approche discrète (ABC 90)	088
2.3. Contraintes algorithmiques à respecter par l'arbitre ABC M	089
2.3.1. Qui divulgue le verdict	089
2.3.2. Support pour la divulgation du verdict	090
2.3.3. Déclenchement du verdict	090
2.3.4. Choix d'un algorithme	091
3. Spécifications de l'ABC M : modifications apportées à l'ABC 90	092
3.1. Mode de fonctionnement du circuit	092
3.2. Divulgation systématique du verdict	094
3.3. Logique supplémentaire de divulgation du verdict de forme décodée	095
3.4. Automate de déclenchement d'arbitrage	099
3.5. Architecture du circuit ABC M	101
4. Validation par simulation du circuit ABC M	103
4.1. Outils de CAO employés	103
4.2. Validation logique	104
4.3. Caractéristiques des simulations	104
4.3.1. Mode SM 90	104
4.3.2. Mode Release When Done (RWD)	105
4.3.3. Mode Release On Request (ROR)	105
4.4. Analyse de synthèse	106
5. Implantation de l'ABC M sur silicium	109
5.1. Plan de masse de l'ABC M	109
5.2. Implantation sur silicium des modifications apportées à l'ABC 90	112
5.2.1. Partie opérative	112
5.2.2. Partie contrôle	112
6. Conclusion	113
Conclusions	115
Bibliographie	119

Annexes	127
Sommaire des annexes	129
Annexe 1. Etude de l'interface côté application de différents gestionnaires de communication de réseaux locaux : analyse et proposition selon la norme IEEE 802	133
Annexe 2. Exemples d'arbitrage de bus	175
Annexe 3. Proposition d'utilisation cohérente des broches du connecteur MULTIBUS et du connecteur DME	183
Annexe 4. Compatibilité électrique	195
Annexe 5. Outils de CAO utilisés pour la validation et l'implantation des circuits intégrés	201
Annexe 6. Résultats des simulations des architectures discrètes utilisant l'ABC 90	221
Annexe 7. Résultats des simulations et de l'implantation de l'ABC M	257
Bibliographie des annexes	297

INTRODUCTION

Aujourd'hui les systèmes de communication sont de plus en plus utilisés. Ces systèmes ont besoin d'organes capables de les gérer : les gestionnaires de communication.

Notre étude sur ces gestionnaires de communication s'est développée à partir de deux études d'ordre général :

. étude de bus série (la transmission de données se fait de façon sérielle, couvrant de grandes distances) utilisés dans les réseaux locaux [BAR 83.1] (on s'est intéressé plus particulièrement aux réseaux locaux industriels [DAN 83]).

. étude de bus parallèles : la transmission de données se faisant en parallèle permet des débits de transfert plus importants; toutefois les distances couvertes sont moindres [BAR 83.2].

En fait, les deux techniques de transmission de données ne sont pas mutuellement exclusives : de plus en plus les gestionnaires d'un réseau local ont besoin d'échanger des informations avec d'autres processeurs à travers un bus parallèle sur le fond de panier, par exemple. Cet échange d'informations fait partie de l'interface entre un gestionnaire et sa (ses) station(s). Une étude sur les fonctionnalités de cette interface a été faite en tenant compte de la norme IEEE 802 [BAR 83.3] (voir annexe 1).

En raison de l'importance grandissante de ces bus parallèles, on s'est donc intéressés à analyser en détails les bus MULTIBUS, UME et SM BUS. L'étude de ces bus de communication parallèle est décrite au chapitre 1, de même que les circuits arbitres de bus correspondants.

L'analyse des similitudes et des différences entre les techniques d'arbitrage employées par les architectures retenues (MULTIBUS, UME, SM 90) en vue d'un circuit arbitre de bus multiprotocole a été menée en deux étapes :

. La première étape consiste à proposer une architecture discrète d'arbitrage bâtie à partir du circuit arbitre de bus ABC 90 (développé pour la structure SM 90) dans d'autres structures multi-micro-processeurs (MULTIBUS, VME). La complexité des architectures discrètes proposées nous a amenés à les valider, en utilisant des outils de CAO logico-fonctionnels comme EPILOG (validation logique) et FIDEL (validation fonctionnelle). L'analyse des simulations faites nous a permis d'affiner les architectures discrètes proposées. Ces architectures se révèlent toutefois coûteuses du point de vue matériel car en plus de l'ABC 90 une logique supplémentaire se révèle nécessaire. Cette première étape est décrite au chapitre 2 et les résultats des simulations sont donnés en annexe 6.

. La seconde étape consiste à proposer un circuit arbitre de bus multiprotocole (ABC-M) qui gère les conflits d'accès au bus sans logique supplémentaire. Ce nouveau circuit se veut compatible avec l'ABC 90 dont il reprend une partie logique conséquente. De plus, il offre d'autres modes de fonctionnement que celui implanté par l'ABC 90 ; c'est à ces nouveaux modes de fonctionnement qu'on doit le caractère multiprotocole du circuit ABC-M.

De même que les architectures discrètes bâties à partir de l'ABC 90 décrites dans le premier chapitre, les architectures intégrées utilisant l'ABC-M ont aussi été validées de façon logico-fonctionnelle. Cette deuxième étape est décrite au chapitre 3 et les résultats des simulations sont données en annexe 7.

1. LES BUS DE COMMUNICATION

1. INTRODUCTION

Les bus de communication de données peuvent trouver quelques domaines d'application principaux :

. bus réseau pour la transmission de données dans les réseaux locaux [COR 81], [HYN 83]. En général la transmission se fait en parallèle pour des réseaux qui couvrent des petites distances et en série pour des distances plus importantes comme ETHERNET [MET 76] par exemple ;

. bus fond de panier pour permettre l'échange d'informations entre plusieurs processeurs (unités centrales ou gestion d'entrées-sorties) [GRO 82], [GRO 84.2] ;

. bus interne pour permettre l'échange d'un microprocesseur avec ses ressources locales, à l'intérieur d'une carte ;

. bus d'instrumentation, comme le GPIB [CIN 81] (norme IEEE 488) qui servent à connecter un ou plusieurs processeurs à un ensemble commun d'instruments ;

. bus d'interface série d'appareils, comme le standard RS 232 C [LES 78] ou RS 422 pour des débits plus importants.

Outre le type d'application envisagée, ce sont la vitesse de transfert des données, ainsi que la distance entre les "organes de communication" qui constituent les facteurs les plus importants pour le choix d'un bus donné.

La figure 1 montre le domaine d'application de différents types de bus de communication de données.

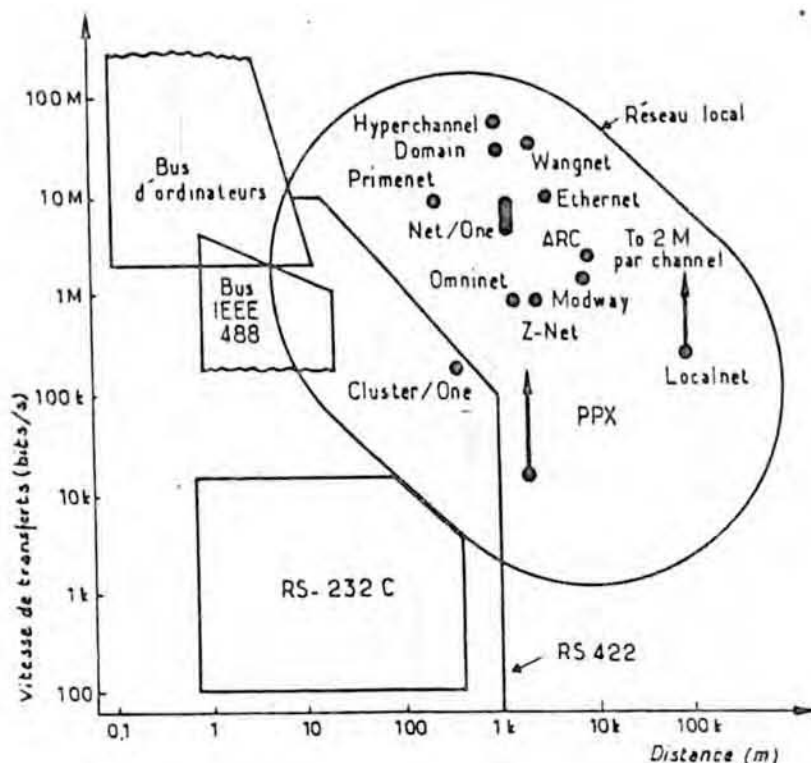


Figure 1 - Domaines d'application de différents types de bus et interfaces [CAT 82]

Dans ce travail, on va s'intéresser particulièrement aux bus fond de panier (appelés bus d'ordinateurs dans la figure ci-dessus) qui sont utilisés pour les interfaces entre processeurs.

Bus fond de panier

Le premier bus fond de panier qui s'est présenté sur le marché a été le S 100, standardisé par l'IEEE sous le nom de P 696 [WAR 83].

Un autre bus, le STD [BIE 82] a pris une partie importante du marché de contrôleurs de procédés industriels [CAM 84].

D'autres bus permettant un adressage mémoire plus important (mots de 16 ou 32 bits) prennent à l'heure actuelle de plus en plus de place sur le marché [BAI 84].

Parmi les 16 bits, on cite MULTIBUS [INT 81] le Q bus [DAL 84] ainsi que le bus de communication (SM bus) de la structure multi-microprocesseurs SM 90 [SM9 82].

Parmi les bus 32 bits, on cite UME [FIS 84] (très utilisé aussi dans les systèmes 16 bits) et MULTIBUS II [DEA 84] [PAC 84], ainsi qu'une version du bus SM 90.

MULTIBUS

MULTIBUS développé par INTEL est devenu, grâce à sa forte présence sur le marché, un standard "de facto" de façon à ce que plusieurs fournisseurs proposent des produits "compatibles MULTIBUS". On présentera plus en détail ce bus au § 1.3. Toutefois, il ne présente pas une architecture très modulaire, vraisemblablement dû au fait qu'il a été conçu sur mesure pour le microprocesseur 8080.

Q BUS

Un autre bus 16 bits très bien implanté sur le marché est le Q bus, développé par DIGITAL pour sa gamme de produits LSI 11.

UME

Le bus Européen le plus répandu est le bus UME [LIL 82] développé par MOTOROLA, MOSTEK & PHILIPS, qui présente une architecture assez sophistiquée. Une gamme d'utilisations de ce bus est le marché du microprocesseur 68 000. Ce bus permet l'utilisation de cartes au format simple ou double EUROPE à travers un connecteur indirect DIN 41612 [HAL 82] qui supporte entièrement des transferts 16 bits. L'extension à 32 bits peut être réalisée à travers un deuxième connecteur [GRO 84.2]. Ce bus est décrit plus en détail au § 1.4.

SM BUS

En France, un autre bus de communication a été proposé, le SM bus pour la structure multimicroprocesseurs hétérogènes SM 90 développée au CNET. Ce bus présente une logique d'arbitrage de bus décentralisée, ainsi que des mécanismes de sécurité (possibilité de doublement des modules, intervention sur la machine en cours de fonctionnement, parités, ...). Le SM bus a été normalisé par les PTT pour les équipements de sécurité. Une description plus détaillée de ce bus est donnée au § 1.2.

Présence des bus fond de panier sur le marché

La figure 2 montre la répartition du marché des bus système en 1983 [DAL 84], ainsi qu'une projection pour 1988 [GR0 84.1]

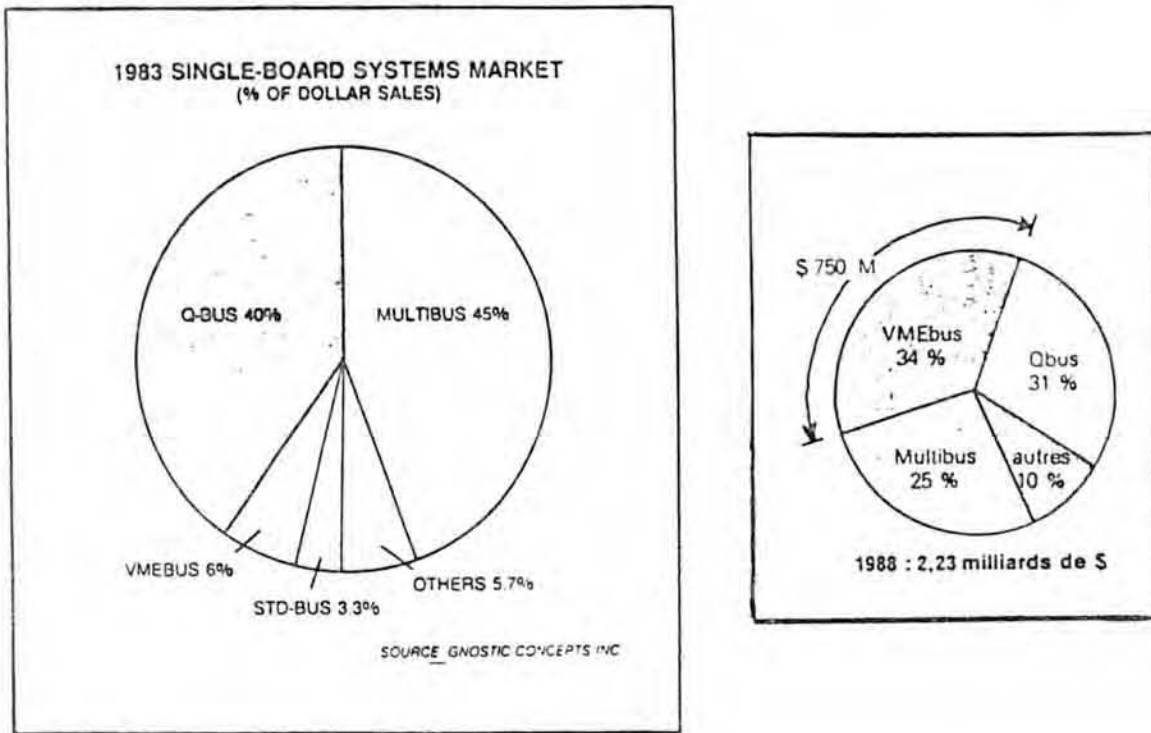


Figure 2 - Marché de différents bus système

Choix de bus

Les critères qui nous ont conduits à porter notre choix sur les bus à analyser comme support de base de ce travail ont été :

- . gamme étendue d'applications envisagées : systèmes multi-micro-processeurs 16 bits homogènes et hétérogènes,
- . présence du bus sur le marché,
- . intérêt de l'architecture de chaque bus,
- . modularité,
- . possibilité de contrôle d'erreurs (pour chercher à éliminer des erreurs [LAP 82]).

À la lumière de ces considérations on a choisi les bus SM BUS, MULTIBUS, VME.

2. BUS SM 90

2.1. Structure du bus

Les spécifications du bus de communication du système multi-microprocesseur SM 90 développé par le CNET, ont suivi les idées directrices suivantes :

. bus suffisamment simple (pas trop de points de connexion), mais général (ne s'appuyant pas sur des notions liées à un microprocesseur particulier), offrant aussi un certain nombre de facilités au niveau de la communication interprocesseurs et de la sécurité, permettant en outre d'implanter un mécanisme d'arbitrage réparti sur chacun des modules susceptibles de prendre l'initiative d'une communication (modules maîtres).

La figure 3 montre l'architecture de la structure SM 90 :

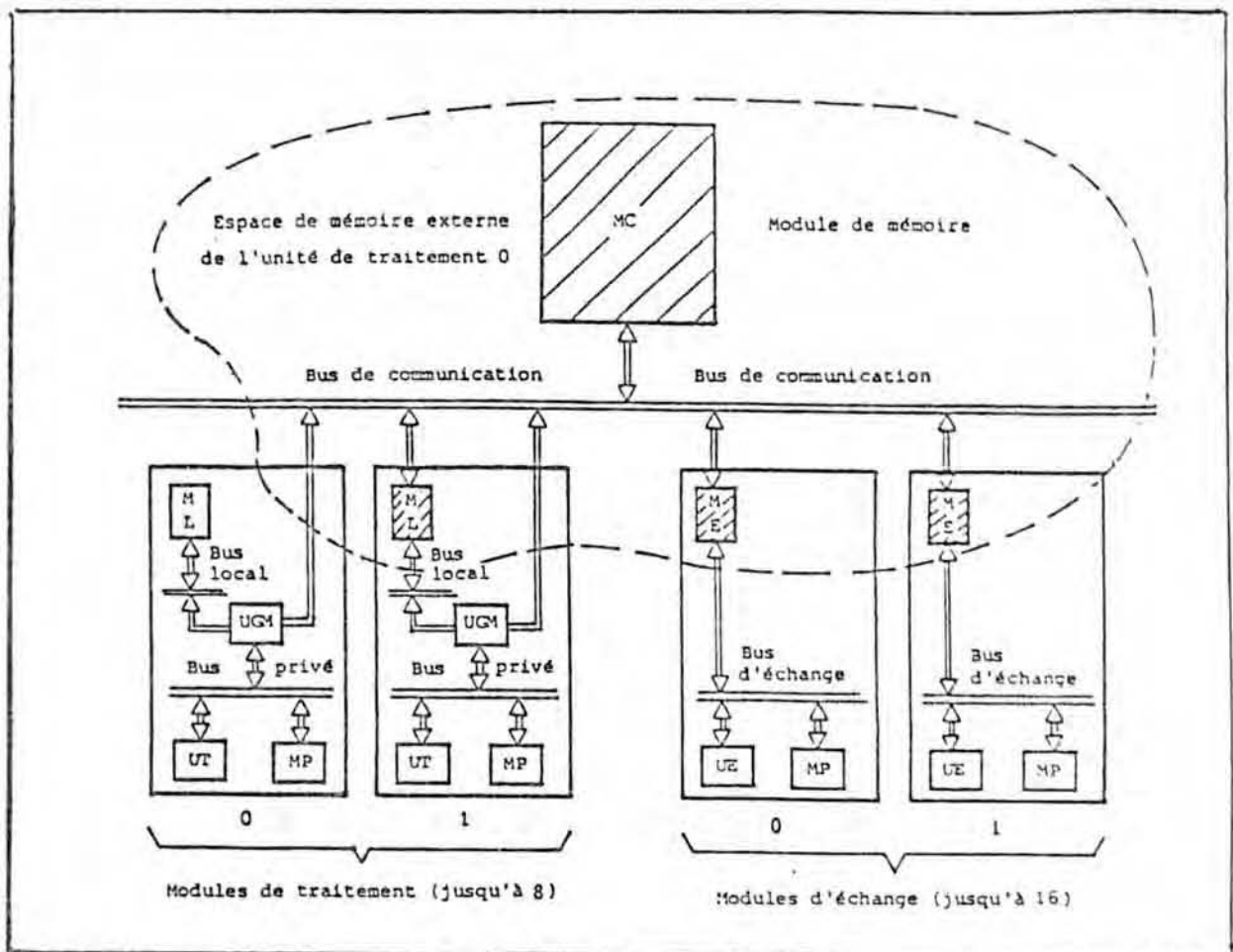


Figure 3 : Organisation de la SM 90

- UT : unité de traitement (microprocesseur 16 bits)
- UE : unité d'échange (microprocesseurs 8 ou 16 bits)
- MP : mémoire privée (accessible par une seule unité)
- ML : mémoire locale obligatoirement connectée au bus local et optionnellement au bus de communication
- ME : mémoire d'échange permettant aux unités de traitement de communiquer avec une unité d'échange (boîte aux lettres)
- MC : mémoire commune, qui peut être éventuellement doublée pour une plus grande disponibilité de la structure
- UGM : unité de gestion de la mémoire, permettant à une unité de traitement d'accéder, au moyen d'un "adressage logique" à sa mémoire locale et à sa mémoire externe (à titre d'exemple, les portions de mémoire externe de l'unité de traitement 0 sont hachurées sur la figure).

D'après cette figure, on constate qu'on a trois types de modules connectés au bus de communication :

- . des modules de traitement (jusqu'à 8),
- . des modules d'échange (jusqu'à 16),
- . un module de mémoire commune.

Pour avoir plus de renseignements sur la structure de la SM 90 dont la description est hors du cadre de ce travail, on se reportera à [FIN 82] [SM9 82].

Les signaux échangés par des bus peuvent être divisés en deux grandes familles :

1) les signaux de transfert, qui comprennent les adresses et les données,

2) les signaux de contrôle qui en fait peuvent être répartis en familles distinctes :

a) lignes de commande, comme écriture/lecture, validation des données sur le bus et autres,

b) lignes de service, comme alarme, initialisation, alimentation, détection d'erreurs, supervision et autres,

c) lignes d'interruption, comme dialogue de l'esclave vers le maître,

d) lignes spécifiques de gestion d'accès au bus, qui servent à régler les conflits causés par les demandes d'accès simultanées au bus.

Vis à vis des signaux qui caractérisent un bus, le bus de communication de la SM 90, appelé SM bus [FIN 81], comporte 64 fils pour la version 16 bits et se présente de la façon suivante :

1) Signaux de transfert :

- . des adresses sur 25 fils (adressage octet),
- . des données sur 16 fils parmi les 25 précédents ; bien qu'adresses et données soient multiplexées (pour limiter le nombre de points de connexion) les techniques utilisées permettent d'assurer au SM bus un débit utile de l'ordre de 5 mégaoctets par seconde (compte tenu d'une durée moyenne donnée d'arbitrage).

2) Signaux de contrôle

a) Lignes de commande :

. des signaux permettant à un module maître ayant obtenu la propriété du bus, d'établir avec un module partenaire un dialogue leur permettant de contrôler l'échange effectué ;

. des signaux liés à un certain nombre d'appels, lancés pourrait-on dire à la cantonnade (diffusion) : c'est-à-dire le module émetteur ne visant pas un module récepteur particulier, mais tous ceux qui peuvent l'entendre à cet instant (à noter que l'initiative de ces appels n'est pas réservée aux modules maîtres).

. des signaux liés aux problèmes de protection mémoire, d'utilisation facultative : comparaison des parités calculées séparément par les partenaires d'un échange, gestion de modules de mémoire doublés permettant d'écrire simultanément dans les deux exemplaires d'un module, les lectures se faisant dans un seul (l'exemplaire actif à cet instant), un basculement sur l'autre exemplaire étant possible en cas d'ennui ;

b) Lignes de service :

. des signaux de type diagnostic, rendant compte d'événements exceptionnels (coupure d'alimentation imminente, bouclage d'un processeur de la structure ...) ; un signal de ce type, dangereux dans la mesure où il déclenche des actions radicales, sera toujours confirmé par un signal de validation véhiculé par un fil du SM bus réservé à cet effet ;

. des signaux pour un dialogue de supervision indépendant du bus parallèle, deux fils du SM bus ayant été réservés pour un bus sériel à venir (protocole analogue à celui défini pour le bus P 896) [PAU 83].

c) Lignes d'interruption :

. le SM bus ne présente pas de lignes d'interruption.

d) Lignes de gestion d'accès au bus :

. des signaux permettant le dialogue entre les arbitres répartis sur les modules maîtres (pour déterminer à quel module attribuer la propriété du bus, parmi les modules maîtres candidats à un échange).

2.2. Connexion d'un module de traitement au bus

La connexion d'un module de traitement au medium se fait de la manière représentée dans la figure 4.

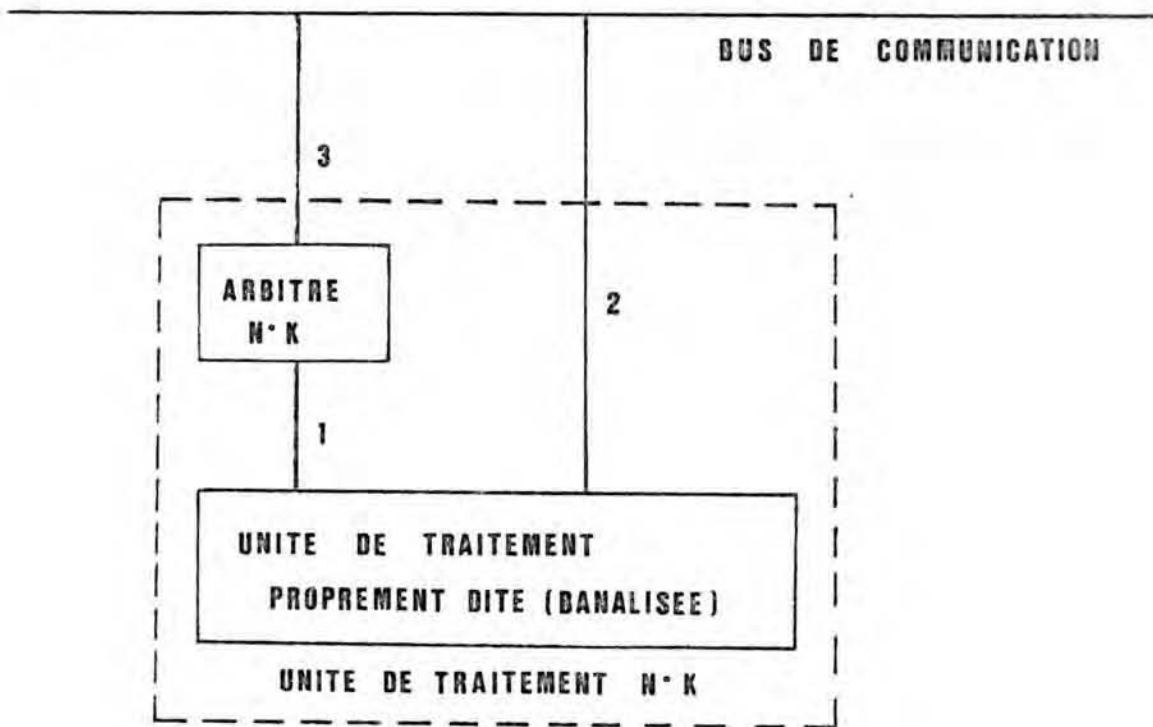


Figure 4 - Connexion d'un module de traitement au SM BUS

D'après cette figure, on peut dégager les interactions nécessaires pour qu'une unité de traitement puisse envoyer ces données à travers le bus de communication.

A l'intérieur d'un module de traitement n° K, par exemple, l'unité de traitement proprement dite (banalisée) fait une demande d'accès au bus à son arbitre (interface 1). Celui-ci dialogue avec les autres arbitres de bus présentés sur la structure (interface 3) jusqu'à ce qu'il obtienne le contrôle du bus. Dès que cette condition est réalisée, elle est retransmise à l'unité de traitement banalisée, au travers de 1. Dès lors, l'unité de traitement en question procède à ses échanges, en envoyant sur le bus de communication (interface 2) tout d'abord les adresses et puis les données.

En annexe 2 on donne les diagrammes d'arbitrage pour le SM Bus.

2.3. Technique d'arbitrage

D'après la figure 3, on peut voir que la SM 90 ne présente pas de carte contrôleur centralisé du système, laquelle aurait pour charge d'assurer l'allocation du bus au module de traitement le plus prioritaire à un instant donné. Cela implique que sur chaque maître on dispose d'une logique d'arbitrage de bus, ce qui rend la méthode d'arbitrage de la SM 90 tout à fait décentralisée.

2.4. Arbitre de bus ABC 90

Initialement, l'arbitre de bus présent sur chaque module de traitement avait été réalisée en composants discrets, mais par la suite un circuit intégré, nommé ABC 90, développé par le CNET/CNS de Meylan [OLI 82], a remplacé la logique en composants discrets. Cet arbitre, pour augmenter la modularité et la disponibilité des structures qui l'utilisent, est conçu pour être employé à raison d'un circuit par module de traitement.

Les arbitres présents sur la structure (jusqu'à 8) travaillent en parallèle pour réaliser un arbitrage chaque fois qu'une ou plusieurs modules de traitement désirent effectuer un échange sur le bus.

Cet arbitrage peut être effectué de deux façons différentes :

- . juste avant l'échange en question (arbitrage visible)
- . au cours de l'échange précédent (arbitrage caché).

Les arbitres utilisent deux types de fils sur le bus :

- . lignes de gestion d'accès : six (ou sept) fils réservés aux dialogues entre arbitres,
- . signaux de transfert : huit fils qui peuvent être destinés à cet effet ou pris parmi les fils d'adresses ou de données, servant à afficher la configuration des demandes.

Configuration non multiplexée

Dans le cas où les huit fils pour la configuration des demandes sont réservés à cet effet, on a alors un bus non multiplexé.

Cette configuration rend l'arbitrage toujours possible : si le signal BPAC (bus arbitrage possible) est activé en permanence (câblé à 1), ce fait rend toujours possible des arbitrages cachés. Si BPAC est câblé à 0, on n'aura que des arbitrages visibles.

Configuration multiplexée

Dans le cas où les huit fils de demande sont pris soit pour des fils d'adresse, soit pour des fils de données, on a alors un bus multiplexé.

L'arbitrage est alors possible :

- . en dehors des échanges (arbitrages visibles) (BPAC étant non nécessaire),
- . pendant un échange (arbitrages cachés) dès que l'unité qui pilote l'échange l'autorise à travers l'activation du signal BPAC. Ce signal sera fourni dès que les fils BA 17-24 sont disponibles.

L'avantage de la solution multiplexé par rapport à la solution non multiplexée découle du fait que la première solution présente un nombre de points de connexion plus réduit. Toutefois, le "timing" des échanges devient plus délicat. L'activation de BPAC ne doit se faire qu'à des moments bien précis.

Il faut préciser qu'actuellement la structure SM 90 emploie le mode multiplexé avec délivrance du signal BPAC [OLI 83.1].

Les principaux signaux sont les suivants (on se reportera à la figure 4 pour visualiser les interfaces) :

Demande d'accès au bus

L'unité qui désire accéder au bus active le signal DAB (demande d'accès au bus) (interface 1).

Affichage des demandes

La configuration des demandes d'accès au bus à un moment donné est affichée sur huit fils appelés $\overline{BA\ 17-24}$. En fait, chacune des huit unités de traitement possibles présentes sur le bus activent un seul fil \overline{BAi} parmi les huit $\overline{BA\ 17-24}$, selon leur adresse (interface 3). Il faut remarquer que chaque arbitre de bus doit initialiser son adresse par câblage à travers les broches UC1-3, donc l'unité de numéro 0 active $\overline{BA\ 17}$, celle de numéro 1, $\overline{BA\ 18}$, et ainsi de suite.

Arbitrage

C'est la combinaison de toutes les demandes présentes sur le bus qui sera présentée aux arbitres de la structure (interface 3). Le signal \overline{BAEQ} activé par le module qui a fait une demande DAB, indique qu'il faut effectuer un arbitrage sur cette configuration de demandes.

Divulcation du verdict

Le numéro du module de traitement autorisé à effectuer le prochain échange est affiché sur les trois fils BM 1-3 (interface 3) et le signal $\overline{BM\ 4}$ valide l'information portée par ces fils (interface 3). L'arbitre le plus prioritaire (égalité entre BM 1-3 et UC 1-3) envoie un signal DBA (demande de bus acceptée) (interface 1) au module de traitement, lequel prend ensuite le contrôle du bus, ce qui est signalé par \overline{DECH} (échange en cours sur le bus (interface 2)).

Algorithme de calcul des priorités

Le verdict est calculé suivant une règle de priorité à la fois fixe et rotative : le paramètre PF (nombre d'unités de traitement en priorité fixe) est câblé sur tous les arbitres ; en fonction de ce paramètre le verdict est calculé de la façon suivante :

. s'il existe des demandes de la part de certains modules "en fixe" (c'est-à-dire ceux de numéros de 0 à PF -1), celui de plus faible numéro est servi ;

. s'il n'existe des demandes que de la part des modules "en rotatif" (c'est-à-dire ceux de numéro PF à 7), la priorité la plus forte est accordée à celui de numéro N + 1, si N est le numéro du dernier module "en rotatif" ayant effectué un échange.

Supervision

Enfin, chaque arbitre comporte une fonction supervision qui, lorsqu'un temps trop long s'écoule entre une demande d'arbitrage ($\overline{\text{BREQ}}$) et le début de l'échange correspondant ($\overline{\text{BECH}}$), provoque un nouvel arbitrage effectué par un autre arbitre à travers l'activation du signal $\overline{\text{BNA}}$ (interface 3).

3. BUS MULTIBUS

3.1. Structure du bus

Proposé initialement par INTEL, le MULTIBUS est devenu un standard officiel de bus, normalisé par IEEE sous le nom de P 796 et repris par de nombreux fabricants de cartes 16 bits, souvent autour de microprocesseurs d'origines différentes, AMC, Forward Technology et NEC. Par ailleurs, INTEL, ainsi que d'autres fournisseurs offrent sur le marché un nombre assez important de cartes (SBC) compatibles MULTIBUS [GIR 82.2], [GIR 82.3], [GIR 82.4].

Dans le cadre de ce rapport, lorsqu'on dit bus MULTIBUS, on veut parler du bus pour les micros 8 / 16 bits, qui est nommé aujourd'hui plutôt MULTIBUS I, dès l'arrivée du bus MULTIBUS II [DEA 84] pour les microprocesseurs 32 bits.

Basé sur le concept maître/esclave et acceptant plusieurs maîtres (jusqu'à 16), il présente les caractéristiques suivantes :

- . capacité d'adressage de 1 mégaoctet, extensible à 16 mégaoctets,
- . bus de données de 16 bits acceptant des micros de 8 et 16 bits,
- . huit niveaux d'interruption,
- . débit maximal de cinq mégatransferts (octets ou mots de 16 bits) par seconde.

La figure 5 montre l'architecture MULTIBUS :

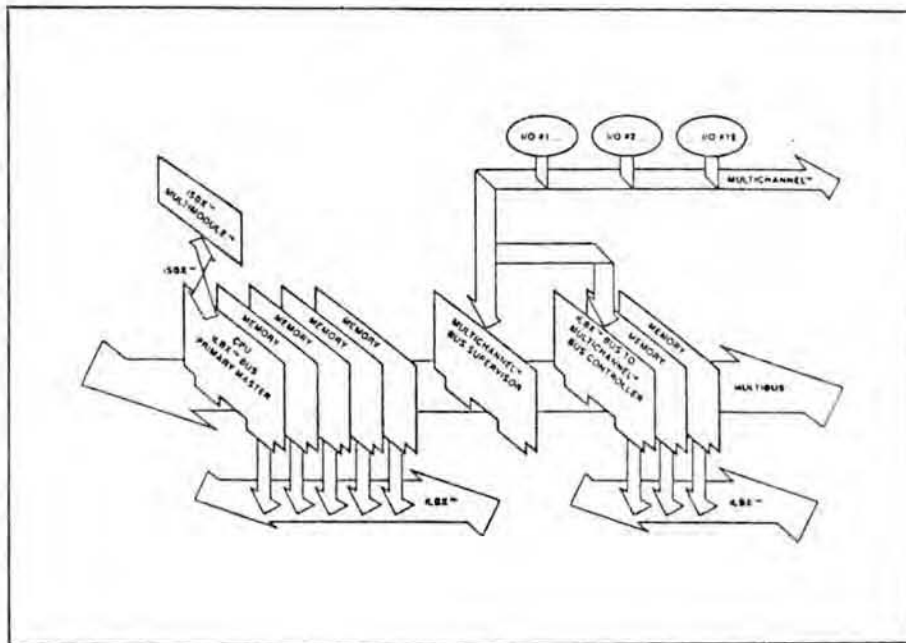


Figure 5 - architecture MULTIBUS

Le MULTIBUS se présente sur deux connecteurs :

. l'un de 86 broches, nommé P1, présente tous les signaux nécessaires pour le MULTIBUS sauf quatre lignes d'extension d'adresse (extension à 16 mégaoctets) ;

. l'autre de 60 broches, nommé P2, présente ces quatre lignes d'adresse en question, plus 56 autres broches pour l'expansion à travers l'utilisation de bus locaux (i LBH : INTEL LOCAL BUS EXTENSION) ou de bus spéciaux d'entrée/sortie (i SBHI/O : INTEL SPECIAL BUS EXTENSION FOR INPUT/OUTPUT). De plus, le connecteur P2 présente des signaux contre les coupures.

Dans le cadre de ce rapport on se restreint à la description des signaux du connecteur P1 :

1) Signaux de transfert :

- . des adresses sur 20 fils,
- . des données sur 16 fils.

Le MULTIBUS ne présente donc pas de multiplexage d'adresses et données ce qui augmente le nombre de points de connexion, mais permet cependant des débits de transfert plus importants (5 mégatransferts (octets ou mots de 16 bits) par exemple).

2) Signaux de contrôle :

a) Lignes de commande :

- . des signaux permettant au module maître du bus de déterminer à son esclave le type d'échange en question : lecture/écriture mémoire, lecture/écriture entrée/sortie, ...

b) Lignes de service :

- . un fil d'initialisation et des fils d'alimentation.

c) Lignes d'interruption :

- . huit niveaux d'interruption et un fil de prise en compte de l'interruption. Les interruptions peuvent être vectorisées ou non vectorisées.

d) Lignes de gestion d'accès au bus :

- . des signaux permettant de résoudre l'allocation de bus qui peut être faite de façon série ou parallèle. La logique de résolution des priorités est synchronisée sur chaque module maître (10 MHz maximum).

3.2. Connexion d'une unité de traitement au bus

La connexion d'une unité de traitement au module maître ou MULTIBUS peut être visualisée à partir de la figure 6 :

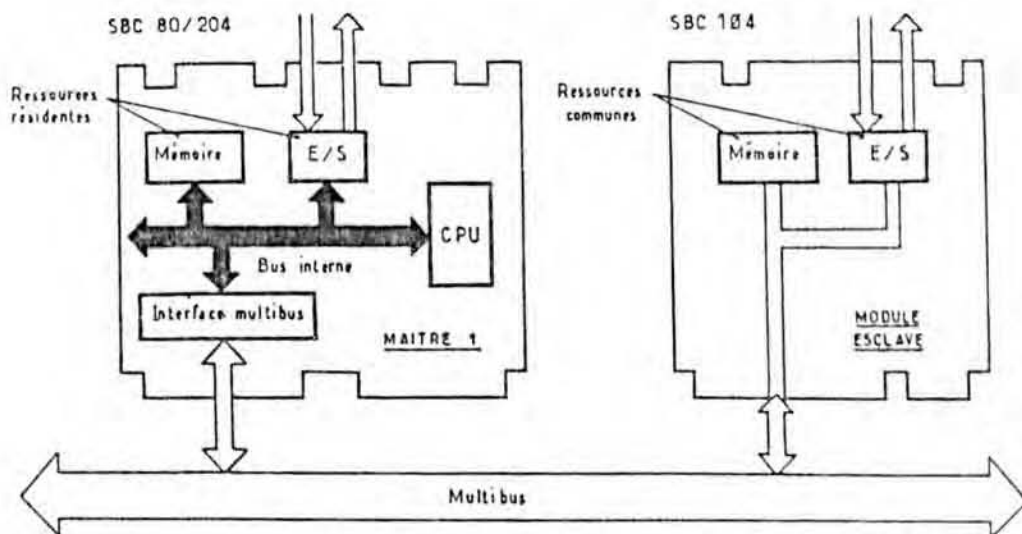


Figure 6 - connexion d'un module maître au MULTIBUS

D'après cette figure, on constate qu'à l'intérieur de chaque carte maître présente sur le bus, il y a un bus interne qui véhicule les informations comme demande de bus, demande de bus acceptée, ainsi que les adresses et les données qui passent à travers l'interface MULTIBUS sur le bus commun ; ces informations seront reprises par le module esclave avec qui l'unité de traitement est en train de dialoguer.

L'interface MULTIBUS de la figure 6 est détaillé dans la figure 7 :

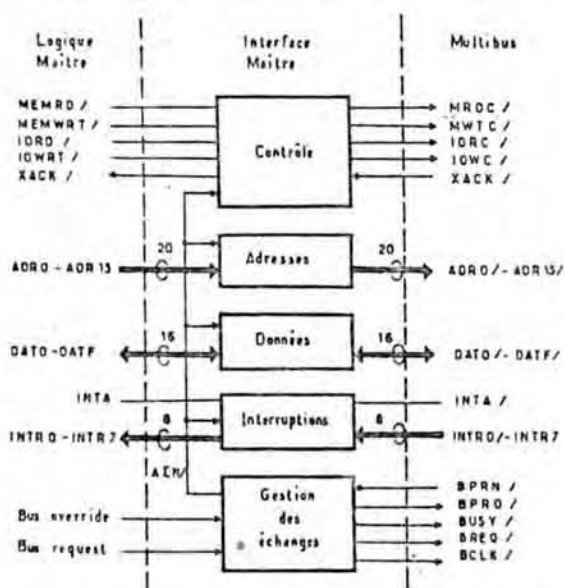


Figure 7 - interface MULTIBUS sur chaque carte maître

3.3. Technique de l'arbitrage

Comme on l'a constaté à partir des figures précédentes, chaque carte maître incorpore un bloc de gestion des échanges, lequel a été initialement implanté en composants discrets et puis par un circuit intégré développé par INTEL, l'arbitre de bus 8289 [REC 80].

La gestion des priorités sur le MULTIBUS peut s'effectuer en parallèle ou en série.

Technique de gestion série des priorités

La position physique d'une carte maître détermine la priorité, car celle-ci est fixée par câblage sur le fond de panier (aucun circuit extérieur n'est nécessaire). Toutefois, ce câblage fige l'emplacement des cartes et la défaillance d'une carte peut être nuisible pour le système. De plus, le nombre de modules maîtres que l'on peut raccorder en série dépend de la période de l'horloge BCLK et des temps de propagation de module maître à module maître. Normalement, à 10 MHz on ne peut raccorder que trois maîtres [SAB 79]. La figure 8 montre la gestion série des priorités.

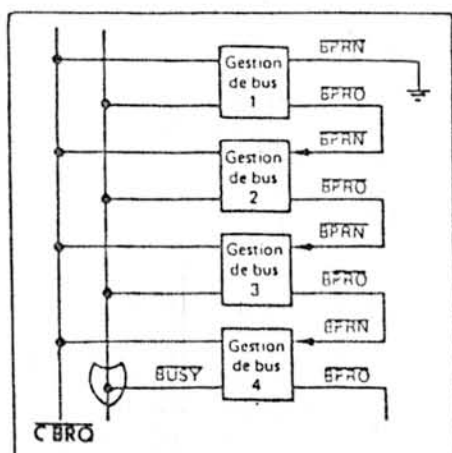


Figure 8 - gestion série des priorités

D'après la figure 8, on voit que cette technique est décentralisée.

Comme on peut voir par cette figure 8, la broche \overline{BPRO} du maître le plus prioritaire est connectée à la broche de demande de bus accepté \overline{BPRN} du maître dont le niveau de priorité est juste en dessous, et ainsi de suite.

Le maître le plus prioritaire à un instant donné a sa broche \overline{BPRN} active et \overline{BPRO} inactive (laquelle empêche l'accès au bus de tous les maîtres moins prioritaires dans la chaîne). Toutefois, un maître moins prioritaire peut demander le bus à travers le signal \overline{CBAQ} (\overline{BPRN} inactif et \overline{BUSY} actif).

L'accès au bus suppose la disponibilité de ce bus, signalée par \overline{BUSY} (ou câblé de tous les signaux \overline{BUSY} de chaque arbitre).

Technique de gestion parallèle des priorités

Cette technique nécessite des circuits extérieurs de gestion des priorités (voir figure 9), ce qui constitue un arbitrage centralisé.

Toutefois, l'arbitrage se fait plus rapidement (< 100 ns) et on peut raccorder au bus jusqu'à huit modules maîtres, chacun avec un niveau de priorité fixe distinct.

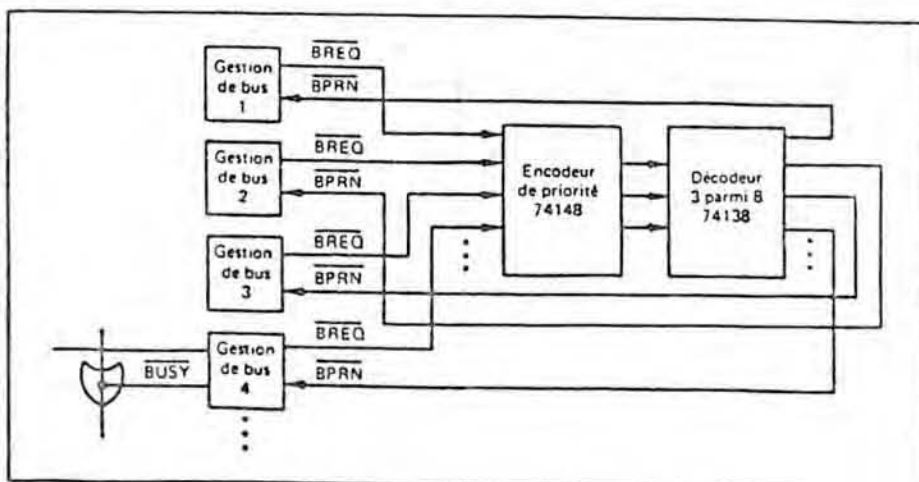


Figure 9 - gestion parallèle des priorités

D'après cette technique, dès qu'un module maître veut accéder au bus, il formule sa demande à travers le signal \overline{BREQ} , laquelle sera prise en charge par un encodeur de priorité 8×3 (composant TTL 74 LS 148). Cet encodeur présente en sortie l'adresse encodée de la carte maître ayant fait la demande la plus prioritaire. Ce verdict est relié à l'entrée d'un décodeur 3×8 (composant TTL 74 LS 138) qui active en sortie (niveau bas) une ligne (\overline{BPRN}) parmi huit. Cette ligne correspond au maître qui a fait la demande \overline{BREQ} la plus prioritaire.

Il faut rappeler que le composant 74 LS 148 ne gère que des niveaux fixe de priorité. Il faut remarquer en outre que lorsqu'un maître obtient l'autorisation d'accès au bus, il n'en dispose pas immédiatement. Il doit attendre que le maître en possession du bus, ait terminé son cycle de transfert, ce qui est signalé sur la broche BUSY.

En annexe 2 on donne les diagrammes d'arbitrage pour le bus MULTIBUS.

3.4. Arbitre de bus 8289

Comme on l'a dit précédemment au § 1.3.3., la gestion des échanges au niveau de chaque module maître peut être remplacée par le circuit 8289. Toutefois, ce circuit est plutôt utilisé dans les systèmes 16 bits.

Malgré son nom, ce circuit n'incorpore pas de logique d'allocation de bus, qui est toujours réalisée par l'ensemble encodeur - décodeur.

Le rôle principal de ce circuit est de décharger le microprocesseur de la tâche d'obtention du bus.

La figure 10 montre une configuration proposée par INTEL pour une carte maître de bus 16 bits (microprocesseur 8086) qui contient le 8289.

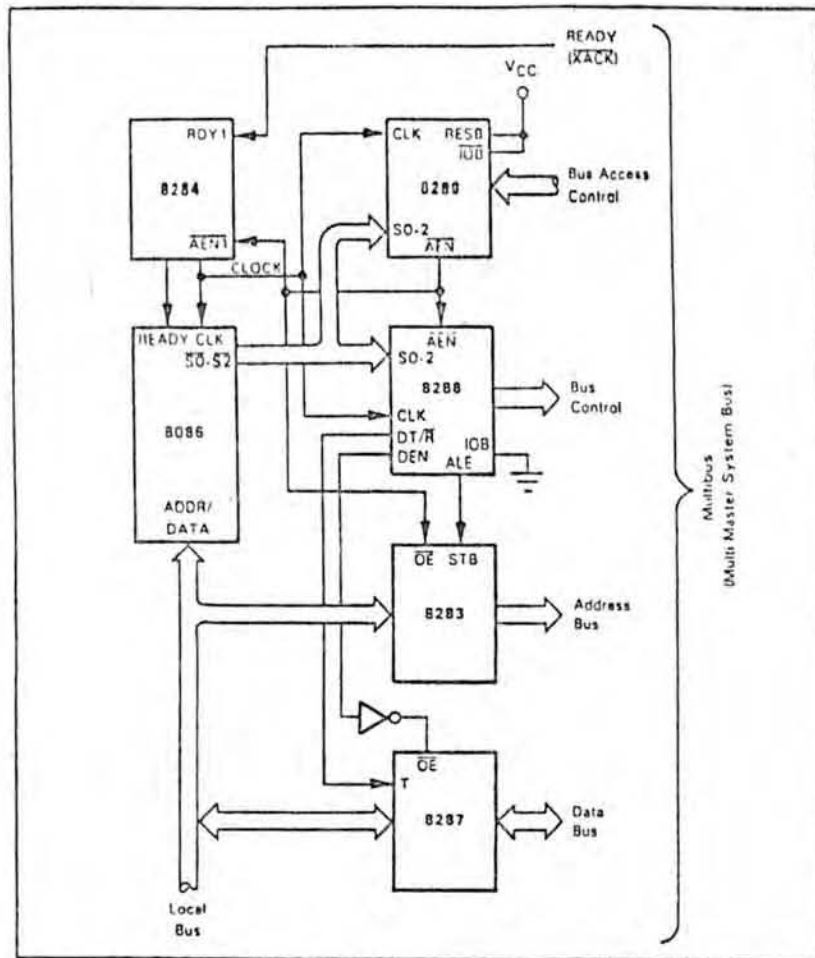


Figure 10 - architecture d'une carte maître dépourvue de ressources locales, contenant l'arbitre de bus 8289

Pour initialiser un cycle de bus, le 8086 force sur les lignes d'état (S0, S1, S2) un état actif (au repos S2 S1 S0 = 1 1 1). Le 8289 et le CPU travaillent de façon synchrone à travers l'horloge CLK (elle est différente de l'horloge de résolution de priorités BCLK décrite auparavant). Si à cet instant le 8289 n'a pas le contrôle du bus (ce n'est pas lui qui force un état bas sur BUSY), il délivre une demande de bus (BREQ) et une demande commune de bus (CBRQ). Ce protocole est tout à fait identique à celui qu'on vient de décrire § 3.3. Chap. 1, Le type de réponse du maître qui détient le bus à CBRQ (relâcher le bus après le transfert en cours, relâcher immédiatement le bus ou ignorer purement et simplement CBRQ) est au choix du concepteur du système bâti autour du bus MULTIBUS.

Le 8289 va maintenir le signal AEN (address enable) inactif jusqu'à ce qu'il récupère le contrôle du bus ; dès lors il active BUSY et AEN.

Dès que la communication maître/esclave s'achève (détection par le maître du signal HACK (figure 7, bloc de contrôle) envoyé par l'esclave), l'unité centrale remet son registre d'état à l'état passif.

Si jamais un maître de bus plus prioritaire essaie de forcer le maître qui possède le bus à le relâcher pendant son transfert (à travers la perte de priorité BPRN), le 8289 va maintenir le contrôle du bus, (BUSY actif) jusqu'à ce que le registre d'état revienne à l'état passif.

L'arbitre 8289 peut fonctionner selon quatre modes différents [INT 84] :

- . mode bus d'entrée/sortie (IOB) lorsque les maîtres disposent des facilités d'entrée/sortie sur les cartes ;

- . mode bus résident (RESB) lorsque les maîtres disposent de mémoire ROM et EPROM à l'intérieur de leurs cartes ;

- . mode bus d'entrée/sortie et bus résident (IOB et RESB) lorsque les modules de traitement sont assez fournis en mémoire et ressources d'entrée/sortie ;

. mode bus unique, lorsque le maître ne dispose d'aucune ressource locale et à ce moment là devra toujours demander le bus système pour accéder à la mémoire ou aux entrées/sorties (figure 8).

En plus de ces modes de fonctionnement, le 8289 reçoit en entrée les signaux (LOCK, ANYRQST et CRQLCK) qui définiront la manière de relâcher le bus.

LOCK activé permet au 8289 de maintenir le contrôle du bus malgré la perte de la plus forte priorité.

ANYRQST activé par câblage, force le 8289 à relâcher le bus au profit d'un maître moins prioritaire, soit à la fin du cycle en cours, soit immédiatement (s'il n'y a pas de cycle de bus en cours), dès que la broche CBAQ est active.

Au moment de relâcher le bus, le 8289 doit désactiver sa demande BREQ.

CRQLCK activé fait que l'on ignore les demandes moins prioritaires CBAQ. De ce fait, un maître plus prioritaire ne relâchera jamais le bus à un maître moins prioritaire.

4. BUS UME

4.1. Structure du bus

Les spécifications du bus UME (de Versa Module Eurocard), soutenues par MOSTEK, MOTOROLA et SIGNETICS, définissent une architecture de bus d'interconnexion de cartes micro-ordinateurs 16/32 bits, principalement des familles de cartes à unité centrale de type 68 000.

Par ailleurs, le UME est en cours de standardisation devant l'IEEE (P 1014) aux Etats Unis et devant l'IEC en Europe [GRO 84.1].

Ce bus, basé aussi sur le concept maître/esclave, présente les caractéristiques suivantes :

- . gestion d'accès au bus parallèle (les priorités peuvent être fixes ou tournantes),
- . quatre niveaux d'accès,
- . cycle indivisible de lecture/modification écriture, permettant l'utilisation de sémaphores pour se réserver une ressource commune dans un environnement multimicroprocesseur,
- . débit de 20 mégaoctets par seconde,
- . sept niveaux d'interruption,
- . notion de cryptage liée à chaque maître,
- . canal de communication série qui assure, en fonctionnement multimicroprocesseur, des échanges de signaux de service ou d'alarme,
- . dispositifs qui accroissent la fiabilité (indicateurs d'erreur de bus, de défaillance système et d'alimentation).

La figure 11 montre l'architecture du bus UME :

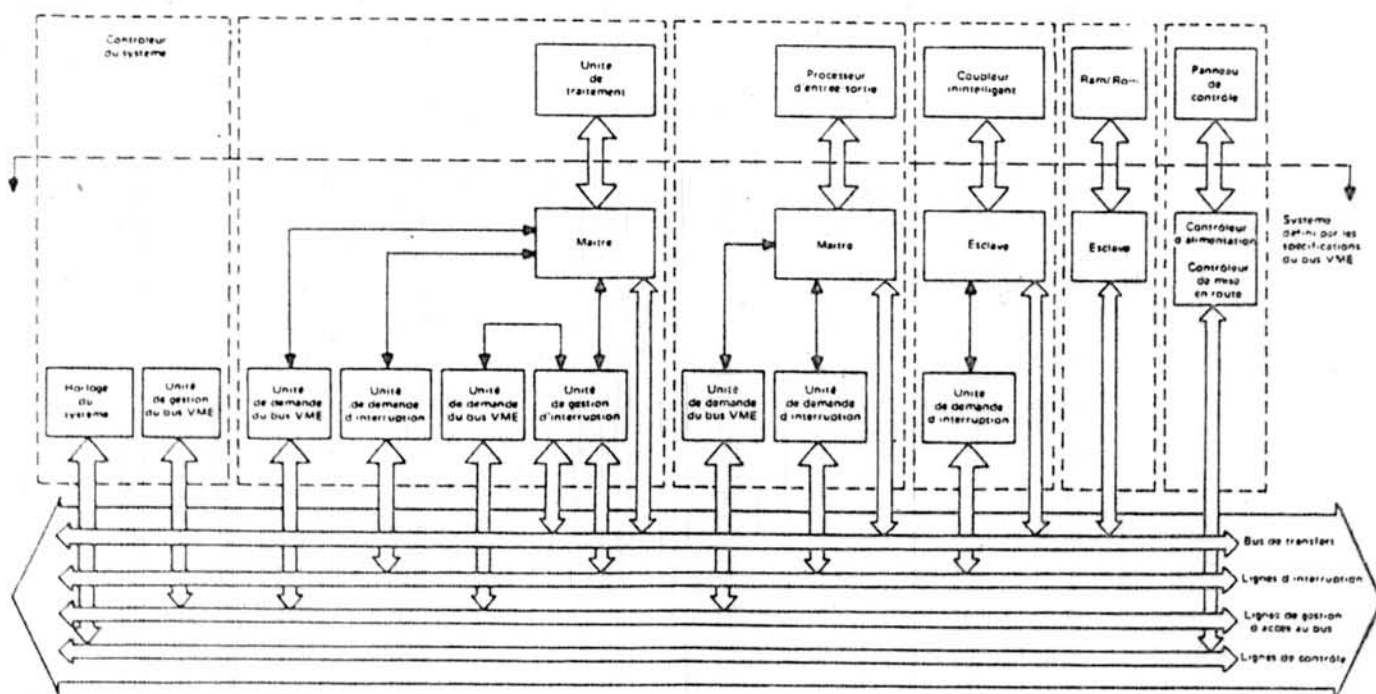


Figure 11 - Principe de connexion des modules au bus UME

D'après cette figure, on constate qu'outre les modules maîtres et esclaves, on dispose de deux cartes spéciales :

- . l'une fait office de contrôleur de système (intégrant l'horloge du système et l'unité de gestion du bus DME),
- . l'autre sert au contrôle d'alimentation et au contrôle de mise en route.

Le bus DME est implanté sur des cartes au format double Europe.

Electriquement, le bus DME se répartit sur deux connecteurs de 96 broches :

- . l'un, nommé P1, permet de traiter complètement des microprocesseurs 16 bits (espace d'adressage de 16 mégaoctets),
- . l'autre, nommé P2, permet une extension d'adresses et données, permettant le raccordement de microprocesseurs 32 bits (espace d'adressage de 4 gigaoctets).

Dans le cadre de ce rapport, on se restreint à la description des signaux du connecteur P1 :

1) Signaux de transfert :

- . des adresses sur 24 fils,
- . des données sur 16 fils.

Le bus DME est alors un bus non-multiplexé.

2) Signaux de contrôle :

a) Lignes de commande :

- . des signaux de contrôle de données, comme lecture/écriture, donnée disponible,
- . des signaux de contrôle d'adresse, dont 6 fils de modification d'adresse ne sont pas disponibles ni sur le SM bus ni sur le MULTIBUS.
Ces signaux de modification d'adresse permettent :
 - . la configuration dynamique du système,
 - . le choix de l'emplacement d'un esclave à l'intérieur de l'espace d'adressage d'un maître,
 - . l'accord de privilèges d'adressage à un esclave, etc ..

b) Lignes de service :

- . des signaux d'initialisation, de panne et d'horloge du système, ainsi qu'un signal (ACFAIL) de coupure de l'alimentation,
- . des signaux liés au bus série (donnée et horloge série).

c) Lignes d'interruption :

- . sept niveaux d'interruption et trois fils pour la prise en compte de l'interruption.

d) Lignes de gestion d'accès au bus

- . des signaux permettant l'allocation du bus au maître plus prioritaire. Le UME dispose de quatre niveaux distincts de priorité d'accès au bus.

4.2. Connexion d'une unité de traitement au bus UME

La connexion d'une unité de traitement au bus UME est décrit dans la figure 12 :

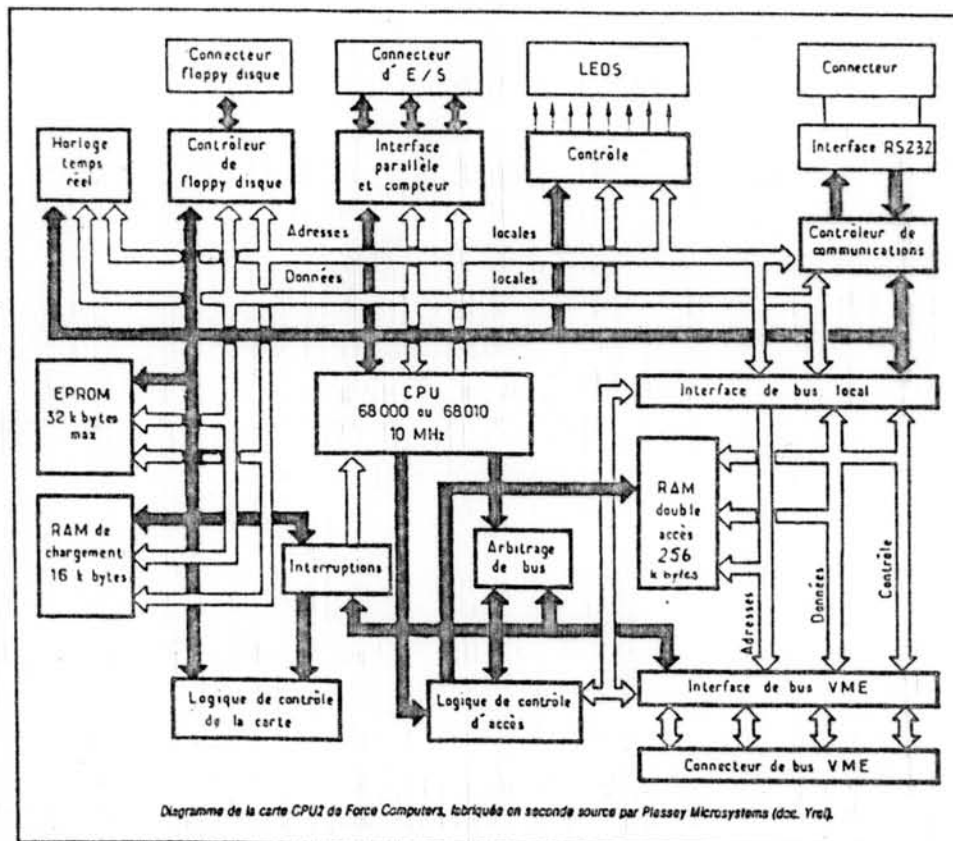


Figure 12 - connexion d'un module maître bâti autour du 68000 à un bus UME, utilisant l'exemple de la carte CPU2 de Force Computers [GR0 84.2]

D'après cette figure, on constate que lorsque le CPU veut s'emparer du bus, il dialogue avec le bloc arbitrage de bus qui se charge de transmettre la demande faite par le CPU à l'unité de gestion centralisée d'accès au bus, positionnée sur la première carte du fond de panier (figure 11).

Dès que l'on obtient la plus forte priorité et ainsi le droit d'accès, l'arbitrage de bus retransmet un signal de demande d'accès au bus accepté au bloc logique de contrôle d'accès, lequel décharge le microprocesseur 68000 de cette tâche.

Dès lors le bloc de logique de contrôle d'accès envoie vers la RAM double accès, les commandes de validation des adresses et données.

4.3. Technique d'arbitrage

Le UME possède quatre niveaux de priorité [MOT] (lignes $\overline{BR3} - 0$). Ces lignes sont activées par les unités de demande d'accès situées sur chaque module maître. C'est à l'unité de gestion d'accès au bus, située sur la carte contrôleur du système, de recevoir ces demandes. Cette unité peut gérer les demandes selon un algorithme de priorités fixes ou un algorithme de priorités tournantes.

Pour la technique de hiérarchie fine, l'unité de gestion alloue le bus à la demande la plus prioritaire (la hiérarchie des priorités va, dans l'ordre décroissant, de $\overline{BR3}$ à $\overline{BR0}$) et génère un niveau bas sur la broche d'autorisation d'accès au bus \overline{BGHIN} correspondante. Cette autorisation d'accès au bus de transfert est conditionnée par la disponibilité de ce bus, signalée par un niveau haut sur la broche \overline{BUSY} . La broche \overline{BCLR} de l'unité de gestion signale au maître en possession du bus, qu'une demande d'accès plus prioritaire est en attente.

Pour la technique de hiérarchie tournante, le niveau alloué à chaque ligne n'est plus fixe, mais tournant ; le maître qui a le niveau de priorité le plus élevé a, après accès au bus, le niveau de priorité le plus bas. Mise à part la différence de fonctionnement au niveau des priorités, les unités de gestion d'accès au bus sont semblables, à part le fait que la ligne \overline{BCLR} est supprimée dans le mode avec priorité tournante.

La figure 13 montre les signaux fournis et reçus par l'unité de gestion d'accès au bus.

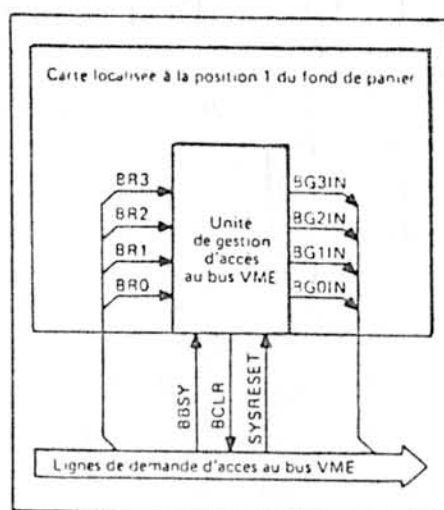


Figure 13 - unité de gestion d'accès au bus VME

L'unité de gestion d'accès au bus peut être remplacée par le circuit arbitre de bus MC 68452 développé par MOTOROLA, [MOT 83] qui traite huit niveaux de demande avec priorités fixes.

Dis à vis de l'unité de demande d'accès au bus VME, elle peut fonctionner selon deux modes :

- . RWD (Release When Done), c'est-à-dire qu'elle libère le bus dès que le maître a terminé un transfert ;
- . ROR (Release On Request), c'est-à-dire qu'elle ne libère le bus, après la fin d'un transfert, que s'il se produit une nouvelle demande.

Cette dernière approche permet de diminuer le nombre de demandes générées par un maître, qui constituent un pourcentage non négligeable du trafic sur le bus.

Dès qu'une unité de demande d'accès au VME reçoit l'autorisation d'accéder au bus, à travers l'activation de sa ligne BGXIN, elle porte la broche BBSY à l'état bas, dès que le maître prend en charge le bus. Il faut dire que l'unité de gestion ne prendra pas en compte d'autre demande d'accès, tant que le bus ne sera pas libre, ce qui empêche des arbitrages cachés.

Comme l'indique la figure 14, les maîtres doivent gérer deux lignes supplémentaires dans le cadre d'une demande d'accès au bus VME. Ces lignes sont :

- . ACFAIL : signale qu'une chute d'alimentation du secteur a été détectée ; dans ce cas, le maître doit rendre le bus quel que soit le transfert qu'il est en train d'exécuter ;

- . BCLR : indique une demande plus prioritaire ; dans ce cas, le maître détermine, suivant le type de transfert qu'il est en train d'exécuter, l'instant où il libérera le bus.

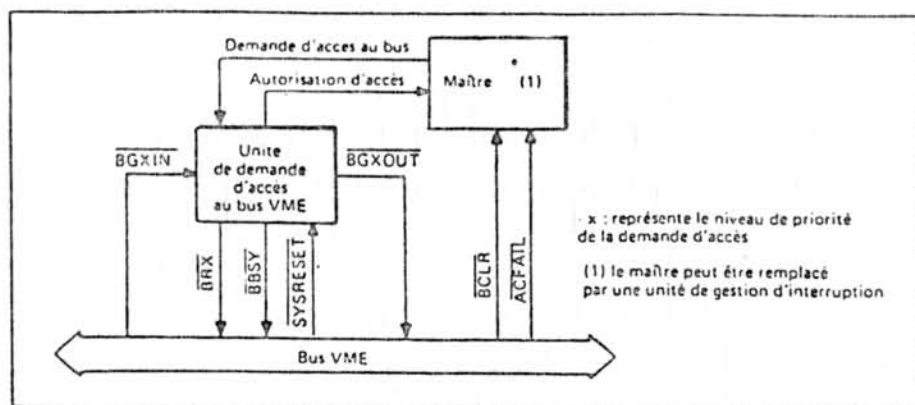


Figure 14 a - Logique de demande d'accès au bus dans le cas où le maître restitue le bus immédiatement après usage : mode RWD

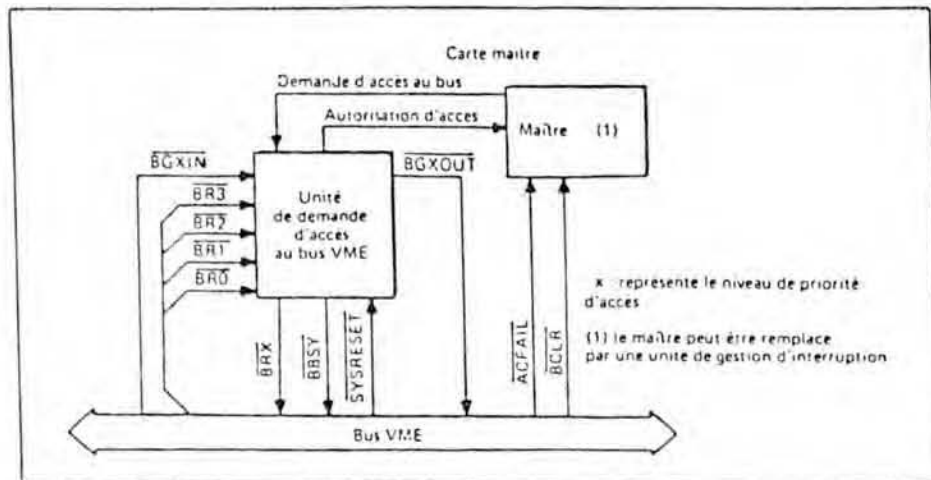


Figure 14 b - Logique de demande d'accès au bus dans le cas où le maître ne restitue le bus que sur demande : mode ROR

Les figures 14a et 14b représentent les modes de fonctionnement de l'unité de demande d'accès au bus VME.

En annexe 2 on donne des diagrammes d'arbitrage pour le bus VME.

4.4. Arbitre de bus MC 68452

Comme on l'a dit au § 4.3. Chap. 1, MOTOROLA a mis sur le marché un circuit arbitre de bus compatible avec le bus VME. Cet arbitre toutefois diffère conceptuellement des autres arbitres qu'on vient de présenter

. il rend des arbitrages centralisés (un seul arbitre pour toutes les cartes connectées au bus VME),

. il remplace l'unité de gestion d'accès au bus UME, tandis que l'arbitre MULTIBUS 8289 joue le rôle d'unité de demande d'accès au bus, (étant présent sur chaque maître, mais n'effectuant pas d'arbitrage du tout (sauf gestion série de résolution de priorités)).

Le figure 15 montre l'utilisation de l'arbitre MC 68452 dans une structure UME :

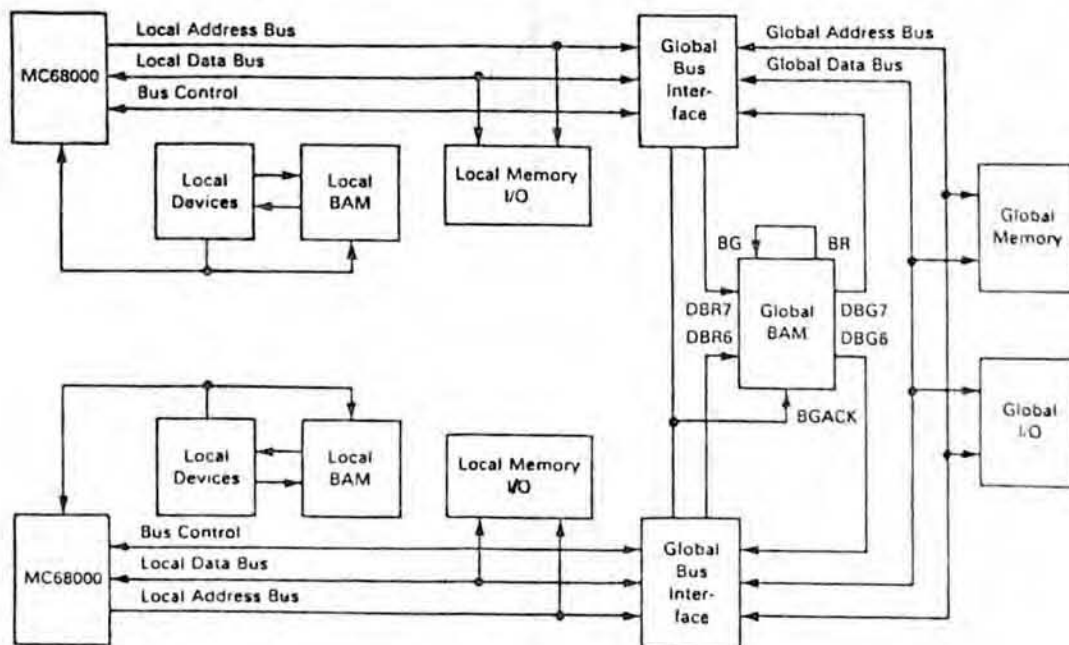


Figure 15 - l'arbitre MC 68452 dans un environnement multimicroprocesseur

Caractéristiques

Les caractéristiques principales de ce circuit sont les suivantes :

- . il réalise l'arbitrage de huit demandes,
- . l'assemblage de circuits MC 68452 permet de traiter plus de demandes,
- . algorithme d'arbitrage fixe,
- . peut fonctionner comme arbitre local de différents circuits (contrôleur de DMA, contrôleur de disque, CRTC, ..) à l'intérieur d'une même carte maître.

Mode de fonctionnement

Le MC 68452 offre une fonction centrale d'arbitrage à travers l'utilisation d'une paire de signaux : demande (\overline{DBR}) et autorisation d'accès au bus (\overline{DBG}) pour chaque unité de traitement (jusqu'à 8).

Lorsqu'une unité de traitement désire accéder au bus, elle active sa broche \overline{DBR} . A partir de ce moment là, l'unité de traitement surveille sa ligne d'autorisation d'accès (\overline{DBG}). Lorsque ce signal devient actif (un seul signal \overline{DBG} est activé à la fois), l'unité de traitement active le signal \overline{BGACK} et commence alors son échange sur le bus. Pour que l'unité maintienne le contrôle du bus tout au long de son transfert, elle doit maintenir le signal \overline{BGACK} actif.

Lorsque le transfert est fini, le maître enlève le signal \overline{BGACK} pour permettre que d'autres unités de traitement puissent accéder au bus.

5. METHODES D'ARBITRAGE

D'après la présentation que l'on vient de faire, des techniques d'arbitrage de la SM 90, du MULTIBUS et du bus VME, on peut dégager un certain nombre de caractéristiques qui nous seront utiles pour la suite :

. Arbitrage centralisé lorsque l'arbitrage est fait par un seul organe présent sur la structure multimicroprocesseur. Cet organe pouvant être réalisé en composants discrets ou remplacé par un circuit intégré spécialisé.

. Arbitrage décentralisé lorsque l'arbitrage est effectué en parallèle par tous les arbitres présents sur la structure. Chaque module de traitement dispose d'un arbitre. Cette solution est plus fiable, augmentant la disponibilité et la modularité.

. Arbitrage visible lorsque l'arbitrage se fait pendant la non occupation du bus. De ce fait, les arbitrages ralentissent le débit de données sur le bus.

. Arbitrage caché lorsque l'arbitrage se fait simultanément avec le transfert en cours. Ce type d'arbitrage est intéressant quand le bus est très chargé.

. Arbitrage déterminé en fonction de niveaux fixes de priorité [JOL 83] (appelés aussi linéaires [COU 79]) lorsque toutes les unités de traitement raccordées au bus possèdent un indice hiérarchique (une priorité) qui est fixé selon un critère spécifique à l'application. La priorité linéaire de chaque unité de traitement peut être fixée par la position physique de la carte (daisy chain) soit par straps. A un instant donné, si plusieurs demandes d'accès au bus se présentent, l'arbitrage rendra le bus à la demande la plus prioritaire.

. Arbitrage déterminé en fonction de niveaux rotatifs de priorité (appelés aussi circulaires [COU 79]) lorsque malgré le fait que chaque unité de traitement possède une priorité spécifique fixée par straps, la dernière unité de traitement qui a exécuté un transfert sur le bus devient la moins prioritaire.

. Arbitrage déterminé en fonction de niveaux mixtes de priorité [OLI 82] lorsque des unités configurées en priorité fixe ou rotative sont raccordées au bus, étant les premières les plus prioritaires que les secondes. Dans le premier groupe (fixes) la priorité est fixée par straps, tandis que dans le second groupe (rotatives) la priorité la plus forte est accordée à l'unité de numéro $N + 1$ si N est le numéro de la dernière unité "en rotatif" ayant effectué un échange.

Le tableau 1 montre comment se répartissent les structures SM 90 MULTIBUS et UME autour de ces concepts.

Structure multi-microprocesseur	SM 90	MULTIBUS	UME
type d'arbitrage			
centralisé			
décentralisé			
visible			
caché			
fixe			
rotatif			
mixte			

Tableau 1 - Méthodes d'arbitrage

Une croix représente le type d'arbitrage employé

6. FONCTIONS D'UN ARBITRE DE BUS

D'après l'analyse de différents arbitres de bus, on peut dégager un certain nombre de fonctions dont les différents arbitres peuvent disposer :

Affichage des demandes

Dès que l'arbitre de bus reçoit une demande de son unité de traitement, il retransmet cette demande à travers le bus de communication de façon à ce que le ou les arbitres présents sur la structure la prennent en compte.

Arbitrage

Dès que le ou les organes d'arbitrage prennent en compte les demandes, ils doivent, selon un algorithme qui leur est propre, déterminer la demande la plus prioritaire parmi les demandes affichées.

Retransmission du verdict

Dès qu'un arbitrage a été fait, il faut que celui qui a gagné en priorité soit notifié de son droit d'accès au bus.

Supervision

Dès que l'unité de traitement qui a eu le droit d'accéder au bus, n'exerce pas ce droit (suite à panne, ..), en n'envoyant pas sur le bus commun ses données, un nouvel arbitrage doit être établi pour que le droit d'accéder au bus soit passé à une unité de traitement non défaillante.

7. CONCLUSION

D'après l'analyse faite sur les différents bus de communication étudiés, ainsi que sur les techniques d'arbitrage associées, on a pu dégager un ensemble de méthodes et de fonctionnalités d'arbitrage dont peut disposer un circuit arbitre de bus.

Parmi les circuits arbitres de bus présentés (ABC 90 pour la SM 90, circuit MC 68452 pour VME et ensemble de composants discrets TTL encodeur et décodeur pour MULTIBUS), seul l'ABC 90 peut réaliser les arbitrages selon toutes les méthodes évoquées (centralisé, décentralisé, visible, caché) en offrant l'ensemble de fonctionnalités dégagées (affichage des demandes, arbitrage, retransmission de verdict et supervision).

Dans le chapitre suivant, on va étudier la compatibilité entre l'ABC 90 et les architectures MULTIBUS et VME.

2 . ETUDE DE COMPATIBILITE
ENTRE LE CIRCUIT ARBITRE DE BUS ABC 90
ET LES BUS DE COMMUNICATION MULTIBUS ET DME :
APPROCHE DISCRETE

1. INTRODUCTION

Comme on l'a vu au chapitre précédent, pour chaque structure de bus les fonctions d'allocation du bus sont toujours réalisées par un organe que nous nommons arbitre. Cet arbitre, comme on vient de le voir, peut être soit un circuit intégré (SM 90 : circuit ABC 90, UME : circuit MC 68452), soit un assemblage de composants discrets (MULTIBUS : encodeur 74LS148 et encodeur 74LS138).

L'analyse de ces arbitres nous a montré qu'ils ont été construits sur mesure pour le bus correspondant. Toutefois, si on peut utiliser un arbitre conçu pour une structure dans une autre, et si cela apporte des avantages, on s'aperçoit de l'intérêt d'étudier la compatibilité des différents arbitres avec les différents bus étudiés.

On précise qu'on entend par compatibilité d'un arbitre H avec des bus Y, Z, le fait qu'il puisse être utilisé comme organe d'allocation de bus, dans les structures pour lesquelles il n'a pas été conçu préalablement (Y, Z, ...). Cela implique le fait d'obéir aux chronogrammes de séquençement des autres bus et que les signaux (électriques) délivrés par l'arbitre H, doivent être compatibles électriquement avec les signaux des bus Y, Z, ... (voir Annexe 4).

Pour que cette compatibilité puisse être possible, nous prenons comme hypothèses de travail :

- . soit d'ajouter des composants extérieurs,
- . soit de modifier éventuellement l'algorithme implanté dans un arbitre H, le transformant en H', lequel serait compatible directement avec les bus Y, Z .. sans pour autant avoir besoin de logique supplémentaire.

Dans ce chapitre nous nous sommes donnés comme hypothèse de chercher d'abord la compatibilité d'un seul arbitre, sans modification de son algorithme, avec les autres types de bus. Toutefois, cette restriction peut être minimisée si on choisit pour H l'arbitre ou l'unité d'arbitrage dont les caractéristiques sont les plus générales et l'algorithme le plus performant.

En l'occurrence, notre choix de H est l'arbitre conçu pour la structure SM 90, le circuit arbitre de bus ABC 90, car son degré d'universalité nous permet d'envisager de l'utiliser dans d'autres structures. En plus des fonctions communes des arbitres, l'ABC 90 apporte des fonctionnalités additionnelles telles que la supervision et l'arbitrage rotatif [OLI 82].

Il n'existe pas à notre connaissance des outils de vérification de compatibilité entre des unités définies et implantées à partir de spécifications basées sur des chronogrammes.

La démarche suivie dans ce travail consiste à utiliser les outils de simulation logico-fonctionnelles disponibles au CNET/CNS pour valider les solutions de compatibilité proposées ci-après. Nous ne prétendons pas qu'elles soient exhaustives.

2. COMPARAISON ENTRE ARBITRES

D'après l'analyse qu'on a faite au chapitre précédent, les fonctions réalisées par les différents arbitres peuvent être schématisées par la figure 16.

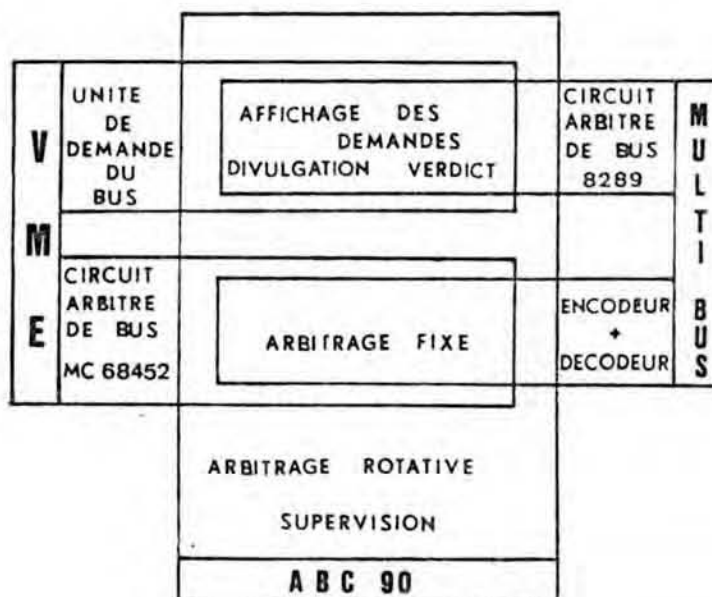


Figure 16 - Fonctions réalisées par les arbitres de bus

On constate aisément que toutes les fonctions réalisées par l'arbitrage dans les structures UME et MULTIBUS sont comprises dans la gamme de fonctions offertes par l'ABC 90.

En outre, l'ABC 90 traite des priorités rotatives, ce qui pourrait également être fait par des composants discrets, mais à un fort prix :

- . nombre élevé de composants,
- . grand nombre de connexions (donc fiabilité moindre par rapport à l'implantation à travers un seul circuit).

En outre, l'ABC 90 présente une fonction de supervision qui n'est pas incluse dans les autres arbitres. Donc, si on se limite pour l'instant au niveau des fonctionnalités, on peut dire que l'arbitre ABC 90 est compatible avec les structures UME et MULTIBUS.

On peut remarquer aussi, d'après cette figure, que l'arbitre réalise des fonctions qui normalement sont accomplies par plusieurs unités ou organes différents. Cela est vrai :

- . pour le UME (unité de demande de bus et circuit arbitre de bus MC 68452)
- . et pour le MULTIBUS (circuit arbitre 8289 et ensemble d'arbitrage encodeur - décodeur).

De ce fait, on peut se poser la question de savoir où mettre l'ABC 90 dans une structure quelconque de compatibilité envisagée.

Il faut envisager de placer l'ABC 90 dans tous les cas de figures possibles. Et l'analyse de solutions type et des résultats de simulation nous montrera les meilleurs emplacements envisageables pour l'arbitre ABC 90.

3. ETUDE DE COMPATIBILITE MULTIBUS - ABC 90

3.1. Proposition d'un ensemble de fils cohérents

A l'égard de la connectique, l'emplacement de l'ABC 90 dans une architecture centralisée n'entraîne aucun besoin de prendre quelques fils des bus standard pour un autre emploi. Toutefois, l'architecture décentralisée, malgré qu'elle soit moins sensible aux pannes et au bruit que l'architecture centralisée, elle exige la redéfinition (ou la prise de quelques fils disponibles) de quelques fils des bus.

L'analyse des signaux du MULTIBUS et de l'ABC 90 nous conduit à définir un ensemble de fils cohérents pour une utilisation ABC 90 - MULTIBUS (architecture décentralisée).

La structure MULTIBUS [INT 81] ne permet pas de multiplexer la configuration de demandes avec les fils d'adresses ou de données :

- . pendant un cycle d'écriture, les adresses et les données sont valides simultanément, ne laissant pas une fenêtre de temps pour que la configuration de demandes soit présentée sur ces fils ;

- . il n'y a pas de signal correspondant à BPAC fourni par la structure MULTIBUS, permettant l'échantillonnage des demandes sur des fils d'adresses ou de données.

De ce fait, le problème est de trouver des fils disponibles sur MULTIBUS.

Ce problème est gênant car en cas de pénurie de broches disponibles sur le connecteur P1, on doit faire un choix entre :

- . prendre i fils dont on a besoin parmi les 24 d'adresse (si on compte les 4 bits supplémentaires introduits par le connecteur P2), ce qui entraîne une perte de l'espace d'adressage,

- . ou prendre statiquement 8 des 16 fils de données, ce qui interdit l'utilisation de microprocesseurs 16 bits sur ces modules.

Toutefois, il faut signaler que dès que le nombre de modules maîtres présents sur le bus diminue, le nombre de broches sur le(s) connecteur(s) MULTIBUS à chercher devient moins critique, diminuant ainsi les conséquences (perte d'espace d'adressage, longueur de mots de 8 bits).

Ces problèmes de perte d'espace d'adressage ou de réduction de la longueur de mot, peuvent être masqués si on utilise le connecteur optionnel P2, lequel contient 10 broches non réservées à l'usage spécifique de chaque système. Toutefois ce connecteur P2 vient d'être normalisé pour le bus iLBx de la famille INTEL [INT 83] dans des applications dont les grosses capacités mémoire sont traitées.

Il faut remarquer cependant que l'utilisateur a le choix entre utiliser le connecteur P2 dans un contexte iLBx ou l'utiliser de façon à mettre à profit les broches non réservées, ce qui est intéressant pour la compatibilité MULTIBUS - ABC 90.

On décrit en annexe 3 une proposition d'utilisation cohérente des broches standard MULTIBUS, dans le cadre de l'utilisation de l'ABC 90 dans une architecture décentralisée.

3.2. Proposition pour rendre possible l'utilisation compatible de l'ABC 90 dans une structure MULTIBUS

On sait, d'après les chronogramme d'arbitrage MULTIBUS montrés en annexe 2, que le MULTIBUS, en général, utilise des arbitrages cachés. Donc l'ABC 90 dans une structure MULTIBUS doit aussi pouvoir permettre ce type d'arbitrage.

L'équivalent de BREQ

Dès que l'ABC 90 sera utilisé comme organe d'arbitrage, il faut lui fournir après les demandes $\overline{BA} 17 - 24$ un signal de demande d'arbitrage (\overline{BREQ}). Dans son fonctionnement normal (ABC 90 dans SM 90), ce signal est activé 40 ns après l'affichage de demande.

Comme le MULTIBUS ne présente pas de signal correspondant à \overline{BREQ} dans le sens ABC 90 (demande d'arbitrage) (voir annexe 3), on doit le fabriquer. On propose l'assemblage de toutes les demandes \overline{BAi} sur une porte AND 8 entrées. Dès qu'il y a une demande active (niveau bas) \overline{BREQ} sera aussi actif. Toutefois en raison de l'implantation interne de l'ABC 90 (automate d'arbitrage), cette broche ne peut pas être activée en permanence.

Pour résoudre ce problème, la solution la plus simple est de désactiver cette porte, dès qu'un échange se fait par le bus (\overline{BECH} à 0). Cette implémentation ne pénalise pas le UME qui ne connaît que des arbitrages visibles, néanmoins pour le MULTIBUS cette implémentation est pénalisante car elle interdit les arbitrages cachés.

On peut utiliser un pulse négatif, fabriqué à partir de \overline{BECH} dont la durée inférieure à \overline{BECH} est calculée de telle sorte que lorsque ce signal est à 0 on remet \overline{BREQ} à 1 (pour que l'ABC 90 soit sensible de nouveau aux demandes \overline{BAi} - nouveau front descendant de \overline{BREQ}).

La figure 17 montre la logique proposée pour le signal \overline{BREQ} .

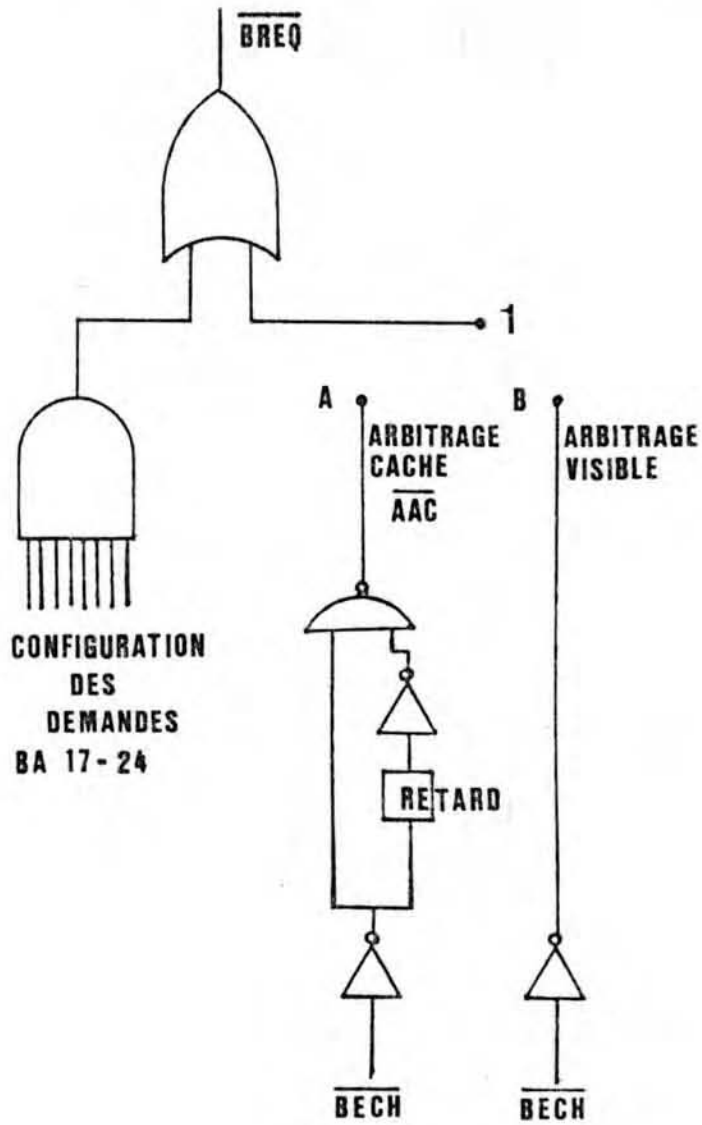
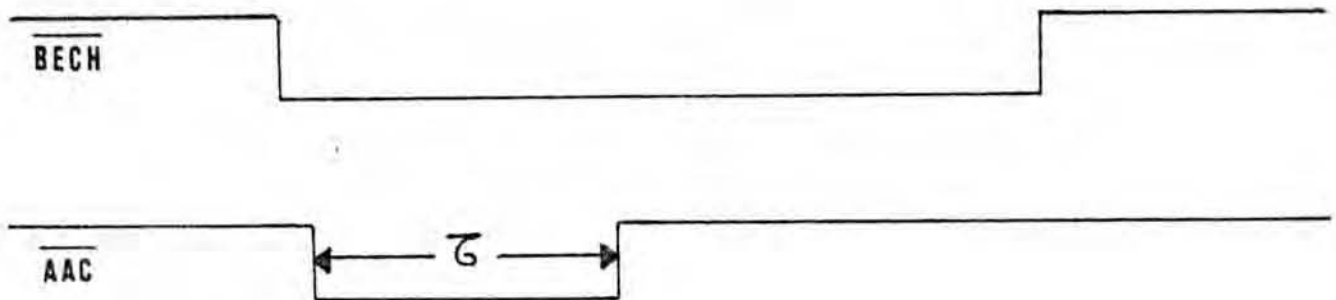


Figure 17 - schéma logique permettant des arbitrages visibles ou cachés

A partir de ce schéma, on obtient le chronogramme suivant pour l'autorisation des arbitrages cachés (qu'on nomme \overline{AAC}) :



La durée du pulse négatif de \overline{AAC} , τ , doit être déterminé en fonction du "timing" de l'ABC 90 : on veut en effet, que dès qu'on autorise de nouveau les arbitrages (front montant de \overline{AAC}), toutes les demandes d'accès au bus soient présentes.

On sait que dès qu'il y a une divulgation de verdict ($\overline{BM4}$), les demandes formulées par les ABC 90 sont retirées (lignes \overline{BAi}) ainsi que le signal de demande d'arbitrage \overline{BREQ} .

Le réaffichage des demandes \overline{BAi} ne s'effectuera qu'à peu près 200 ns après la prise du bus (\overline{BECH}) (selon le séquençement de la partie contrôle de l'ABC 90).

. Pour les modules ne comportant pas d'arbitre de bus (cartes SBC) les demandes faites ne seront enlevées qu'après avoir été satisfaites.

D'après ces considérations, on a pris pour τ , un temps de l'ordre de 200 ns.

Supervision

Comme on l'avait montré au § 2.4. Chap. 1, l'ABC 90 dispose d'une fonction supervision, laquelle est activée dès qu'un temps donné (réglé par une capacité externe, broche KP de l'arbitre) est dépassé entre le front descendant d'une demande d'arbitrage (\overline{BREQ}) et le front descendant du signal d'occupation du bus (\overline{BECH}).

Dès que la supervision est activée (signal \overline{BNA} : actif bas) les demandes (\overline{BA} 17-24) faites par les modules contenant l'arbitre, ainsi que le signal de demande d'arbitrage (\overline{BREQ}), sont retirées et représentées dès que le signal \overline{BNA} devient de nouveau inactif.

Le chronogramme de supervision de l'ABC 90 est montré ci-dessous (figure 18) :

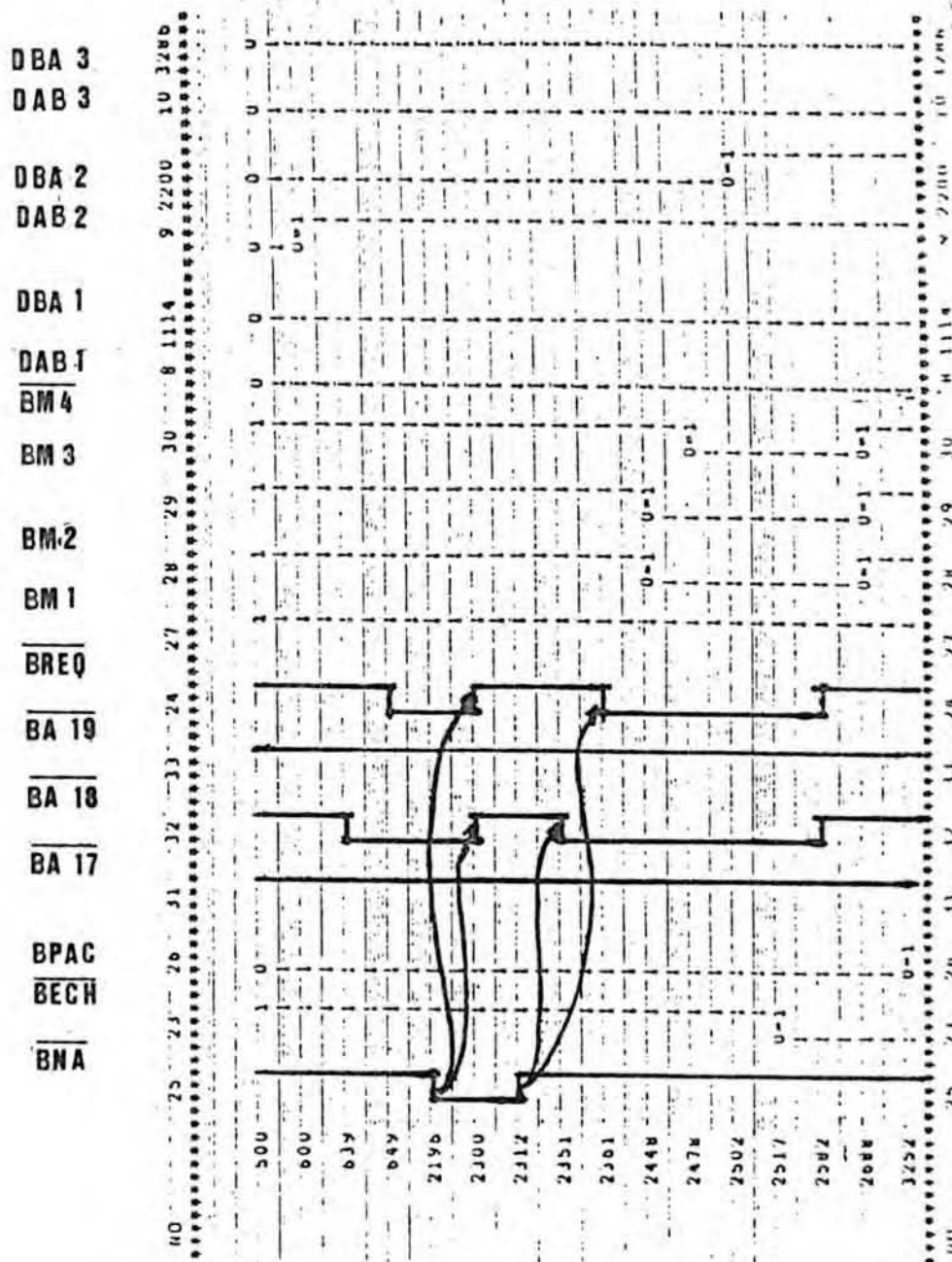


Figure 18 - Chronogramme de supervision pour la structure SM 90

Comme le signal \overline{BNA} est particulier à l'ABC 90, les modules SBC (cf. § 3.1. Chap. 1) ne vont pas retirer ses demandes à la suite de \overline{BNA} . Cela a pour conséquence que le signal \overline{BREQ} , tel qu'il est montré à la figure 15, restera toujours actif, n'autorisant pas la fonction de supervision.

Pour autoriser la fonction de supervision, on relie le signal de supervision \overline{BNA} , à travers une porte ETNON, soit au signal de bus occupé (\overline{BECH} : n'autorisant ainsi que les arbitrages visibles, soit au signal d'autorisation d'arbitrage caché (signal \overline{AAC} généré à partir de \overline{BECH} (voir figure 17) : permettant les arbitrages cachés).

La figure 19 montre le schéma logique qui sert à valider la fonction de supervision, laquelle peut être mise à profit pour les deux types d'arbitrages : visible et caché.

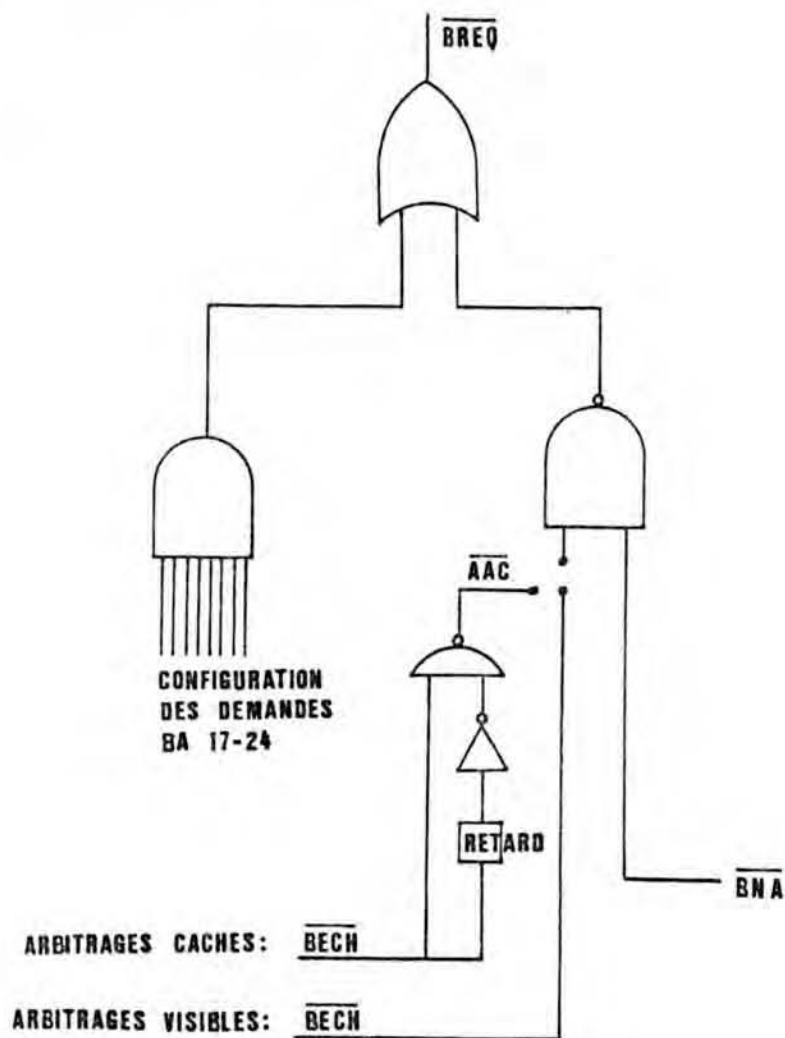


Figure 19 - Schéma logique permettant la supervision

On souligne que ce schéma logique est assez souple car il permet :

- . des arbitrages visibles,
- . ou des arbitrages cachés,
- . et la supervision.

3.3. Architectures compatibles proposées

Comme on l'a vu au § 3. du chapitre 1, vis à vis de la gestion d'accès au bus, le MULTIBUS présente deux sortes de modules :

. le circuit 8289, sur chaque unité de traitement (UT) 16 bits ou bloc logique pour les unités de traitement 8 bits, dont le rôle est d'afficher les demandes et retransmettre le verdict vers l'unité de traitement. Voir figure 16.

. logique externe de résolution de priorités, constitué d'un encodeur 8×3 et un décodeur 3×8 .

Il n'est pas intéressant de placer l'ABC 90 à la place du 8289, car dans cette configuration il ne réalise pas la fonction d'arbitrage. En outre cet emplacement n'est même pas possible (câblage statique de BREQ à VDD).

Si on le place dans le deuxième cas de figure, l'ABC 90 remplacera, tout en offrant des avantages considérables, la logique centralisée d'arbitrage MULTIBUS. Pour cette architecture, l'ABC 90 se constituera donc, malgré sa destination d'origine en organe d'arbitrage centralisé. Cela constitue donc une architecture centralisée (§ 3.3.1. Chap. 2). De même, on propose au § 3.3.2. Chap. 2 une architecture décentralisée.

3.3.1. Architecture centralisée

L'architecture centralisée MULTIBUS est montrée à la figure 20. Dans cette architecture on profite de la fonctionnalité d'arbitrage de l'ABC 90. En fait les ABC 90 présents sur la carte centrale d'arbitrage (figure 20) ont le rôle de réaliser les arbitrages.

Comme on peut le constater d'après cette figure, on ne connecte pas d'unité de traitement à ces arbitres (fonction affichage de demandes inhibée : broche DAB câblée à zéro).

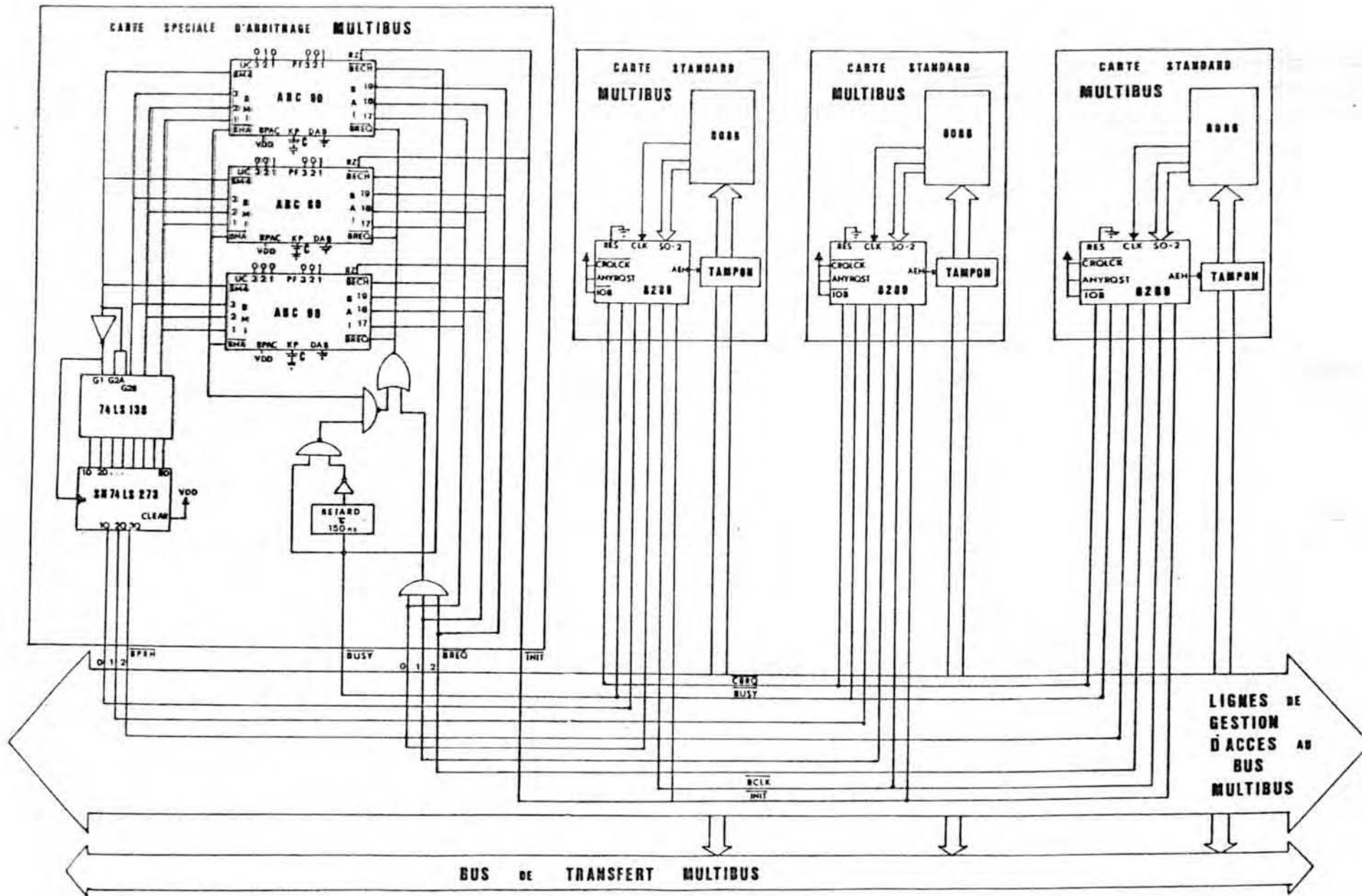


FIG. 20 : Architecture MULTIBUS : Arb. Centralisé - ABC 90

Nombre d'ABC 90 en fonction du nombre d'UT raccordées au bus

Il faut préciser que sur la carte centrale d'arbitrage il y aura au moins autant d'ABC 90 qu'il y a d'unités (MULTIBUS) de traitement connectées au bus, car sinon dans certains cas, le verdict ne sera pas divulgué.

Rappelons que l'ABC 90 ne divulgue le verdict que si le dernier transfert sur le bus a été effectué par l'unité de traitement auquel il est associé.

Comme dans notre cas, à chaque ABC 90 sur cette carte il n'y a pas d'unité de traitement (dans le sens de l'architecture SM 90) associée (DAB est câblée à 0), le but recherché est d'avoir une correspondance exacte entre les adresses des unités de traitement (qui ne contiennent que le circuit 8289) et les adresses des ABC 90 sur la carte centrale d'arbitrage.

De cette manière, chaque fois qu'une unité de traitement, d'adresse H, prend le bus, un arbitre de bus ABC 90 possédant la même adresse, placé sur la carte centrale d'arbitrage divulguera l'adresse de la prochaine unité de traitement qui aura le droit de procéder à un transfert sur le bus.

Décodage du verdict

Le verdict calculé par l'ABC 90 sera délivré de manière codée sur 3 bits (BM1-3). Ce verdict sera validé par un signal de validation BM4 actif 50 ns après la divulgation du verdict.

Ce verdict ne peut pas être connecté directement aux différentes unités de traitement sur le bus, car celles-ci n'ont qu'un seul signal (actif bas) de reconnaissance d'avoir obtenu la plus forte priorité.

On est donc amenés à utiliser un décodeur (3 x 8) (74 LS 138) à partir du ET CABLE des signaux BM1-3 issus de chaque arbitre.

Le décodage sera validé dès que le signal BM4 devient actif. Si le circuit 74 LS 138 n'est pas activé (entrées G1 : BM4, G2A : BM4, G2B : BM4) il affiche 1 à toutes ses sorties.

Les sorties de ce décodeur sont connectées dans le même ordre que les demandes BA 17-24 aux broches BPRN de chaque unité de traitement.

Enregistrement du verdict

D'après l'analyse des chronogrammes d'arbitrage MULTIBUS (voir annexe 2), on constate que le signal d'autorisation d'accès au bus BPRN ne devient inactif, sur un unité de traitement qui possède le bus, que si une autre unité de traitement plus prioritaire gagne le droit d'accéder au bus, en voyant sa broche BPRN active. Bien entendu, il n'y a qu'un seul signal BPRN actif à la fois.

Toutefois, dès que le signal de validation du verdict n'est plus actif, les sorties du décodeur sont toutes mises à 1, ce qui ne correspond pas à la fonctionnalité MULTIBUS.

On est donc amenés à interposer un registre 8 bits (SN 74 LS 273) à la sortie du décodeur, lequel n'enregistre les sorties du décodeur que lorsque le signal de validation du verdict est activé (entrée CLOCK du 74 LS 273). On s'assure ainsi qu'on aura toujours un signal BPRN actif parmi 8.

Initialisation

Les adresses des unités de traitement raccordées au bus peuvent être n'importe lesquelles de 0 à 7. Toutefois, à l'initialisation (signal RZ à zéro pendant au minimum 400 ns), c'est toujours à l'arbitre de bus ABC 90 d'adresse 0 présent sur la carte centralisée, de divulguer le verdict.

Si jamais il n'y a pas d'arbitre configuré à l'adresse 0 (ou s'il est défaillant) la fonction de supervision permettra à l'arbitre correspondant à l'unité de traitement la plus prioritaire de divulguer le verdict. Toutefois, cette supervision ralentit le bus. Aussi, s'avère-t-il intéressant que la carte centralisée ait un ABC 90 avec l'adresse 0.

Avantages de cette architecture

- . Possibilité d'avoir des niveaux rotatifs de priorité,
- . supervision des arbitrages rendus, incorporation donc de mécanismes de contrôle et de récupération d'erreurs.

Inconvénients

. Cette architecture est plus chère en matériel (circuits ABC 90, décodeur, registre 8 bits, portes logiques) que la solution centralisée d'origine MULTIBUS (encodeur et décodeur) ;

. les arbitrages (mesures entre affichage de demandes \overline{BA}_i et \overline{BM}_4 plus le temps de traversée du décodeur) sont plus lents : environ 150 ns pour l'architecture proposée contre 40 ns pour l'encodeur et décodeur. Toutefois, cet inconvénient devient mineur dès qu'on fait des arbitrages cachés.

3.3.2. Architecture décentralisée

On peut aussi proposer une architecture qui réalise des arbitrages décentralisés (voir figure 21) :

Cette architecture dispose :

1. d'un ABC 90 sur chaque carte maître spéciale utilisant toutes ses potentialités : affichage, arbitrage, retransmission du verdict et supervision. On aurait ainsi plusieurs ABC 90 travaillant en parallèle à la manière de la structure SM 90, chacun sur une carte spéciale MULTIBUS ;

Le dialogue entre arbitres nécessite toutefois 6 fils du bus fond de panier qui ne sont pas offerts par le bus MULTIBUS § 3.1. Chap. 2). En outre, pour que les ABC 90 travaillent en parallèle, il faut qu'ils voient tous la configuration de demandes sur le bus (l'ABC 90 peut accepter jusqu'à 8 demandes).

Donc les cartes spéciales MULTIBUS contenant l'ABC 90 requièrent une redéfinition des broches du connecteur, ce qui entraîne une perte de potentialité de cette carte. Une proposition d'utilisation cohérente des broches du connecteur MULTIBUS est donnée en annexe 3.

2. des cartes standard contenant le circuit 8289,

3. logique centralisée d'arbitrage (encodeur + décodeur), activée dès qu'un module maître du bus ne contenant pas l'ABC 90 prend le bus.

Il faut signaler que pour l'ABC 90 l'arbitre qui doit divulguer le verdict est celui qui a effectué le dernier échange sur le bus. Donc si le module maître en question ne comportait pas d'arbitre, la divulgation du verdict sera faite par les circuits encodeur-décodeur.

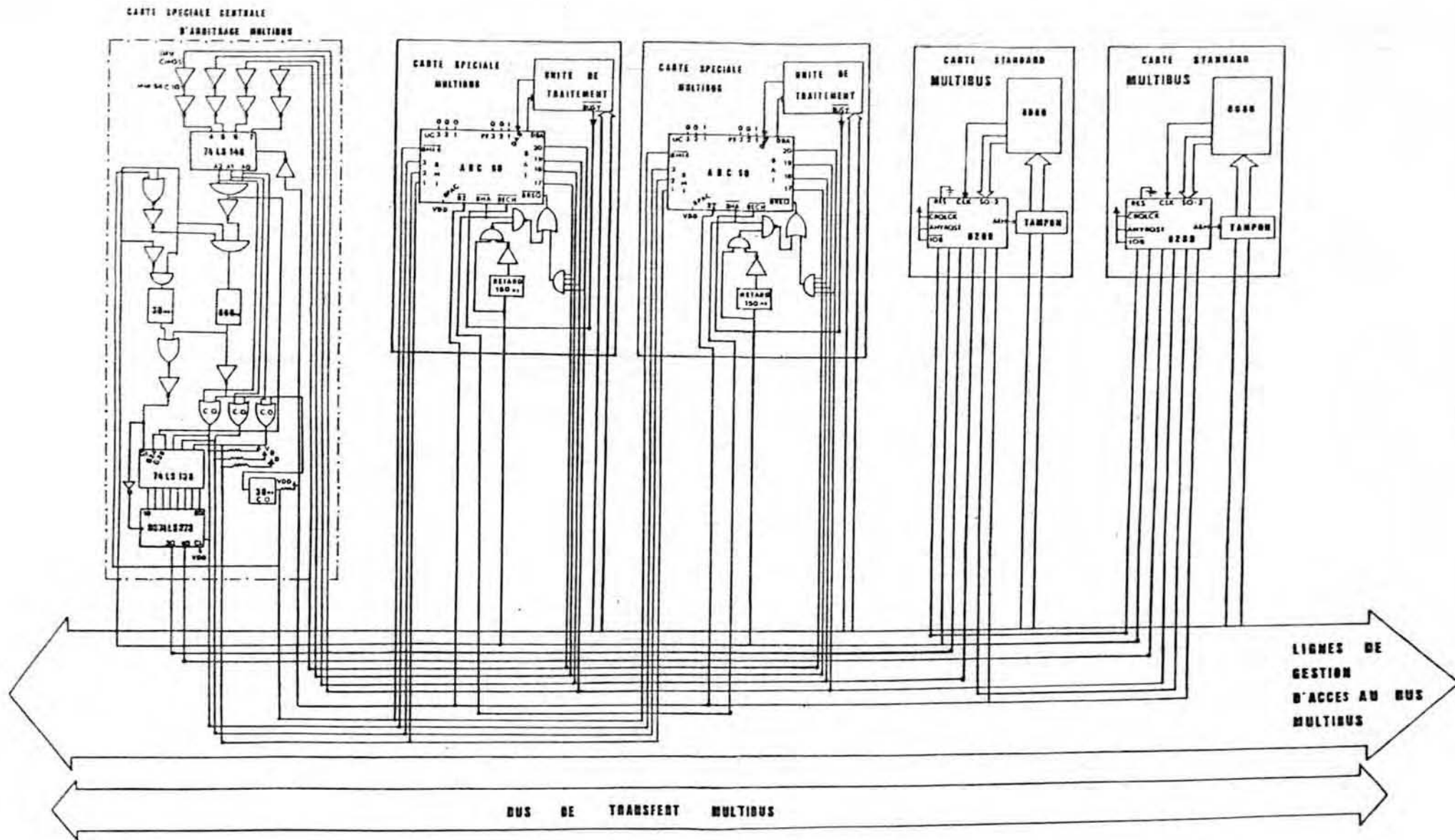


FIG. 21 : Architecture MULTIBUS : Arb. Décentralisé - ABC 90

Configuration des demandes

Comme cette architecture peut avoir ses arbitrages faits par deux organes différents, il faut bien entendu présenter la configuration de demandes BA 17-24 aux arbitres de bus ABC 90 présents sur la structure ainsi que pour l'encodeur.

Toutefois, en raison de la vitesse différente des circuits encodeur (36 ns maximum) et ABC 90 (environ 120 ns entre BAi et BM4), pour éviter des problèmes de basculement, on doit prendre la précaution suivante :

. interposer à chaque entrée de l'encodeur (BAi) deux portes inverseur CMOS MM 54 C 10 connectées en série (retard typique de 60 ns chacune) pour que le temps de réponse de l'encodeur plus portes soit à peu près égal à celui nécessaire à l'arbitre pour divulguer le verdict.

Validation du verdict

Pour le verdict, on fera un ET CABLE de tous les verdicts. Bien entendu, il ne faut pas que les deux types d'organe d'arbitrage délivrent leurs verdicts au même moment, ce qui entraînerait des conflits à cause de la différence d'algorithme d'arbitrage implanté sur chacun d'eux (fixe pour l'encodeur-décodeur et rotatif par exemple pour l'ABC 90).

On doit donc réaliser une exclusion mutuelle entre les verdicts délivrés par ces deux types d'arbitre. La solution proposée est schématisée à la figure suivante :

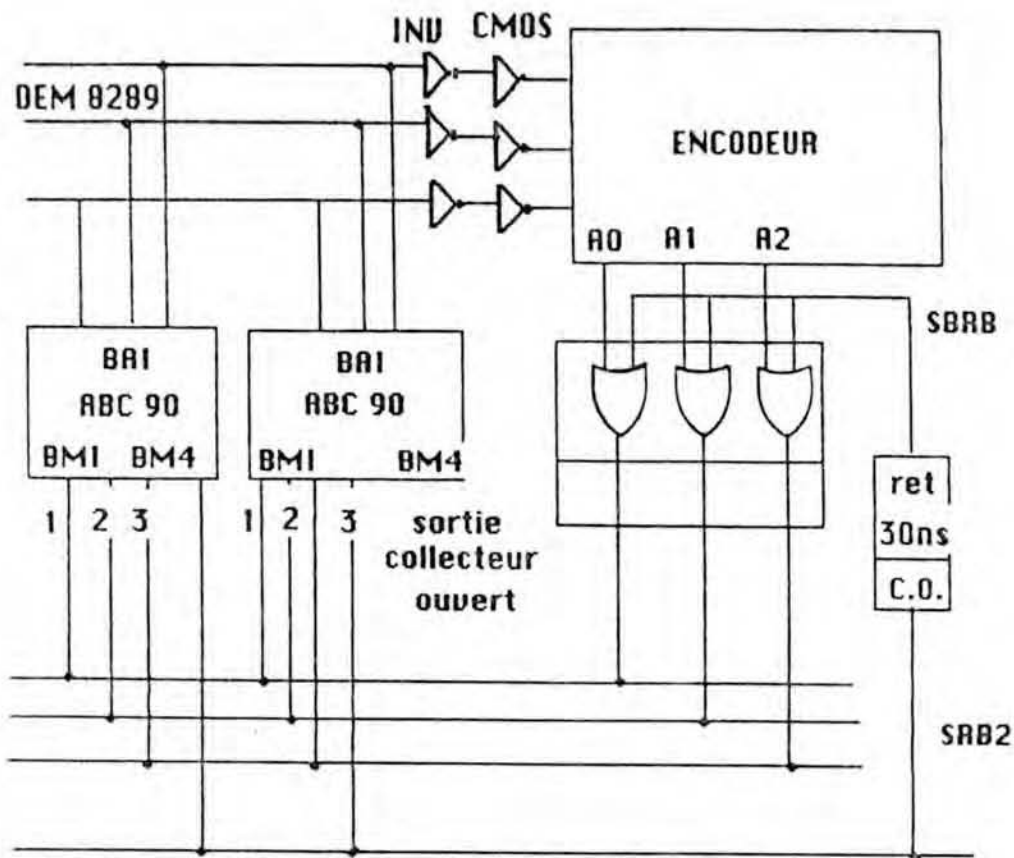


Figure 22 - architecture décentralisée :
divulgateion et validation du verdict

Pour assurer l'exclusion mutuelle entre les deux types de verdict, on propose le schéma logique suivant :

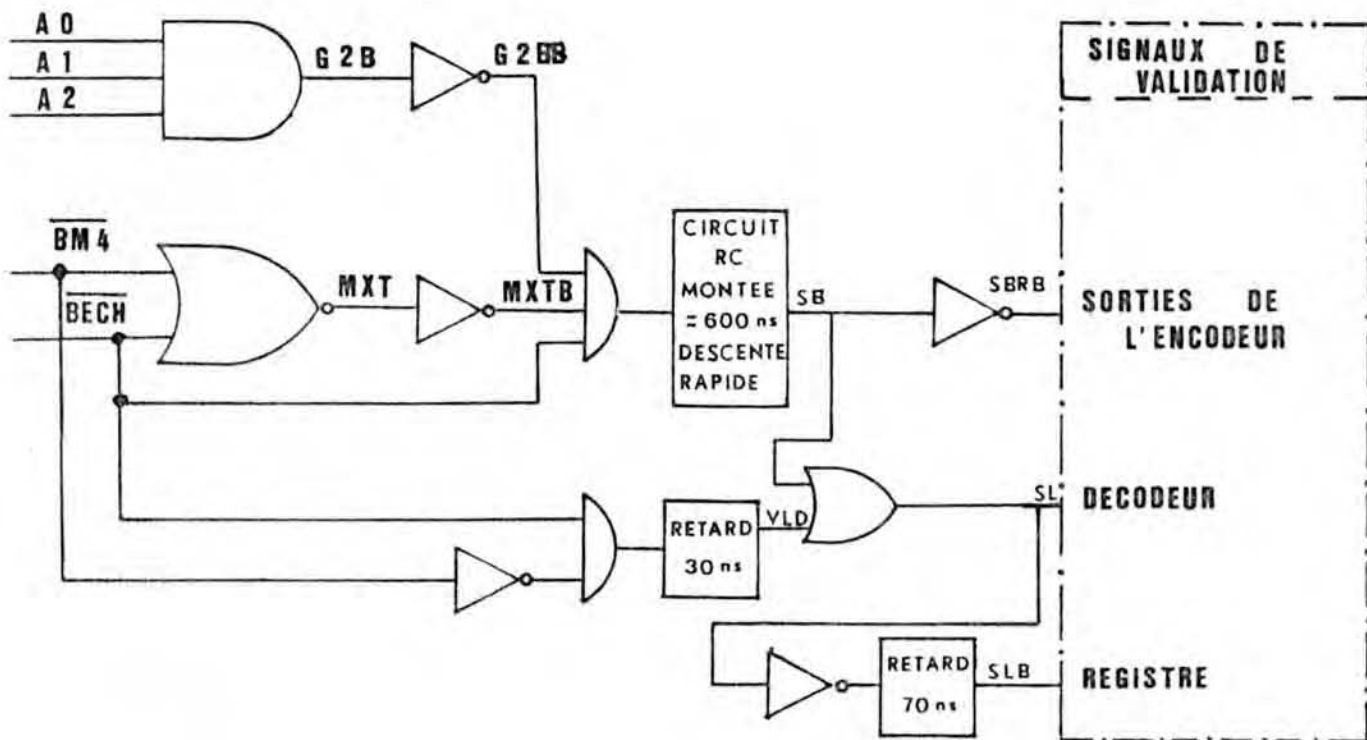


Figure 23 - Schéma logique pour l'obtention
des signaux de validation

Décodage du verdict

Le décodeur aura en entrée le verdict calculé par les ABC 90 (signal VLD actif) ou par l'encodeur (signal SBRB actif). Donc le signal de validation du décodeur résulte simplement du OU de ces deux signaux (SL) montrés dans la figure 23.

Enregistrement du verdict

De même que pour les considérations faites au § 3.3.1. Chap. 2, on doit enregistrer les sorties du décodeur pour avoir un fonctionnement MULTIBUS cohérent. Le signal d'enregistrement correspond à la négation de SL retardé de 70 ns (signal CLOCK du registre).

Initialisation

Voir § 3.3.1. Chap. 2.

Avantages de cette architecture

- . Possibilité de configurer les unités de traitement en priorité fixe ou tournante ;
- . modularité et disponibilité, on aura plusieurs arbitres travaillant en parallèle sur des cartes différentes ;
- . supervision de l'arbitrage (mécanismes de contrôle et de récupération des erreurs).

Inconvénients

- . Architecture très chère en matériel (bien plus importante que pour l'architecture centralisée),
 - . les arbitrages (mesurés entre BA_i et BPRN) sont plus lents : environ 200 ns si l'arbitrage est délivré par un arbitre, environ 700 ns si l'arbitrage est délivré par l'encodeur (dû à l'exclusivité entre les deux types d'arbitrage : signal SB).
- Toutefois, dès que l'arbitrage est caché, cet inconvénient a son importance diminuée.

Remarques :

On a aussi une architecture hybride qui consisterait à faire des arbitrages décentralisés et centralisés, laquelle est toutefois très chère en matériel et difficile à mettre au point.

En fait, cette architecture consiste à grouper les deux propositions précédentes, car on dispose :

- . cartes standard SBC contenant le circuit 8289,
- . cartes spéciales MULTIBUS contenant l'ABC 90 : arbitrage décentralisé,
- . carte centralisée d'arbitrage, contenant autant d'ABC 90 qu'il y a de cartes standard SBC. Ces arbitres remplacent le schéma classique encodeur + décodeur de l'arbitrage centralisé MULTIBUS.

4. ETUDE DE COMPATIBILITE UME - ABC 90

4.1. Proposition d'un ensemble de fils cohérents

De même qu'au § 3.1. Chap. 2, l'utilisation de l'ABC 90 dans une architecture décentralisée nécessite la redéfinition de quelques fils du bus avec un autre emploi. C'est l'analyse de ces fils qu'on traite dans cette section.

L'analyse des échanges de la structure UME [GIR 82.1] étant donné que le bus UME n'accepte que des arbitrages visibles, on a été amenés à choisir pour la configuration de demandes de l'arbitre (BA 17-24) une configuration démultiplexée. Cela veut dire qu'on doit prendre les signaux B_{AI} ainsi que d'autres signaux propres au dialogue entre ABC 90 (cas de l'architecture décentralisée) parmi les fils disponibles du bus UME à cet effet. Cette condition n'est pas tellement contraignante pour le UME car ce bus dispose d'un nombre assez important de fils appartenant à la famille 2d (voir § 4.1. Chap. 1).

Toutefois si ce nombre de fils disponibles à définir avec un autre emploi n'est pas suffisant, on peut utiliser des broches sur le connecteur P2 du bus UME (ce connecteur permet l'utilisation de microprocesseurs 32 bits).

Néanmoins de façon analogue au bus ILB_H associé à MULTIBUS, un bus pour les échanges mémoire rapides a été défini pour le système UME : le bus UMH [GRO 84.1]. Ce bus prend les rangées A et C du connecteur 96 broches P2. Malgré la normalisation du connecteur P2, nombreux sont les systèmes qui continuent à l'utiliser dans une application spécifique.

Une proposition d'utilisation cohérente des broches du connecteur UME dans une architecture décentralisée est présentée en annexe 3.

4.2. Proposition pour rendre possible l'utilisation compatible de l'ABC 90 dans une structure DME

Comme on l'a dit au § 3.2. Chap. 2, le DME n'admet que des arbitrages visibles. Dès qu'on veut utiliser la supervision, on est amenés alors à suivre les mêmes propositions que dans le cas du MULTIBUS, schéma logique figure 19.

4.3. Architectures compatibles proposées

Comme on l'a vu au § 4. Chap. 1, vis à vis de la gestion d'accès au bus, le DME présente deux sortes de modules :

- . unité de demande de bus placée sur chaque unité de traitement, dont le rôle est d'afficher les demandes et retransmettre le verdict (voir figure 16),

- . unité de gestion d'accès au bus qui peut être implémentée soit par des composants discrets, soit par le circuit arbitre de bus MC 68452. Cette unité a pour rôle d'effectuer des arbitrages.

De même que pour l'utilisation compatible de l'ABC 90 avec MULTIBUS (§ 3. Chap. 2), on peut proposer deux types d'architectures compatibles ABC 90 - DME : centralisée (§ 4.3.1. Chap. 2) et décentralisée (§ 4.3.2. Chap 2).

4.3.1. Architecture centralisée

Si on utilise l'ABC 90 à la place de l'unité de gestion d'accès au bus placée sur la carte contrôleur du système, l'ABC 90 constituera donc l'organe d'arbitrage centralisé pour la structure DME (voir figure 24). Il n'y aura pas non plus d'unité de traitement (dans le sens SM 90) associée à cet arbitre. Les seules fonctionnalités actives de l'ABC 90 seront :

- . arbitrage,
- . supervision.

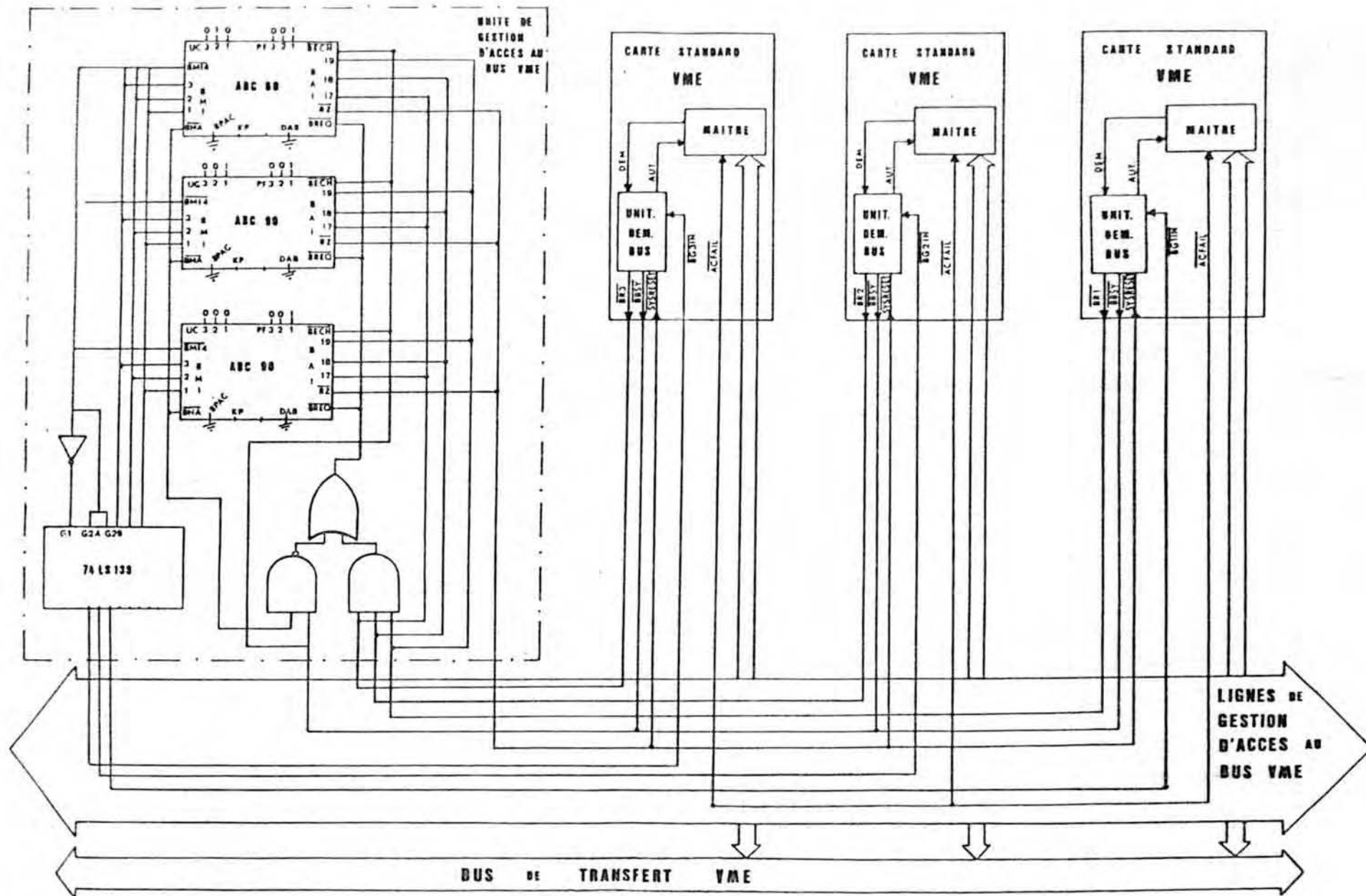


FIG. 24 : Architecture VME : Arb. Centralisé - ABC 90

Nombre d'ABC 90 en fonction du nombre d'UT raccordées au bus

Comme on l'a dit pour l'architecture centralisée MULTIBUS, il y aura autant d'ABC 90 sur la carte contrôleur du système qu'il y a d'unités de traitement connectées au bus.

Décodage du verdict

On aura aussi besoin d'un décodeur à partir du ET CABLE des signaux BM1-3 issus de chaque arbitre à l'instar de la proposition pour le MULTIBUS, car les unités de traitement UME n'acceptent qu'un seul signal d'autorisation d'accès au bus (BG x IN).

La condition logique d'activation du décodeur est aussi semblable à celle prévue par l'architecture centralisée MULTIBUS.

Initialisation

Les mêmes considérations concernant l'arbitre d'adresse 0 faites à propos de l'initialisation pour l'architecture MULTIBUS - ABC 90 sont aussi valables pour les architectures UME - ABC 90, qu'elles soient centralisées ou décentralisées.

Avantages de cette architecture

- . Possibilité d'avoir des niveaux rotatifs de priorité,
- . supervision des arbitrages rendus.

Inconvénients

. Architecture plus chère en matériel (des ABC 90, circuit décodeur, logique supplémentaire) que la solution MC 68452. Toutefois, il faut préciser que cette architecture est moins encombrante que celle pour le MULTIBUS, car dans le cadre UME - ABC 90 on n'a pas besoin ni de la logique permettant des arbitrages cachés, ni de registres ;

. Les arbitrages sont plus lents : environ 220 ns pour l'architecture proposée contre 60 ns pour le circuit MC 68452.

Il faut rappeler que pour la structure UME les arbitrages sont effectués avant les échanges, donc on ralentit le bus.

4.3.2. Architecture décentralisée

On peut aussi proposer une architecture qui réalise des arbitrages décentralisés (voir figure 25). Cette architecture dispose :

1. d'un ABC 90 sur chaque carte maître spéciale à la place de l'unité de demande du bus UME utilisant toutes ses potentialités : affichage, arbitrage, retransmission du verdict et supervision. On aura alors plusieurs ABC 90 travaillant en parallèle, de manière décentralisée ;

De même que pour l'architecture décentralisée MULTIBUS, l'arbitrage décentralisé des arbitres requiert une redéfinition de la fonctionnalité de quelques broches du connecteur UME. En annexe 3.2. on montre une proposition d'utilisation cohérente des broches du connecteur UME des cartes spéciales UME contenant l'ABC 90.

2. des cartes standard UME, ne contenant pas d'ABC 90 ;

3. logique centralisée d'arbitrage réalisant un algorithme fixe de priorités dès qu'une unité de traitement standard UME prend le bus.

Configuration des demandes, validation du verdict et décodage du verdict

Les mêmes considérations faites au § 3.3.2. Chap. 2, sont valables. Toutefois le type d'arbitrage en question pour l'architecture UME - ABC 90 ne peut être que visible.

Avantages de cette architecture

. Liberté totale de configuration de chaque unité de traitement,
. modularité et disponibilité, arbitrages parallèles réalisés par des organes implantés sur des unités de traitement distinctes,
. incorporation de mécanismes de contrôle et de récupération des erreurs dûe à la supervision.

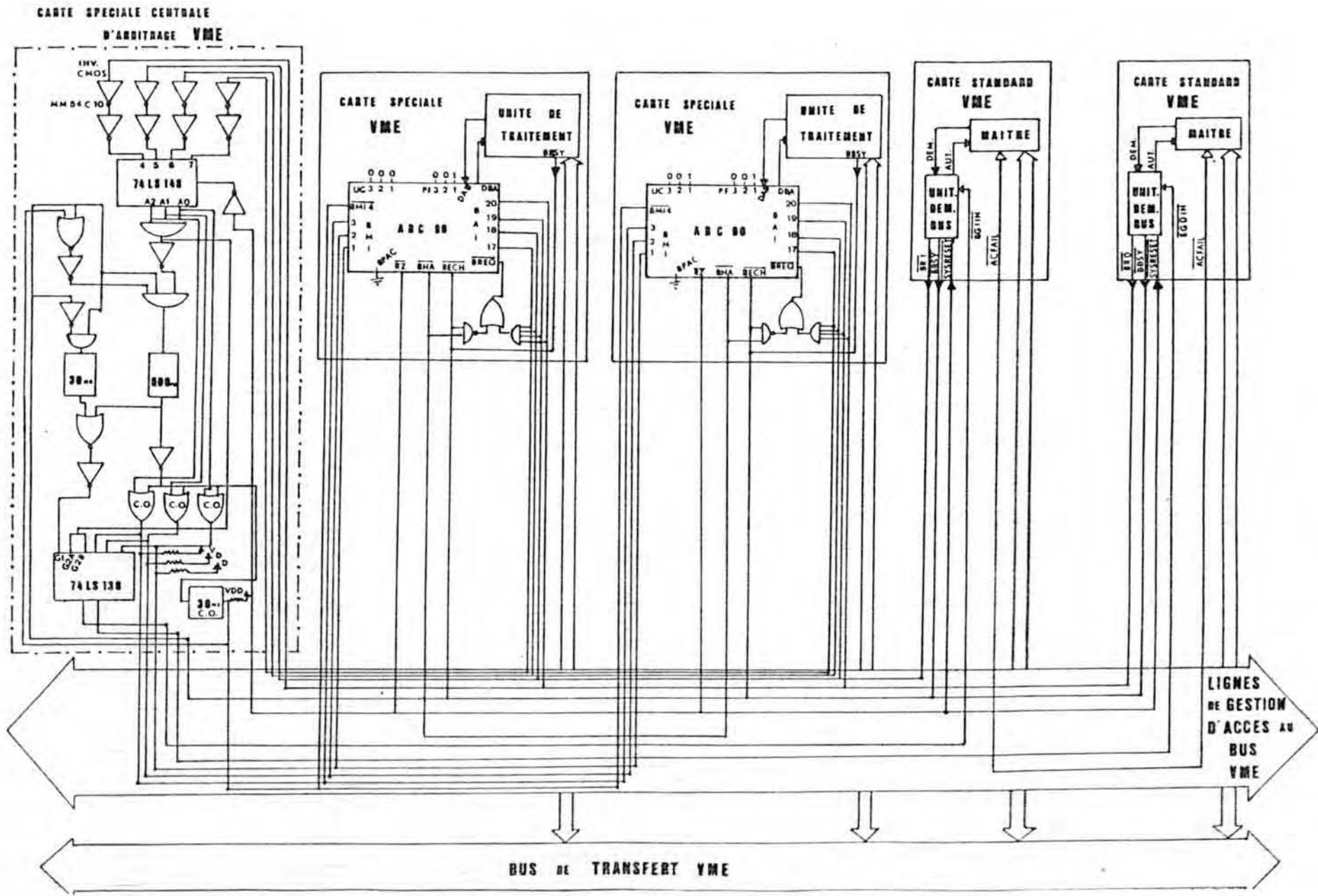


FIG. 25 : Architecture VME : Arb. Décentralisé - ABC 90

Inconvénients

. Architecture très chère en matériel (bien plus importante que pour l'architecture UME centralisée, figure 24),

. ralentissement considérable du bus dès que l'arbitrage est rendu par l'encodeur (de l'ordre de 700 ns, à cause de l'exclusivité entre les deux types d'arbitrages : signal SB).

Dès que l'arbitrage est rendu par un ABC 90 ce ralentissement est moins important (de l'ordre de 200 ns).

Remarque

Mutatis mutandus (voir § 3.3.2. Chap. 2) on peut aussi proposer une architecture hybride pour la structure UME avec des arbitres de bus ABC 90. Cette architecture est intéressante car à l'exception du décodeur, tous les circuits destinés à l'arbitrage sont des ABC 90.

5. VALIDATION PAR SIMULATION DES ARCHITECTURES PROPOSEES

5.1. Approches de validation

L'analyse de compatibilité qu'on vient de faire nous a amenés à proposer des architectures matérielles d'arbitrage incluant l'ABC 90. Ces architectures surtout les décentralisées (§ 3.3.2. et 4.3.2. Chap. 2) s'avèrent complexes en raison de la quantité et de la diversité des circuits employés.

De ce fait, les propositions architecturales faites ne tiennent leur valeur qu'à partir du moment où elles sont validées.

Cette validation peut être réalisée selon deux approches différentes :

- . matérielle
- . logicielle.

La première solution, malgré la portée des résultats qu'on peut obtenir (validation complète des architectures : logique et électrique) se révèle extrêmement coûteuse car on aurait besoin de systèmes multi-microprocesseurs bâtis autour du MULTIBUS et autour du UME, lesquels ne sont pas disponibles au CNET. En outre, la dernière version de l'ABC 90 ne vient d'être échantillonnée que récemment.

De même, la validation matérielle entraîne des temps de développement considérables.

En raison de ces contraintes, on n'a pas retenu pour la validation des architectures proposées dans ce travail, l'approche matérielle.

La seconde solution permet aussi la validation des architectures dans des conditions intéressantes (coût et temps de développement). Cette validation logicielle consiste à démontrer que la logique proposée amène à un comportement cohérent tant pour l'ABC 90 que pour les unités de traitement UME ou MULTIBUS.

Cette validation logique ne prend pas en compte les aspects électriques (tels que charge que l'ABC 90 peut supporter, type de sortie TTL ou drain ouvert).

Comme les outils de validation électrique disponibles au CNET [ASS 84], tels que SPICE ou ELDO [HEN 84] ne se révèlent pas appropriés à la validation électrique des architectures proposées, on a mené l'étude de compatibilité électrique (annexe 4) à partir de spécifications.

5.2. Validation logique

Comme il n'y a pas d'outils de validation de compatibilité entre unités décrites par des chronogrammes, on a utilisé des outils logico-fonctionnels disponibles au CNET.

En fait, malgré la disponibilité d'une description complète de l'ABC 90 sur le simulateur logique EPILOG [EPI 79] (sous forme de portes), le manque de description d'implantation du circuit 8289 (MULTIBUS) ainsi que de l'unité de demande d'accès au bus UME (on ne connaît que leur comportement sous forme de chronogrammes), nous a amenés à utiliser le simulateur fonctionnel FIDEL récemment développé par l'équipe Parallélisme, Communication et Circuits Intégrés du Laboratoire IMAG, en collaboration avec le CNET [HAZ 84].

La simulation fonctionnelle peut être regardée comme une solution aux problèmes de grande taille mémoire et temps CPU importants par rapport à la simulation logique, surtout pour des circuits complexes.

En outre, l'utilisation du simulateur fonctionnel est intéressante en raison de la compatibilité intégrale entre EPILOG et FIDEL (en fait, pour l'instant, FIDEL s'insère dans EPILOG).

Une description de ces outils de CAO employés pour la validation logico-fonctionnelle est donnée en annexe 5.

5.3. Caractéristiques des simulations

Dans le cadre des simulations réalisées on a donc utilisé :

- . description EPILOG de l'ABC 90
- . description FIDEL du circuit 8289 (annexe 6)
- . description FIDEL de l'unité de demande d'accès au bus UME (voir annexe 6)
- . description globale et simulation de chaque architecture sous EPILOG, tenant compte des retards des circuits et du temps d'établissement des signaux sur le bus.

Remarque :

Les conditions d'initialisation et la configuration de ces circuits est donnée en annexe 6.

Les architectures simulées sont celles présentées dans ce chapitre :

figure 20 - architecture centralisée MULTIBUS - ABC 90

figure 21 - architecture décentralisée MULTIBUS - ABC 90

figure 24 - architecture centralisée UME - ABC 90

figure 25 - architecture décentralisée UME - ABC 90.

5.4. Analyse des résultats

En plus de la validation des architectures proposées, le résultat des simulations nous permet d'identifier quelques caractéristiques principales d'arbitrage, telles que le temps d'arbitrage, les algorithmes d'arbitrage que l'architecture peut supporter, nécessité ou non d'utiliser la fonctionnalité de supervision de l'ABC 90 ...).

Les résultats des simulations sont donnés en annexe 6. L'analyse de ces résultats permet de dégager les caractéristiques principales de l'utilisation compatible de l'ABC 90 avec MULTIBUS ou UME, lesquelles sont données dans le tableau 2.

L'analyse comparative de ces caractéristiques permet de mettre en relief (tableau 3) les avantages et les inconvénients de l'utilisation de l'ABC 90 dans les structures MULTIBUS et UME.

TABLEAU 2
CARACTERISTIQUES DE L'UTILISATION COMPATIBLE DE L'ABC 90

Caractéristiques de l'utilisation de l'ABC 90	MULTIBUS	
	Centralisé (fig 20)	décentralisé (fig 21)
1. Temps d'arbitrage	200 ns * (BAi, BREQ et BPRN)	200 ns * (verd. ABC 90) 700 ns (encodeur)
2. Période d'horloge BCLK	400 ns	400 ns
3. Type d'arbitrage	CACHE	CACHE
4. Supervision	NECESSAIRE	NECESSAIRE
5. Algorithme d'arbitrage	LIBRE : fixe rotatif ou mixte	LIBRE : (verd. ABC 90) FIHE (verd. encodeur)
6. Coût en matériel	<u>Carte centrale d'arbitrage</u> contient : <ul style="list-style-type: none"> . autant d'ABC 90 qu'il y a d'UT (cartes standard MULTIBUS) . capacité KP (supervision) . décodeur SN74LS138 . registre SN74LS273 . portes pour validation . arbitrage caché (retard) . portes TTL (ET, OU ..) 	chaque <u>carte spéciale</u> contenant l'ABC 90 contient : <ul style="list-style-type: none"> . un ABC 90 . portes . capacité KP (supervision) La <u>carte centrale d'arbitrage</u> contient : <ul style="list-style-type: none"> . inverseurs CMOS MM54C10 . encodeur SN74LS148 . décodeur SN74LS138 . portes pour validation . arbitrage caché . portes TTL

	UME	
	Centralisé (fig 24)	Décentralisé (fig 25)
1.	200 ns * (BA#i, CBA et BGIN)	200 ns * (ABC 90) 700 ns (encodeur)
2.	-	-
3.	VISIBLE	VISIBLE
4.	OPTIONNELLE	OPTIONNELLE
5.	LIBRE	LIBRE (verd. ABC 90) FIHE (verd. encodeur)
6. Carte centrale d'arbitrage contient :	<ul style="list-style-type: none"> . autant d'ABC 90 qu'il y a d'UT (cartes standard UME) . décodeur SN 74 LS 138 . portes 	Chaque <u>carte spéciale</u> contenant l'ABC 90 contient : <ul style="list-style-type: none"> . un ABC 90 . portes La <u>carte centrale d'arbitrage</u> contient : <ul style="list-style-type: none"> . inverseurs CMOS MM54C10 . encodeur SN74LS148 . décodeur SN74LS138 . portes TTL

* La version actuelle de l'ABC 90 porte à 240 ns l'arbitrage.

**TABEAU 3
AVANTAGES ET INCONVENIENTS
DE L'UTILISATION COMPATIBLE DE L'ABC 90**

Utilisation de l'ABC 90 dans les structures MULTIBUS et UME	Avantages
C E N T R A L I S E E	<p>1. MULTIBUS</p> <ul style="list-style-type: none"> . Possibilité d'avoir des niveaux rotatifs de priorité . Mécanismes de contrôle et de récupération d'erreurs (supervision)
D E C E N T R A L I S E E	<p>2. UME</p> <ul style="list-style-type: none"> . Possibilité d'avoir des niveaux rotatifs de priorité . Mécanismes de contrôle et de récupération d'erreurs (supervision)
D E C E N T R A L I S E E	<p>3. MULTIBUS</p> <ul style="list-style-type: none"> . Possibilité de configurer les UT en priorité fixe ou tournante . Modularité de disponibilité (plusieurs ABC 90 peuvent travailler en //) . Mécanismes de contrôle et de récupération des erreurs
D E C E N T R A L I S E E	<p>4. UME</p> <ul style="list-style-type: none"> . Liberté totale de configuration de chaque UT . Modularité et disponibilité . Mécanismes de contrôle et de récupération des erreurs

	Inconvénients
C E N T R A L I S E E	<p>1. MULTIBUS</p> <ul style="list-style-type: none"> . Architecture chère en matériel . Besoin d'augmenter la période d'horloge BCLK à 400 ns : ralentissement d'arbitrage
D E C E N T R A L I S E E	<p>2. UME</p> <ul style="list-style-type: none"> . Architecture chère en matériel . Arbitrages visibles plus lents
D E C E N T R A L I S E E	<p>3. MULTIBUS</p> <ul style="list-style-type: none"> . Architecture chère en matériel (plus chère que pour l'architecture centralisée MULTIBUS - ABC 90) . Besoin d'augmenter la période d'horloge BCLK à 400 ns : ralentissement d'arbitrage . Besoin de redéfinition de broches du connecteur MULTIBUS avec un autre emploi, ce qui est gênant pour les cartes spéciales contenant l'ABC 90
D E C E N T R A L I S E E	<p>4. UME</p> <ul style="list-style-type: none"> . Architecture chère en matériel (plus chère que pour l'architecture centralisée UME - ABC 90) . Arbitrages visibles plus lents . Ralentissement considérable du bus dès que c'est à l'encodeur de divulguer le verdict . A partir de 4 UT, besoin de redéfinition de broches du connecteur UME avec un autre emploi (problème pour les cartes spéciales contenant l'ABC 90)

6. CONCLUSION

D'après l'analyse qu'on vient de faire de différents arbitres de bus à usage multi-micro-processeur pour les structures SM 90, UME et MULTIBUS, on a démontré que l'arbitre de bus ABC 90 est le plus intéressant pour permettre la supervision des arbitrages rendus, ainsi que l'allocation de niveaux rotatifs de priorité aux différentes unités de traitement.

On a proposé et validé deux architectures de base pour chaque proposition de compatibilité (MULTIBUS - ABC 90 et UME - ABC 90), l'une centralisée, l'autre décentralisée.

Même si les architectures décentralisées permettent une modularité et une sûreté de fonctionnement accrues par rapport aux architectures centralisées (ce qui est intéressant pour les architectures multi-micro-processeurs) elle se révèlent toutes les deux chères du point de vue matériel en raison de la logique supplémentaire implantée en composants discrets.

L'analyse des modifications logiques apportées à l'arbitre de bus ABC 90 nous conduit à envisager de modifier ce circuit en le transformant en circuit multi-protocoles compatible avec les structures UME et MULTIBUS, tout en gardant ses puissantes fonctionnalités d'origine.

De plus, ce nouveau circuit pourra effectuer des arbitrages où l'ABC 90 les réalise avec perte de performances (cas des architectures décentralisées).

Néanmoins, même si ce circuit multi-protocoles gagne en temps d'arbitrage par rapport à l'utilisation de l'ABC 90, les deux approches (ABC 90 et circuit multi-protocoles) sont plus lentes que les techniques d'arbitrage d'origine. De même, les architectures décentralisées (surtout MULTIBUS) imposent une redéfinition de quelques broches des connecteurs standard des cartes contenant soit l'ABC 90 soit le circuit multi-protocoles.

CHAPITRE 3

PROPOSITION D'UN CIRCUIT ARBITRE DE BUS DE COMMUNICATION MULTI-PROTOCOLES :

L'ABC M

1. INTRODUCTION

Une architecture multi-microprocesseurs bâtie autour d'un bus exige des arbitrages pour assurer qu'un seul processeur à un instant donné ait le droit de réaliser des transferts à travers le bus.

Ces arbitrages peuvent être réalisés de deux manières différentes :

- . par logiciel : les arbitrages prennent plusieurs cycles machine [NEL 84],

- . par matériel : un ensemble de composants discrets, voire un circuit intégré spécialisé, sont conçus pour allouer le bus d'une manière plus rapide.

On constate donc que l'intérêt majeur de l'approche matérielle découle de leur rapidité.

Deux sortes de solutions matérielles se présentent :

- . l'une par des composants discrets non spécialisés ; inconvénient ; coût dû au nombre de boîtiers, nombre important de connexions ;

- . l'autre par un circuit intégré spécialisé, lequel en raison de son caractère spécifique peut disposer d'algorithmes d'arbitrage plus complexes (traiter les niveaux rotatifs de priorité, par exemple) + des mécanismes de contrôle d'erreurs et de récupération des erreurs (mécanismes cherchant à éliminer des erreurs) [LAP 82] (si une unité désignée comme la plus prioritaire ne reconnaît pas son droit d'accéder au bus, une fonctionnalité de supervision peut déclencher un nouvel arbitrage, par exemple), + des mécanismes qui permettent le parallélisme entre arbitres, ce qui autorise son utilisation dans un environnement d'arbitrage décentralisé.

Comme on l'a vu au § 2. Chap. 2, seul l'ABC 90 présente la possibilité de traiter les niveaux rotatifs de priorité. En outre, ce circuit présente la supervision et permet des arbitrages décentralisés.

Comme on l'a montré au chapitre 2, ce circuit est compatible avec les structures MULTIBUS et DME, avec un surplus de composants discrets additionnels. Donc, malgré les avantages de son utilisation dans ces structures, cela peut coûter cher, surtout si on veut réaliser des arbitrages décentralisés [BAR 04].

Pour pouvoir répondre à cette contrainte matérielle, nous avons analysé les modifications éventuelles à apporter à l'ABC 90 pour qu'il devienne directement compatible avec les structures MULTIBUS et DME.

C'est la proposition d'un circuit arbitre de bus multiprotocoles appelé ABC M ("ARBITRE DE BUS DE COMMUNICATION MULTIPROTOCOLES") bâti à partir de l'ABC 90 qui fait l'objet de ce chapitre.

2. PROPOSITION D'UN CIRCUIT ARBITRE DE BUS DE COMMUNICATION MULTIPROTOCOLES : ABC M

2.1. Architectures dans lesquelles on peut placer l'ABC M

Ce nouveau circuit permettra d'effectuer des arbitrages pour les structures multi-microprocesseurs MULTIBUS ou DME. De même, il pourra être aussi utilisé dans la structure SM 90 car l'ABC M est entièrement compatible avec l'ABC 90 (§ 3.1. Chap. 3).

Il faut remarquer que lorsqu'un architecte système veut concevoir un système multi-microprocesseurs autour d'un bus préalablement choisi, il aura à sa disposition des cartes disponibles sur le marché, compatibles avec le bus en question, dites cartes standard.

Il sera amené à concevoir d'autres cartes processeur qui rendront son système spécifique ; elles sont appelées cartes spéciales. C'est sur cette deuxième catégorie de cartes qu'on vise à placer l'ABC M.

On peut donc proposer pour l'ABC M les architectures suivantes :

- . dans la structure SM 90, pour des arbitrages décentralisés (voir figure 26),
- . dans la structure UME pour effectuer :
 - des arbitrages centralisés (architecture normale UME, § 4. Chap. 1) (voir figure 27),
 - des arbitrages décentralisés (voir figure 28),
- . dans la structure MULTIBUS, pour effectuer :
 - des arbitrages centralisés (architecture normale MULTIBUS, § 3. Chap. 1) (voir figure 29),
 - des arbitrages décentralisés (voir figure 30),
- . à l'intérieur d'une unité de traitement, en allouant le bus interne à des ressources locales, comme DMA, contrôleur d'interruptions et autres (voir figure 31).

Cette liste n'est pas exhaustive.

On doit remarquer que l'utilisation de l'ABC M dans les structures multi - processeurs MULTIBUS et UME (architectures décentralisées) doit tenir compte de la disponibilité de fils sur les bus pour passer les nouveaux signaux nécessaires à l'ABC M comme on l'a fait avec l'ABC 90 (voir annexe 3).

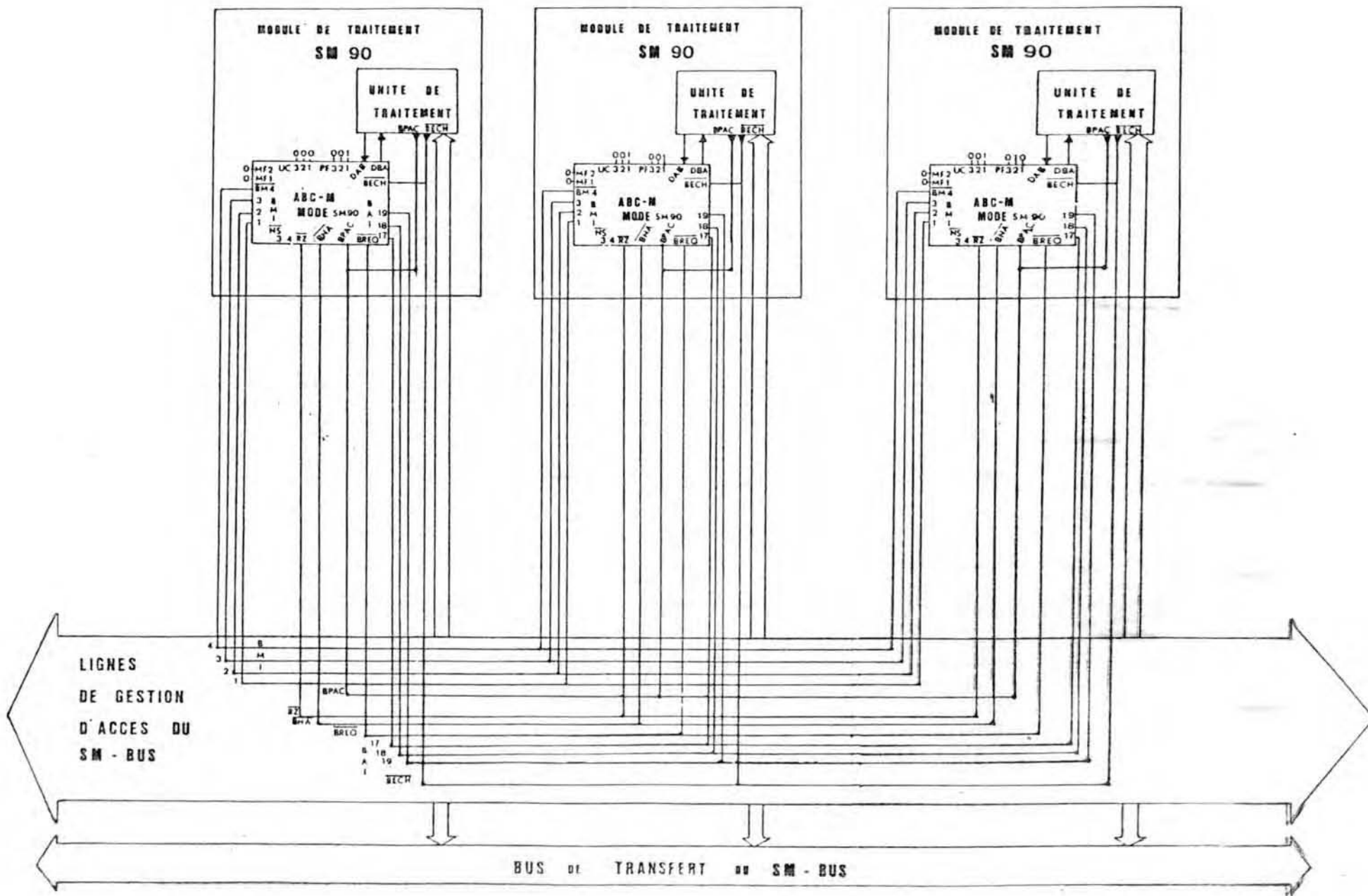


FIG. 26 : Architecture SM 90 : Arb. Décentralisé - ABC-M (MODE SM 90)

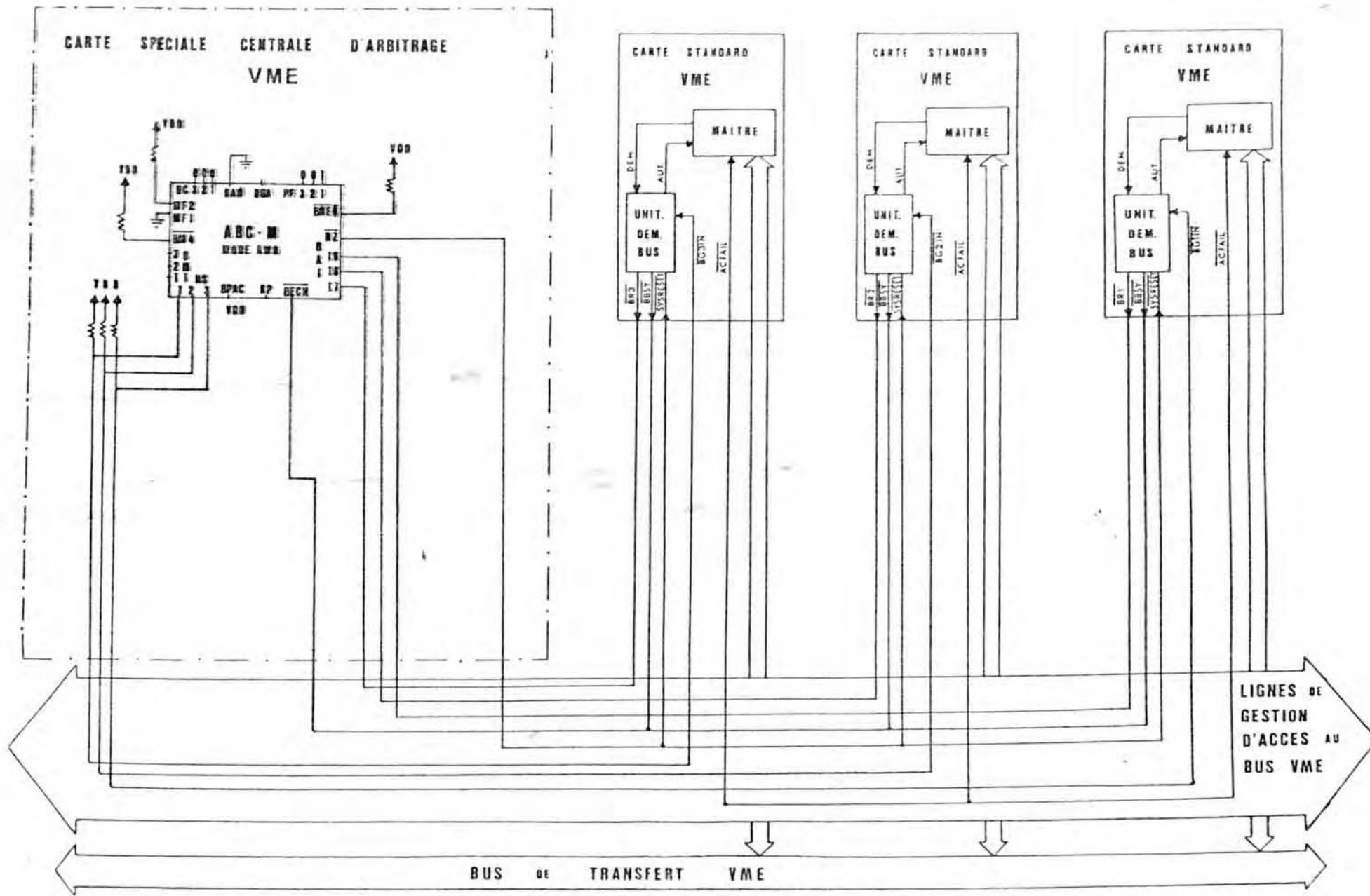


FIG. 27 : Architecture VME : Arb. Centralisé ABC-M (MODE RWD)

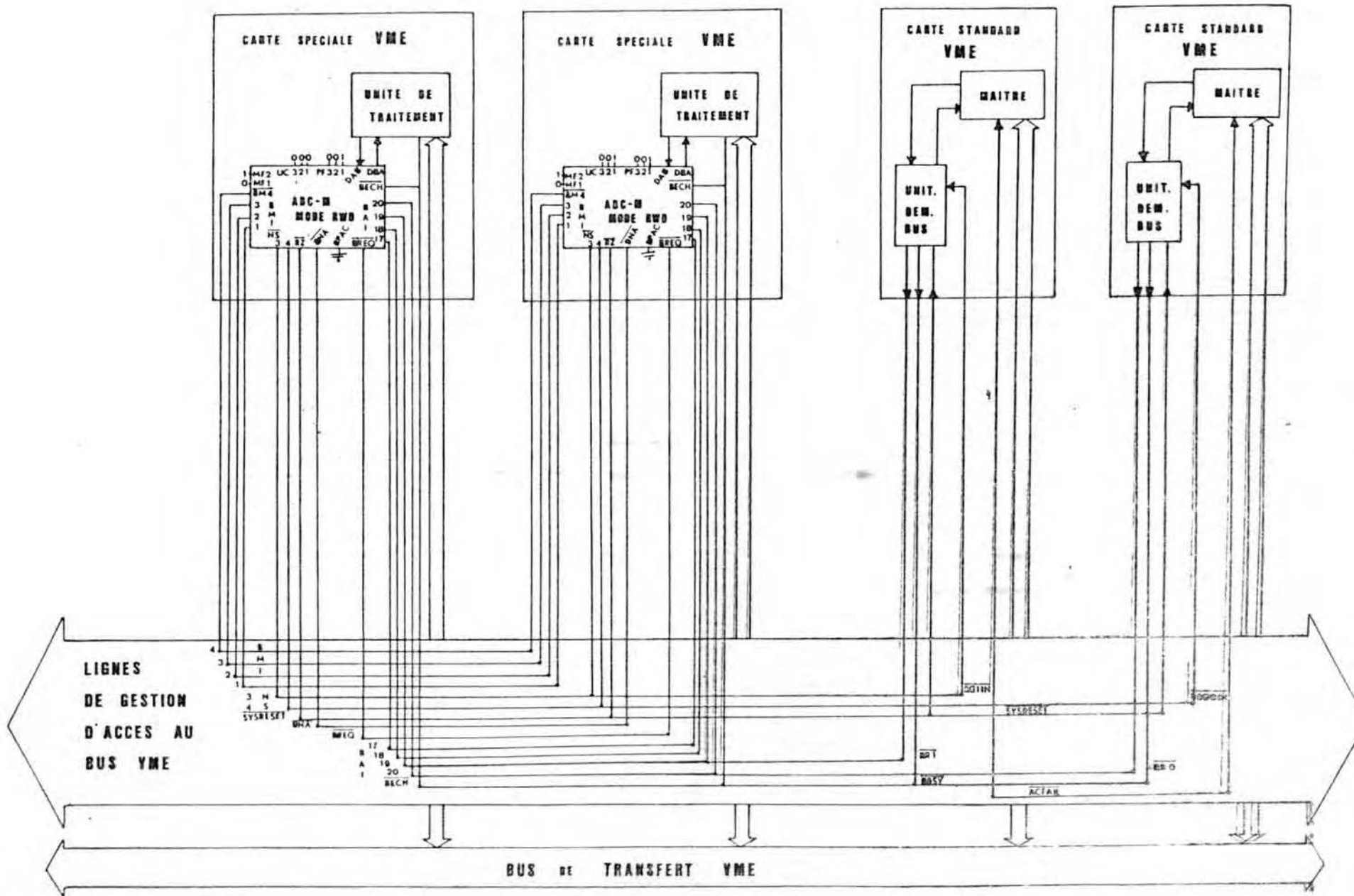


FIG. 28 : Architecture VME : Arb. Décentralisé ABC-M (MODE RWD)

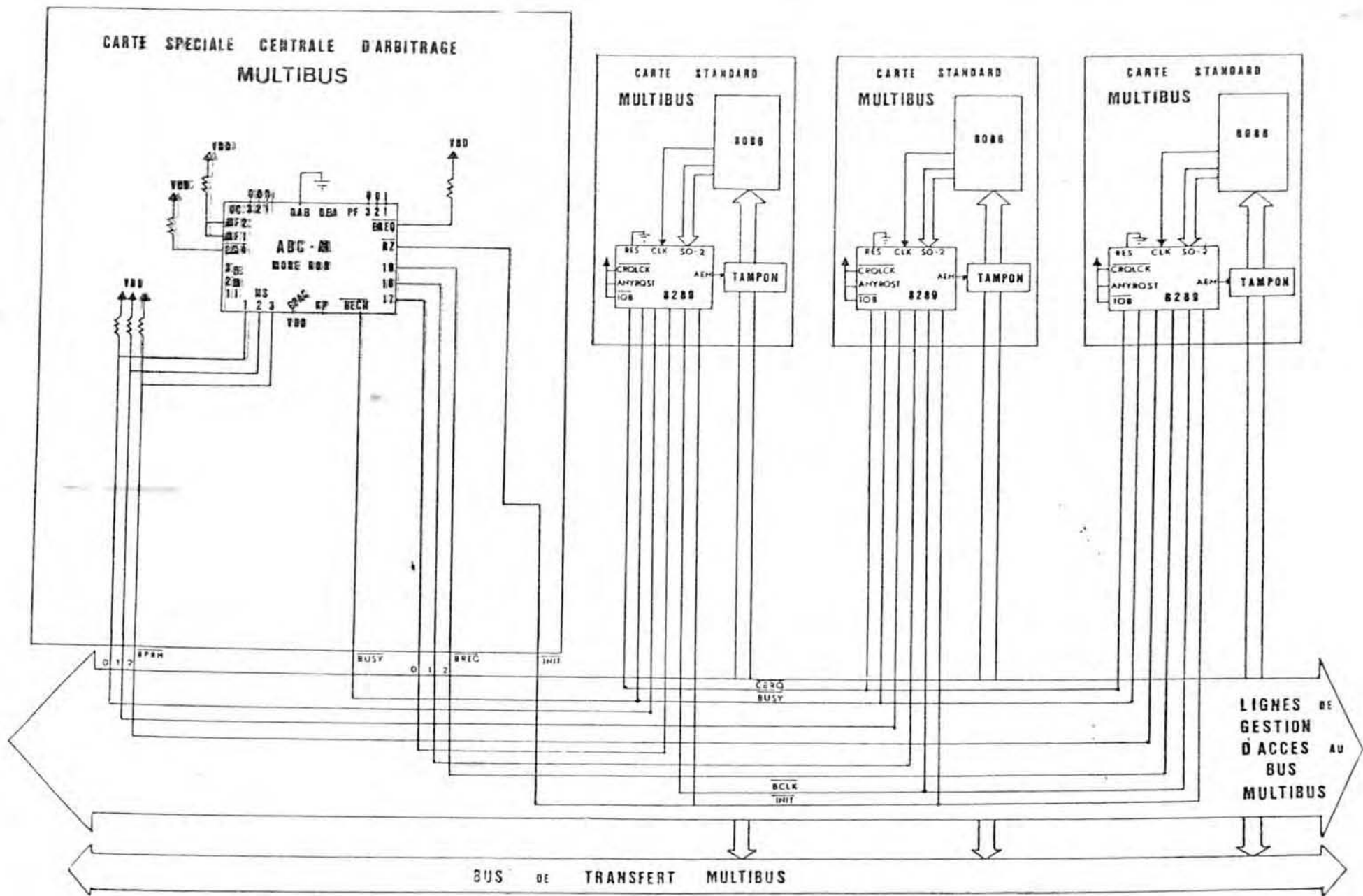


FIG. 29 : Architecture MULTIBUS : Arb. Centralisé - ABC-M (MODE ROR)

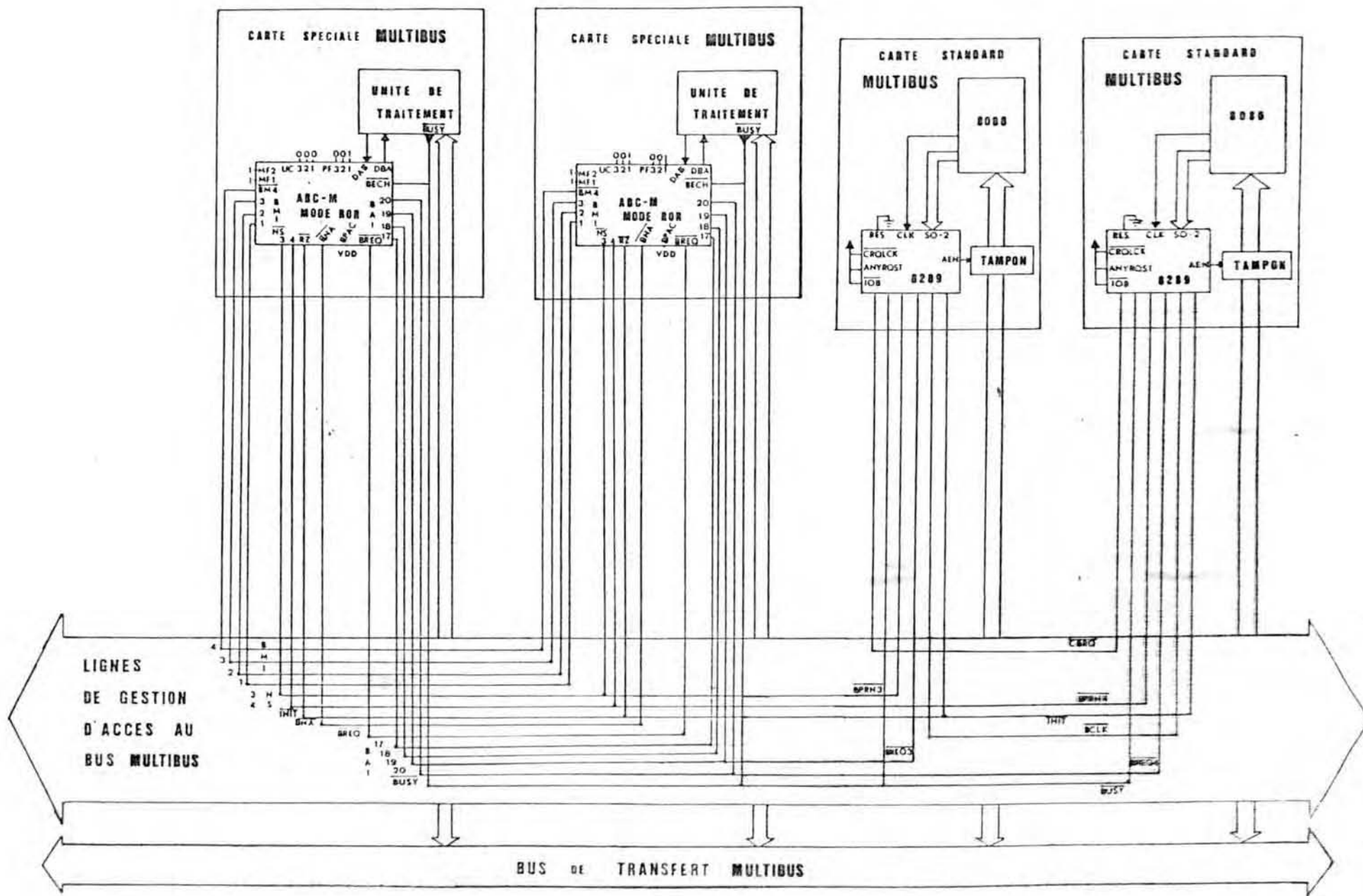


FIG. 30 : Architecture MULTIBUS ; Arb. Décentralisé ABC-M (MODE ROR)

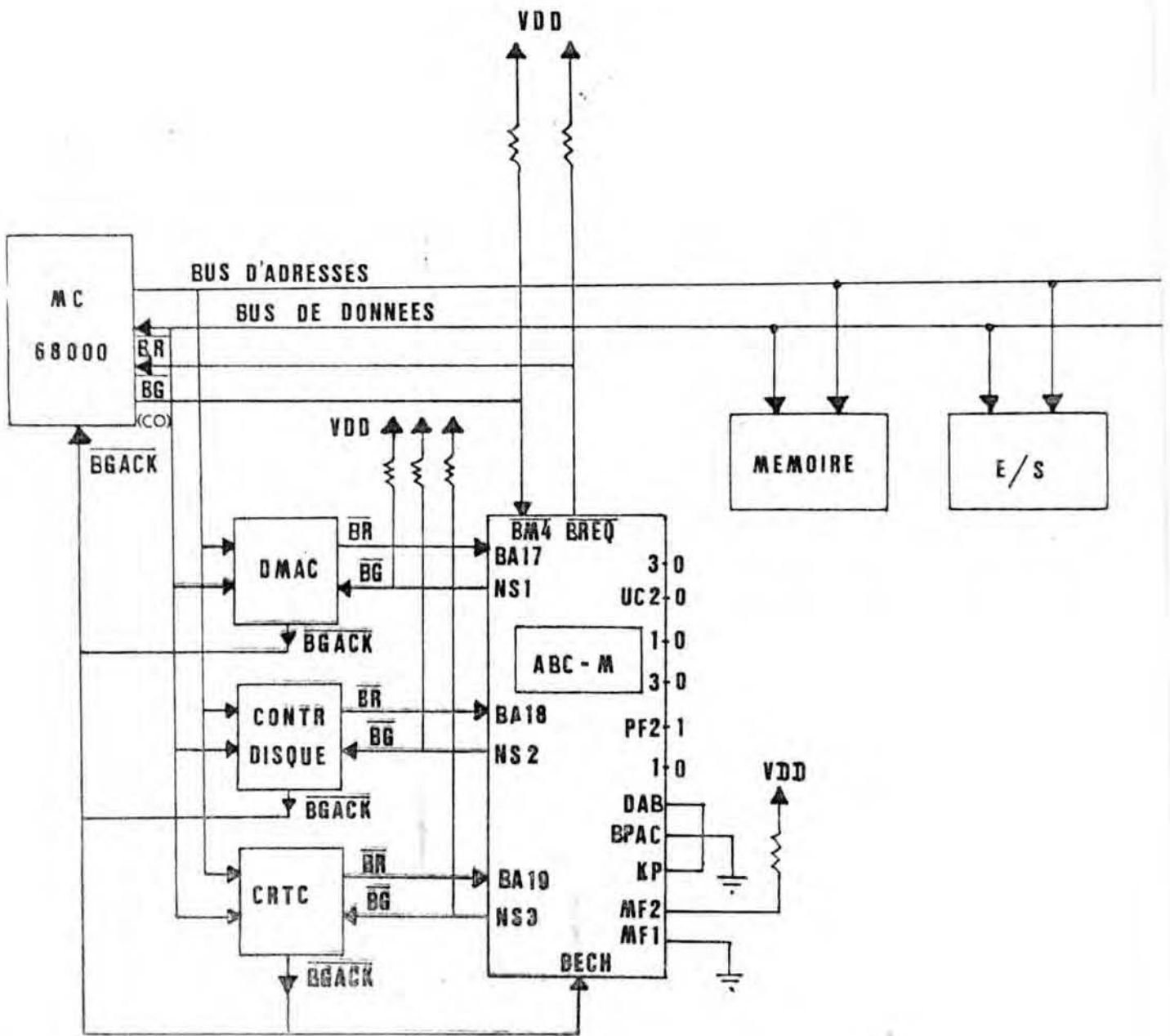


Figure 31 - utilisation de l'ABC M comme arbitre de bus local

2.2. Avantages de l'utilisation de l'ABC M par rapport à l'approche discrète (ABC 90)

Les avantages de l'utilisation de l'ABC M dans les structures UME et MULTIBUS peuvent être résumés en quelques mots :

- . intégration,
- . coût bas du matériel,
- . mécanismes de contrôle et de récupération d'erreurs [LAP 82] (supervision des arbitrages rendus),
- . possibilité de traiter les niveaux rotatifs de priorité ainsi que de réaliser des arbitrages cachés,
- . mécanismes de redondance de service distribuée (dès que plusieurs ABC M sont présents (architecture décentralisée), tous ces arbitres divulguent le verdict simultanément) [LAP 82].

Si on compare l'utilisation de l'ABC M dans les structures MULTIBUS et UME par rapport à celle de l'ABC 90 (§ 3.3. , 4.3. Chap. 2), on constate que la première solution offre en plus des avantages de l'utilisation de l'ABC 90 les avantages suivants :

Avantages :

- . solution moins chère du point de vue matériel ; un seul ABC M remplace l'ABC 90 plus des composants discrets supplémentaires,
- . à la différence de l'ABC 90, l'ABC M, pour divulguer systématiquement le verdict (§ 3. Chap. 3), peut effectuer des arbitrages pour n'importe quel nombre d'unités de traitement (jusqu'à 8, centralisées ou décentralisées), connectées au bus, tandis qu'il faudrait autant d'ABC 90 qu'il y a d'unités de traitement lorsqu'on utilise l'ABC 90. Cela rend envisageable l'utilisation de l'ABC M comme organe centralisé d'arbitrage.

2.3. Contraintes algorithmiques à respecter par l'arbitre ABC M

Comme on l'a montré aux § 3.3.1. et 4.3.1. Chap. 2, les contraintes principales d'utilisation de l'ABC 90 dans les structures MULTIBUS et UME sont :

. le fait qu'il n'y a qu'un seul ABC 90 qui divulgue le verdict (celui associé à l'unité de traitement qui a effectué le dernier échange sur le bus) ; cela mène à avoir autant d'ABC 90 qu'il y a d'unités de traitement raccordées au bus (§ 3.3.1., 4.3.1. Chap. 2) ;

. le fait que le verdict soit délivré de façon codée sur trois bits (BM 1-3), tandis que MULTIBUS et UME n'ont qu'un seul signal correspondant, \overline{BPRN} et \overline{BGxIN} , respectivement ;

. le fait que l'ABC 90 ne déclenche des arbitrages (signal \overline{BREQ}) que si une demande locale d'accès au bus (DAB) a été faite.

2.3.1. Qui divulgue le verdict ?

Pour résoudre le premier problème, on est amenés à modifier l'algorithme implanté dans l'ABC 90. Les solutions que l'on a envisagées sont les suivantes :

a) tous les arbitres présents sur la structure divulguent le verdict simultanément ;

b) . un seul arbitre divulgue le verdict,
. ou c'est le dernier qui a eu le bus (pour les arbitres configurés sur des cartes spéciales),

. ou c'est l'arbitre d'adresse 0 (UC) dès qu'une des conditions suivantes se produit :

+ une unité de traitement ne contenant pas d'arbitre configurée sur carte standard prend le bus ;

+ à l'initialisation.

Pour démarquer les U.T. configurées sur des cartes standard par rapport à celles sur cartes spéciales, on utilisera le paramètre PF (nombre d'ABC 90 configurées en priorité fixe), de façon à ce que toutes les unités de traitement contenant un ABC 90 (cartes spéciales) soient configurées en priorité fixe (de 0 à PF-1).

L'inconvénient est évident : le concepteur système n'a pas entière liberté de choix de configuration de ses U.T. selon une priorité fixe ou tournante.

c) Un seul arbitre divulgue le verdict :

. à l'initialisation c'est à l'arbitre d'adresse 0 de divulguer le verdict (comme pour l'ABC 90),

. en cours de fonctionnement c'est à l'arbitre d'adresse 1 et ainsi de suite, selon un algorithme modulo PF-1.

On aurait ainsi une structure de pile pour la divulgation du verdict.

L'inconvénient est le manque de liberté de configuration comme pour la solution précédente.

2.3.2. Support pour la divulgation du verdict

Pour résoudre le second problème (nécessité d'avoir un verdict décodé 1/8), on est obligé d'introduire une logique supplémentaire de sortie, laquelle peut être semblable pour les trois solutions possibles présentées ci-dessus.

2.3.3. Déclenchement du verdict

Pour résoudre le troisième problème (avoir un arbitrage déclenché pour n'importe quel type de demande (locale ou pas)), on est amenés soit à introduire une logique complexe, soit à introduire un nouvel automate (§ 3.4. Chap. 3): Toutefois la solution à ce problème est commune aux propositions algorithmiques a, b et c.

2.3.4. Choix d'un algorithme

L'analyse de ces nouveaux algorithmes ainsi que les modifications architecturales correspondantes, nous ont amenés à choisir la 1ère solution a, car :

- . elle permet une entière liberté de choix pour l'assignation de priorités fixes ou rotatives pour chaque unité de traitement,

- . au niveau de la conception du nouveau circuit intégré bâti autour de l'ABC 90, elle amène à des modifications moins importantes que dans le cas des autres solutions.

On doit remarquer que la divulgation systématique du verdict entraîne un changement du concept de supervision, tel qu'il est perçu pour l'ABC 90. En fait la fonctionnalité de supervision de l'ABC 90 permet aussi d'éviter quelques cas de panne liés à la divulgation du verdict : si un arbitre donné doit divulguer le verdict et qu'il ne le fait pas, un autre arbitre présent sur la structure réalise l'arbitrage.

Avec la divulgation systématique du verdict par tous les arbitres présents sur la structure (algorithme a), cette "supervision" devient implicite.

Néanmoins, la supervision reste utile pour l'algorithme a : dès qu'un temps surveillé entre une demande d'arbitrage et la prise du bus par l'unité de traitement qui en a bénéficié s'écoule, le signal de supervision devenant actif va déclencher un nouvel arbitrage en faveur d'une autre U.T. (non défaillante).

3. SPECIFICATIONS DE L'ABC M : MODIFICATIONS APORTEES A L'ABC 90

Les modifications principales portées à l'ABC 90 le transforment en un véritable circuit arbitre de bus de communication multi-protocoles (ABC M) sont les suivantes :

1) Augmentation du nombre de broches du circuit, de 28 à 38, avec l'inclusion de :

- . deux broches nommées MF1 et MF2 qui définiront par câblage le mode de fonctionnement du circuit,
- . huit broches nommées-NS qui présentent le verdict décodé (1 ligne NS active parmi 8) pour les structures UME ou MULTIBUS.

Cette modification est détaillée au § 3.1. Chap. 3.

2) Possibilité d'avoir une divulgation permanente ou non du verdict (cas de l'utilisation de l'ABC M dans la structure SM 90) selon le mode de fonctionnement choisi.

Cette modification est détaillée au § 3.2. suivant.

3) Introduction d'une logique supplémentaire liée à la publication du verdict de façon décodée. De plus, cette logique doit tenir compte des différents modes de fonctionnement du circuit.

Cette modification est détaillée au § 3.3.

4) Introduction d'un nouvel automate de déclenchement d'arbitrage.

Cette modification est détaillée au § 3.4.

3.1. Modes de fonctionnement du circuit

L'ABC M présente deux broches MF1, MF2 qui définissent le mode de fonctionnement du circuit (tableau 4) :

Tableau 4
Modes de fonctionnement de l'ABC M

MF2	MF1	MODE SM 90
1	1	L'ABC M fonctionne à l'instar de l'ABC 90 dans une structure SM 90. Un seul ABC M divulgue le verdict à la fois. Ce verdict est présenté de manière codée (SM 90) et décodée (1 parmi 8) signaux non tenus en compte par la SM 90.
MF2	MF1	MODE "RELEASE WHEN DONE" (Dès que le maître a fini son transfert, il libère le bus)
0	1	Chaque ABC M présent sur la structure divulgue le verdict (BM 1-3). Ce verdict est aussi décodé en 1 parmi 8 (broches NS). Dès qu'il y a une prise de bus ($\overline{\text{BECH}}$) et que le signal de demande d'arbitrage (BREQ) est inactif ce verdict (BM 1-3) ainsi que le verdict décodé disparaissent.
ou		
1	0	Ce mode de fonctionnement peut être utilisé tant pour la structure MULTIBUS que pour la structure UME
i.e. MF2≠MF1		
MF2	MF1	MODE "RELEASE ON REQUEST" (Après la fin d'un transfert, le maître qui a eu le bus ne le libère que s'il se produit une nouvelle demande)
0	0	Chaque ABC M présent sur la structure divulgue le verdict (BM 1-3). Ce verdict est aussi décodé en 1 parmi 8 (broches NS). Le verdict BM 1-3 disparaît dès qu'il y a une prise de bus et que le signal de demande d'arbitrage est inactif, tandis que les signaux NS sont valables jusqu'à l'apparition d'un nouvel arbitrage (signal BM-4) de façon à ce qu'il y ait toujours un maître ayant droit d'accès au bus. Ce mode de fonctionnement est le plus fréquent pour la structure MULTIBUS. La structure UME est prévue aussi pour marcher en "RELEASE ON REQUEST".

La définition de ces trois modes de fonctionnement différents (montrée à la figure 32) consiste simplement en un decodeur 2 x 4 pour lequel on n'a besoin que de deux états en sortie :

. NM1 actif bas, pour le mode de fonctionnement SM 90

Remarque : Le câblage extérieur de MF1 et MF2 à 1 n'est pas nécessaire, car on présente en interne cette possibilité.

. NM4 actif bas, pour le mode de fonctionnement RELEASE ON REQUEST

. le mode RELEASE WHEN DONE est sélectionné dès que NM1 et NM4 sont à l'état haut.

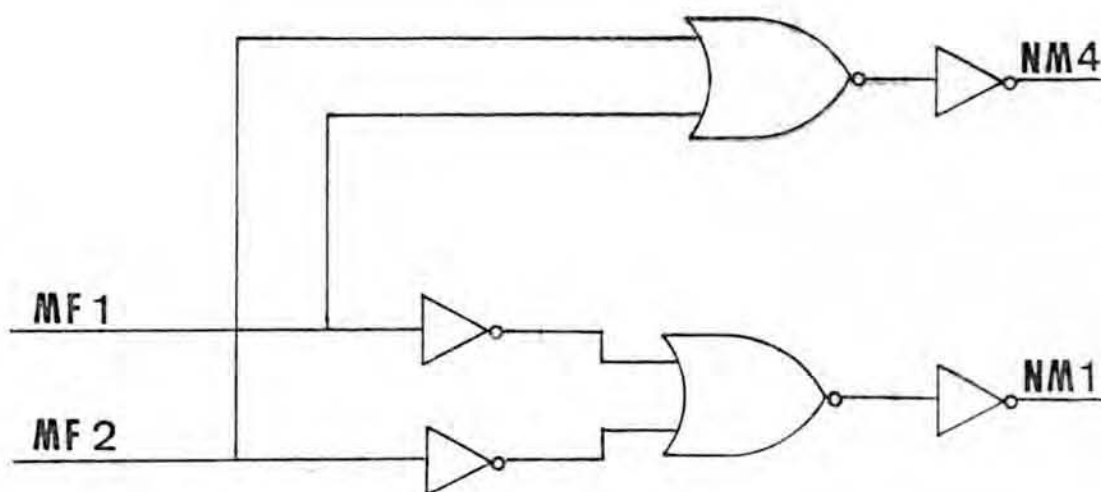


Figure 32 - Logique associée aux modes de fonctionnement

3.2. Divulgaration du verdict

On rappelle que pour tous les modes de fonctionnement, sauf celui de la SM 90, les ABC M divulguent systématiquement le verdict, de façon à ce que le signal NARUC (qui active l'encodeur de divulgation du verdict BM 1-3 en sortie) soit actif en permanence.

Dès que le mode SM 90 est choisi, NARUC copie le signal NARUA de l'ABC 90 (actif dès que le verdict BM 1-3 est identique au câblage de l'adresse de l'arbitre (broches UC)).

U F 9 0 8
BIB. INCA
CHATELAIN

La logique associée est donnée ci-dessous :

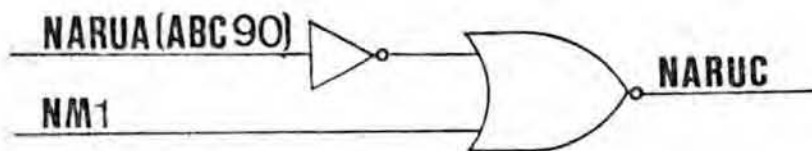


Figure 33 - Logique de divulgation du verdict

3.3. Logique supplémentaire de divulgation du verdict de forme décodée

Le verdict calculé par l'arbitre ABC 90 est porté sur 3 bits : BM 1-3. Toutefois les modules de demande d'accès aux bus MULTIBUS et UME ne reçoivent qu'un signal de demande d'accès au bus accepté : \overline{BGxIN} dans le cas UME et \overline{BPRN} dans le cas MULTIBUS. Le module s'identifie comme le plus prioritaire. De ce fait on est donc amenés à décoder le verdict porté par BM 1-3 en validant un signal parmi huit (signaux NS).

Néanmoins, le problème se complique car les deux modes "RELEASE WHEN DONE" (RWD) et "RELEASE ON REQUEST" (ROR) imposent des timings différents pour les signaux NS :

. mode RWD : dès qu'une unité de traitement se reconnaît comme le plus prioritaire, elle prend le bus et relâche sa demande ; à partir de ce moment là le verdict (BM 1-3), le signal de validation du verdict BM4 et le verdict décodé (NS 1-8) peuvent disparaître ;

. mode ROR : dès qu'une unité de traitement se reconnaît comme la plus prioritaire elle prend le bus, mais ne relâche pas sa demande. Son signal de demande d'accès au bus accepté (NS) ne sera désactivé que si un autre verdict plus prioritaire sera divulgué. Ce mode ROR exige ainsi une mémorisation du verdict décodé.

En raison des différentes possibilités offertes par l'ABC M à l'égard de la divulgation du verdict, le problème de la validation de la logique associée à ces fonctionnalités s'est compliqué. De plus, la validation de la nouvelle logique supplémentaire doit tenir compte de la description de l'ABC 90 réalisée en portes logiques à travers le simulateur logique EPILOG [EPI 79].

Pour la synthèse de cette logique supplémentaire on a effectué la simulation de plusieurs architectures différentes. Un des facteurs qui a contribué à la "difficulté" du problème de synthèse logique a été notre décision de ne pas introduire de nouveaux retards au circuit.

La solution adoptée est montrée dans la figure 34.

PARTIE SPECIFIQUE DE L'ABC - M

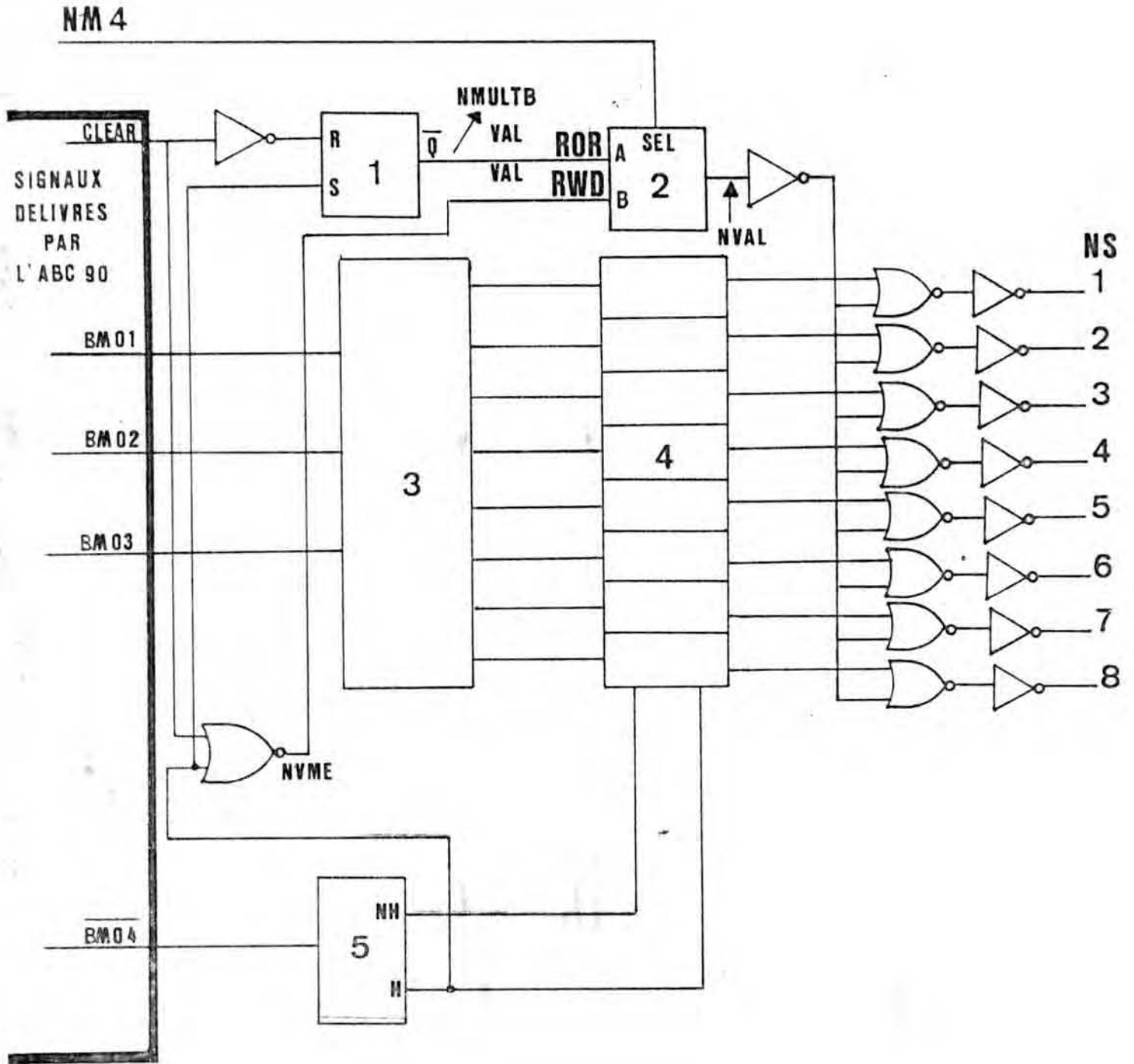


Figure 34 - Logique supplémentaire de divulgation du verdict de forme décodée, spécifique à l'ABC M

U F R G S
BIBLIOTECA
CPD/PGCC

Liste des signaux utilisés dans la figure 34

Délivrés par l'ABC 90		Délivrés par l'ABC M	
BMO 1-3	Verdict calculé à l'intérieur de l'ABC 90	NM4	Sélection d'un mode de fonctionnement (§ 3.1. chap. 4) NM4 Haut : sélection mode RWD NM4 bas : mode ROR
BMO4	Validation du verdict BMO 1-3	NS 1-8	Nouvelles sorties de demandes d'accès au bus acceptées. Un seul signal NSi est actif à la fois
CLEAR	Signal issu de la partie contrôle de l'ABC 90 à la suite de la remise à 0 (RZ) Remarque : on a pris <u>CLEAR</u> à la place de <u>RZ</u> pour des raisons de connectique.		

Liste de cellules employées dans la logique supplémentaire de divulgation du verdict de forme décodée (figure 34)

1. LATCH bâti à partir de bascule R-S ETNON, on utilise la sortie inversée Q
2. Multiplexeur (2 voies vers 1),
 $SORTIE = A. SEL + B. SEL$
à l'entrée A de ce multiplexeur on relie le signal de validation des sorties NS en mode ROR (VAL ROR dans la figure 34)
à l'entrée B de ce multiplexeur on relie le signal de validation des sorties NS en mode RWD (VAL RWD dans la figure 34)
le signal de sélection est NM4 (§ 3.1. Chap. 4).
3. Décodeur 3 x 8 sans signal de validation
4. Registre maître/esclave 8 bits sensible à niveau
Ce registre a besoin de 2 phases d'horloge non recouvrantes.
5. Générateur de phases d'horloge non recouvrantes
Les phases d'horloge sont générées à partir du signal BMO4 délivré par l'ABC 90.

Remarque :

Comme l'ABC M est conçu en technologie NMOS, on a décidé de réaliser notre synthèse à partir des portes OUNON, car elles autorisent des dimensions plus réduites des transistors [MEA 80].

3.4. Automate de déclenchement d'arbitrage

Un autre problème qui entraîne la nécessité de portes externes pour l'utilisation de l'ABC 90 dans les structures UME ou MULTIBUS (fig. 20, 21, 24 et 25), est le fait que ce circuit ne génère un signal de demande d'arbitrage BREQ que si une demande d'accès au bus local (DAB) a été faite.

Néanmoins, dans le cadre de l'ABC M on doit générer ce signal BREQ pour n'importe quel type de demande : locale (DAB) ou extérieure (carte standard UME ou MULTIBUS ne contenant pas d'ABC M).

Pour résoudre ce problème on a été amenés tout d'abord à se fournir un signal de demande de bus général (DBUS), actif haut dès qu'une demande se produit (au minimum un signal BA_i quelconque actif bas).

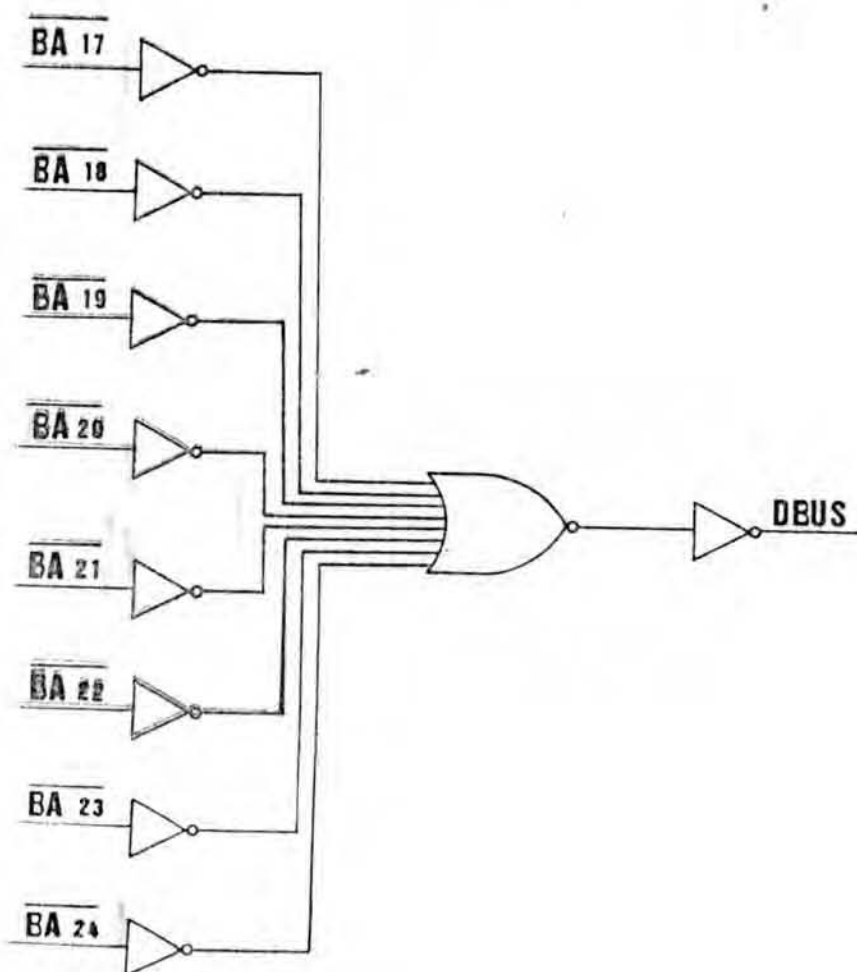


Figure 35 - Logique de demande générale de bus

A partir du signal de demande de bus général (DBUS) et des autres signaux (BM4, BECH, BPAC, RZ, BNA) qui interviennent sur le signal de demande locale d'arbitrage de l'ABC 90 (BREQ de l'ABC 90), on a cherché à produire le signal correspondant de demande d'arbitrage général pour l'ABC M (BREQ de l'ABC M) de deux façons distinctes :

- . par une approche opérative, laquelle a été abandonnée, car elle nécessite l'introduction d'un temps de retard,
- . par l'introduction d'un nouvel automate de déclenchement d'arbitrage. C'est cette solution qui a été retenue pour l'ABC M.

Cet automate, décrit dans un formalisme proche des réseaux de Petri [MOR 78], le GRAFCET [ITI 82], a été conçu de façon à ce qu'il puisse marcher en parallèle avec l'automate d'affichage de demandes (ABC 90) [OLI 83]. En fait, l'automate d'affichage (demande locale DAB) devient un sous-ensemble de l'automate de déclenchement d'arbitrage (demande générale de bus DBUS). La génération de BREQ à partir de ces deux automates est montrée à la figure 36.

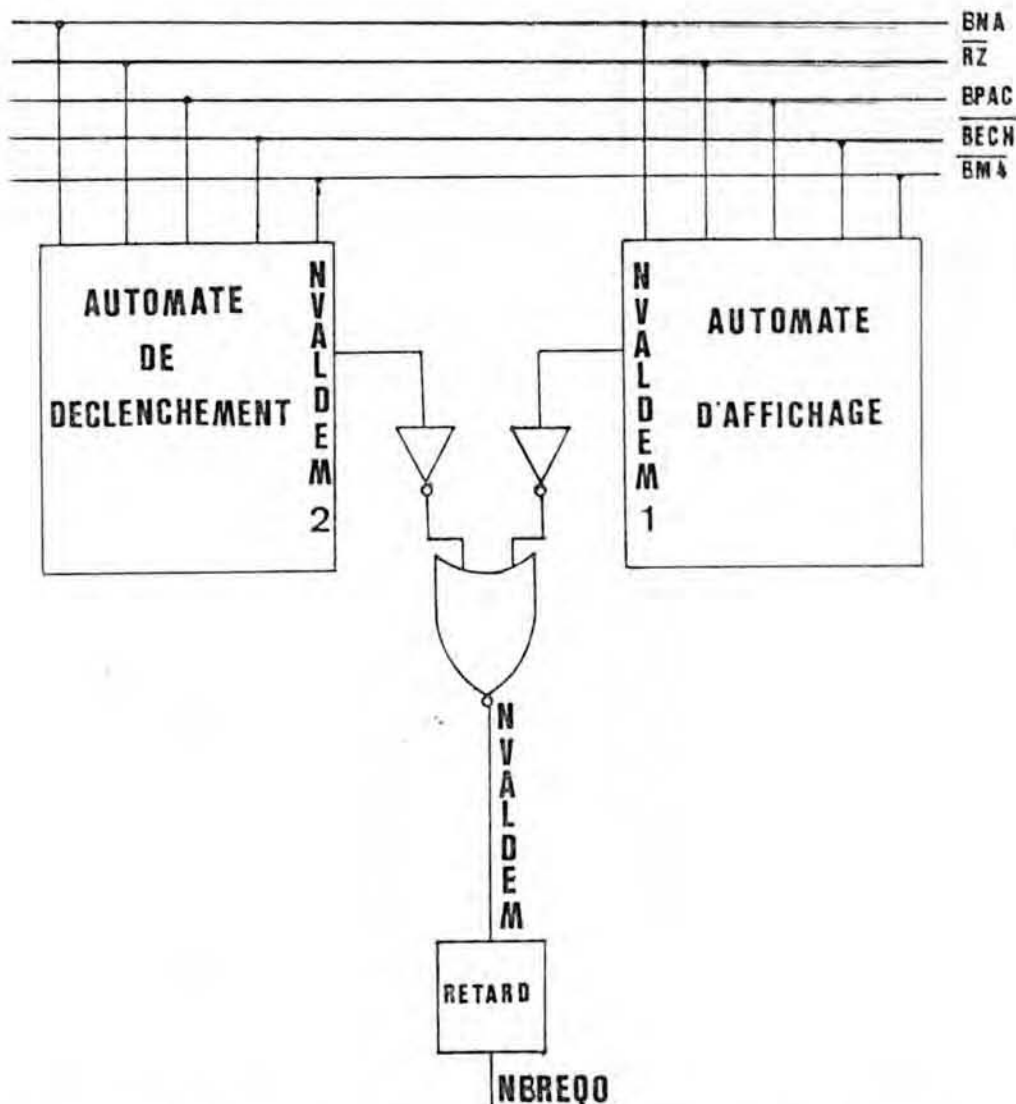


Figure 36 - Génération du signal de demande d'arbitrage BREQ à partir de deux automates de l'ABC M

3.5. Architecture du circuit ABC M

En ce qui concerne l'architecture, le nouveau circuit se différencie surtout de l'ABC 90 (dont l'architecture est montrée à la figure 37), par l'introduction d'une nouvelle partie liée au déclenchement d'arbitrage. En outre, la partie opérative d'arbitrage doit contenir la logique supplémentaire de divulgation de verdict. L'architecture de l'ABC M est montrée à la figure 38.

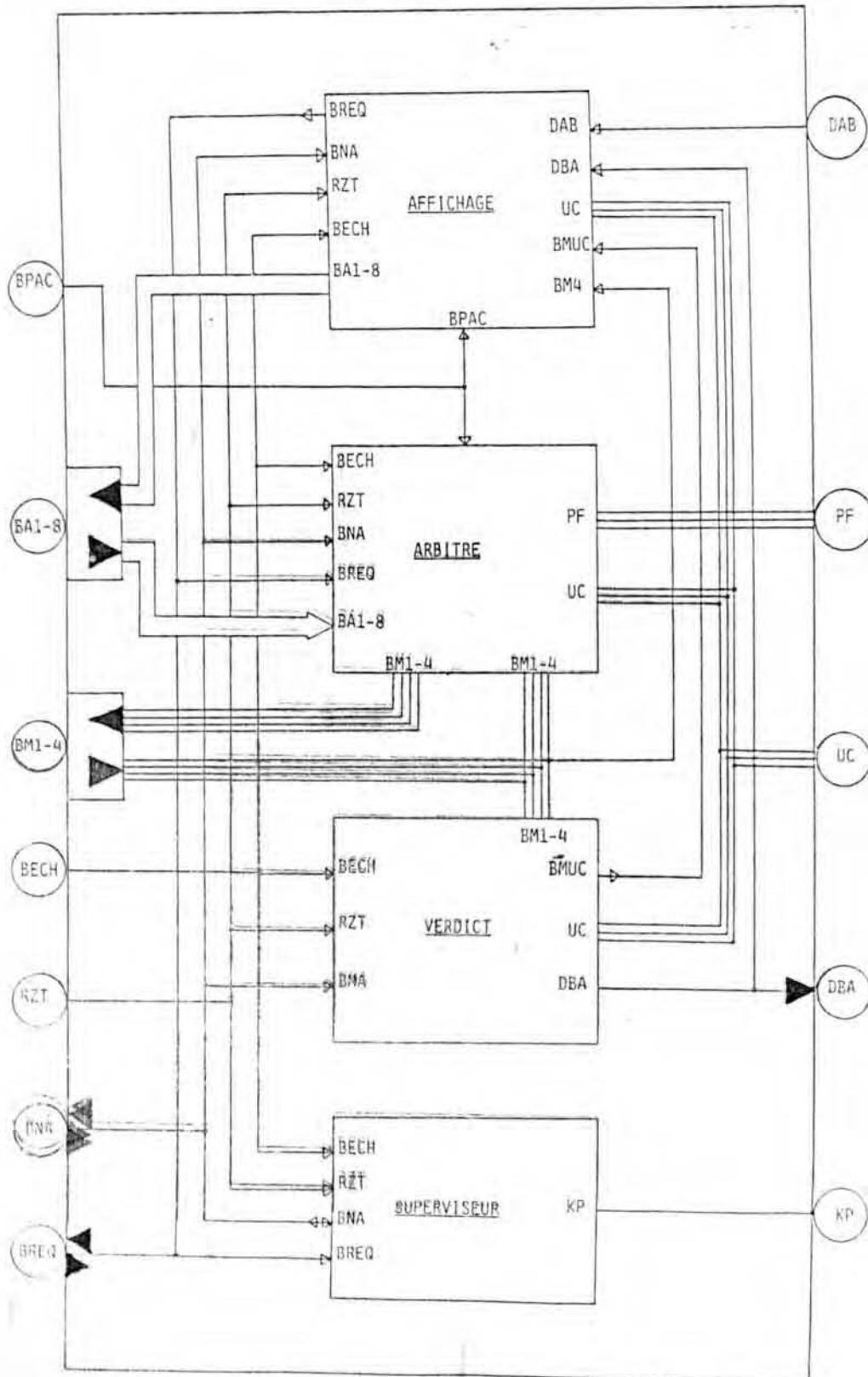


Figure 37 - Architecture du circuit ABC 90

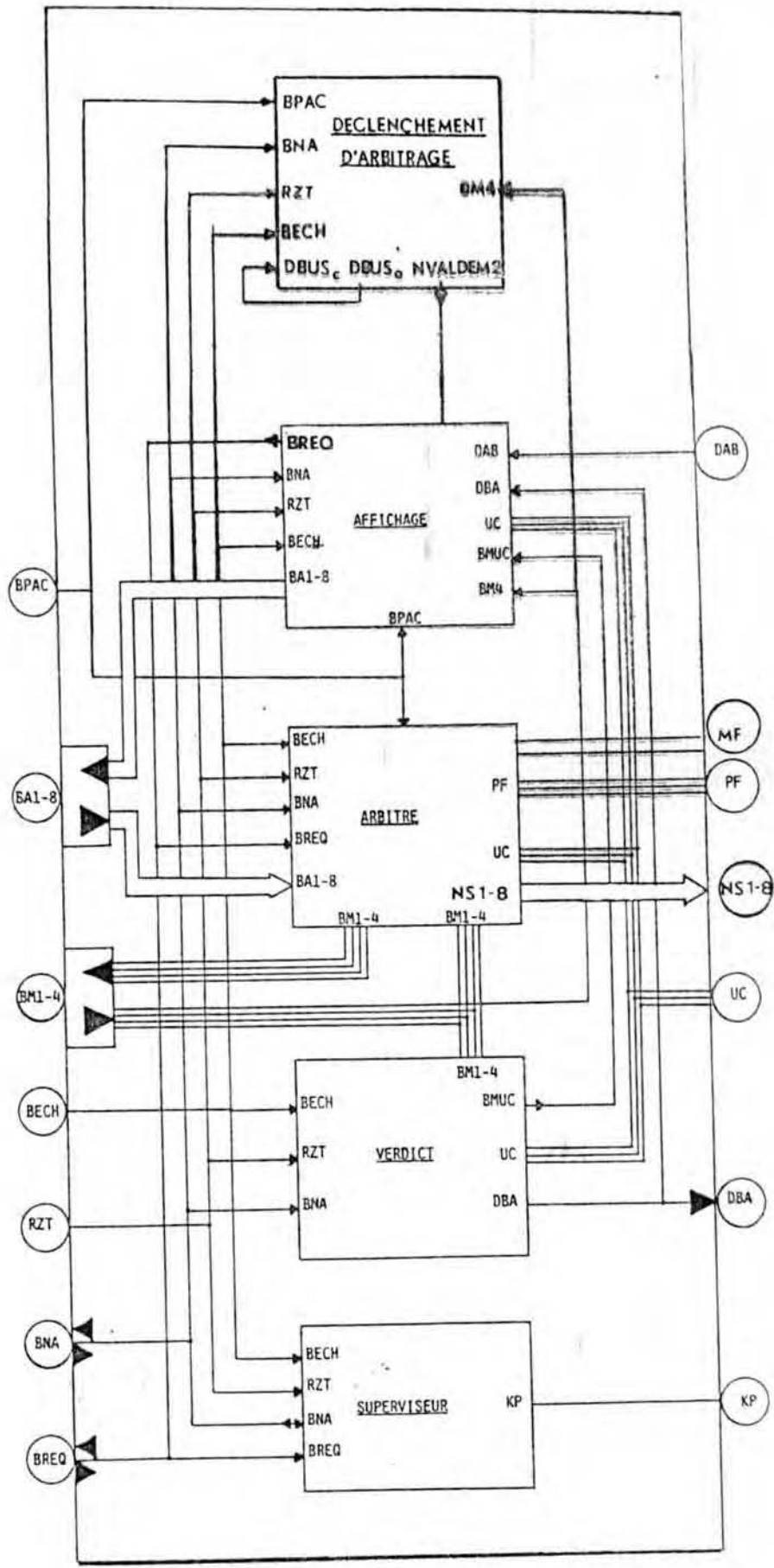


Figure 38 - Architecture du circuit ABC M

4. VALIDATION PAR SIMULATION DU CIRCUIT ABC M

4.1. Outils de CAO employés

La complexité des modifications introduites à l'ABC 90 le transformant en circuit arbitre de bus de communication multi-protocoles exige une validation pour assurer le bon fonctionnement de ce nouveau circuit. De même que pour la validation des architectures discrètes (§ 5. Chap. 2), une validation électrique de l'ABC M n'est pas possible avec les outils de CAO disponibles [ASS 84], car ceux-ci ne permettent la simulation que de petits circuits (contenant peu de transistors).

La compatibilité électrique de l'ABC M sera donc démontrée (en annexe 4) à la lumière de la documentation technique.

L'outil de CAO le plus approprié aux buts de validation qu'on s'est donnés est le simulateur logique EPILOG [EPI 79], car outre ses potentialités (introduction des temps de retard, adaptabilité aux circuits intégrés), on dispose au CNET d'une description complète de l'ABC 90 sur ce simulateur. De ce fait, il ne nous a fallu qu'introduire les modifications logiques apportées à l'ABC 90.

De même que pour la validation des architectures discrètes proposées (§ 5. Chap. 2), on doit aussi utiliser le simulateur fonctionnel FIDEL [HAZ 84] pour pouvoir valider les modes de fonctionnement RWD et ROR de l'ABC M, car dans ces modes de fonctionnement l'ABC 90 se trouve soit dans un contexte MULTIBUS, soit dans un contexte UME.

Remarque :

En raison de la non disponibilité d'un schéma d'implantation logique du circuit 8289 (contexte MULTIBUS) ou de l'unité de demande d'accès au bus (contexte UME), on les décrit sous FIDEL .

4.2. Validation logique

Comme on l'a vu au § 3. Chap. 3, les spécifications de l'ABC M impliquent 4 modifications principales au circuit ABC 90 (§ 3.1., 3.2., 3.3., 3.4. Chap. 3).

On a essayé de valider séparément chacune de ces modifications (quelques unes de ces modifications ne peuvent être validées que globalement). La validation la plus délicate a été celle de la logique supplémentaire de divulgation du verdict de forme décodée (§ 3.3. Chap. 3) en raison du choix fait de ne pas introduire d'autres retards au circuit.

Le nouvel automate de déclenchement d'arbitrage (§ 3.4. Chap. 3) a aussi été validé sous EPILOG. La méthode de synthèse de cet automate sur silicium est donnée en annexe 7.

Malgré l'intérêt de valider séparément chacune des modifications introduites dans l'ABC 90 (affinement de la logique proposée pour chaque modification), seule la validation globale du circuit pourra déceler des problèmes d'interface des nouvelles modifications entre elles, ainsi qu'avec la description de l'ABC 90 existante.

4.3. Caractéristiques des simulations

Comme on l'a vu au § 4.1. Chap. 3, l'ABC M présente trois modes de fonctionnement. On a donc procédé à la validation de chacun de ces modes.

4.3.1. Mode SM 90

Pour le mode SM 90, l'architecture simulée (figure 26) consiste en la structure multi-microprocesseur SM 90 dont la seule différence par rapport à la structure SM 90 courante, est le remplacement de l'ABC 90 par l'ABC M.

Les résultats des simulations (donnés en annexe 7) démontrent que l'ABC M est entièrement compatible avec l'ABC 90.

Il faut remarquer que l'architecture simulée a été entièrement décrite sous EPILOG.

4.3.2. Mode RELEASE WHEN DONE (RWD)

Ce mode de fonctionnement de l'ABC M permet son utilisation indistinctement dans les structures MULTIBUS ou UME.

La validation de ce mode a été faite à travers l'utilisation de l'ABC M dans une structure UME, car c'est ce mode de fonctionnement qui est décrit plus en détails dans la littérature [GIR 82.1].

L'architecture simulée (figure 28) est décentralisée, car c'est elle qui peut déceler tous les problèmes de compatibilité (fonctionnement simultané d'ABC M en parallèle, interface entre ABC M et unités de demande d'accès au bus UME).

Les résultats des simulations sont donnés en annexe 7.

Remarques :

- . l'ABC M a été décrit sous EPILOG,
- . l'unité de demande d'accès au bus UME sur chaque carte UME standard a été décrit sous FIDEL (voir annexes 6 et 7),
- . l'architecture globale (cartes spéciales contenant l'ABC M et cartes standard UME) a été décrite et simulée sous EPILOG.

4.3.3. Mode RELEASE ON REQUEST (ROR)

Ce mode de fonctionnement permet que l'ABC M soit employé tant dans une structure MULTIBUS que dans une structure UME.

Toutefois, la validation de ce mode a été faite en utilisant l'ABC M dans une structure MULTIBUS, car c'est le mode de fonctionnement le plus répandu dans les systèmes MULTIBUS existants [GIR 81] et [SAB 79].

De même que pour la validation du mode RWD, on a utilisé une architecture décentralisée (figure 30) pour la validation du mode ROR.

Les résultats des simulations sont donnés en annexe 7.

Remarques :

- . l'ABC M a été décrit sous EPILOG
- . le circuit 8289 sur chaque standard MULTIBUS a été décrit sous FIDEL (annexes 6 et 7)
- . l'architecture globale (figure 30) a été décrite et simulée sous EPILOG.

4.4. Analyse de synthèse

Les résultats de simulation de l'ABC M dans ces différents modes de fonctionnement (annexe 7) nous permet de dégager les caractéristiques principales (tableau 5) de l'utilisation de l'ABC M dans différentes structures multi-microprocesseurs.

Si on compare les caractéristiques de l'utilisation compatible de l'ABC M (tableau 5) à celles de l'ABC 90 (tableau 2), on note les avantages de la solution intégrée (ABC M) :

. coût matériel réduit

remarque : l'arbitrage caché intégré enlève le besoin d'un retard externe (§ 3.2. Chap. 2) ;

. entière liberté de configuration de priorité de chaque unité de traitement.

On rappelle que les architectures décentralisées utilisant l'ABC 90 présentent une dégradation de performances dès que c'est l'encodeur externe qui divulgue le verdict : retard de l'ordre de 700 ns et verdict basé uniquement sur des niveaux fixes de priorité ;

. temps d'arbitrage constant (de l'ordre de 250 ns) et inférieur en moyenne aux temps d'arbitrage pour les architectures décentralisées discrètes.

Par rapport aux techniques normales d'arbitrage pour chaque structure (circuits encodeur - décodeur : MULTIBUS, circuit MC 68452 : UME et circuit ABC 90 : SM 90), les avantages et inconvénients de l'utilisation de l'ABC M sont montrés au tableau 6.

TABEAU 5
CARACTERISTIQUES DE L'UTILISATION COMPATIBLE
DE L'ABC M

Caractéristiques d'utilisation de l'ABC M	Modes de fonctionnement		
	SM 90 (fig 26)	RWD, Applic. UME (fig 28)	ROR, Applic. MULTIBUS (fig 30)
Temps d'arbitrage	≈ 240 ns (DAB, DBA)	≈ 250 ns (BR, BGIN)	≈ 250 ns (BREQ, BPRN)
Période d'horloge	-	-	400 ns (BCLK)
Type d'arbitrage	Visible ou caché	Visible	Caché
Supervision	Optionnelle	Optionnelle	Nécessaire
Algorithme d'arbitrage	Fixe, rotatif ou mixte	Fixe, rotatif ou mixte	Fixe, rotatif ou mixte
Coût en matériel	un ABC M sur chaque carte module de traitement	un ABC M sur chaque carte spéciale	. un ABC M sur chaque carte spéciale . une capacité KP pour la supervision

TABLERU 6
AVANTAGES ET INCONVENIENTS
DE L'UTILISATION COMPATIBLE DE L'ABC M

Utilisation de l'ABC M dans les structures SM 90, UME, MULTIBUS	Avantages	Inconvénients
MULTIBUS	<ul style="list-style-type: none"> · Solution matérielle pas chère (1 ABC M contre 1 encodeur + 1 décodeur) · Entière liberté de choix de configuration des UT 	<ul style="list-style-type: none"> · Besoin d'augmenter la période d'horloge BCLK à 400 ns : ralentissement des arbitrages
C E N T A R	<ul style="list-style-type: none"> · Introduction de mécanismes de contrôle et récupération d'erreurs (supervision) 	
L U M E	<ul style="list-style-type: none"> · Solution matérielle pas chère (1 ABC M contre 1 arbitre MC 68452) · Possibilité d'avoir des niveaux rotatifs de priorité (y compris mixtes) · Introduction de mécanismes de contrôle et récupération d'erreurs (supervision) 	<ul style="list-style-type: none"> · Arbitrages visibles plus lents (240 ns) que circuit MC 68452 (60 ns)

D S M 90	<ul style="list-style-type: none"> · Du fait que l'ABC M divulgue le verdict sous forme <u>décodée</u> (NS 1-8) cela peut être utilisé pour valider des circuits branchés sur un bus local (§ 2.1. Chap. 3) 	<ul style="list-style-type: none"> · L'ABC M se présente sur un boîtier plus grand que l'ABC 90 (40 broches dont 38 utilisées, contre 28, respectivement)
E C		
E M U L T I B U S	<ul style="list-style-type: none"> · <u>Solution très économique</u> (comparer coût matériel tab. 5 et 2) par rapport à l'architecture discrète · <u>Liberté de configuration des UT</u> (La solution ABC 90 imposait un arbitrage fixe délivré par l'encodeur) · Introduction de mécanismes de contrôle et récupération d'erreurs (plusieurs ABC M divulguent le verdict simultanément + supervision) · Modularité et disponibilité 	<ul style="list-style-type: none"> · Besoin important de redéfinition du connecteur P1 (cartes spéciales MULTIBUS) ou utilisation de broches du connecteur P2 · Besoin d'augmenter la période d'horloge BCLK à 400 ns
I U M E	<ul style="list-style-type: none"> · <u>Solution très économique</u> (comparer coût matériel tab. 5 et 2) par rapport à l'architecture discrète · <u>Liberté totale</u> de conf. de chaque UT · Introduction de mécanismes de contrôle et récupération d'erreurs (plusieurs ABC M divulguent le verdict simultanément + supervision) · Modularité et disponibilité 	<ul style="list-style-type: none"> · A partir de 4000 besoin de redéfinition des broches du connecteur P1 (cartes spéciales UME) ou utilisation de broches du connecteur P2 · Arbitrages visibles plus lents (240 ns contre 60 ns (MC 68452)). Toutefois ce temps d'attente est constant (il n'y a pas de retardée / 200 ns) solution ABC 90 UME)

5. IMPLANTATION DE L'ABC M SUR SILICIUM

5.1. Plan de masse de l'ABC M

Une fois le circuit ABC M validé, on est passés à l'implantation sur silicium des modifications proposées. Ce nouveau circuit est conçu dans la même technologie que le circuit ABC 90 (technologie HMOS d'Eurotechnique (NMOS longueur du canal de 3,5 μm).

A partir du plan de masse de l'ABC 90 (figure 39) on a cherché à introduire nos modifications (10 broches en plus, nouvelles parties opératives, nouvel automate) de façon à respecter les contraintes suivantes :

- . minimiser la surface additionnelle ajoutée,
- . réduire les fissures du plan de masse de l'ABC 90 en essayant de diminuer les problèmes inhérents de connectique,
- . placer les nouveaux plots (MF1, MF2, NS 1-8) de façon contigüe, réduisant ainsi les longueurs de connexions (source de retards dans les circuits intégrés [BAR 82]) pour l'implantation de la logique supplémentaire de divulgation du verdict,
- . avoir une distribution homogène des plots sur les quatre côtés du circuit,
- . emplacement VDD et VSS symétriques.

D'après ces considérations, le plan de masse proposé pour l'ABC M est montré à la figure 40.

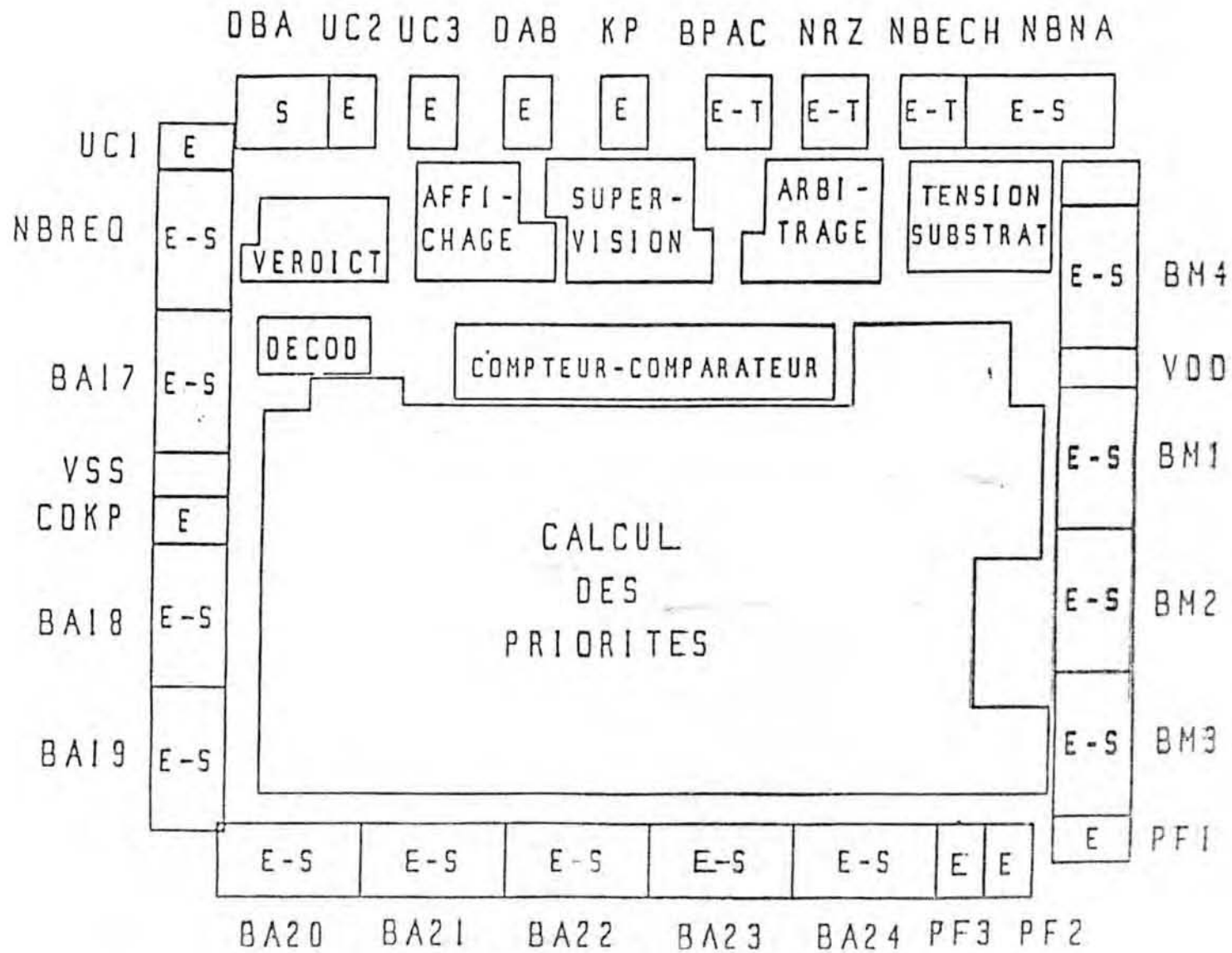


Figure 39 - plan de masse de l'ABC 90

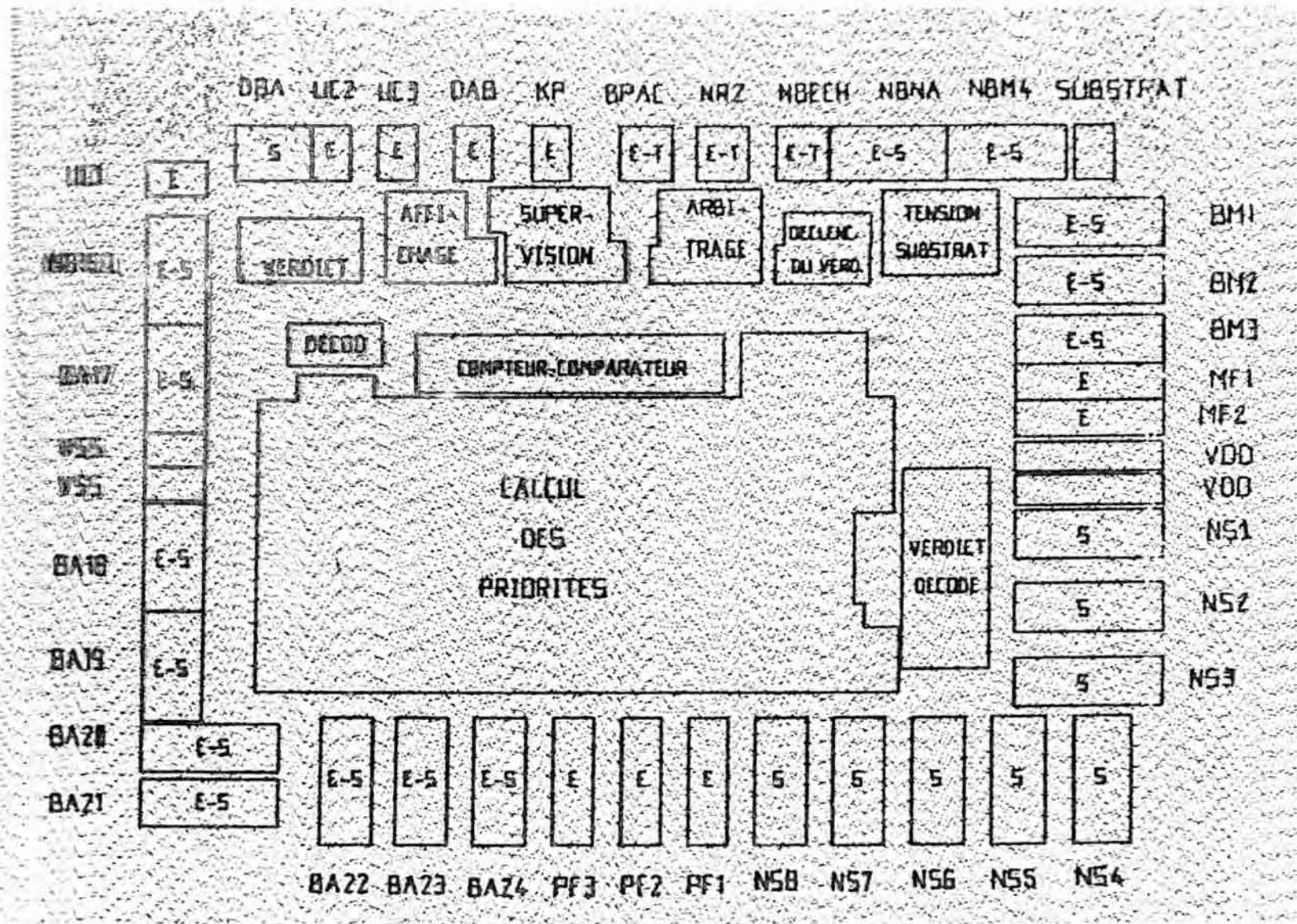


Figure 40 - plan de masse de l'ABC M

5.2. Implantation sur silicium des modifications apportées à l'ABC 90

La synthèse des modifications introduites pour l'implantation sur silicium (§ 3. Chap. 3) peut être réalisée de deux façons différentes [SUZ 81] :

. synthèse de partie opérative :

- cas de la modification liée aux modes de fonctionnement du circuit (§ 3.1. Chap. 3),
- cas de la divulgation systématique du verdict (§ 3.2. Chap. 3),
- cas de la logique supplémentaire de divulgation du verdict de forme décodée (§ 3.3. Chap. 3) ;

. synthèse de partie contrôle :

- cas de l'automate de déclenchement d'arbitrage (§ 3.4. Chap. 3).

5.2.1. Partie opérative

Pour la synthèse de partie opérative, on a utilisé le langage CRASH [ROS 82] qui permet une description symbolique du type STICK [MEA 80]. Cette description symbolique à partir du circuit décrit en transistors, permet une implantation plus rapide qu'une implantation au micron. L'augmentation en surface par rapport à celle-ci est de l'ordre de 10 %. La méthodologie de synthèse de la partie opérative est décrite plus en détails en annexe 5.

5.2.2. Partie contrôle

Une méthode automatique de synthèse de partie contrôle à partir d'une description sous GRAFCET a été développée au CNET [OLI 83-2]. Cette méthode (annexe 5) a permis la compilation rapide sur silicium de l'automate de déclenchement d'arbitrage.

6. CONCLUSION

Après avoir considéré les principales approches d'arbitrage telles que les méthodes logicielle et matérielle, on conclut que cette deuxième méthode donne des temps d'arbitrage plus intéressants.

Dans cette deuxième voie on a proposé un circuit intégré arbitre de bus multiprotocoles : l'ABC M bâti à partir de l'ABC 90 (arbitre de bus de la structure SM 90).

Après avoir analysé plusieurs architectures multiprotocoles que l'ABC M puisse utiliser, on a dégagé les principaux avantages de cette utilisation :

- . incorporation de mécanismes de contrôle et de récupération d'erreurs (mécanismes cherchant à éliminer des erreurs [LAP 82])
- . incorporation de mécanismes de redondance de service [LAP 82] dans la divulgation systématique du verdict
- . potentialités de l'algorithme d'arbitrage implanté (traitement de niveaux fixes, rotatifs et mixtes de priorité)
- . solution économique du point de vue matériel

D'après l'analyse qu'on vient de faire sur différentes algorithmes d'arbitrage possibles pour le circuit arbitre de bus multiprotocoles, on a démontré que la divulgation systématique du verdict par tous les arbitres présents sur la structure, se révèle le meilleur algorithme pour l'ABC M, car il permet une entière liberté de configuration de chaque unité de traitement (fixe ou rotative). L'implantation de cet algorithme implique des modifications du circuit ABC 90 dont l'ABC M est dérivé.

Les modifications apportées consistent surtout dans l'introduction d'une logique supplémentaire liée à la divulgation du verdict (partie opérative) ainsi qu'un nouvel automate lié au déclenchement d'arbitrage (partie contrôle).

Ces modifications étant complexes nous avons été obligés de les valider. On a conclu que la validation la plus adaptée à l'ABC M est la validation logique qui se réalise par des outils logico-fonctionnels (EPILUB & FIBEL).

Une fois l'ABC M validé (interactions : modifications à apporter à l'ABC 90, résultats des simulations) on a décidé de les implanter sur silicium.

CONCLUSIONS

Après avoir étudié les systèmes de communication, on s'est aperçu de l'importance grandissante dans ces systèmes du support de communication, lequel véhicule les informations entre les processeurs du système.

Ceux-ci étant nombreux à pouvoir utiliser ce support, une gestion d'accès au médium est nécessaire.

Après avoir étudié quelques supports de communication parallèle ainsi que la technique d'arbitrage associée, on a conclu que l'arbitre de bus de la structure SM 90 peut être utilisé dans les structures multi-processeurs MULTIBUS et UME de façon avantageuse (fonctionnalités en plus par rapport aux techniques d'arbitrage d'origine pour ces structures).

Néanmoins, cette utilisation compatible de l'ABC 90 s'est révélée coûteuse du point de vue matériel, ce qui nous a amenés à proposer un circuit arbitre de bus multiprotocoles (ABC M).

Grâce à ses différents modes de fonctionnement possibles, ce circuit peut être utilisé dans les architectures multiprocesseurs SM 90, MULTIBUS et UME.

La conception du circuit ABC M s'est bornée essentiellement à modifier le circuit arbitre de bus ABC 90, conçu au CNET de Grenoble pour la structure SM 90.

Les modifications introduites ont été validées par le simulateur logique EPILOG.

Pour l'implantation de ces modifications sur silicium, on a utilisé des outils de CAO appartenant à la base de données de conception de VLSI, disponible au CNET Grenoble, appelée CASSIOPEE.

Pour la validation globale des architectures multiprocesseurs dont le fonctionnement est dans la plupart des cas décrit en termes de chronogrammes, on a choisi de les simuler. On a donc utilisé le simulateur logico-fonctionnel FIDEL. Soulignons en passant l'absence ou l'inexistence d'outils permettant de valider de telles architectures autrement que par simulation.

La validation logique des architectures discrètes (utilisation de l'ABC 90 et logique discrète supplémentaire) et architectures intégrées (utilisation de l'ABC M), nous a permis de conclure que la deuxième solution est plus intéressante du point de vue matériel et architectural car l'algorithme implanté dans l'ABC M permettant la divulgation systématique du verdict, rend son utilisation envisageable dans les systèmes cherchant à avoir une meilleure sûreté de fonctionnement (introduction de mécanismes de contrôle et de récupération d'erreurs ainsi que de la redondance de service).

La validation des architectures intégrées nous a permis d'envisager la conception du circuit intégré ABC M en technologie HMOS. La cuisson du circuit doit être faite au CNET Grenoble.

En raison du caractère multiprotocoles de l'ABC M, ce circuit se présentera comme une solution très envisageable, une fois disponible, en tant qu'organe d'arbitrage de plusieurs systèmes multi-microprocesseurs, comme SM 90, MULTIBUS et DME, dont l'importance ne cesse de croître.

BIBLIOGRAPHIE

- [ASS 84] ASSAEL, J.; SENN, P.; TAWFIK, M.
La chaîne de conception analogique de CASSIOPEE
Séminaire sur les circuits intégrés analogiques - GCIA - Paris
- Octobre 84
- [BAI 84] BAILEY, C.
Hardware, software trends expand multiuser system
performance. Mini-Micro Systems - Juin 84
- [BAR 82] BARONE, D.A.B.
Choix d'un siliciure pour la fabrication de circuits intégrés
MOS
Mémoire pour l'obtention du Diplôme d'Etudes Approfondies
(DEA) - ENSERG - Grenoble - Juillet 82
- [BAR 83-1] BARONE, D.A.B.
Etude des bus de communication série
Journées d'études internes au département Architecture des
Micro-Systèmes - CNET CNS - Grenoble - Juin 82
- [BAR 83-2] BARONE, D.A.B.
Etude des bus de communication parallèle
Journées d'études internes au département Architecture des
Micro-Systèmes - CNET CNS - Grenoble - Juin 83
- [BAR 83-3] BARONE, D.A.B.
Etude de l'interface côté application de différents
gestionnaires de communication de Réseaux Locaux :
analyse et proposition selon la norme IEEE 802
Rapport technique interne à l'équipe Parallélisme,
Communication et Circuits Intégrés - laboratoire Génie
Informatique - Juin 83
- [BAR 84] BARONE, D.A.B.
Etude de compatibilité du circuit arbitre de bus ABC 90 avec
les structures multi-microprocesseurs MULTIBUS et UME
Rapport de recherche - lab. Génie Informatique - à paraître
- [BIE 82] BIEWER, M.
Le bus STD : conception modulaire pour les micros 8 bits
Minis et Micros n° 167 - Juin 82
- [CAM 84] CAMPO, M.
The evolution of the STD bus
Electronic Engineering - Mai 84

- [CAT 82] **CATIER, E**
Les réseaux d'ordinateurs
Electronique Industrielle - n° 39 - 1982
- [COR 81] **CORNAFION**
Systèmes Informatiques Distribués
Ed. Dunod Informatique - Paris 1981
- [CIN 81] **CINARE, F.**
Instrumentation intelligente décentralisée : principes et
emploi du GPIB
Minis et Micros - n° 145
- [COU 79] **COURVOISIER, M.**
A component for multimicrocomputer structure : a
programmable arbiter
Proceedings of Compcon Fall - Septembre 79
- [DAN 83] **DANG, M.; MAZARE, G.; MICHEL, M.**
Local Area Networks for Distributed Process Control Systems
Synchronization, Control and Communication in Distributed
Computing Systems - Academic Press - Novembre 1983
- [DAL 84] **DALRYMPLE, R.**
Interface buses dominate single-board computer market
Mini-Micro Systems - Juin 84
- [DEA 84] **DEASTON, J.**
Structure des bus dans l'architecture MULTIBUS 2
Electronique Industrielle - n° 63 - Janvier 84
- [EPI 79] **EPILOG**
Manuel de référence - Janvier 79
- [FIN 81] **FINGER, U.; LIGNERES, P.**
Structures pluriprocésseurs SM 90 : le bus de communication,
le bus local
Note technique NT/PRA/06E/SIT/367 du CNET - Mars 81
- [FIN 82] **FINGER, U.; MEDIGUE, G.**
Architecture multiprocésseur : l'exemple de la SM 90
Minis et Micros - Octobre 82
- [FIS 84] **FISCHER, W.; ROPER, P.**
Versatile bus suits realtime processor applications
Computer Design - Juin 84

- [GIR 81] GIROD, D.
Caractéristiques et fonctionnement maître-esclave du
MULTIBUS. Minis et Micros - n° 150
- [GIR 82-1], [GIR 82-2], [GIR 82-3], [GIR 82-4] GIROD, D.
Le bus UME : description du fonctionnement dynamique
Minis et Micros - n° 162 - Avril 82
- Panorama des cartes au format MULTIBUS : cartes maîtres et
extension mémoire. Minis et Micros - n° 164 - Mai 82
- Panorama des cartes au format MULTIBUS : E/S numériques et
analogiques. Minis et Micros - n° 166 - Juin 82
- Panorama des cartes au format MULTIBUS : conversion A/N,
N/A et autres cartes. Minis et Micros - n° 168 - Juillet 82
- [GRO 82] GROSS, C.
Les défis des standards des cartes industrielles
Electronique Industrielle - n° 30 - Mars 82
- [GRO 84-1], [GRO 84-2] GROSS, C.
De nouvelles extensions pour le bus UME
Electronique Industrielle - n° 66 - Mars 84
- Plessey et Force Computers s'associent pour les cartes UME
Electronique Industrielle - n° 71 - Mai 84
- [HAL 82] DE HALLEUX, P.
Connecteurs pour circuits imprimés : DIN 41612 et auto
-dénudants
Composants Mécaniques, Electriques et Electroniques - n° 139
Décembre 82
- [HAZ 84] EL HAZEM, H.; MAZARE, G.; POIZE, M.; PUISSOCHET, A.
Functional modeling for logic simulation
Proceedings of the International Conference on Computer
Design (ICCD 84) - New-York - Octobre 84
- [HEN 84] HENNION, B.; SENN, P.
ELDO : Présentation de ce nouveau simulateur électrique fin
suivie d'exemples d'application sur des cellules à capacités
commutées
Séminaire sur des circuits intégrés analogiques - GCIA - Paris
- Octobre 84

- [HYN 83] HYNES, S.
Bus systems and networks
Electronic Engineering - Mars 83
- [INT 81] INTEL MULTIBUS Specifications
Manual Order Number : 9800683 - 03
Copyright 1978, 1979, 1981 Intel Corporation
- [INT 83] INTEL Corporation
MULTIBUS Data Book, 1983
- [INT 84] INTEL Corporation
Microsystem Components Handbook - Vol 2 - 1984
- [ITI 82] ITICSOHN, P.
Représentation des automatismes séquentiels : le GRAFCET
Electronique Industrielle - n° 34 - Mai 1982
- [JOL 83] JOLY, R.
Interconnexion par bus en environnement multiprocesseur :
"Principes de communication et techniques d'arbitrage"
Thèse de Docteur - Ingénieur, ENST, Juin 83
- [LAP 82] LAPRIE, J.C.; COSTES, R.
Dependability : A Unifying Concept for Reliable Computing
Proceedings of the 1982 International Symposium on Fault
Tolerant Computing - 1982
- [LES 78] LESEA, A.; ZAKS, R.
Techniques d'interface aux microprocesseurs
Ed. SYBEH - Europe - Paris - 1978
- [LIL 82] LILEN, H.
UME le nouveau bus standard pour format Europe
Electronique Industrielle - n° 27 - Février 82
- [MEB 80] MEAD, C.; CONWAY, L.
Introduction to VLSI systems
Addison-Wesley Publishing Company - U.S.A. - 1980

- [MET 76] METCALFE, R.M. ; BOGGS, D.R.
Ethernet : Distributed Packet Switching for Local Computer
Networks
Communications of the ACM - vol 19 - n° 7 - Juillet 76
- [MOR 78] MORALLA, M.; PULOU, J.; SIFAKIS, J.
Réseaux de Pétri synchronisés
R.A.I.R.O. Automatique - vol 12 - n° 2 - 1978
- [MOT] MOTOROLA
UME bus : specification manual
M68KUMEB (D1) Microsystems
- [MOT 83] MOTOROLA Semiconductors
MC68452 Advance Information - 1983
- [NEL 84] NELSON, J.C.C.; REFAI, M.K.
Design of a hardware arbiter for multi-microprocessor
systems
Microprocessors ans Microsystems - vol 8 - n° 1 - 1984
- [OLI 82] OLIVE, V.; ROUQUIER, D.
Conception d'un circuit intégré arbitre de bus "ABC 90"
Actes du Congrès de l'AFCEC - Octobre 1982
- [OLI 83-1] OLIVE, V.; ROUQUIER, D.
Arbitre de bus de communication pour la SM 90 : l'ABC 90
Rapport interne - CNET CNS - 1983
- [OLI 83-2] OLIVE, V.; ROUQUIER, D.
A systematic method for the synthesis of control parts
defined by GRAFCET
Proceedings of the IFIP 83 - Paris - 1983
- [PAC 84] PACKER, S. J.; BHASKER, N.
Message passing supports multiple processor design
Computer Design - Juin 1984
- [PAU 83] PAUKER, M.
Trois bus européens : UPL, UME, P096
Acta Electronica - 25 - n°1 - 1983

- [REC 80] RECTOR, R.; GALEXY
The 8086 Book
Osborne - Mc Graw Hill - 1980
- [ROS 82] ROSET, A.
CRASH : une méthode de conception symbolique pour le NMOS
Note technique NT/CNS/CCI/05 - CNET CNS - Juillet 82
- [SAB 79] SABATIER, A.
L'utilisation du MULTIBUS
Electronique et Automatique Industrielle - n° 264 - Mars 79
- [SM9 82] Equipe SM 90
Description générale de la SM 90
Note technique NT/PAA/06E/SML/703 du CNET - Mars 82
- [SUZ 81] SUZIM, A.
Etude des parties opératives à éléments modulaires pour
processeurs monolithiques
Thèse de Docteur - Ingénieur - INPG - Novembre 81
- [WAR 83] WARREN, C.
A mix of standard and proprietary buses marks the latest
microcomputer systems
Electronic Design - Mars 83

ANNEXES

SOMMAIRE DES ANNEXES

ANNEXE 1 - ÉTUDE DE L'INTERFACE CÔTÉ APPLICATION DE DIFFÉRENTS GESTIONNAIRES DE COMMU- NICATION DE RÉSEaux LOCAUX : ANALYSE ET PROPOSITION SELON LA NORME IEEE 802	133
1.1. Analyse de quelques réseaux industriels	135
1.1.1. FACTOR	135
1.1.1.1. Caractéristiques du réseau	135
1.1.1.2. Caractéristiques principales du Coupleur Réseau	137
1.1.2. CARTHAGE	140
1.1.2.1. Caractéristiques du réseau	140
1.1.2.2. Caractéristiques principales du Coupleur Réseau	140
1.1.3. REBUS	145
1.1.3.1. Caractéristiques du réseau	145
1.1.3.2. Coupleur Réseau	148
1.2. Norme IEEE 802	151
1.2.1. Etat de la normalisation	151
1.2.2. Description de la norme IEEE 802	152
1.2.2.1. Couche "LOGICAL LINK CONTROL" (LLC)	153
1.2.2.2. Couche "MEDIUM ACCESS CONTROL" (MAC)	154
1.2.2.3. "Couche PHYSIQUE"	154
1.2.2.4. Primitives de service échangées entre couches	155
1.2.3. Configuration de chacune des cartes coupleurs des réseaux étudiés selon la norme IEEE 802	156
1.2.4. Remarques	159
1.3. Proposition d'une interface en termes de primitives de services spécifiées par la norme IEEE 802 et indépendantes du support	161
 ANNEXE 2 - EXEMPLES D'ARBITRAGE DE BUS	 175
2.1. SM 90	177
2.2. MULTIBUS	179
2.3. VME	180
 ANNEXE 3 - PROPOSITION D'UTILISATION COHERENTE DES BROCHES DU CONNECTEUR MULTIBUS ET DU CONNECTEUR VME	 183
3.1. Proposition d'utilisation cohérente des broches du connecteur MULTIBUS	185
3.1.1. Rappel	185
3.1.2. Correspondance entre les broches du circuit ABC 90 avec celles du connecteur MULTIBUS	188

3.2. Proposition d'utilisation cohérente des broches du connecteur VME	189
3.2.1. Rappel	189
3.2.2. Correspondance entre les broches du circuit ABC 90 avec celles du connecteur VME	192

ANNEXE 4 - COMPATIBILITE ELECTRIQUE 195

4.1. Compatibilité électrique de l'ABC 90	197
4.2. Compatibilité électrique de l'ABC-M	198

ANNEXE 5 - OUTILS DE CAO UTILISES POUR LA VALIDATION ET L'IMPLANTATION DE CIRCUITS INTEGRES 201

5.1. Introduction	203
5.2. Types d'outils employés	203
5.2.1. Outils de validation	203
5.2.1.1. Langages et simulateurs fonctionnels	203
5.2.1.2. Simulateurs logiques	203
5.2.1.3. Simulateurs électriques	204
5.2.2. Outils d'implantation et de vérification	204
5.3. Outils de validation	205
5.3.1. FIDEL	205
5.3.1.1. Description d'un modèle en FIDEL	205
5.3.1.2. Interface FIDEL-EPILOG	206
5.3.2. EPILOG	205
5.3.3. ELDO	207
5.4. Outils d'implantation, vérification et génération des masques	207
5.4.1. Méthode de conception symbolique CRASH	207
5.4.2. Méthode de conception symbolique MDMOS	208
5.4.3. Outil de vérification des règles de dessin DRC	208
5.4.4. Outils de génération de masques et vérification globale	209
5.5. Exemple de conception d'un multiplexeur	210
5.6. Tableaux de récapitulation des outils de CAO utilisés dans le cadre de ce rapport	216

ANNEXE 6 - RESULTAT DES SIMULATION DES ARCHITECTURES DISCRETES UTILISANT L'ABC 90 221

6.1. Introduction	223
6.1.1. Outils de CAO employés	223
6.1.2. Rappel des architectures	225
6.2. Résultats pour l'architecture centralisée MULTIBUS-ABC 90	226

6.2.1.	Caractéristiques de la simulation	226
6.2.2.	Chronogrammes EPILOG de validation de l'architecture centralisée MULTIBUS-ABC 90	228
6.2.2.A.	Chronogrammes correspondant aux stimulis décrits en 6.2.1.	228
6.2.2.B.	Exemple d'intervention de la fonctionnalité de supervision : panne ou inexistence d'un arbitre ABC 90 à l'adresse 0	232
6.2.3.	Remarques	235
6.3.	Résultats pour l'architecture décentralisée MULTIBUS-ABC 90	235
6.3.1.	Caractéristiques de la simulation	235
6.3.2.	Chronogrammes EPILOG de validation de l'architecture décentralisée MULTIBUS-ABC 90	237
6.3.3.	Remarques	242
6.4.	Résultats pour l'architecture centralisée VME-ABC 90	242
6.4.1.	Caractéristiques de la simulation	242
6.4.2.	Chronogrammes EPILOG de validation de l'architecture centralisée VME-ABC 90	244
6.4.2.A.	Chronogrammes correspondant aux stimulis décrits en 6.4.1.	244
6.4.2.B.	Exemple d'intervention de la fonctionnalité de supervision : panne ou inexistence d'un arbitre ABC 90 à l'adresse 0	247
6.4.3.	Remarques	250
6.5.	Résultats pour l'architecture décentralisée VME-ABC 90	250
6.5.1.	Caractéristiques de la simulation	250
6.5.2.	Chronogramme EPILOG de validation de l'architecture décentralisée VME-ABC 90	252
6.5.3.	Remarques	256

ANNEXE 7 - RESULTATS DES SIMULATIONS ET DE L'IMPLANTATION DE L'ABC-M 257

7.1.	Validation des modifications apportées à l'ABC 90 le transformant en circuit multi- protocoles (ABC-M)	259
7.1.1.	Modes de fonctionnement du circuit	250
7.1.2.	Divulgateion systématique du verdict	261
7.1.3.	Logique supplémentaire de divulgation du verdict de forme décodée	262
7.1.4.	Logique associée à l'automate de déclenchement d'arbitrage	266
7.1.4.1.	Validation de la génération d'un signal de demande géné- rale de bus	266
7.1.4.2.	Validation de l'automate de déclenchement d'arbitrage	267
7.1.4.3.	Validation de la logique asso- ciée à la génération du signal de demande d'arbitrage BREQ	270

7.1.5. Remarques	271
7.2. Validation des architectures multimicro- processeurs	271
7.2.1. Mode SM 90	271
7.2.1.1. Description de l'architecture simulée	272
7.2.1.2. Chronogramme EPILOG de validation du mode de fonc- tionnement SM 90 de l'ABC-M	274
7.2.1.3. Remarques	277
7.2.2. Mode "Release when done"	277
7.2.2.1. Description de l'architecture simulée	278
7.2.2.2. Chronogramme EPILOG de validation du mode de fonc- tionnement "Release when done" (RWD) de l'ABC-M (architecture VME)	279
7.2.2.3. Remarques	283
7.2.3. Mode "Release on request"	283
7.2.3.1. Description de l'architecture simulée	284
7.2.3.2. Chronogramme EPILOG de valida- tion du mode de fonctionnement ROR de l'ABC-M (architecture MULTIBUS)	286
7.2.3.3. Remarques	291
7.3. Synthèse sur silicium de modifications apportées à l'ABC 90	292
7.3.1. Méthode de synthèse de l'automate de déclenchement d'arbitrage	292
7.3.1.1. GRAFCET	292
7.3.1.2. Méthode de synthèse sur silicium	294
7.3.1.3. Plan de masse de l'automate	295

ANNEXE 1

ETUDE DE L'INTERFACE COTE APPLICATION DE
DIFFERENTS GESTIONNAIRES DE COMMUNICATION
DE RESEAUX LOCAUX : ANALYSE ET PROPOSITION
SELON NORME IEEE 802

1.1. ANALYSE DE QUELQUES RÉSEAUX LOCAUX INDUSTRIELS

1.1.1. FACTOR

1.1.1.1. CARACTÉRISTIQUES DU RÉSEAU

Factor est un réseau local industriel basé sur la technique CSMA/CD [ITI 84] et a été développé par la Société APSIS de Grenoble en collaboration avec l'IMAG et avec le LAAS (Laboratoire d'Automatique et d'Analyse de Systèmes) à Toulouse. Ce réseau, montré schématiquement à la figure A1.1, correspond à une volonté de maîtriser le développement des réseaux de communication et de choisir les équipements qui s'y raccordent sans subir les contraintes imposées par le parc installé.

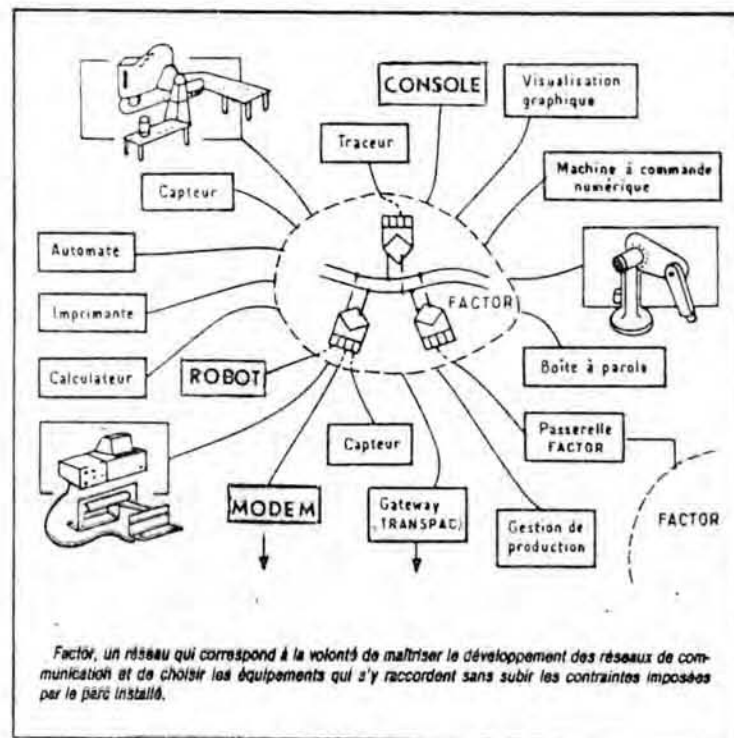


Figure A1.1 - Contexte d'utilisation du réseau Factor.

L'architecture de Factor est fondée sur les concepts d'architecture de systèmes distribués énoncés par l'ISO [ROS 84]. Factor supporte les sept couches définies par le modèle OSI.

La technique Factor est basée sur le CSMA/CD [TOB 80] présentant l'inconvénient commun à toutes les techniques bâties autour de cette méthode, de ne pas pouvoir assurer un temps de réponse en cas de surcharge.

Nonobstant, on ajoute des priorités soit aux niveaux des équipements (équipements privilégiés (P) et non privilégiés (NP)) et au niveau des messages (flot express (FE), messages d'alarme par exemple, et flot normal (FN), messages d'informations normaux) pour que l'on puisse être capable de passer quelques messages plus rapidement que d'autres.

Le tableau A1.1 ci-dessous montre comment sont définis les niveaux de priorité :

NIVEAU DE PRIORITE	TYPE DE COMBINAISON (EQP-FLOT)
0	FE
1	ACQUITTEMENT NIVEAU 2
2	P + FN
3	NP + FN

Tableau A1.1 - Définition de niveaux de priorité.

Les règles de base de cette technique sont les suivantes :

- 1ère) Toutes les stations voient toutes les collisions.
- 2ème) Toutes les stations sont en écoute permanente.

Dans le cas d'une collision, les stations qui l'ont produite brouillent le câble pendant un temps de collision ($TC = 2\tau$, où $\tau = 10 \mu s$ est le temps pour que le signal se propage d'un bout à l'autre d'un câble mesurant 2 km).

Les stations qui ne l'ont pas produite attendent ce même temps, mais en ne brouillant pas le câble.

A la fin de ce temps de brouillage, toutes les stations attendent un temps aléatoire (ΔT), qui est fonction de la priorité de chaque station.

Les stations qui n'ont rien à émettre rentrent également dans le processus d'attente en prenant en compte la priorité la plus faible (priorité de niveau 3) afin de ne pas perturber la stratégie de reconfiguration du médium.

Ce temps aléatoire (ΔT) a été défini de la façon suivante :

$$\Delta T = X_n + n.a. [0, y.60 \mu s]$$

où X_n est l'étalon de temps fixe lié à chacun des niveaux de priorité, selon le tableau A1.2 ci-après :

NIVEAU DE PRIORITE	ETALON
X_0	$20 \mu s$
X_1	$20 \mu s + Y.60 \mu s$
X_2	$20 \mu s + Y.120 \mu s$
X_3	$20 \mu s + 180 \mu s$

Tableau A1.2 - Etalons liés aux niveaux de priorité.

La définition a été faite ainsi pour que l'on ait une intersection vide entre les différents niveaux de priorité.

n.a. est un nombre aléatoire qui varie entre 0 et $60 \mu s$, dans le cas d'une première collision.

Y : [1,8] définit le nombre de collisions successives, de sorte que à chaque collision, l'intervalle de valeurs où l'on tire la valeur aléatoire augmente, ce qui réduit la probabilité de collisions successives.

Le nombre maximum de collisions successives admis est de huit, ce qui donne une attente maximale, dans le cas d'une trame de priorité 3, de l'ordre de 1,94 ms :

$$TC + \Delta T = 20 \mu s + 180 \mu s \times 8 + 8.60 \mu s.$$

1.1.1.2. CARACTÉRISTIQUES PRINCIPALES DU COUPLEUR RESEAU

Le coupleur du réseau Factor se présente matériellement sur une seule carte Multibus. Toutefois, cette carte est composée de trois parties (voir figure A1.2).

a) COUPLEUR RESEAU PROPREMENT DIT

Il réalise les fonctions suivantes :

- contrôleur de communications multi-protocoles (micro-processeur 8751).
- génération de signaux de contrôle de commande (micro-processeur 8751).
- interface vers l'unité de contrôle au médium (UCM) (circuit 2652). Ce circuit traite la trame HDLC [MAC 79]

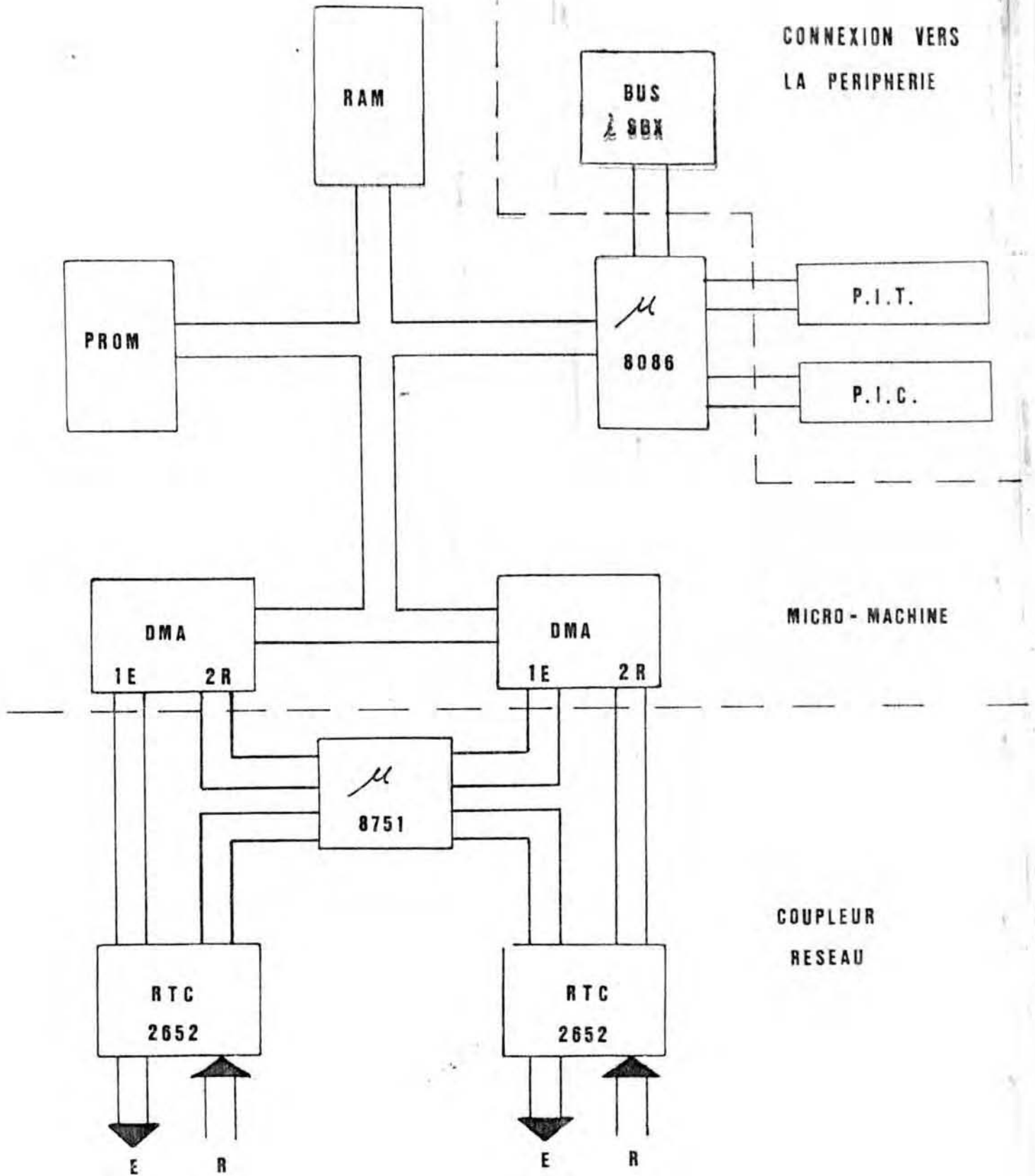


Figure A1.2 - Présentation de la carte coupleur Factor.

On doit remarquer que pour augmenter la sûreté de fonctionnement, Factor emploie deux mediums (câble coaxial) permettant un transfert de 2 x 2 Mbits/s [ITI 84], ce qui entraîne un doublement des circuits 2652 présents sur la carte coupleur.

b) MICRO-MACHINE

C'est la micro-machine qui émet toutes les trames reçues. Les trames sont enregistrées sur la mémoire RAM. Pour décharger le microprocesseur 8086, qui se charge aussi de la gestion de périphériques (voir 1.1.2.3.), la carte Factor dispose aussi de deux circuits DMA (8257) (chacun a une voie en EMISSION et deux en RECEPTION) pour faire le transfert direct des trames depuis ou vers la RAM.

c) CONNEXION VERS LA PERIPHERIE

Chaque carte Factor présente une interface au bus iSBX permettant la connexion jusqu'à 4 équipements. Cela se fait de la manière montrée à la figure A1.3 [SAB 81].

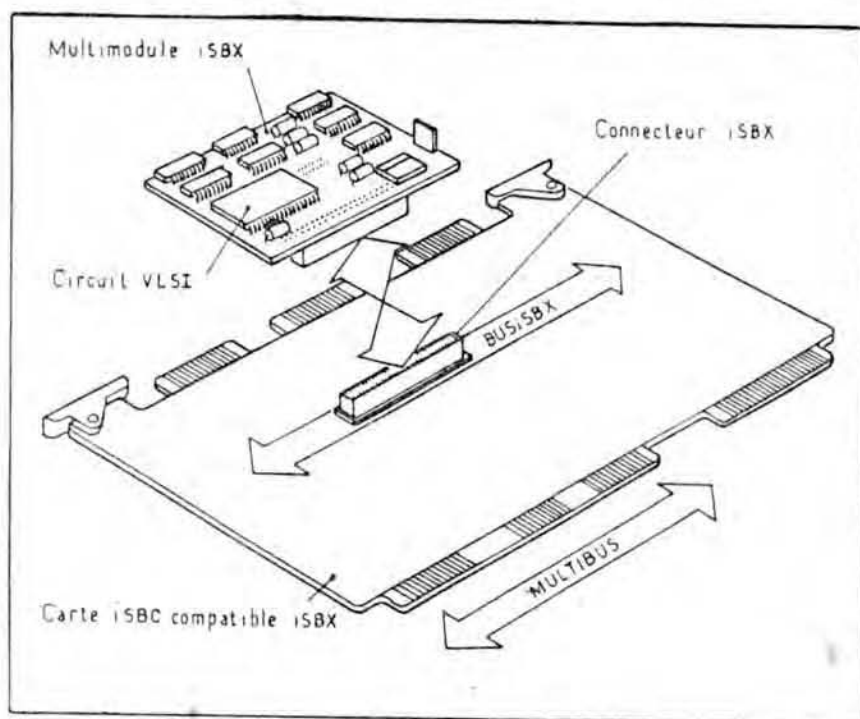


Figure A1.3 - Carte iSBC supportant le bus iSBX.

De plus, la carte Factor présente des circuits spécifiques pour la gestion des interruptions (circuit PIC : 8259 A-2) et la gestion des timeurs (circuit PIT).

1.1.2. CARTHAGE

1.1.2.1. CARACTÉRISTIQUES DU RÉSEAU

Le but du projet CARTHAGE [REN 83] (Commutateur à Accès Réparti pour Terminaux Hétérogènes Adapté à la Gestion d'Entreprises) développé au CCETT (Centre Commun d'Etudes de Télédiffusion et Télécommunications) à Rennes est d'associer sur une topologie en boucle à fibre optique (apparentée au concept de Réseau Local d'Entreprise) à la fois des fonctions de transmission et de commutation de données et des fonctions de commutation téléphonique répartie, telles que celles offertes habituellement par les PABX en étoile. Le synoptique du réseau CARTHAGE est montré à la figure A1.4.

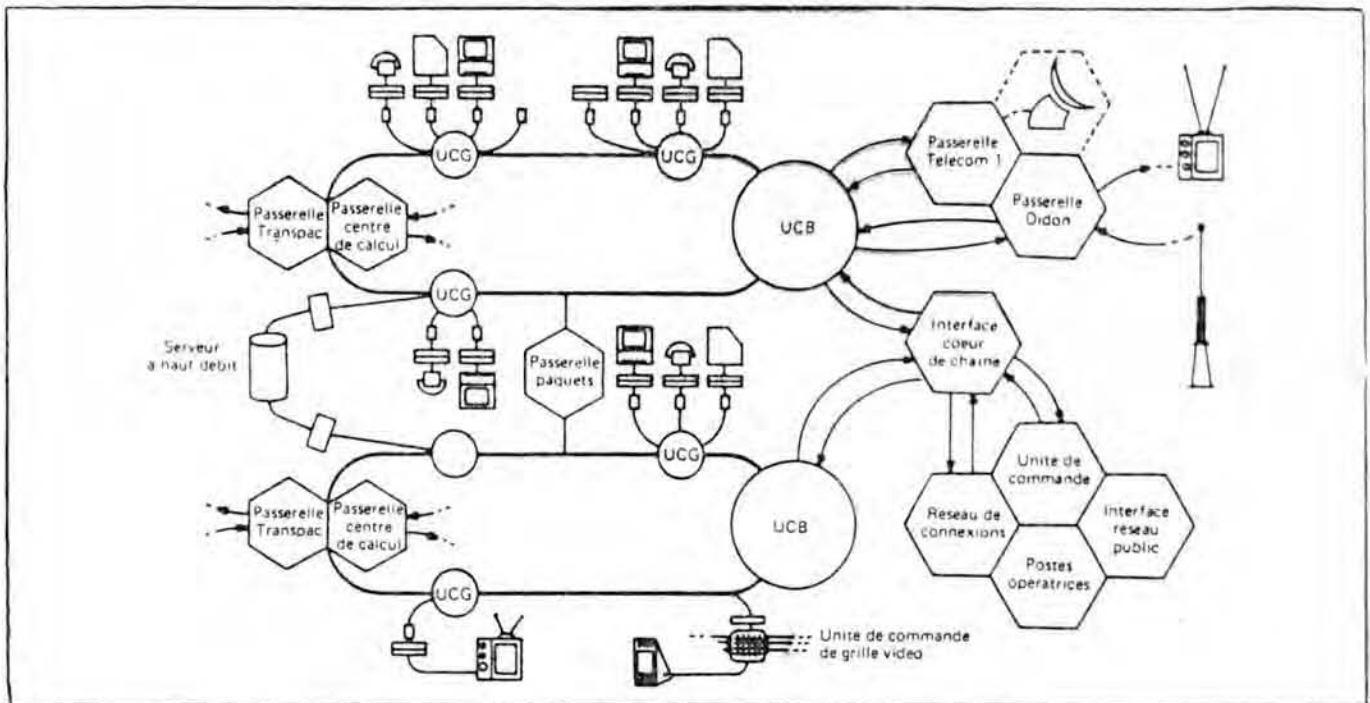


Figure A1.4 - Synoptique du réseau CARTHAGE mis en place au CCETT.

Ainsi, le projet CARTHAGE dans sa version de base à 8 Mbits/s a été désigné pour desservir un centre de 500 personnes, soit un trafic téléphonique pouvant atteindre une centaine d'échanges et un trafic de données de 2 Mbits/s, ou même plus puisque la dynamique du partage entre débit de parole et débit de données est totale. Pour des raisons de fiabilité, la boucle à fibre optique est doublée.

Ce réseau, en plus, grâce à un choix approprié du format des paquets, est capable de supporter des services diffusés du type DIDON-ANTIOPE de façon à permettre pour certaines applications l'utilisation de terminaux grand public bon marché.

La boucle CARTHAGE offre à la fois des facilités de commutation de circuits de parole à 64 Mbits/s selon une technique temporelle classique, de circuits pour données synchrones de 2,4 Mbits/s à 48 Mbits/s et un canal pour la commutation de paquets de données au sens de datagrammes et de circuits virtuels.

La technique d'accès au médium retenue pour ce service est une technique de "jeton" circulant sur la boucle. Le rôle dévolu à ce jeton est triple, il permet bien sûr l'allocation tournante du droit de parole sur la boucle sans contention, mais il permet aussi, grâce à un mécanisme de chaînage de paquets élémentaires, que l'on désignera par segments de données, d'insérer une hiérarchie entre stations connectées sur la boucle. Le principe de cette hiérarchie est basé sur le droit de chaîner un certain nombre de segments de données, c'est-à-dire de garder le droit de parole pendant un certain nombre de trames négociable avec l'Unité de Contrôle de Boucle ou configurable au démarrage du système.

La troisième caractéristique du jeton est de permettre un fonctionnement en mode diffusé.

La recherche d'un système de communication le plus ouvert possible a conduit à retenir pour CARTHAGE une structure de multiplex de transmission temporel offrant aussi bien des canaux transparents en mode circuit que des accès en mode paquet. Ce multiplex est basé sur une structure de trame de durée 125 μ s, la trame étant elle-même divisée en 128 intervalles de temps (IT). La réservation d'un IT par trame constitue donc un canal à 64 Kbits/s. Par ailleurs, la possibilité d'organiser des trames en multitrames de 20 trames permet de multiplexer dans les IT à 64 Kbits/s des sous-canaux (circuits) de données de 2400 bits/s à 48 Kbits/s.

La synchronisation de ce multiplex est assurée par un mot de synchronisation porté par l'ITO de chaque trame.

Pour ce qui est du mode paquet, le mode communication point à point repose sur une enveloppe aussi proche que possible d'HDLC [MAC 79]

La trame [LEF 82] prend l'aspect montré ci-dessous (figure A1.5) :

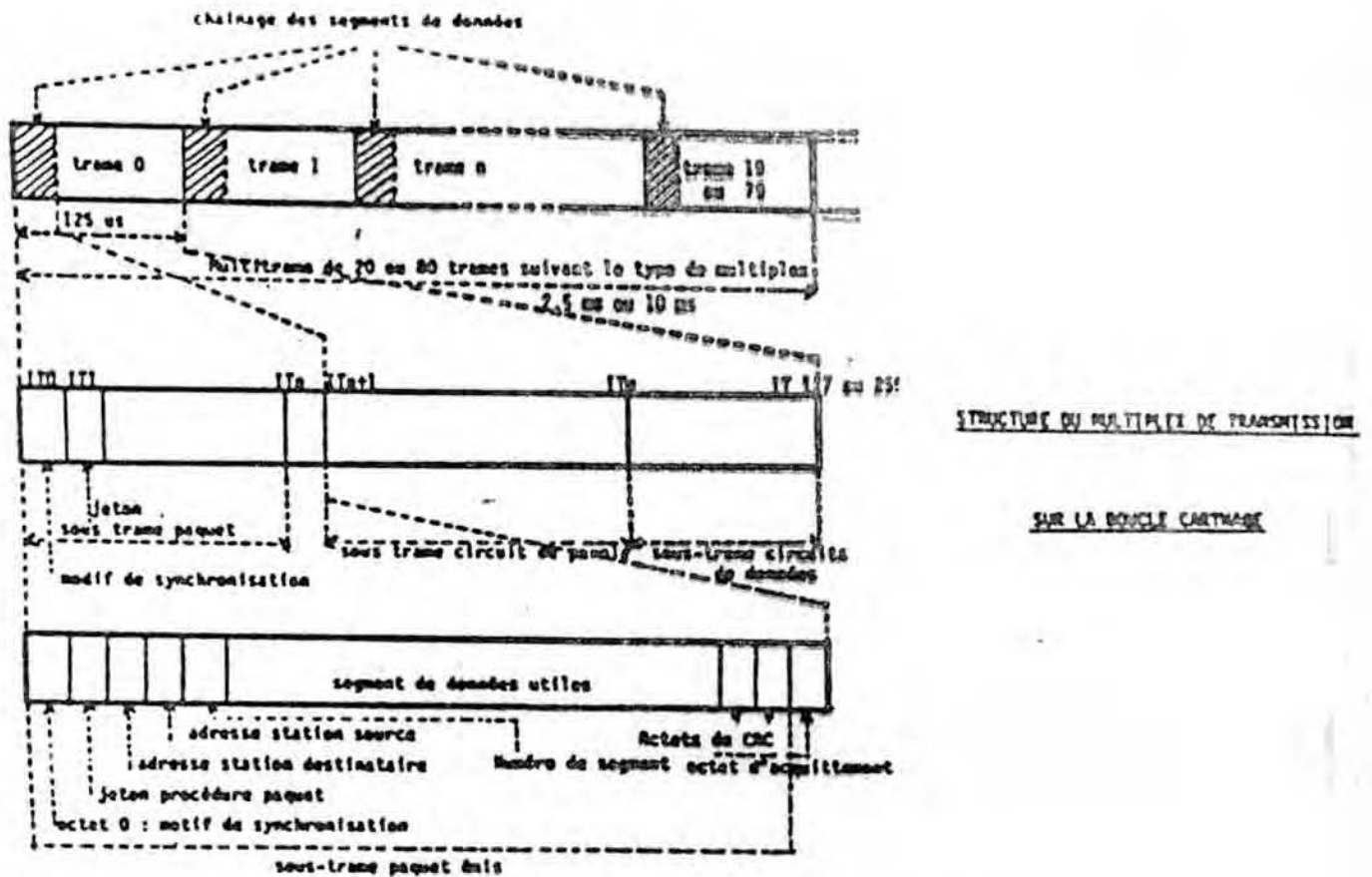


Figure A1.5 - Constitution de la trame CARTHAGE.

On peut donc constater que chaque trame est subdivisée en trois sous-trames, l'une réservée aux communications en mode paquet, l'autre à la transmission de données en mode circuit et la dernière à la transmission de la parole sur canaux téléphoniques.

1.1.2.2. CARACTÉRISTIQUES PRINCIPALES DU COUPLEUR RÉSEAU

La carte COUPLEUR de PAQUETS (COP) du Réseau CARTHAGE se compose des automates et mécanismes suivants :

- l'automate de synchronisation,
- l'automate de procédure de la couche LIAISON de DONNEES (niveau 2) de la norme ISO [RUT 82],
- la mémoire tampon de réception en double accès,
- les files d'attente en réception et en émission,
- la boîte aux lettres d'émission en double accès.

Le schéma de cette carte est montré à la figure A1.6. Elle comporte environ 110 boîtiers.

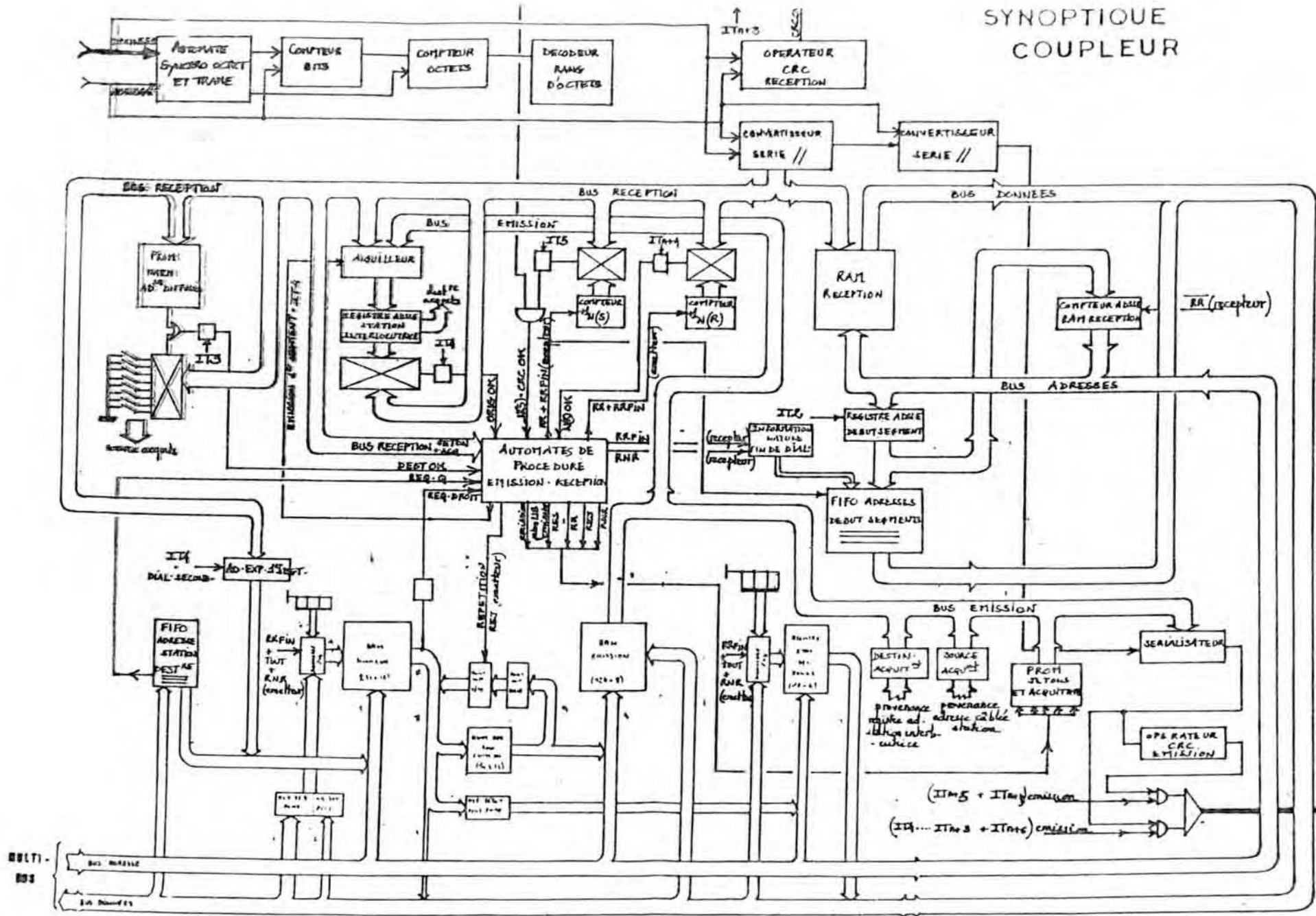


Figure A1.6 - Présentation de la carte coupleur CARTHAGE.

a) FONCTIONNALITE DU COUPLEUR EN RECEPTION

La partie réception, représentée à la figure A1.7, comporte une mémoire RAM de 2 Koctets, les compteurs d'adressage et une FIFO.

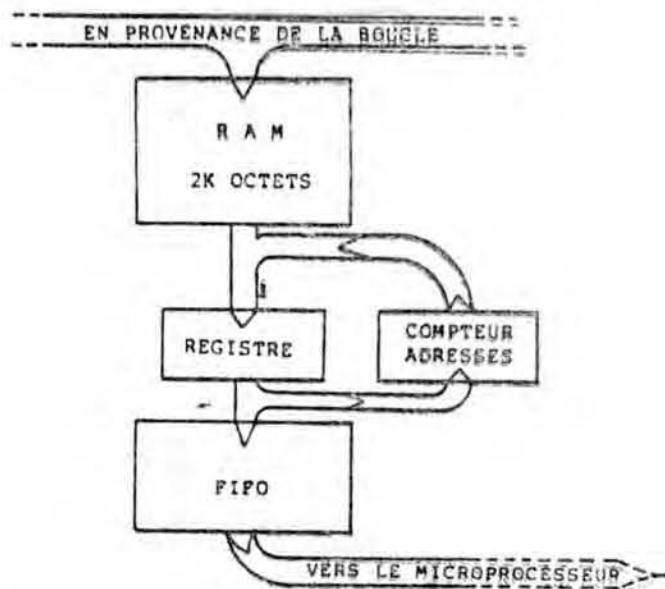


Figure A1.7 - Synoptique de réception du coupleur CARTHAGE.

b) FONCTIONNALITE DU COUPLEUR EN EMISSION

L'émission des paquets est relativement complexe, du fait de la sophistication de la procédure d'émission tendant à l'utilisation optimale de la ressource de communication : (néanmoins, dans la mesure où elle est complètement prise en charge par l'automate de procédure, elle est transparente à l'utilisateur) :

- utilisation en full duplex du canal paquet : à la réception d'un premier segment d'une station distante, la station locale qui peut émettre vers cette station (paquet en attente) le fait immédiatement en même temps qu'elle acquitte le segment reçu.
- contrôle d'erreur : un segment non acquitté par la station distante devra être répété alors même que le suivant est déjà transmis. Ceci implique un mécanisme de reprise.

Le synoptique de l'émission est montré à la figure A1.8.

En plus des parties EMISSION et RECEPTION de la carte coupleur décrites ci-dessus, cette carte comprend aussi :

- une partie destinée à la sérialisation désérialisation des données, calcul de CRC [MAC 79], qui sera réalisée par un circuit intégré prédiffusé spécifique, avec le rôle, pourtant, d'assurer la fonction du coupleur synchronisé.

- une partie destinée au traitement de la procédure ligne, qui sera réalisée par un deuxième C.I. à base de portes prédiffusées. Ce circuit reçoit du premier les octets et la référence temporelle des "segments" de données et lui transmet les octets à émettre.

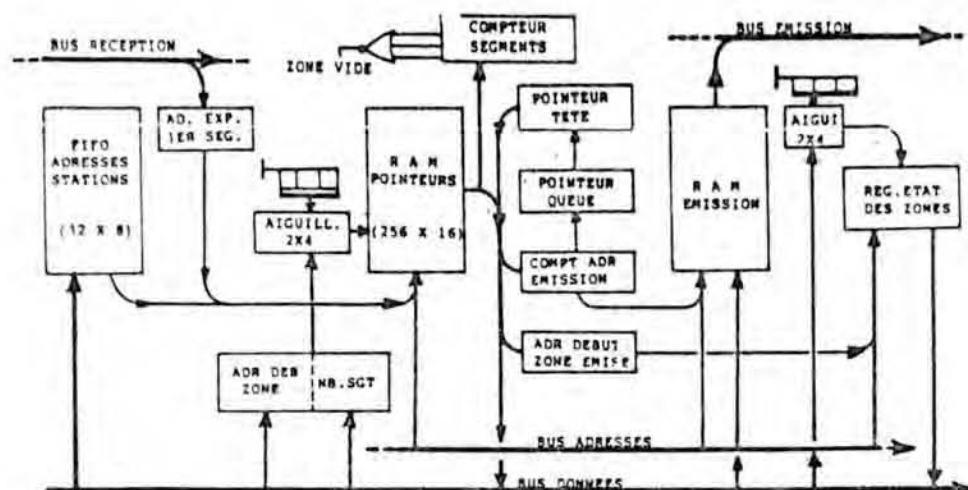


Figure A1.8 - Synoptique d'émission du coupleur CARTHAGE.

1.1.3. RÉBUS

1.1.3.1. CARACTÉRISTIQUES DU RÉSEAU

Le réseau REBUS a été conçu pour les systèmes de communication adaptés à la commande coordonnée des processus industriels. Ce réseau REBUS [AYA 82] fait partie du système de contrôle en temps réel pour des applications industrielles, actuellement fabriqué et commercialisé par la Société SEREG-SCHLUMBERGER, sous le nom de MODUMAT 800.

L'architecture du système REBUS est montrée à la figure A1.9.

On peut remarquer que le système présente deux types de matériels distincts :

- 1) Matériel Salle Technique (partie basse de la figure A1.9) :
 - Régulateur Multiboucles programmables (R.M.P.),
 - Automates programmables,
 - Acquéreurs de données programmables.

2) Matériel Salle de Contrôle (partie haute de la figure A1.9) :

- Consoles de visualisation banalisées ou spécialisées programmables,
- Imprimante pour journaux de bord,
- Enregistreurs graphiques programmables,
- Calculateurs.

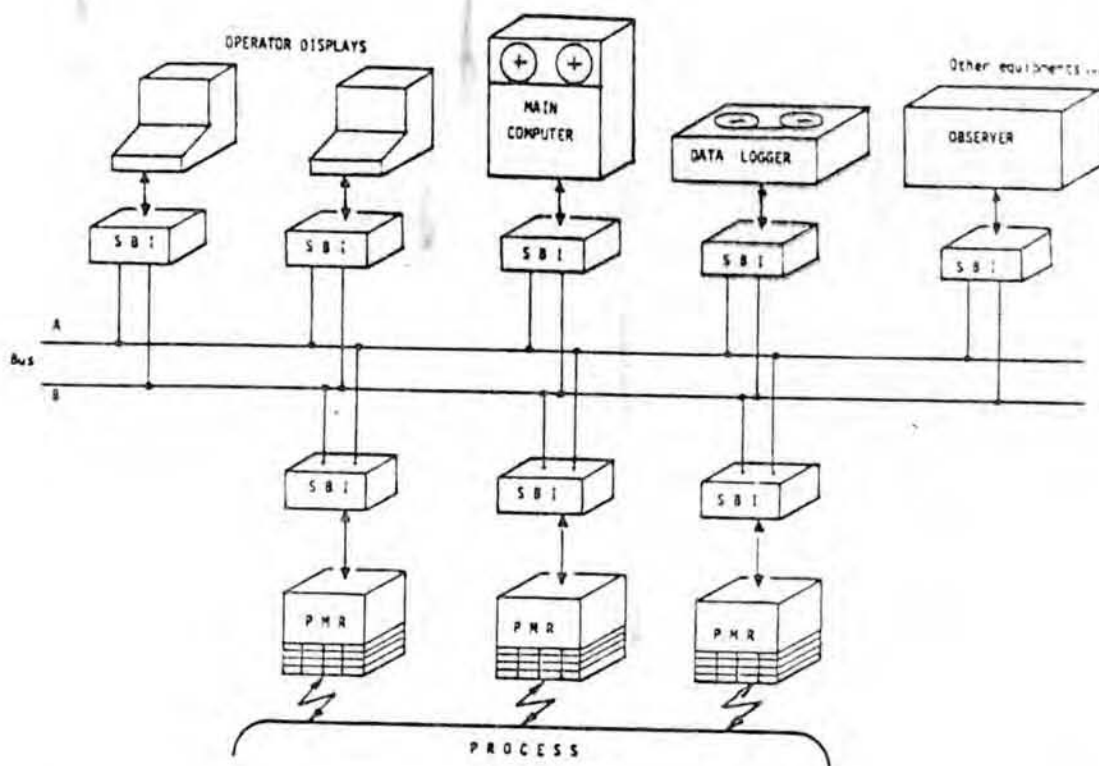


Figure A1.9 - Architecture du système REBUS.

On peut remarquer aussi que le support de transmission (paire bifilaire torsadée blindée) a été doublé pour accroître la fiabilité.

A tout instant, un "bus" suffit pour la transmission de l'information ; le deuxième est donc en redondance, et il sera actionné en cas de défaillance du premier, par des mécanismes distribués de reconfiguration.

Le rôle et la responsabilité de chaque équipement à l'égard de la reconfiguration varie selon qu'il est esclave, maître ou contrôleur.

L'EQUIPEMENT ESCLAVE : répond automatiquement sur la voie qui l'interroge et n'a aucune responsabilité de reconfiguration.

L'EQUIPEMENT MAÎTRE : tente d'obtenir les informations dont il a besoin sur la voie où il a reçu sa délégation. Lorsqu'il n'arrive pas à obtenir une réponse d'un esclave non réputé en panne au bout de n tentatives, il décide alors de son propre gré de l'atteindre sur l'autre voie.

Le CONTRÔLEUR : distribue les délégations à chaque cycle alternativement sur une voie puis l'autre. Il peut alors constater qu'une voie n'est plus utilisable ; il le signale alors et reste sur l'autre voie tant qu'il ne détecte pas la remise en fonctionnement de la première.

La vitesse de transmission sur le bus série est de 512 Kbauds en utilisant les trames HDLC. Il faut remarquer que les concepteurs du système REBUS ont choisi des solutions robustes et simples comportant un maximum de sécurité au détriment d'un débit accru.

L'algorithme d'accès au médium est basé sur la technique de jeton [KEN 82], circulant sur un anneau virtuel, montré à la figure ci-dessous. L'ordonnancement [COR 80] est virtuel, car seulement le jeton suit le chemin virtuel ; les messages transmis par une station qui détient le jeton peuvent être envoyés à n'importe quelle station et aucune organisation physique du médium de communication ne doit être assumée au préalable.

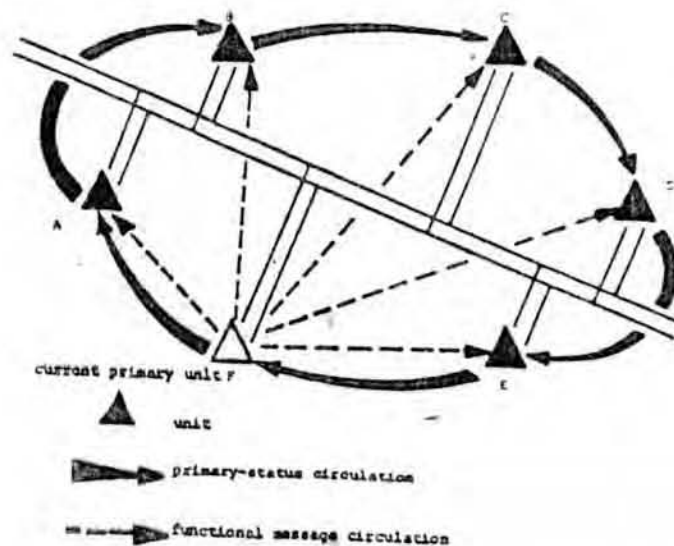


Figure A1.10 - Synoptique de l'algorithme d'accès au bus réseau.

Pour que le protocole d'accès au médium soit fiable et robuste, trois propriétés doivent être satisfaites :

1) A un instant donné, il existe une seule station primaire (qui détient le jeton) (EXCLUSION).

Cette station a accès à toutes les autres stations qui sont alors secondaires, par une procédure d'appel réponse.

La station primaire relâche le jeton au bout du temps qui lui a été alloué, ou lorsqu'elle n'a plus de trame à émettre.

Il existe un contrôleur fixe qui, après chaque relâche donne la main à un nouveau primaire pour un certain temps. En outre, ce contrôleur gère l'enchaînement des primaires grâce à une table préétablie et vérifie les relâches.

2) La perte du jeton ne peut arriver que temporairement (ROBUSTESSE).

3) Toutes les stations "saines" doivent être capables de recevoir le jeton (FIABILITE).

1.1.3.2. COUPLEUR RÉSEAU

Considérations générales

Comme on avait montré à la figure A1.9, tous les équipements (employés soit en Salles Techniques, soit en Salles de Contrôle) sont raccordés aux bus par le biais de cartes INTERFACE BUS SERIE (SBI à la figure).

Cette carte qui est montrée à la figure A1.11 est composée de trois parties [DIA 79] :

- une partie d'interface avec le bus,
- une partie de traitement,
- et une partie d'interface avec le processeur à connecter au système.

Ce découpage est tout à fait fonctionnel car il permet une grande souplesse : si le bus change, par exemple devient optique, seule la partie d'interface bus sera à modifier ; si le processeur diffère, par exemple devient à 16 bits, seule l'interface arrière est à modifier (actuellement les microprocesseurs de visualisation et de régulation sont à 8 bits).

On va analyser maintenant plus en détail chacune de ces trois parties.

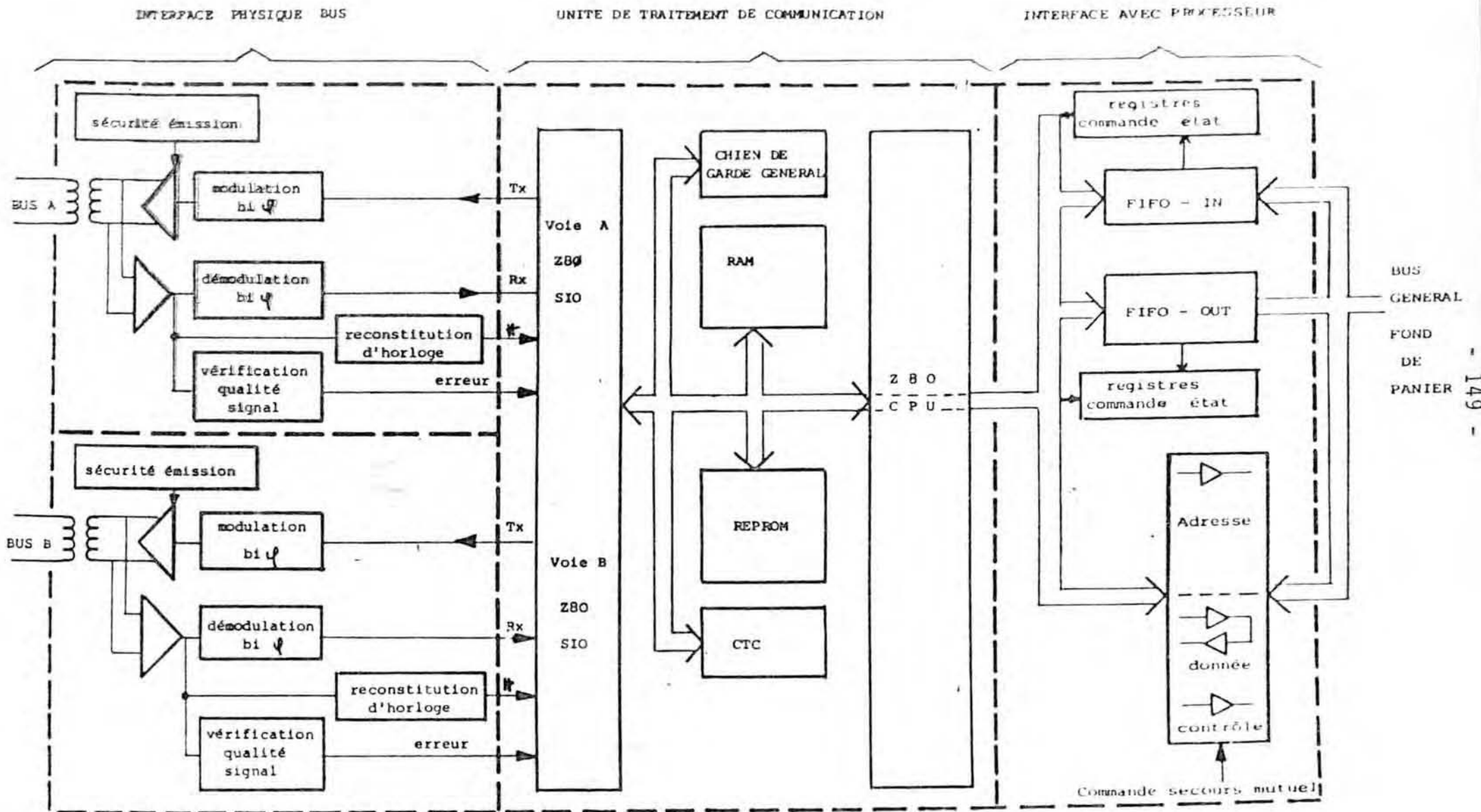


Figure A1.11 - Carte d'interface bus série.

INTERFACE AVEC LE PROCESSEUR DE L'EQUIPEMENT

Elle permet de recevoir les demandes du processeur de l'équipement et lui renvoie les réponses. Elle se compose de deux parties : Interface d'Echange et Interface Bus Processeur.

INTERFACE D'ECHANGE

Correspond à la partie haute de l'Interface avec Processeur montré à la figure A1.11.

L'interface d'échange est utilisée en fonctionnement normal et comprend deux registres à 8 bits utilisés respectivement en FIFO entrée (écrit par l'équipement et lu par l'interface) et FIFO sortie (écrit par l'interface et lu par l'équipement). Ces deux registres sont chargés et lus de manière asynchrone. A chacun d'eux est associé un registre de commande d'état composé de trois bits : Autorisation-lecture, Autorisation-écriture et Commande ou donnée (0 = DONNEES, 1 = COMMANDES). Les deux bits d'autorisation permettent d'effectuer la synchronisation entre le processeur équipement et le processeur interface. Ces registres d'état sont lus de manière asynchrone.

C'est l'équipement qui prend l'initiative des transferts ; pour cela, il lit le statut de l'interface : information disponible, type d'information, prêt à recevoir.

La procédure d'échange est donnée selon le schéma montré à la figure A1.12.

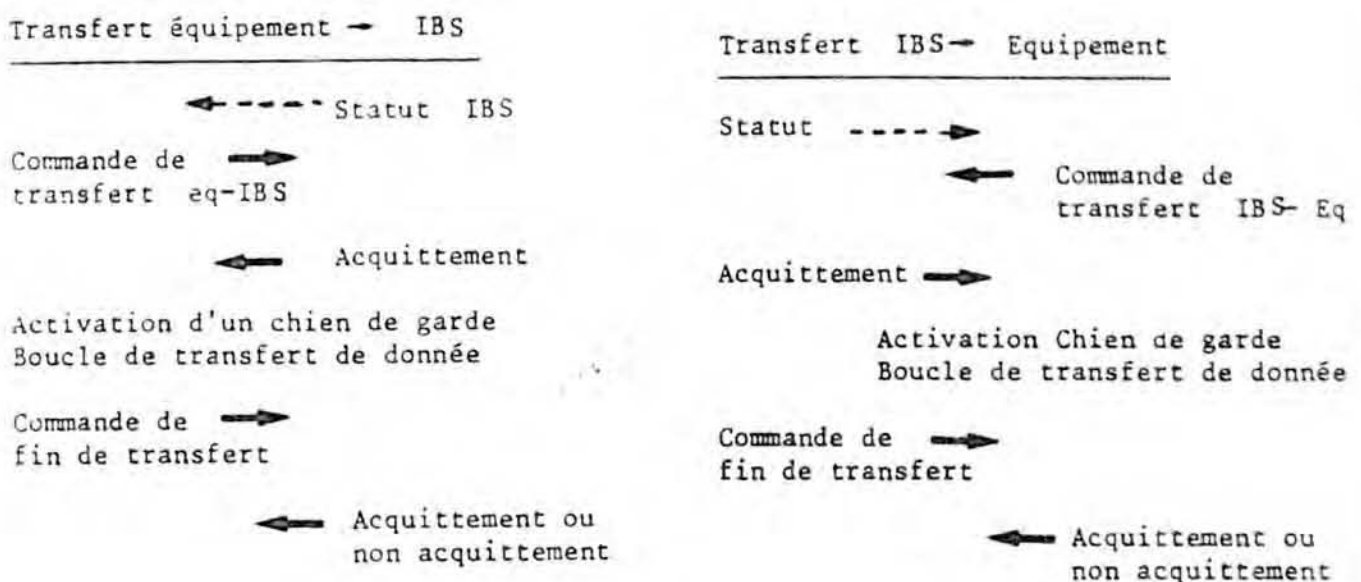


Figure A1.12

1.2. NORME IEEE 802

1.2.1. ÉTAT DE LA NORMALISATION

Plusieurs constructeurs se sont lancés, à la suite de XEROX (créatrice d'ETHERNET [MET 76]) dans l'étude et le développement d'architecture de réseaux locaux, mais le problème d'une architecture standard et normalisée reste entier.

Néanmoins, plusieurs comités internationaux se sont penchés sur le problème de normalisation de réseaux locaux.

Parmi eux, on peut citer l'ISO [RUT 82] (International Standard Organization), laquelle par l'intermédiaire de son sous-comité SC 16 a proposé une norme d'Interconnexion des Systèmes Ouverts (OSI), très répandue car cette norme, due au caractère d'"ouverture" du système, permet à tous constructeurs qui la respectent de se connecter avec d'autres.

Cette norme est divisée en sept couches. Chacune de ces couches réalise des fonctions spécifiques, lesquelles vont être mises à profit par des couches de niveaux supérieurs.

Néanmoins, cette norme n'est pas tout à fait applicable aux réseaux locaux, car elle a été faite pour les réseaux généraux.

Pour les réseaux locaux, la norme la plus adaptée est celle qui est spécifiée par l'IEEE [DAN 82] (Institute of Electrical and Electronics Engineers) à travers son comité 802, laquelle est compatible avec la norme ISO-OSI citée précédemment.

Le but de cette commission est de développer un standard permettant la transmission de trames de messages entre systèmes informatiques de conception courante, à travers un support partagé entre ces systèmes quelle que soit leur architecture. Le comité s'est assigné également une démarche d'approche par couche, conformément au modèle de l'ISO relatif à l'interconnexion des systèmes ouverts.

Pour fournir un standard de ce type, il faut tout d'abord spécifier les niveaux bas de l'architecture (niveau "Physique" et niveau "Liaison de Données") et procéder à une certaine extension avec aménagement du troisième niveau (niveau "Réseau") du modèle ISO. Le but du réseau n'est pas d'offrir une liaison très évoluée et donc compliquée entre plusieurs machines d'architectures différentes, mais d'offrir plutôt une interface de liaison d'un niveau relativement bas entre systèmes informatiques se partageant le réseau.

Au niveau européen, l'ECMA (European Computer Manufacturers Association) a défini un standard compatible à ISO-OSI et maintenant, elle a défini une norme pour les réseaux locaux, norme compatible avec IEEE 802.

Nous l'avons choisie car elle est la plus adaptée aux réseaux locaux. De ce fait, on va essayer de configurer chacun des réseaux retenus (FACTOR, REBUS, CARTHAGE) selon cette norme [BAR 83-2].

1.2.2. DESCRIPTION DE LA NORME IEEE 802

Les caractéristiques principales de la norme IEEE 802 sont les suivantes :

1) Deux méthodes d'accès sont supportées par la norme : CSMA/CD et jeton circulant.

2) Deux topologies sont supportées : bus et anneau physique.

- Pour le bus, on peut avoir soit du CSMA/CD [TOB 80], soit du jeton circulant [KEN 82].

- Pour l'anneau physique, on ne peut avoir que le jeton circulant.

3) Les protocoles de niveau 2 (LIAISON DE DONNEES) de la norme ISO sont sensés être les mêmes pour toutes les méthodes d'accès et topologies.

4) Au-dessous de la sous-couche d'accès de niveau 2, c'est-à-dire au niveau 1, il y a plusieurs types de couches physiques : CSMA/CD sur bus, jeton sur bus et jeton sur anneau. Bien qu'il y ait des similarités entre les medias physiques de ces trois catégories, le but principal est d'optimiser les medias par rapport à la méthode d'accès et à la topologie, la recherche des caractéristiques communes étant regardée comme de deuxième priorité. Chaque catégorie peut avoir plusieurs medias (par exemple, CSMA/CD et jeton sur bus pouvant avoir media coaxial en bande de base ou bande large, et le jeton sur anneau peut se présenter sur paires torsadées, câble coaxial et fibre optique).

Ci-après, la figure A1.13 montre la structuration en couches de la norme IEEE 802. On montre également la norme OSI pour qu'on puisse constater les relations entre elles [DAN 82].

Par la suite, on va essayer de caractériser chacune des couches de la norme IEEE 802.

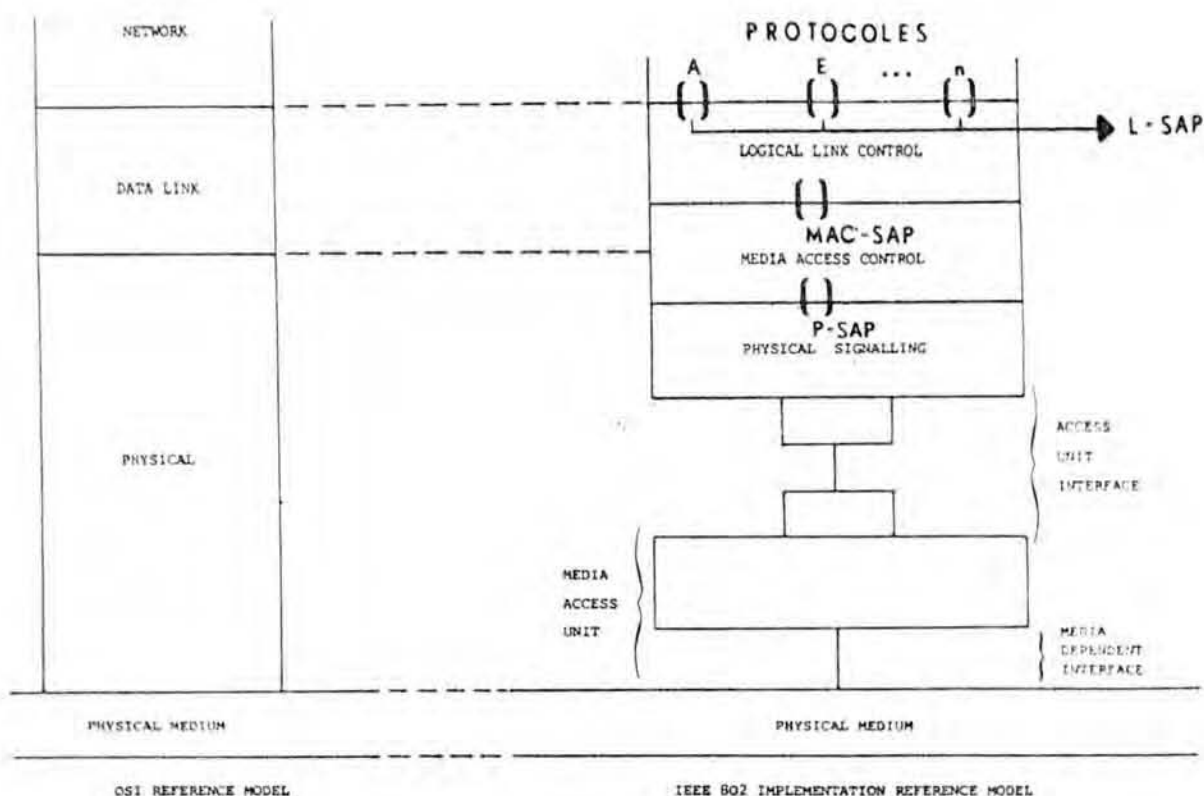


Figure A1.13 - Comparaison des normes OSI et IEEE 802.

1.2.2.1. COUCHE "LOGICAL LINK CONTROL" (LLC)

Cette couche supporte deux types de service de liaisons :

MODE SANS CONNEXION

Les trames sont échangées entre différentes stations LLC sans avoir besoin d'établir une liaison logique entre paires de points d'accès au LLC (L-SAP : "link layer service access point").

Au niveau LLC, les trames ne sont pas acquittées, et il n'y a pas non plus aucun contrôle de flux ou procédures de récupération d'erreurs.

Ce mode devrait être utilisé quand, par exemple, le niveau 4 de la norme ISO (niveau Transport) assure le séquençement et la reprise.

MODE AVEC CONNEXION

Une liaison logique est établie entre paires de points d'accès LLC (L-SAPs) avant tout échange de données.

Dans le mode de transfert de données, les trames sont transmises/délivrées en séquence. La politique de reprise

(error recovery) et de contrôle de flux sont faites au niveau LLC.

En fait, le LLC IEEE 802 supporte plusieurs protocoles de la couche réseau différents (A, B, ... N, dans la figure A1.13 par le biais de L-SAPs différents).

Ce mode devrait correspondre au service rendu par un protocole comme HDLC, pour des couches plus élevées assez élémentaires.

1.2.2.2. COUCHE "MEDIUM ACCESS CONTROL" (MAC)

La couche MAC réalise une des deux fonctions en supportant la couche LLC : CSMA/CD et jeton.

Le MAC réalise la fonction d'adressage des stations destinataires. Par rapport au modèle OSI, le MAC réalise des fonctions de niveau 2 (LIAISON DE DONNÉES) comme le calcul du champ de contrôle CRC et aussi des fonctions qui relèvent du niveau 1 (PHYSIQUE) comme le traitement du fanion et l'algorithme de contention ("back-off").

1.2.2.3. "COUCHE PHYSIQUE"

La couche physique de la norme OSI correspond à plusieurs couches de la norme IEEE 802 (voir figure A1.13).

La "couche physique" a pour but la transmission et la réception de bits entre P-SAPs. En fait, cette couche fournit la capacité de transmettre et recevoir des signaux modulés relatifs à certains canaux de fréquence, dans le cas de bande large, ou un simple canal de fréquence dans le cas de bande de base.

Les sous-couches de la norme IEEE 802 qui correspondent à la couche PHYSIQUE de la norme ISO (modèle OSI) sont décrites ci-après :

. SOUS-COUCHE MAC

En fait, le MAC appartient à la fois aux couches de niveaux 2 et 1 du modèle OSI.

. SOUS-COUCHE DE SIGNALISATION PHYSIQUE (PSS)

Les services fournis par cette sous-couche permettent à la sous-couche locale MAC de transmettre et recevoir trames à/de entités MAC distantes. Le PSS contribue aussi avec l'entité de gérance locale du PSS et supporte les services demandés par l'entité de gérance (par exemple, répertoria-tion de statistiques, initialisation, etc ...).

. SOUS-COUCHE UNITE D'INTERFACE D'ACCES

Cette sous-couche remplit le rôle d'interface physique entre PSS et la sous-couche M.A.U. (Media Access Unit). En plus, cette unité est conçue de façon que les différences, parmi les différents medias, soient transparentes aux DTE (Data Terminal Equipment). La sélection de signaux logiques de contrôle et les procédures fonctionnelles sont conçues avec cette finalité.

. SOUS-COUCHE UNITE D'ACCES AU MEDIUM (M.A.U.)

M.A.U., c'est le dispositif qui fournit l'interface entre l'unité d'interface d'accès et le medium. En fait, le M.A.U. fournit les moyens pour faire la transformation de signaux logiques en électriques et vice versa.

1.2.2.4. PRIMITIVES DE SERVICE ÉCHANGÉES ENTRE COUCHES

Les couches et sous-couches de la norme IEEE 802 exécutent en fait des services. Les services réalisés par une couche sont les capacités qu'elle offre à la couche de niveau juste au-dessus. Pour qu'une couche puisse réaliser ces services, elle s'appuie sur les services offerts par la couche au-dessous.

Les services sont spécifiés par la description d'un flot d'informations à l'interface entre deux couches adjacentes. Ce flot d'informations est modelé par des événements discrets et instantanés à l'interface, qui caractérisent la fourniture d'un service. Chacun de ces événements consiste à passer une primitive de service d'une couche à une autre à travers un point d'accès de service (SAP) associé à l'interface.

Il faut bien voir que ces primitives de service sont une abstraction car elles spécifient seulement les services rendus sans faire aucune référence aux moyens à travers desquels ces services sont réalisés.

Par la norme IEEE 802, les services rendus par les couches sont spécifiés par la description des primitives de service et paramètres qui caractérisent ce service.

D'après la norme, les primitives peuvent être classifiées en trois types différents :

. REQUEST

La primitive REQUEST est passée d'une couche N à une couche N-1 pour demander qu'un service soit fait.

. INDICATE

La primitive INDICATE est passée d'une couche N-1 à une couche N pour indiquer qu'un événement interne à la couche N-1 est significatif pour la couche N. Cet événement

peut être relié logiquement à une demande REQUEST faite ailleurs (station distante) ou peut être la conséquence d'un évènement interne à la couche N-1.

RESPONSE

La primitive RESPONSE est passée de la couche N-1 à la couche N pour lui donner les résultats associés à une demande REQUEST faite précédemment.

Ces primitives sont illustrées à la figure A1.14 ci-dessous :

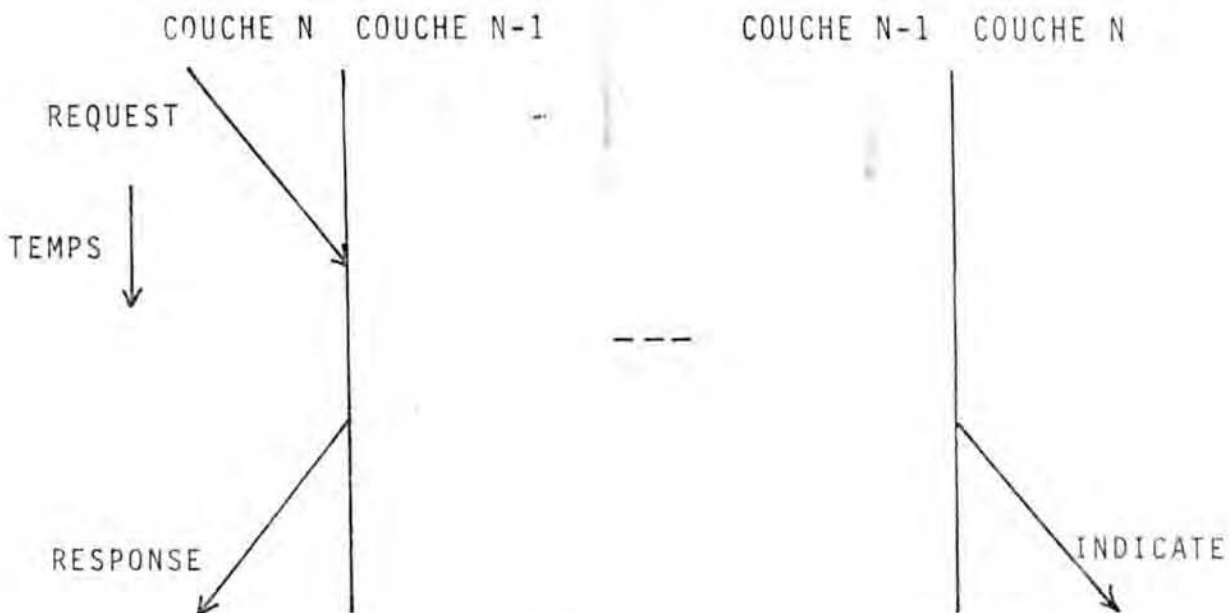


Figure A1.14 - Primitives de service échangées entre couches.

1.2.3. CONFIGURATION DE CHACUNE DES CARTES COUPLEURS DES RÉSEAUX ÉTUDIÉS SELON LA NORME IEEE 802

D'après l'analyse qu'on vient de faire de différents réseaux locaux, au niveau de leur fonctionnalité et de la norme la plus adaptée aux réseaux locaux industriels, la norme IEEE 802, on résume dans la tableau A1.1 les caractéristiques principales des techniques retenues.

Au niveau des cartes coupleurs associées à chaque réseau, l'étude de la norme IEEE 802 nous permet de configurer chacune d'entre elles selon cette norme.

De ce fait, on montre :

- dans la figure A1.15, la configuration du coupleur Factor selon la norme IEEE 802.
- dans la figure A1.16, la configuration du coupleur CARTHAGE dans cette norme.

- dans la figure A1.17, la configuration du coupleur REBUS selon la même norme.

Type de proposition	CARTHAGE	FACTOR	REBUS
Equipement	V24/X24/X21	V24// /HDLC/GPIB Méth.access sess.	V24
5	Rien	ECMA Session	Rien
4	ECMA/ISO c10 - c12	ECMA/ISO c13	ECMA/ISO c12
3	compatible X25.3	MAP résenu, redondance à gérer	MAP résenu redondance à gérer
2.LLC	compatible HDLC	compatible HDLC	compatible HDLC
2.MAC	JETON	CSMA.CD	JETON VIRTUEL
1	Bdb Manchester	Bdb Manchester	Bdb Manchester
Medium	fibre	coaxial	paire

Tableau A1.1 - Caractéristiques principales des réseaux locaux étudiés.

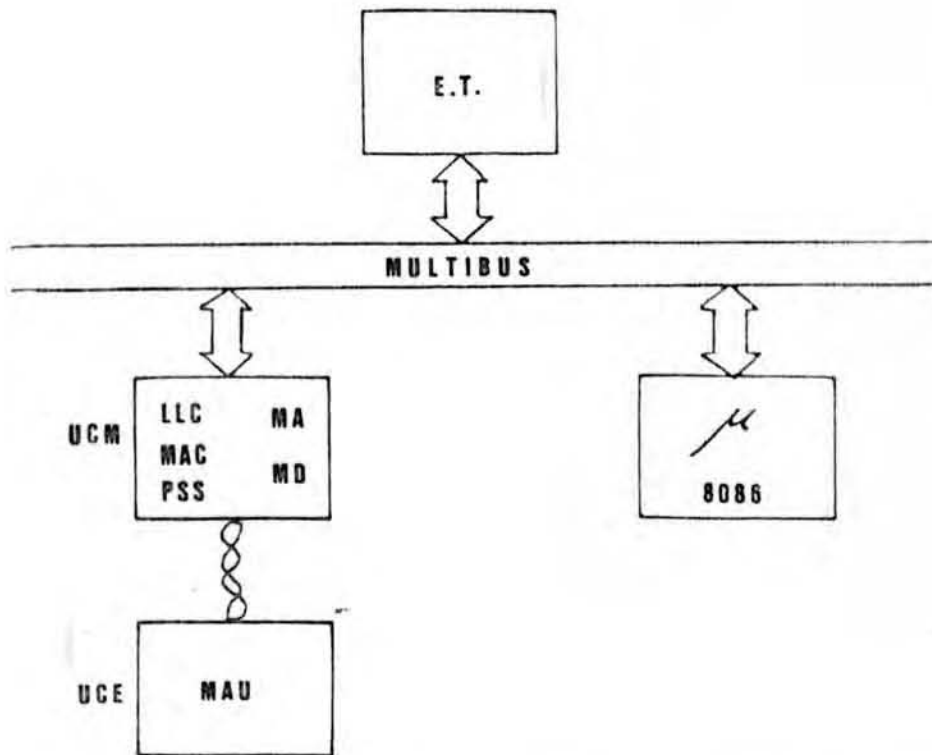


Figure A1.15 - Configuration du coupleur FACTOR selon norme IEEE 802.

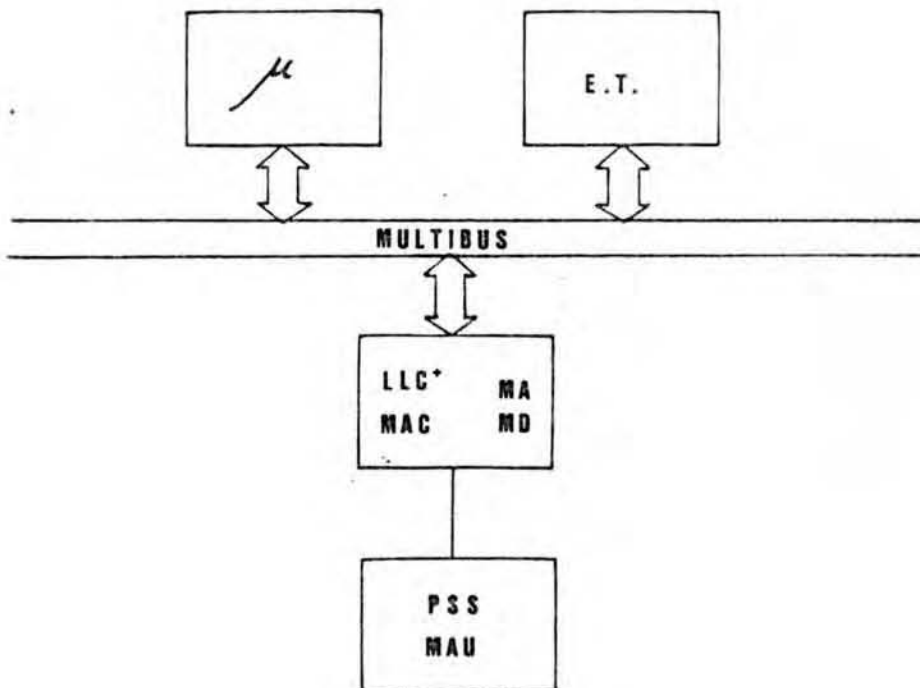


Figure A1.16 - Configuration du coupleur CARTHAGE selon norme IEEE 802.

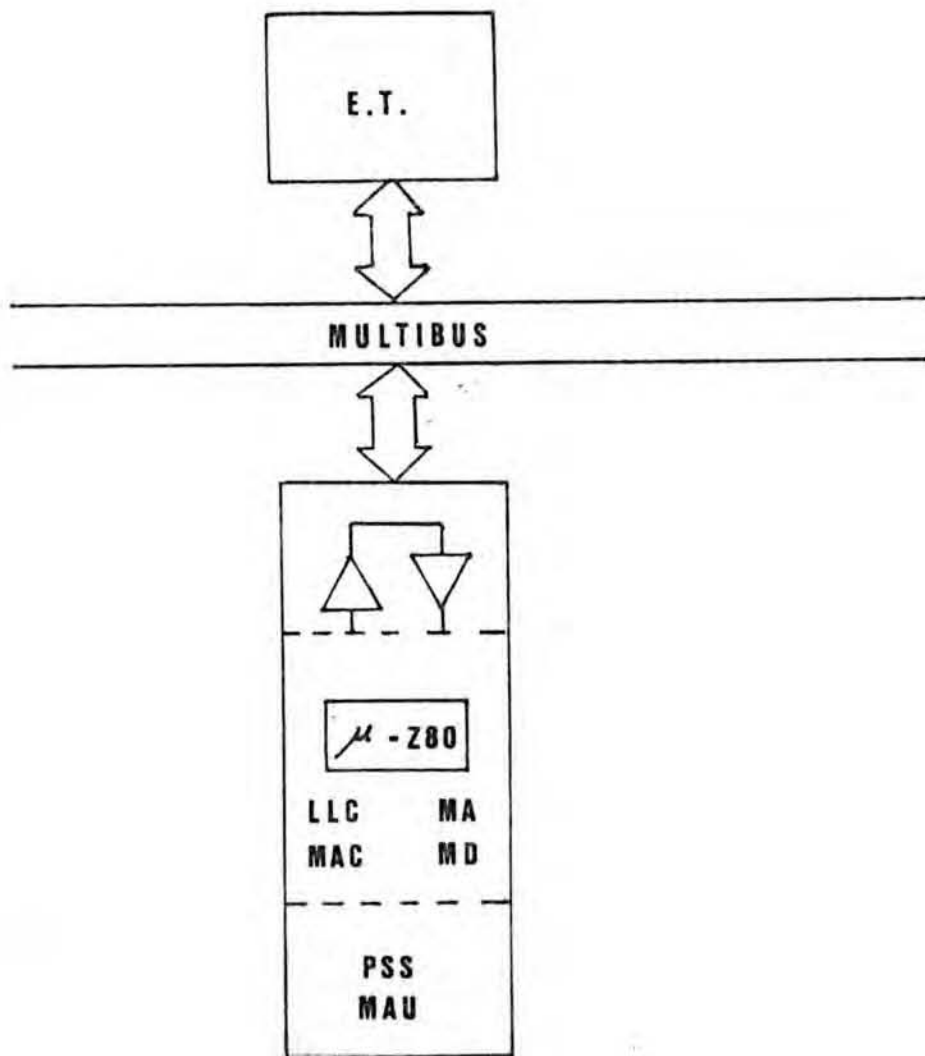


Figure A1.17 - Configuration du coupleur REBUS selon norme IEEE 802.

1.2.4. REMARQUES

D'après l'analyse faite de différents coupleurs, il est possible d'envisager un découpage physique entre circuits dédiés à la communication de sorte que, à chaque technique d'accès ou bus (CSMA/CD, jeton physique, jeton virtuel), on associe un circuit spécifique noté GT (gérant de transmission). Il est d'autre part possible d'imaginer qu'un circuit VLSI nommé GC (gérant de communication) puisse intégrer les protocoles de niveaux 2 (norme IEEE 802) lesquels ne dépendent pas de la technique d'accès au bus (LLC et quelques niveaux au-dessus):

Dans les figures A1.18 et A1.19, on montre des architectures possibles pour ces circuits :

- dans la figure A1.18, l'interface entre les 2 circuits GT et GC est parallèle.

- dans la figure A1.19, l'interface entre les 2 circuits GT et GC est en série.

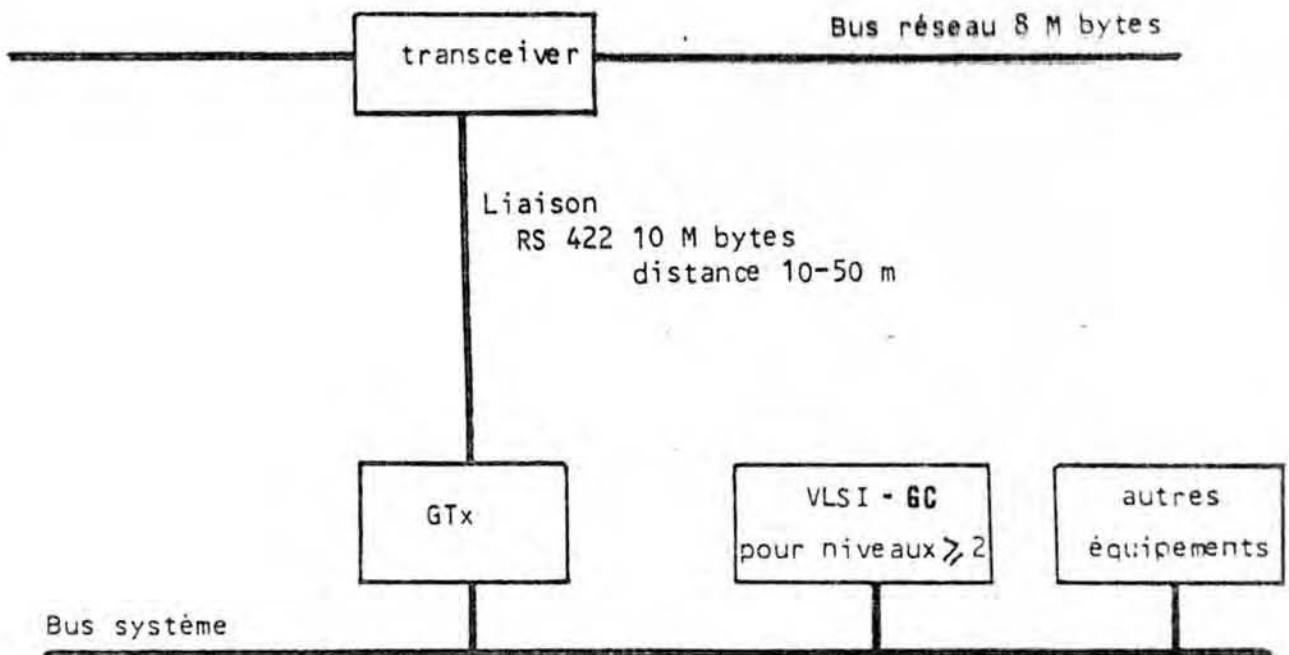


Figure A1.18 - Interface parallèle entre les circuits gérants de communication (GC) et de transmission (GT).

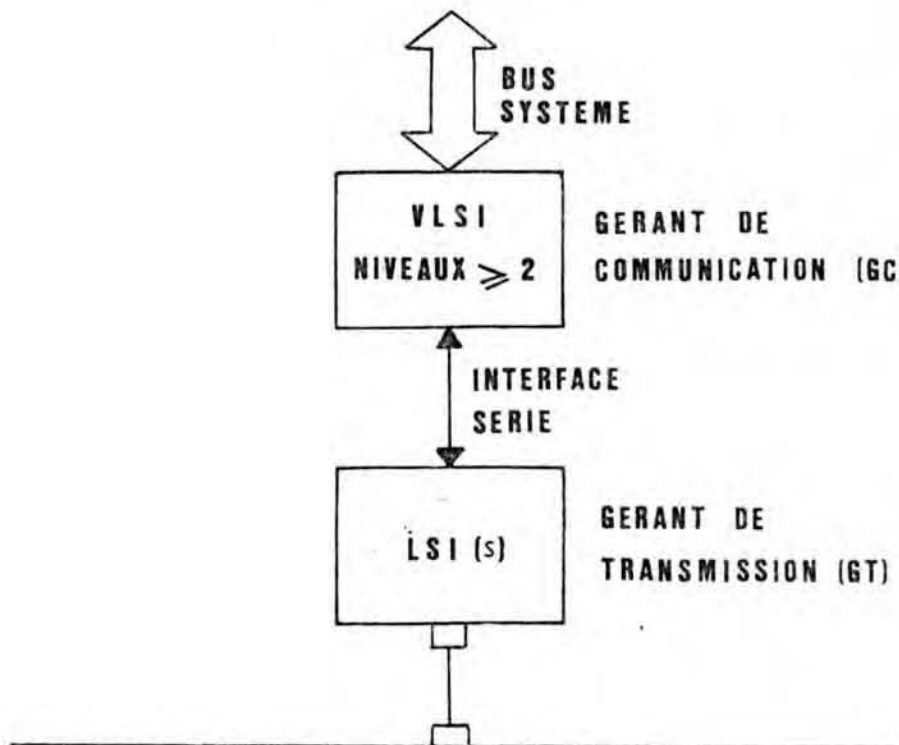


Figure A1.19 - Interface série entre les circuits gérants de communication (GC) et de transmission (GT).

Dû à l'intérêt de ces circuits, on propose ci-après une interface logique entre les niveaux LLC et MAC, pour les réseaux analysés.

1.3. PROPOSITION D'UNE INTERFACE EN TERMES DE PRIMITIVES DE SERVICES SPECIFIEES PAR LA NORME IEEE 802 ET INDÉPENDANTES DU SUPPORT

Comme on a vu au § 1.1.2.2., le passage d'informations à travers les couches de la norme IEEE 802 se fait à travers des primitives de service.

L'analyse de primitives de services utilisés par les différents réseaux analysés (FACTOR, CARTHAGE et REBUS) et par les normes (IEEE 802, ETHERNET et ECMA/TC 24 [EZM 82], en considérant le circuit proposé par IBM pour cette dernière) nous a permis de proposer une interface logique commune aux techniques et normes qu'on vient de citer. Néanmoins, on a dû proposer d'autres primitives assignées dans les tableaux suivants (A1.2 à A1.9) et aussi d'autres paramètres dans les primitives existantes, pour pouvoir passer des services qui ne sont pas spécifiés par la norme IEEE 802.

TABLEAU 2

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
*MA.STATUS.REQUEST status-options :							
- charge Nb messages de service total de trames [VARIABLE ENTIERE]	---	---	---	---	---	---	OUI
- inhibition de réac- tion (liée à l'ob- servateur réseau) [VARIABLE BOOLEENNE]	---	---	---	---	---	---	OUI
- réaction automati- que et immédiate sous forme de messa- ge d'erreur [VARIABLE BOOLEENNE]	---	---	NON : il n'y a que des ac- quittements positifs	---	ACK négatifs (RNC) REJETS de segment	---	OUI
- retransmission jusqu'à N fois [VARIABLE ENTIERE]	---	---	3	---	REPMAX	OUI	3
- réaction automati- que en ACQUITTEMENT [VARIABLE BOOLEENNE]	---	:La procédure d'ACK est du ressort des couches du niveau haut non implémen- tées au sein d'ETHERNET	---	---	Dernier IT de la sous-trame paquet	---	:NON : atten- dre message d'erreur seu- lement

TABLEAU 3

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
*MA.STATUS.RESPONSE status-options :							
- informations ex- traites à partir de l'observation de char- ge	---	---	---	---	---	---	OUI
- inhibition de réac- tion a pu être prise en compte ou pas	---	---	---	---	---	---	OUI
-réponse à la demande de réaction automati- que et immédiate sous forme de message d'er- reur	---	---	---	---	---	---	
-réponse à la demande de retransmission jusqu'à N fois	---	---	---	---	---	---	
- réaction automati- que en acquittement a pu être faite ou pas	---	---	---	---	---	---	

TABLEAU 4

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
*MA.EVENT.INDICATE évent. options :							
- incohérence de com- portement dans l'in- terface	---	---	Cas d'alarme: peut arriver dans les pé- riodes de test	---	OUI	---	OUI
- état de saturation du réseau [VARIABLE BOOLEENNE]	---	---	---	---	OUI , FIFO moitié pleine	OUI	OUI

TABLEAU 5

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
*MA.DATA.REQUEST Paramètres :							
Adresse destination [Nb ent. caractères]	1 à 70 : Cette @ peut être : : -physique : -logique : .sous groupe : .dif. gén.	0 à 1er BM d' 60 : 0=@ PHYS : 1=@ LOG : Dif. ← : Gén. ↓ : 48x'1' Sous : Groupe	10 (PHYS) : 'FF'=@ Groupe	60	10 (PHYS)	10	: 60 : 5 premiers 0 : =@ LOGIQUE : derniers 0 : =@ PHYSIQUE
Adresse source [Nb ent. caractères]	1 à 70 : CRC CSMA/CD : 40 : TOKEN 20	0 à 60	10	10	10	10	10
Data-unit [NB ent. caractères]							
- destination-service- accès-point	: 10=@ du proto: : cole lié au : LLC qui est : en train d'é- : changer l'in- : formation						
- source-service- accès-point	: 10=@ du proto: : cole avec le- : quel on se : communique						

TABLEAU 5 (suite)

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
- champ de contrôle	10= pareil à HDLC	20 : type de paquet : à transmettre : type de pro- : tocole niveau : usager	10 contient le type de commande de la trame	20 : priorité : jeton : comptage : réseau prio- : rité : EXT, SIG, : EXP	10:n° de seg- ment	10= type de trame : trame I INFO : trame S : SUPERVISION : trame U NON : NUMEROTEES	10= type de trame HDLC
champ d'information	√ NOMBRE ENT. d'OCTETS entre 46 et 15000	46 et 15000	2560	4K0	32,64 ou 960	√ NOMBRE ENT. de CARACTERES	2560
<u>Data-size</u> [NB ent. caractères]							
<u>transit-service</u> :	OUI	OUI	: Nb de la tra- : me HDLC, con- : tient la tail- : le du champ : INF0 en Nb : d'octets	---	: ler 0 de : données	: NB de la tra- : me HDLC	: 10
- <u>normal</u> [BOOLEAN]	OUI	OUI	---	OUI	---	OUI	: correspond à : priorité 0
- <u>avec priorité</u> [BOOLEAN] priorité [0, n]	---	---	: 4 niveaux	: 8 niveaux	---	---	: 256 (10)

TABLEAU 5 (suite)

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
- <u>demande d'acquitte- ment</u> [BOOLEAN]	---	---	:Une trame :INFO demande :ACK :Une trame ACK :ne demande :pas ACK	---	OUI	OUI	
*- <u>demande d'informa- tion sur le(s) re- ceveur(s)</u> [BOOLEAN]	---	---	---	:Compteur = Nb :de stations	OUI	:Demande faite :uniquement :par le contrô- :leur	OUI (solu- :tion IBM)
*- <u>péremption</u>	---	---	---	---	---	---	OUI
*- <u>urgence</u>	---	---	---	---	---	OUI	OUI

TABLEAU 6

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
*MA.DATA.RESPONSE Paramètres :				:Primitive op- :tionnelle :les paramè- :tres associés :avec cette :primitive dé- :pendent de :l'implémenta- :tion			
<u>Transmission-status</u> :							
- <u>transmission com- plète</u> [BOOLEAN]	OUI	OUI	---		---	OUI	:Norme :IEEE 802
- <u>débordement du Nb de tentatives d'accès au médium</u> [BOOLEAN]	16	16	:8 tentatives		OUI	OUI	: 8
:P.ex. CSMA/CD → Nb de :collisions successi- :ves							
- <u>transmission time-out</u> [BOOLEAN]	OUI	OUI	:OUI (1,5 ms)	OUI	OUI	OUI	:Norme :IEEE 802
*- <u>réponse aux servi- ces demandés à travers MA.DATA. REQUEST (trans. serv.)</u>				:Compteur=Nb :st prés;erreur :de CRC → :C const. : :pas erreur ≠ C+I :de CRC : :A = { copie : C+I : saturé : C const.			

TABLEAU 6 (suite)

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
- information sur le(s) receveur(s). [ENTIER]	---	---	---		OUI	Réponse au contrôleur pendant phase d'interroga- tion	Option du type IBM
- indication d'ACK reçu [BOOLEAN]	---	---	OUI		OUI	OUI	OUI
- indication de pé- remption [BOOLEAN]	---	---	---	---	---	---	OUI
- indication d'urgen- ce [BOOLEAN]	---	---	---		---	---	OUI

U F R G S
BIBLIOTECA
CPD/RGCC

TABLEAU 7

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
*MA.DATA.INDICATE Paramètres :							
Adresse destination [Nb ent. caractères]	1 à 70 Cette @ peut être : - physique - logique .sous-group .dif.gén.	60	10	60	10	10	60 dont les 5 premiers =>@L le dernier =>@PH
Adresse-source [NB ent. caractères]	1 à 70	60	10	10	10	10	10
Data-unit :							
-destination service- access-point	10=@ du proto: cole lié au LLC qui est en train d'é- changer l'in- formation						
- source-service- access-point	10=@ du proto: cole avec le- quel on se communique						

TABLEAU 7 (suite)

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
-champ de contrôle	10= pareil à HDLC	20	10			10	10=> HDLC
-champ d'information	∇ Nb entier d'0 s		2560		32,64 ou 960	∇ Nb entier de caractères	2560
<u>Data-size</u> [Nb ent. caractères]	Comprise entre 640 et 15180 [Bornes de la trame com- plète]		Nb de la tra- me HDLC con- tient la tail- le du champ INFO en Nb d'Octets		Premier 0 de données IT5	OUI	10
<u>Réception status</u>				La primitive MA.DATA. INDICATE ne présente pas réception status			
- <u>réception complète</u> [BOOLEAN]	OUI	OUI	OUI		OUI	OUI	Norme IEEE 802
- <u>réception trop lon- gue</u> [BOOLEAN]	OUI	OUI	OUI (chien de garde)		---	OUI	Norme IEEE 802

TABLEAU 7 (suite)

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
*- forme incorrecte de la trame : pas correctement déli- mitée [BOOLEAN]	---		OUI		OUI	OUI	OUI
*- mauvais CRC [BOOLEAN]		OUI	OUI		OUI	OUI	OUI
*- pas d'égalité avec data-size [BOOLEAN] ou Nb d'octets non- entiers		OUI	Taille Nb octets incor- recte		OUI	OUI	OUI

TABLEAU 8

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
* MA.RESET.REQUEST [BOOLEAN] efface les données contenues dans le Data-unit qui sont en train d'être traitées par le MAC	OUI	---	---	OUI	---	---	Envisagé
* MA.RESET.RESPONSE [BOOLEAN] informe si l'efface- ment a pu être réali- sé ou pas Paramètre : - reset-status : [OK, PAS OK]	OUI	---	---	OUI	---	---	Envisagé
* MA.ADDRESS-SET adresse-source	---	OUI=64@ de groupe possi- bles	Changement de l'@ physique dans un regis- tre du cir- cuit program- mable 2652 RTC	---	NON:table d'adresses lo- giques stati- que (PROM)	Changement de l'@ physique dans un regis- tre du Circ. programmable SIO	OUI

TABLEAU 8 (suite)

PRIMITIVE DE SERVICE POINT A POINT	NORME IEEE 802	ETHERNET	FACTOR	IBM ECMA/TC24/ 82/201	CARTHAGE	REBUS	PROPOSITION
* MA.ADDRESS-RESET adresse-source	---	OUI	---	---	:NON:table :d'adresses lo: :giques stati- :que (PROM)	---	OUI
* MA.ADDRESS-READ adresse-source	---	OUI	:On n'a que :des adresses :physiques	---	---	---	OUI

ANNEXE 2

EXEMPLES D'ARBITRAGE DE BUS

2.1. SM 90 [OLI 83-1]

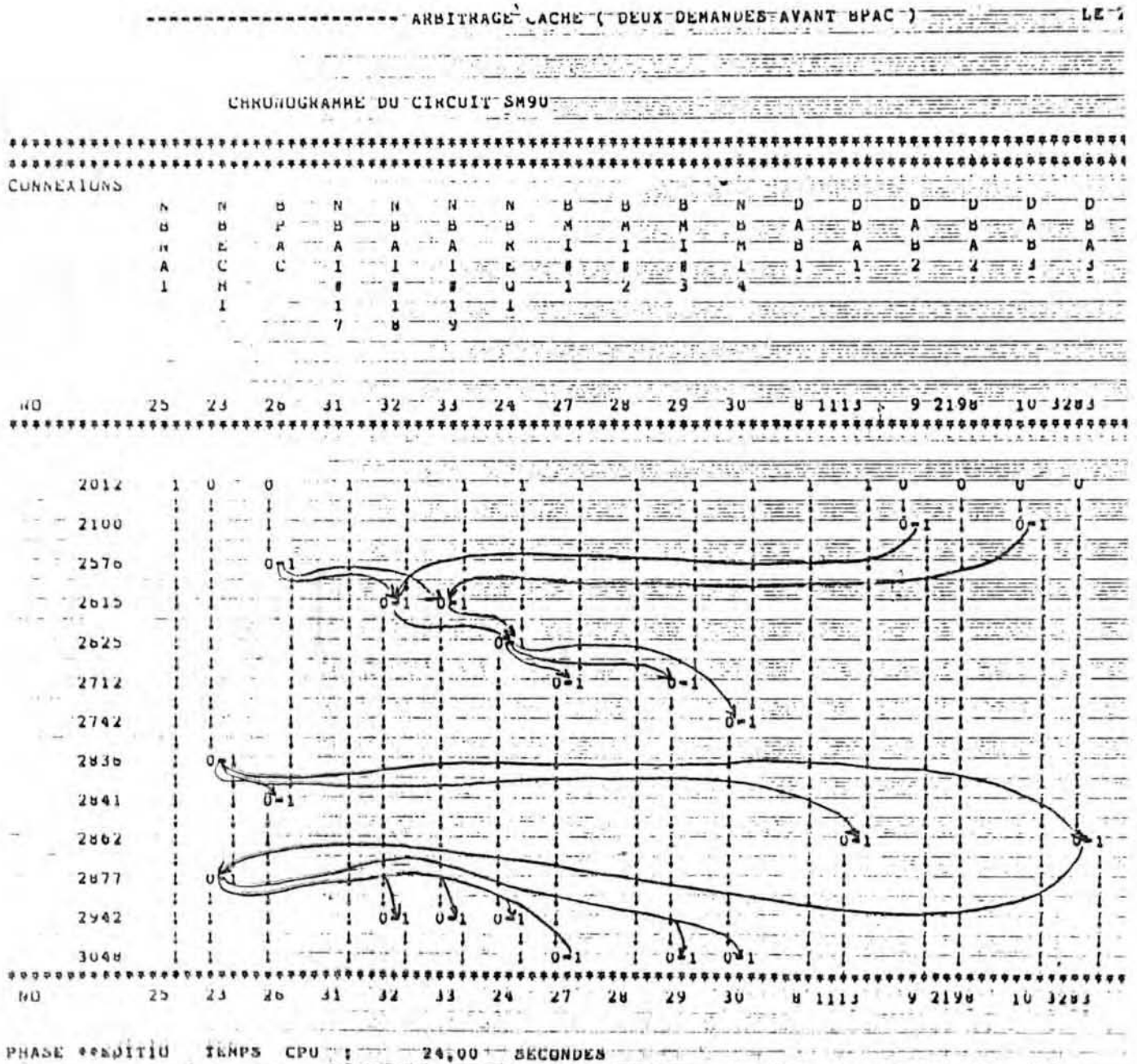


Figure A2.1 - Chronogramme EPILOG [EPI 79] d'arbitrage pour la structure SM 90.

La figure A2.1 représente un cycle d'arbitrage ou bus de la SM 90. Trois modules de traitement sont raccordés au bus d'adresses 0, 1 et 2. Celui d'adresse 0 est configuré en priorité fixe et les modules d'adresses 1 et 2 sont configurés en priorité rotative.

Dans l'exemple choisi, au début de la séquence montrée dans la figure A2.1, le module de traitement configuré à l'adresse 0 occupe le bus. Au préalable, le module de traitement d'adresse 1 a déjà effectué l'échange sur le bus. Dès que les fils BA 17-24 deviennent disponibles (ils étaient pris pour les adresses au début de l'échange, ce qui n'est pas montré sur le chronogramme), le module maître du bus à cet instant (celui d'adresse 0) active le fil BPAC (Bus Arbitrage Caché Possible), de sorte que les demandes d'accès au bus faites par les modules de traitement configurés aux adresses 1 et 2 (DAB2 et DAB3, respectivement) soient présentées sur le bus (signaux NBAI#18 et NBAI#19 sur le chronogramme).

Les signaux BAI actifs entraînent un signal de demande d'arbitrage NBREQ dont le front descendant enregistre dans tous les arbitres ABC 90 présents sur la structure, la configuration de demandes au bus.

Dû au fait, que les demandes d'accès au bus sont formulées par des modules configurés en priorité rotative, étant donné que la dernière unité rotative à prendre le bus est celle d'adresse 1, le verdict d'arbitrage (signaux BMI#1, 2 et 3) rend l'unité d'adresse 2 la plus prioritaire (demande NBAI#19) (voir § 1.2.4.).

Le verdict est validé par le front descendant du signal NBMI 4.

On remarque que le verdict a été délivré pendant l'occupation du bus, ce qui caractérise un arbitrage caché (voir § 1.5.)

Dès que le transfert effectué par le module de traitement 0 est fini (front montant de NBECHI), les arbitres du bus ABC 90 sur chaque module de traitement retransmettent le verdict à leur unité de traitement associée (cf. signaux DBA 1, DBA 2, DBA 3). De ce fait, l'unité de traitement du module de traitement (voir figure 4, § 1.2.2.) d'adresse 2, voit sa demande de bus acceptée (DBA 3 actif haut).

A partir de ce moment là, cette unité peut commencer son échange à travers le bus (front descendant de NBECHI). Ce front descendant déclenche la libération de signaux de demande de bus (NBAI), demande d'arbitrage (NBREQI) ainsi que les signaux de verdict (BMI#1, BMI#2, BMI#3) et validation du verdict (NBMI 4)

Remarque

Pour faciliter la comparaison avec le chronogramme montré, on a mis un N devant les signaux actifs bas (dans le texte du chapitre 2, on met une barre).

2.2. MULTIBUS [GIR 81]

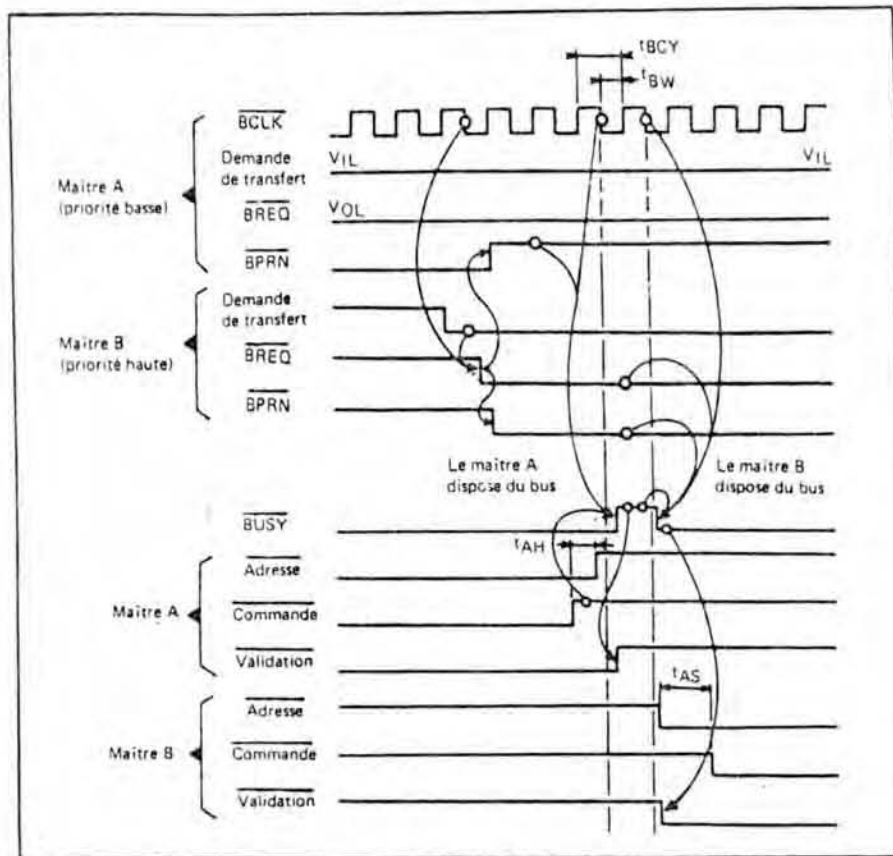


Figure A2.2 - Chronogramme de la prise de contrôle du bus par un maître plus prioritaire.

La figure A2.2 représente le mécanisme d'accès au Multibus dans un environnement multimicroprocesseur. Le mode de gestion des priorités adopté est celui de la résolution parallèle.

Néanmoins, le chronogramme dans le cas d'une résolution sérielle serait similaire à celui de la figure A2.2.

Dans l'exemple choisi, le module maître A a un niveau de priorité d'accès au bus inférieur à celui du module maître B. Le module B génère une demande d'accès au bus, pendant que le module A dispose de ce bus.

La demande d'accès au bus, formulée par le module B, est synchronisée sur le flanc descendant BCLK et donne BREQ. La broche BPRN du module A passe à l'état haut. Le module A termine le cycle de bus en cours et fait passer la broche BUSY à l'état haut, indiquant ainsi la disponibilité du bus. Le passage à l'état haut de BUSY suit le passage à l'état haut du signal de commande (MRDC, IORC, MWTC ou IOWC) et est synchronisé sur le flanc descendant de BCLK. Le passage de BUSY à l'état haut entraîne la déconnexion du module A du MULTIBUS (invalidation des drivers) et l'accès à ce bus par le module B. La prise en charge du bus par le module B est synchronisé sur BCLK et provoque la mise à l'état bas de BUSY.

Le module A peut, sous certaines conditions conserver le contrôle du MULTIBUS malgré une demande d'accès formulée par le module B. Dans ce cas, il génère un signal interne de verrouillage qui maintiendra **BUSY** à l'état bas aussi longtemps qu'il le souhaitera. Une telle caractéristique est intéressante lorsqu'un maître veut tester si une ressource est disponible et, dans l'affirmative, se la réserver (sémaphore).

2.3. VME [GIR 82]

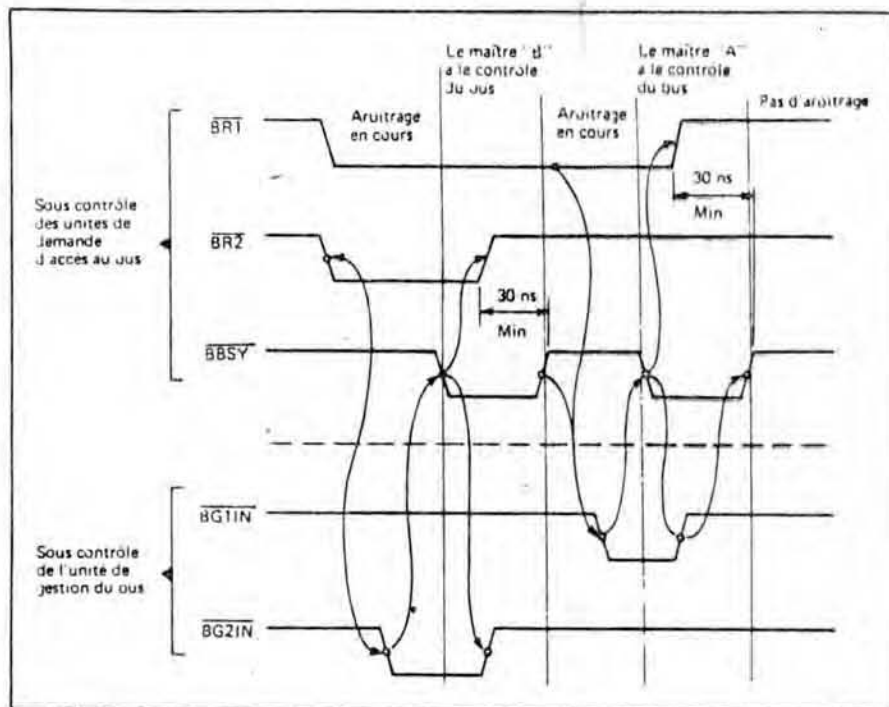


Figure A2.3 - Demande d'accès au bus par des maîtres ayant des niveaux de priorités différents.

Au début de la séquence, les unités de demande d'accès au bus du maître A et du maître B ont adressé à l'unité de gestion du bus une demande d'accès, respectivement sur les lignes $\overline{BR1}$ et $\overline{BR2}$. Supposons que ces demandes d'accès arrivent simultanément sur l'unité de gestion du bus. Celle-ci détecte l'unité de demande qui a le niveau d'accès le plus élevé (le maître B) et valide la ligne $\overline{BG2IN}$. Dès que l'unité de demande du bus du maître B reçoit l'autorisation d'accès au bus, elle porte sa broche \overline{BBSY} à l'état bas, se réservant le bus. Elle libère ensuite la ligne $\overline{BR2}$ et informe son

→ maître que le bus est à sa disposition.

Lorsque l'unité de gestion du bus détecte un niveau bas sur sa broche \overline{BBSY} (indiquant une occupation du bus), elle libère $\overline{BG2IN}$. Elle ne prendra pas en compte d'autre demande d'accès, tant que le bus ne sera pas libre. Lorsque le maître B a terminé son transfert, son unité d'accès au bus remet la broche \overline{BBSY} à l'état haut, à condition que $\overline{BG2IN}$ soit à l'état haut et qu'un temps d'au moins 30 ns se

soit écoulé entre l'instant de retour de BRZ à l'état haut et l'instant de libération du bus.

L'unité de gestion du bus, constatant que le bus est libre, l'accorde au maître A, en portant sa broche BGIN à l'état bas et la même opération que précédemment recommence.

ANNEXE 3

PROPOSITION D'UTILISATION COHERENTE DES BROCHES
DU CONNECTEUR MULTIBUS ET
DU CONNECTEUR VME

3.1. PROPOSITION D'UTILISATION COHÉRENTE DES BROCHES DU CONNECTEUR MULTIBUS

3.1.1. RAPPEL

Le dialogue entre arbitres nécessite 8 fils dont 6 ne trouvent pas de correspondants sur le connecteur MULTIBUS.

Par ailleurs, pour que l'ABC 90 puisse arbitrer, il faut qu'il dispose de la configuration de demandes (8 au maximum).

La structure MULTIBUS ne permet pas de multiplexer la configuration de demandes avec les fils d'adresses ou de données [GIR 81] :

. pendant un cycle d'écriture, les adresses et les données sont valides simultanément, ne laissant pas une fenêtre de temps pour que la configuration des demandes soit présentée sur ces fils ;

. il n'y a pas de signal correspondant à BPAC fourni par la structure Multibus, permettant l'échantillonnage des demandes sur des fils d'adresses ou de données.

De ce fait, le problème est de trouver des fils disponibles sur MULTIBUS. Une proposition d'utilisation cohérente des broches du connecteur MULTIBUS est présentée dans le tableau qui suit.

TABLEAU A3.1

TYPES DE SIGNAUX	SIGNAUX DU BUS MULTIBUS	SIGNAUX DE L'ARBITRE ABC 90 ;OLI 83-1;
Ligne de service	INIT (initialisation)	\overline{RZ} (remise à zéro) Tout à fait pareil à INIT. On relie toutes les broches INIT du MULTIBUS (\overline{RZ} de l'ABC 90 relié à INIT)
Lignes de gestion d'accès au Bus	<p>\overline{BREQ} (demande d'accès au bus) Cette broche de chaque module maître (jusqu'à 8) est reliée à une entrée spécifique de l'arbitre (broches BA 17-24) ce qui spécifie le niveau de priorité de chaque maître. Cette broche joue le rôle d'une ligne BA_i dans l'ABC 90.</p> <p>\overline{CBRQ} (demande d'un module moins prioritaire) Cette broche sert à signaler au maître du bus qu'un autre maître moins prioritaire est en attente.</p> <p>\overline{BPRN} (validation d'accès au bus) Suivant le même ordre qu'on a relié \overline{BREQ} à l'arbitre on relie les broches \overline{BPRN} à la sortie du décodeur externe de priorités.</p>	<p>BA 17-24 (configuration des demandes) Pour cette configuration des demandes, l'ABC 90 a besoin jusqu'à 8 broches. On peut prendre soit parmi les 24 broches d'adresse (réduction de l'espace d'adressage), soit parmi les 16 fils de données (restriction à l'emploi des 16 bits). Plus le nombre de broches prises est petit (nombre plus réduit de modules maîtres standards et spéciaux en tout), plus grand sera l'espace d'adressage des cartes spéciales contenant l'arbitre.</p> <p>L'ABC 90 n'a pas de signal semblable à \overline{CBRQ} et il n'est pas pris en considération par l'ABC 90. Donc cette broche est disponible. \overline{BREQ} (demande d'arbitrage). On peut prendre la broche \overline{BREQ}.</p> <p>$\overline{BM1-3}$, $\overline{BM4}$ (verdict et validation du verdict) Ces broches pour les cartes spéciales sont configurées en entrée/sortie et sont en aval par rapport à l'encodeur. Du fait que sur les cartes spéciales on n'a pas besoin de \overline{CBRQ}, \overline{BPRN}, \overline{BCLK} et \overline{BPRQ}, on les utilise respectivement en connexion avec $\overline{BM1}$, $\overline{BM2}$, $\overline{BM3}$ et $\overline{BM4}$.</p>

TABLEAU A3.1 (suite)

TYPES DE SIGNAUX	SIGNAUX DU BUS MULTIBUS	SIGNAUX DE L'ARBITRE ABC 90 (OLI 83-1)
Lignes de gestion	<p><u>BUSY</u> (occupation du MULTIBUS) Quand un maître dispose du MULTIBUS, il le signale en validant la ligne BUSY</p>	<p><u>BECH</u> (échange en cours) Ce signal est tout à fait pareil à BUSY et est utilisé en entrée/sortie. On relie donc toutes les broches BUSY du MULTIBUS (broches BECH de l'ABC 90 relié à BUSY).</p>
d'accès au Bus	<p><u>BCLK</u> (horloge du MULTIBUS) Ce signal sert à synchroniser (front descendant) la demande de transfert BREQ associée.</p>	<p>Du fait que l'ABC 90 est asynchrone, on n'a pas besoin de lui fournir BCLK. Donc, la broche BCLK du connecteur MULTIBUS devient disponible.</p>
	<p><u>BPRO</u> (validation série d'accès au bus) Cette broche sert à valider ou inhiber le module de priorité inférieure (gestion série des priorités).</p>	<p><u>BNA</u> (indication de nouvel arbitrage) On peut prendre soit le signal de FIN de transfert avancé <u>AACK</u> (spécifique du 8080), soit un fil d'adresse.</p>

3.1.2. CORRESPONDANCE ENTRE LES BROCHES DU CIRCUIT ABC 90 AVEC CELLES DU CONNECTEUR MULTIBUS

D'après ce tableau, la correspondance entre les broches du circuit ABC 90 avec celles du connecteur MULTIBUS est donnée dans le tableau A3.2 qui suit :

TABLEAU 'A3.2

CARACTERISTIQUES DE CORRESPONDANCE BROCHES ABC 90 BROCHES CONNECTEUR MULTIBUS	BROCHES DE L'ABC 90	BROCHES DU CONNECTEUR MULTIBUS
: Broches du connecteur VME dont on n'a : pas besoin de redéfinir leur signifi- : cation.	: RZ	: INIT
	: BECH	: BUSY
: Broches disponibles sur le connecteur : MULTIBUS des cartes spéciales contenant : l'ABC 90.	: BNA	: BREQ
	: BM1	: CBRQ
	: BM2	: BPRN
	: BM3	: BCLK
	: BM4	: BPRO
: Broches du connecteur qu'on a besoin : de redéfinir.	: BA _i ($2 \leq i \leq 8$)	: ($2 \leq i \leq 8$) bro- : ches d'adres- : se ou de : données

Remarque :

Le standard MULTIBUS normalise le connecteur. L'implantation du bus fond de panier (lequel peut être réalisé soit par une carte imprimée, soit par "wrapping") est laissée au gré de l'utilisateur.

Pour les broches du connecteur MULTIBUS dont on a besoin d'enlever la signification d'origine et d'en proposer une autre (groupe 3 du tableau précédent) on a le choix entre :

. prendre les i fils manquants parmi les 24 d'adresse (si on compte les 4 bits supplémentaires introduits par le connecteur P2), ce qui entraîne une perte de l'espace d'adressage.

. ou prendre statiquement 8 des 16 fils de données ce qui interdit l'utilisation de microprocesseurs 16 bits sur ces modules.

Remarque :

L'utilisation compatible de l'ABC-M exige en plus des broches nécessaires à l'ABC 90, dont on vient de discuter, un nombre de broches correspondant au nombre de cartes standards MULTIBUS contenant le 8289 (chaque bit NS de l'ABC-M correspond à BPRN de chaque carte).

On aura donc besoin de i broches, étant i égal au nombre de cartes standards MULTIBUS présentes.

3.2. PROPOSITION D'UTILISATION COHÉRENTE DES BROCHES DU CONNECTEUR VME

3.2.1. RAPPEL

L'analyse des échanges de la structure VME [GIR 82], et comme le bus VME n'accepte que des arbitrages visibles, on a été amenés à choisir pour la configuration de demandes de l'arbitre (BA 17-24) une configuration démultiplexée.

Cela veut dire qu'on doit prendre jusqu'à 8 broches parmi les broches disponibles du connecteur VME à cet effet. Cette condition n'est pas tellement contraignante pour le VME, car ce bus dispose d'un nombre assez important de fils appartenant à la famille 2d (voir § 1.5.1.)

Une proposition d'utilisation cohérente des broches du connecteur VME est donnée dans le tableau A3.3 suivant.

TABLEAU A3.3

TYPES DE SIGNAUX	SIGNAUX DU BUS VME	SIGNAUX DE L'ARBITRE ABC 90
Lignes de service	<p>$\overline{\text{SYSRESET}}$ (initialisation)</p>	<p>$\overline{\text{RZ}}$ (remise à zéro) Tout à fait pareil à $\overline{\text{SYSRESET}}$. On relie en <u>ET CABLE</u> toutes les broches $\overline{\text{SYSRESET}}$ et $\overline{\text{RZ}}$.</p>
service	<p>$\overline{\text{ACFAIL}}$ (coupure de l'alimentation "secteur") Ce fil est activé lorsqu'une chute d'alimentation du secteur a été détectée.</p>	<p>Il n'y a pas de correspondance chez l'arbitre et ce fil $\overline{\text{ACFAIL}}$ ne concerne que l'unité de traitement associée à l'arbitre.</p>
Lignes de gestion d'accès au Bus	<p>$\overline{\text{BCLR}}$ (demande d'un module plus prioritaire) Cette broche est supprimée dans le mode avec priorité tournante.</p>	<p>Il n'y a pas de correspondance avec la broche $\overline{\text{BCLR}}$.</p>
Lignes de gestion d'accès au Bus	<p>$\overline{\text{BR3}} - \overline{\text{BR0}}$ (demande d'accès au bus) Chacune de ces quatre broches a un niveau de priorité différent. Sur chacune de ces lignes, on peut raccorder d'autres modules à travers le "daisy chain" : broches $\text{BG3 OUT} - \text{BG0 OUT}$.</p>	<p>$\text{BA17} - 24$ (configuration des demandes) Pour cette configuration des demandes, l'ABC 90 a besoin jusqu'à 8 broches. On peut prendre aisément les 4 broches de demande ($\overline{\text{BR3}} - \overline{\text{BR0}}$) et les 4 broches d'autorisation d'accès au bus en série ($\text{BG3 OUT} - \text{BG0 OUT}$) ; on a alors 8 niveaux de demandes VME : $\overline{\text{BR3}}$, $\overline{\text{BR0}}$ et $\overline{\text{BR3'}}$, $\overline{\text{BR0'}}$.</p> <p>$\overline{\text{BREQ}}$ (demande d'arbitrage) On peut prendre un fil parmi les fils d'extension sur le 2ème connecteur d'extension à 32 bits et 4 gigaoctets de mémoire (32 bits).</p>

TABLEAU A3.3 (suite)

TYPES DE SIGNAUX	SIGNAUX DU BUS VME	SIGNAUX DE L'ARBITRE ABC 90
Lignes de gestion	<p>$\overline{BG3\ IN} - \overline{BG0\ IN}$ (autorisation d'accès au bus) Une demande d'accès au bus \overline{BRx} est accordée par l'activation de la broche $\overline{BGx\ IN}$ correspondante</p>	<p>$\overline{BM1-3}$, $\overline{BM4}$ (verdict ($\overline{BM1-3}$) et validation du verdict ($\overline{BM4}$)) On prend les broches $\overline{BG3\ OUT} - \overline{BG0\ OUT}$ pour divulguer le verdict et le valider.</p>
d'accès au Bus	<p>\overline{BBSY} (occupation du VME) Cette broche est portée à l'état bas dès qu'un maître prend en charge le bus VME.</p>	<p>\overline{BECH} (échanges en cours) Ce signal est tout à fait pareil à \overline{BBSY} et est utilisé en E/S. On fait donc un <u>OU CABLE</u> des broches \overline{BBSY} et \overline{BECH}.</p>
		<p>\overline{BNA} (indication de nouvel arbitrage) Ce signal est activé lorsqu'un temps réglé par une capacité externe KP s'écoule. Comme pour \overline{BREQ} on peut prendre aussi un fil parmi les 24 fils d'adresse sur le connecteur P1 ou parmi les 8 d'extension sur P2.</p>

3.2.2. CORRESPONDANCE ENTRE LES BROCHES DU CIRCUIT ABC 90 AVEC CELLES DU CONNECTEUR VME

D'après ce tableau, la correspondance entre les broches du circuit ABC 90 et celles du connecteur MULTIBUS sont données dans le tableau A3.4 qui suit :

TABLEAU A3.4

CARACTERISTIQUES DE CORRESPONDANCE BROCHES ABC 90 - BROCHES CONNECTEUR VME	BROCHES DE L'ABC 90	BROCHES DU CONNECTEUR VME
Broches du connecteur dont on n'a pas besoin de redéfinir la signification.	RZ	SYSRESET
	BECH	BBSY
	BA _i (i ≤ 4)	BA3 - BRØ
Broches disponibles sur le connecteur VME des cartes spéciales contenant l'ABC 90	BM1	BGØ OUT
	BM2	BG1 OUT
	BM3	BG2 OUT
	BM4	BG3 OUT
Broches du connecteur qu'on a besoin de redéfinir.	BNA	BCLR (bro- che disponi- ble dans le mode avec priorité tournante)
	BA _i > 4	Fils d'adres- ses ou de données

On constate donc que jusqu'à 4 unités de traitement connectées au bus, il n'y a pas besoin de redéfinition des broches du connecteur, à l'exception près de BNA. Si les 4 unités de traitement sont configurées en priorité rotative, la broche BCLR devient disponible et on fait correspondre BNA et BCLR. Donc l'utilisation de l'ABC 90 avec la structure VME (architecture décentralisée) se fait sans aucun besoin de dégradation de potentialités (la redéfinition des broches du

connecteur MULTIBUS peut entraîner la perte de l'espace d'adressage ou une réduction de la longueur du mot).

Remarque :

L'utilisation compatible de l'ABC-M exige en plus des broches nécessaires à l'ABC 90, dont on vient de discuter, un nombre de broches correspondant au nombre de cartes standards VME (chaque bit NS de l'ABC-M correspond à BGxIN de chaque carte).

ANNEXE 4

COMPATIBILITE ELECTRIQUE

4.1. COMPATIBILITÉ ÉLECTRIQUE DE L'ABC 90

Toutes les propositions qu'on vient de présenter ne seraient valables qu'à partir du moment où la compatibilité électrique entre les signaux de l'arbitre et ceux du bus VME ou MULTIBUS est assurée.

Parmi les solutions présentées, les plus contraignantes du point de vue électrique sont celles où l'on fait un arbitrage décentralisé (§ 2.3.3.2. et 2.4.3.2.), car on a à "driver" un nombre de portes plus important.

Pour les solutions centralisées, la compatibilité électrique est plus souple car on peut mettre éventuellement des drivers, soit de sortie, soit d'entrée, à la carte contrôleur du système où sont placés le(s) arbitre(s). Pour les solutions décentralisées, dès qu'on fait face à une charge très importante à driver, on est amené à utiliser des portes TTL LS qui consomment moins.

Voici quelques caractéristiques fondamentales à respecter pour avoir une compatibilité électrique : [INT 81], [MOT], [OLI 83].

- . BREQ (entrée : Trigger de Schmitt ;
sortie : Drain Ouvert)

A cette broche de l'ABC 90, on relie pour toutes propositions présentées une porte AND 8 entrées des demandes d'accès au bus. Comme cette broche a une sortie drain ouvert, il faut que la sortie du AND soit aussi en collecteur ouvert.

- . BM1 à 3 (entrée : Trigger de Schmitt ;
sortie : Drain Ouvert)

Pour toutes les propositions, ces broches sont communes à tous les arbitres et à l'entrée du décodeur directement (architecture centralisée) ou indirectement INVERSEURS CMOS (architecture décentralisée). Pour celui-ci, on suggère le décodeur 3 x 8 (SN 54 LS 138) car sa consommation en courant est très réduite (pour $V_{IL} = 0,4$ V on a $I_{IL} = 0,4$ mA) tandis que le décodeur TTL standard SN 54 S 138 en consomme le quintuple (pour $V_{IL} = 0,5$ V on a $I_{IL} = -2$ mA).

Comme la sortie de ces broches est en collecteur ouvert, cela va imposer que la sortie de l'encodeur, pour les architectures décentralisées, se présente aussi en collecteur ouvert, ce qui représente une porte supplémentaire.

- . BM4 (entrée : Trigger de Schmitt ;
sortie : Drain Ouvert)

Pour les propositions décentralisées, comme on l'a dit au § 2.4.3.2., il faut que le signal de validation du verdict publié par l'encodeur, SB, se présente aussi en collecteur ouvert pour qu'on puisse faire le OU CABLE du signal SBRP avec BM4.

- . BNA (entrée : Trigger de Schmitt ;
sortie : Drain Ouvert)

Cette broche relie uniquement les arbitres, donc il n'y a aucun problème de compatibilité électrique vis-à-vis de cette broche.

- . BAi (entrée : Trigger de Schmitt ;
sortie : Drain Ouvert)

Comme on a pu le constater dans les propositions présentées, on relie à ces broches des demandes faites par les modules arbitres et celles faites par les modules standards, comme le VME ou le MULTIBUS. Les broches de demande du VME (BRO-BR3) sont aussi configurées en Drain Ouvert, tandis que celles pour le MULTIBUS (BPRN) sont du type TTL standard ($IOLMIN = 16 \text{ mA}$, $IOHMIN = -400 \text{ mA}$, $COMIN = 300 \text{ PF}$), qui n'exigent donc pas des résistances de terminaison. Donc, on doit lier à chaque demande formulée par une unité de traitement MULTIBUS (BREQ) une porte sortie collecteur ouvert.

La charge même, dans le pire des cas (1 module standard + 7 cartes spéciales), le driver VME collecteur ouvert ($A VOL = 0,7 \text{ V}$ on fournit $IOL = 40 \text{ mA}$), ou le driver MULTIBUS TTL standard, sont capables de tenir.

- . BECH (entrée : Trigger de Schmitt)

Cette broche ne concerne pas l'arbitre car elle est configurée en entrée seulement, mais plutôt l'unité de traitement associée aux arbitres, car c'est à cette unité d'activer BECH dès qu'elle prend possession du bus.

Donc, le driver de sortie de la broche BECH à la sortie de chaque unité de traitement associée à un arbitre, doit pouvoir driver dans le pire des cas, soit 7 modules VME (broche BBSY consomme -400 mA à $0 < V$ entrée $< 0,5$), soit 7 modules MULTIBUS (broche BUSY consomme -2 mA à l'état bas) plus une ou plusieurs portes AND selon la logique supplémentaire proposée. Les solutions tout à fait décentralisées sont plus contraignantes pour ce driver (on peut avoir une porte AND associée à chaque module contenant l'arbitre) tandis que les solutions centralisées n'exigent qu'une seule porte AND associée à BECH.

On rappelle que, étant donné que ce driver est situé sur l'unité de traitement, son dimensionnement sur mesure n'est pas un problème, car il va être placé sur une carte spéciale.

On doit aussi considérer le cas inverse, c'est-à-dire un module MULTIBUS ou un module VME doivent être capables de driver jusqu'à 7 arbitres. Toutefois, ce n'est pas un problème, car chaque arbitre n'a besoin de que $50 \mu\text{A}$ ($I IL$) ce qui peut être largement fourni par les drivers en collecteur ouvert VME ou MULTIBUS.

4.2. COMPATIBILITÉ ÉLECTRIQUE DE L'ABC-M

Les mêmes considérations faites en 4.1. sont valables. La seule différence par rapport à la compatibilité électrique de l'ABC 90, c'est l'adjonction de 10 broches, dont 8 correspondent à des signaux véhiculés par le bus.

Les broches présentées en plus par l'ABC-M sont les suivantes : NS 1-8 (sortie : Drain Ouvert).

Ces broches correspondant au verdict décodé en un seul bit parmi 8, vont être reliées directement, soit à BPRN de chaque carte standard MULTIBUS, soit à BGxIN de chaque carte standard VME.

Toutes ces sorties sont du type Drain Ouvert (pour permettre la divulgation simultanée par tous les ABC-M présents sur la structure).

Les plots de sortie NS ont été dimensionnés pour une charge de 200 pF.

. MF1, MF2 (entrées : normales)

Ces broches sont câblées de façon identique sur chaque ABC-M et n'entrent pas dans l'enjeu de compatibilité électrique.

. RZ (entrée : Trigger de Schmitt)

Cette broche n'intéresse l'arbitre qu'en entrée. Alors le problème de driver le courant nécessaire est laissé au niveau de l'unité de traitement, comme on vient de l'expliquer pour la broche BECH.

Le MULTIBUS (broche INIT) comme le VME (broche SYSRESET) présentent cette broche en collecteur ouvert.

. BPAC (entrée : Trigger de Schmitt)

Cette broche est tout à fait spécifique des arbitres et donc même si elle était utilisée, elle ne poserait pas de problème de compatibilité électrique. Toutefois, comme on l'a dit, cette broche est câblée dans chaque arbitre, soit à 0 (arbitrage caché impossible), soit à 1 (arbitrage caché toujours possible) selon l'utilisation envisagée du bus.

. DAB, DBA (entrée : normale ;
sortie : push-pull, respectivement)

Ces broches ne concernent que l'interface entre l'arbitre et son unité de traitement associée, n'entrant donc pas dans l'enjeu de compatibilité électrique.

De même, les broches de l'arbitre UC1 - 3, KP, VDD et GND ne posent aucune contrainte pour la compatibilité électrique.

On rappelle qu'à toutes les sorties Drain Ouvert de l'arbitre on doit raccorder une résistance de terminaison.

ANNEXE 5

OUTILS DE CAO UTILISES
POUR LA VALIDATION ET IMPLANTATION
DE CIRCUITS INTEGRES

UFRGS
BIBLIOTECA
CPD/PGCC

5.1. INTRODUCTION

Dans cette annexe, on se propose de montrer les outils de conception assistée par ordinateur (CAO) utilisés dans ce travail.

Ces outils appartiennent à deux familles principales :

. Outils de validation, lesquels servent à démontrer à travers des simulations que les architectures proposées discrètes (voir ch. 2) ou intégrées (voir ch. 3) sont valables.

. Outils d'implantation et de vérification, lesquels servent à décrire sur Silicium, des circuits décrits par portes ou par transistors.

Dans le cadre de notre travail, on a utilisé des outils de cette famille pour l'implantation des modifications apportées à l'ABC 90 pour le transformer en circuit multiprotocoles (voir § 3.3.)

5.2. TYPES D'OUTILS DE CAO EMPLOYÉS

Lorsque l'on est amené à concevoir des circuits intégrés, on est obligé d'utiliser des outils de conception assistés par ordinateur (CAO) dans les différentes étapes de conception car ils permettent une rapidité de conception ainsi qu'une fiabilité accrue [ETI 83].

Ces outils peuvent être classés de la manière suivante :

5.2.1. OUTILS DE VALIDATION

5.2.1.1. LANGAGES ET SIMULATEURS FONCTIONNELS

Conçus pour aider la définition architecturale du circuit ainsi que pour permettre la validation fonctionnelle.

Dans le cadre de notre travail, on a utilisé le simulateur logico-fonctionnel FIDEL [HAZ 84].

5.2.1.2. SIMULATEURS LOGIQUES

Leur rôle est de permettre la vérification de synthèse, d'analyser les temps de propagation ainsi que de simuler les pannes.

Parmi ces simulateurs logiques, on distingue EPILOG [EPI 79] développé par THOMSON-EFCIS et disponible au CNET, lequel a été utilisé dans ce travail.

5.2.1.3. SIMULATEURS ÉLECTRIQUES

Ils servent à valider électriquement (temps de propagation, capacité que le circuit simulé peut attaquer, ...) le circuit conçu. Les simulateurs électriques sont utilisés plutôt dans les cas où en plus de la validation logique (on peut simuler des centaines de portes), on est dans un environnement électrique délicat (charge qu'on peut attaquer, fréquence maximale d'utilisation, ...). On n'utilise donc les simulateurs électriques que pour un nombre pas trop élevé de transistors (de l'ordre des dizaines, en général).

Dans le cadre de notre travail, on a utilisé le simulateur électrique performant ELDO [HEN 84] développé au CNET.

5.2.2. OUTILS D'IMPLANTATION ET DE VÉRIFICATION

Ces outils sont utilisés pour l'implantation de circuits intégrés. Ces outils comportent des logiciels graphiques interactifs sur des écrans de haute définition, et permettent de digitaliser le dessin du circuit (forme symbolique ou réelle (au micron)), de stocker les cellules, de les corriger et les rappeler pour les connecter ensemble.

Parmi ces outils, il y a aussi ceux qui permettent la vérification des règles de dessin (DRC : DESIGN RULES CHECKING).

Comme outils de dessin de circuit, on a utilisé - pour les parties logiques qu'on a été amené à introduire dans le cadre de la conception de l'arbitre ABC-M - le langage CRASH [ROS 82] qui permet un dessin à partir d'un symbolisme de type STICK [MEA 80].

On remarque que, dans le cadre de la conception de l'ABC 90, l'outil d'implantation symbolique le plus utilisé a été le MDMOS [CON 82].

L'implantation des parties en MDMOS et d'autres en STICK pose des problèmes d'interfaçage.

Une fois le dessin symbolique fait, on doit réaliser la vérification des règles de dessin symbolique à travers un programme de DRC. Dans le cadre de notre travail, on a utilisé le programme de vérification des règles de dessin de circuits décrits par CRASH [BON 84].

Tous les programmes qu'on vient de citer sont disponibles sur la base de données CASSIOPEE [BEY 82] (voir § 5.2.3.) disponible sur l'ordinateur VAX 780 du CNET.

Dès qu'on obtient une implantation correcte (pas d'erreurs détectées par le DRC-CRASH), comme le circuit a été décrit par des symboles, on doit disposer d'un outil de CAO qui génère les masques de circuits intégrés (implantation réelle au micron).

Cet outil n'est pas disponible pour l'instant sur la base CASSIOPEE. Toutefois, il l'est sur l'ordinateur CALMA.

Sur cet ordinateur, on dispose aussi d'un programme de DRC qui vérifie les règles de dessin au micron.

Dans la suite de cette annexe, on va décrire les outils utilisés dans le cadre de notre travail :

5.3. OUTILS DE VALIDATION

5.3.1. FIDEL [HAZ 84]

La simulation fonctionnelle peut être regardée comme une solution aux problèmes de grande taille, mémoire et temps CPU importants par rapport à la simulation logique, surtout pour des circuits complexes.

La simulation fonctionnelle permet la simulation de modèles qui correspondent à des fonctions logiques semblables, comme un multiplexeur, un décodeur, un registre, etc ...

Dès qu'on a à décrire un modèle en FIDEL, on peut le regarder comme une "boîte noire" avec ses entrées, sorties et états internes.

5.3.1.1. DESCRIPTION D'UN MODÈLE EN FIDEL

FIDEL se présente comme un langage fonctionnel de haut niveau.

La description d'un modèle en FIDEL présente trois parties distinctes :

. En-tête du modèle avec liste de paramètres d'entrée et sortie.

. Déclarations des variables où on spécifie les attributions (entrée ou sortie), taille de vecteur, des variables déclarés à l'en-tête du modèle ainsi que les initialisations et attributions (entières) des variables internes.

En outre, on peut initialiser les ratios du modèle. Les entrées ne peuvent pas être initialisées car elles sont délivrées par EPILOG.

. Partie algorithmique où on spécifie l'algorithme qui représente la fonctionnalité du modèle comme une relation entre les paramètres d'entrée, les conditions qui doivent être satisfaites et les événements en sortie à être générés.

Il faut remarquer que cette partie algorithmique peut prendre en compte des événements simultanés (parallélisme : PAR-ENDPAR dans le modèle) ainsi que les contraintes temporelles comme temps de set up, de maintien (hold) et largeur minimum de pulse.

5.3.1.2. INTERFACE FIDEL-EPILOG

A partir d'une description FIDEL d'un modèle, le compilateur FIDEL décrit en PASCAL génère un code exécutable, lequel sera interprété par le simulateur EPILOG. A cet effet, un interpréteur FIDEL décrit en FORTRAN a été incorporé au simulateur hôte.

De cette manière, le simulateur appelle le modèle dans un des cas de figure suivants :

- modifications des paramètres d'entrée du modèle, ce qui génère des nouveaux évènements en sortie.
- évènements temporels internes, ce qui entraîne une mise à jour des variables en question.

5.3.2. EPILOG

EPILOG [EPI 79] est un ensemble de programmes de simulation de circuits logiques. Il a été développé par la Société THOMSON-EFCIS.

La description d'un circuit à simuler se fait à partir des éléments standards (portes ET, OU, ET NON, OU NON, monostable, etc ...) reconnus par le langage de description, ou des sous-circuits (combinaison des éléments standards) stockés en bibliothèque privée ou bibliothèque publique.

Pour augmenter la précision des résultats, l'utilisateur peut choisir le niveau de finesse de la modélisation temporelle (pas de retard, retards typiques, retards typiques de montée et de descente, retards minimaux et maximaux de montée et de descente, retards fixes ou variables en cours de simulation).

Pour la description d'un circuit, l'utilisateur peut manipuler les entités suivantes :

- * CIRCUIT : le circuit en cours de traitement (description, simulation ou édition).
- * BLOC : circuit déjà stocké en bibliothèque et utilisé pour la description d'un circuit.
- * MODULE : élément standard du langage de description (portes, bascules élémentaires par exemple).
- * MODELE : élément logique particulier décrit par l'utilisateur sous forme d'algorithme (sous-programme FORTRAN).
- * CONNEXION : équipotentielle ou branche d'équipotentielle du circuit identifié par un nom.

Le circuit une fois décrit est stocké dans la bibliothèque privée de l'utilisateur.

Il faut ensuite définir la séquence d'entrée à appliquer au circuit.

Enfin, il faut choisir le mode d'édition des résultats qui dépendent du type de simulation.

Les types de simulation offerts par EPILOG sont :

- simulation logique utilisant trois valeurs 0, 1 et X, X rendant compte des indéterminations dues :
 - . à l'initialisation,
 - . à l'application de configurations ou de séquences interdites sur certains éléments.
- simulation fine utilisant quatre valeurs 0, 1, Ø et X, Ø rendant compte des indéterminations de transition d'un signal passant de 0 à 1, ou de 1 à 0 avec une certaine incertitude.
- simulation de défauts utilisant trois valeurs 0, 1 et X, et servant à déterminer l'efficacité d'une séquence d'entrée vis-à-vis de la détection d'un ensemble de défauts susceptibles d'affecter le circuit.

C'est la simulation logique que l'on a utilisée dans ce travail.

5.3.3. ELDO

ELDO [HEN 84] est un simulateur électrique très performant permettant l'analyse de circuits MOS complexes. Dû au fait que les simulations électriques ELDO sont très intéressantes à l'égard du temps CPU, par rapport au simulateur électrique SPICE (aussi disponible au CNET), il permet la simulation de circuits contenant un nombre assez large de transistors (des circuits de plus de 1 000 transistors ont déjà été simulés).

5.4. OUTILS D'IMPLANTATION, VÉRIFICATION ET GÉNÉRATION DE MASQUES

5.4.1. MÉTHODE DE CONCEPTION SYMBOLIQUE CRASH

CRASH [ROS 82] (Conception Rapide Aidée par un Symbolisme Habile) est une méthode de conception de circuits intégrés adaptée à la technologie NMOS seulement.

Cette méthode permet le dessin rapide du circuit et un calcul rapide des caractéristiques électriques et temporelles du bloc considéré. De plus CASSIOPEE, permet à partir d'une description graphique d'un circuit type au sens CASSIOPEE, une génération automatique de la description élec-

trique du circuit servant au simulateur électrique performant ELDO [HEN 84].

Les symboles employés par CRASH sont simples et proches de la description électrique du circuit. L'accroissement de taille des blocs se situe autour de 10 %, si on compare avec une implantation au micron.

5.4.2. MÉTHODE DE CONCEPTION SYMBOLIQUE MDMOS

L'avantage premier de l'implantation symbolique MDMOS [CON 82], pour la conception de circuits intégrés, est la réduction de l'encombrement dû aux transistors lors de la constitution de portes élémentaires (NAND ou NOR).

De même, l'implantation des portes en est très facilitée.

La porte MDMOS n'a aucune fonction logique élémentaire (NAND ou NOR) par elle-même. Elle n'est que la contraction dans une même unité des transistors appartenant à des portes élémentaires différentes et ayant leur grille commune.

Ceci permet une modulation de la porte et une implantation en ligne.

On ne retrouve la structure de porte élémentaire que par regroupement d'éléments de portes MDMOS distinctes.

5.4.3. OUTIL DE VÉRIFICATION DES RÈGLES DE DESSIN DRC

Le DRC symbolique STICK dans CASSIOPEE [BON 84] est l'implantation directe des règles de garde en symbolisme CRASH.

Les règles technologiques sont donc vérifiées en technologie XMOS1 d'EUROTECHNIQUE.

On doit remarquer que le DRC STICK CASSIOPEE est différent du DRC CALMA. Or celui-ci travaille sur des règles exprimées en micron, alors que le DRC STICK travaille sur un pas LAMBDA de 3.25 microns ou un pas de TETA de 3.75 microns. De cette façon, le DRC STICK majore les règles, ce qui est contraignant.

D'autre part, les notions d'équipotentiels, connexions, transistors qui apparaissent dans un DRC symbolique n'apparaissent pas dans CALMA qui ne connaît que des masques.

5.4.4. OUTILS DE GÉNÉRATION DE MASQUES ET VÉRIFICATION GLOBALE

Comme on l'avait dit en 5.2.2., pour le moment CASSIOPEE (dont les outils de description de circuit sont représentés dans la partie haute de la figure A5.1) ne permet pas de terminer tout le processus de conception, car la génération automatique de dessin au micron (GENERE sur figure A5.1), le DRC micron, ainsi que la génération de bandes magnétiques pour les masqueurs sont réalisés par le système graphique CALMA (représenté dans la partie basse de la figure A5.1). L'interface CASSIOPEE-CALMA se fait par le programme GENCAL.

La figure A5.1, ci-dessous, montre donc les étapes de la conception (simulations de validation non comprises) d'un circuit intégré au CNET GRENOBLE.

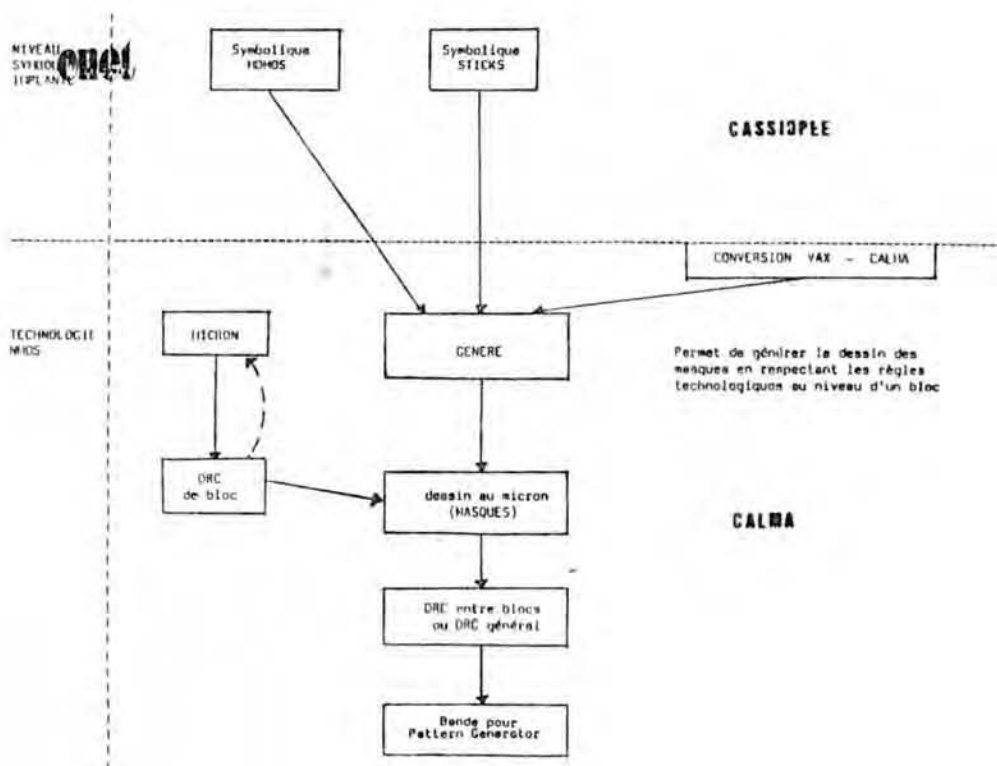


Figure A5.1 - Chaîne de conception au CNET.

5.5. EXEMPLE DE CONCEPTION D'UN MULTIPLEXEUR

A la figure A5.2, on montre en guise d'exemple la suite d'étapes de conception d'un circuit intégré, en l'occurrence le multiplexeur employé dans la logique supplémentaire de divulgation du verdict de forme décodé (voir § 3.3.) :

- . Figure A5.2-a : schéma logique du multiplexeur.
- . Figure A5.2-b : schéma électrique du multiplexeur.
- . Figure A5.2-c : implantation STICK du multiplexeur.
- . Figure A5.2-d : masques du multiplexeur.
- . Figure A5.2-e : photographie du multiplexeur (côté droit de la figure où sont montrés 3 multiplexeurs).
- . Figure A5.2-f : photographie d'un montage pour mesurer le temps de propagation du multiplexeur en fonction d'un changement du signal de sélection.
- . Figure A5.2-g : description électrique du circuit simulateur ELDO.
- . Figure A5.2-h : résultat de simulation ELDO.
- . Figure A5.2-i : test statique sous pointes du multiplexeur.
- . Figure A5.2-j : réponse dynamique du multiplexeur à 20 MHz, à une variation du signal de sélection (entrées A et B constantes).

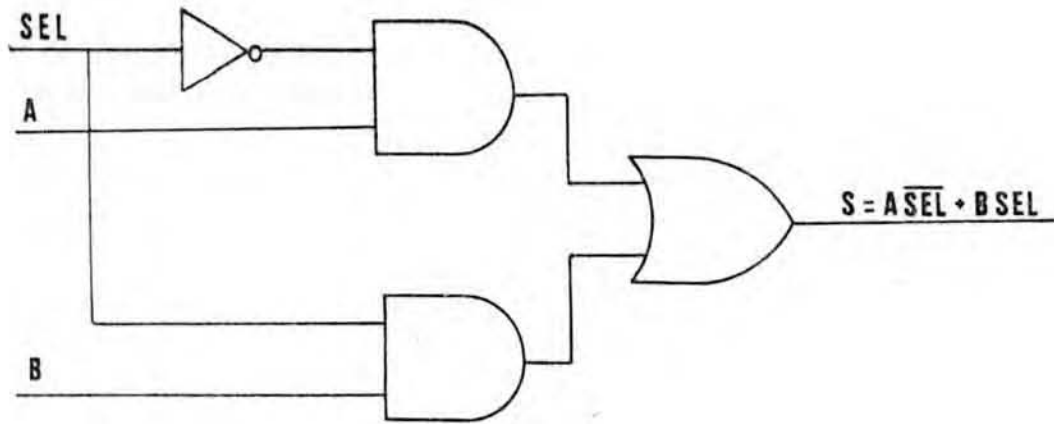


Figure A5.2-a - Schéma logique du multiplexeur.

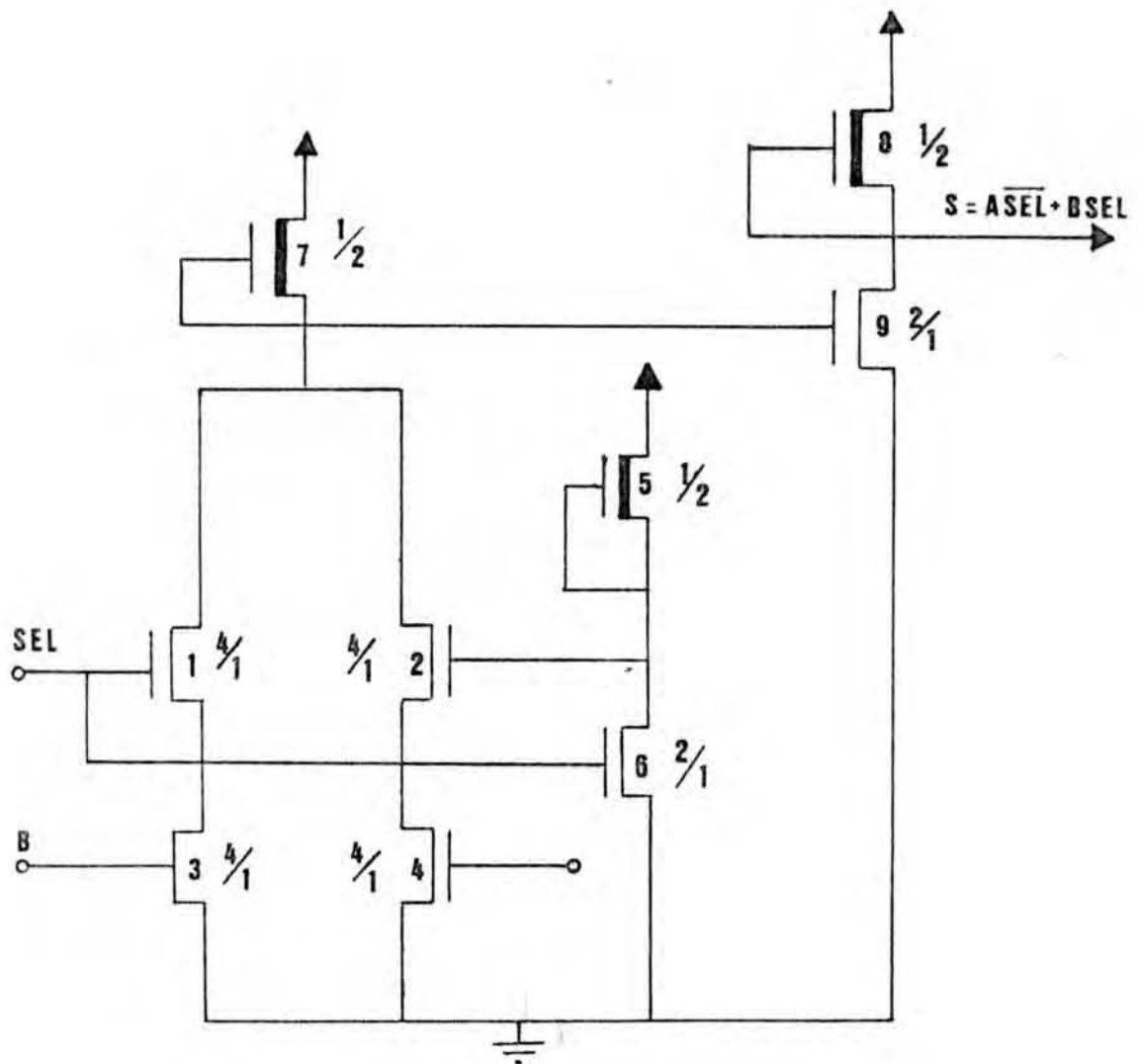


Figure A5.2-b - Schéma électrique du multiplexeur.

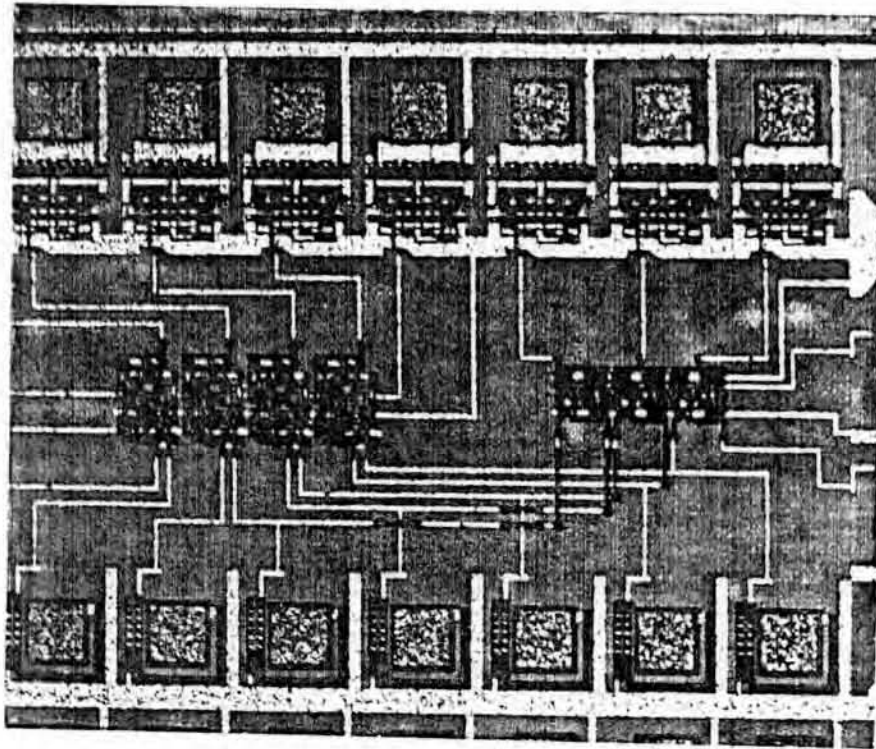


Figure A5.2-e - Photographie du multiplexeur (côté droit de la figure ; en fait, on montre 3 multiplexeurs.

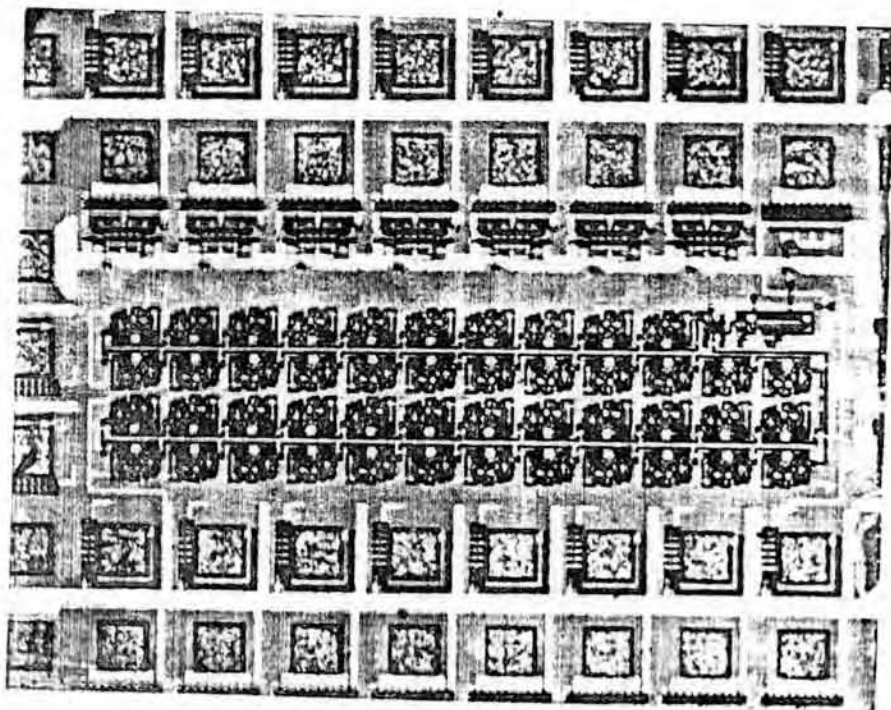


Figure A5.2-f - Photographie d'un montage pour mesurer le temps de propagation du multiplexeur en fonction d'un changement du signal de sélection.


```
(* 12-JAN-1984  DESCRIPTION FILE OF IS-      *)
* INCLUDE TOPCO:FILE:1001STICK.PAK*
* SUBS-CIRCUIT INVE+SFUR (11,12)

(*10-C *) TRA(N,1, 3.50000E+00, 7.50000E+00,12      ,-1      ,-2)
(*11-S *) TRA(N,3, 7.50000E+00, 3.50000E+00,0      ,11      ,12      ,-2)

(*CAP INV *) CAP(T2,0,3.94625E-02)
```

```
CIRCUIT (
A      ,B      ,SORT      ,SORTB      ,RELG      ,VINA      ,VINR      ,VINSELG )
,N21      ,N33      ,N36      ,N37      ,VINA      ,VINR      ,VINSELG )

(*5-C *) TRA(N,1, 3.50000E+00, 7.50000E+00,N36      ,N36      ,N1      ,-2)
(*8-C *) TRA(N,1, 3.50000E+00, 7.50000E+00,-1      ,SORT      ,SORT      ,-2)
(*7-C *) TRA(N,1, 3.50000E+00, 7.50000E+00,-1      ,N21      ,N21      ,-2)
(*6-S *) TRA(N,3, 7.50000E+00, 3.50000E+00,N3A      ,SELG      ,0      ,-2)
(*9-S *) TRA(N,3, 7.50000E+00, 3.50000E+00,0      ,N21      ,SORT      ,-2)
(*1-S *) TRA(N,3, 1.50000E+01, 3.50000E+00,N21      ,SELG      ,N33      ,-2)
(*3-S *) TRA(N,3, 1.50000E+01, 3.50000E+00,0      ,R      ,N33      ,-2)
(*2-S *) TRA(N,3, 1.50000E+01, 3.50000E+00,N37      ,N36      ,N21      ,-2)
(*4-S *) TRA(N,3, 1.50000E+01, 3.50000E+00,N37      ,A      ,0      ,-2)
(*EQUIFB *) CAP(R      ,0, 1.89000E-03)
(*EQUIIT *) CAP(SORT      ,0, 2.34075E-02)
(*EQUIRO *) CAP(SELG      ,0, 3.25625E-02)
(*EQUIV *) CAP(N21      ,0, 4.92188E-02)
(*EQUIHB *) CAP(N33      ,0, 2.12625E-02)
(*EQUIHA *) CAP(N3A      ,0, 3.94625E-02)
(*EQUILB *) CAP(N37      ,0, 2.12625E-02)
(*EQUIPA *) CAP(A      ,0, 1.97750E-03)
INVERSFUR( SORT, SORTA )
INVERSFUR( VINA, A )
INVERSFUR( VINR, P )
INVERSFUR( VINSFLG, SELG)
```

Figure A5.2-g - Description électrique du circuit : simulateur ELDO.

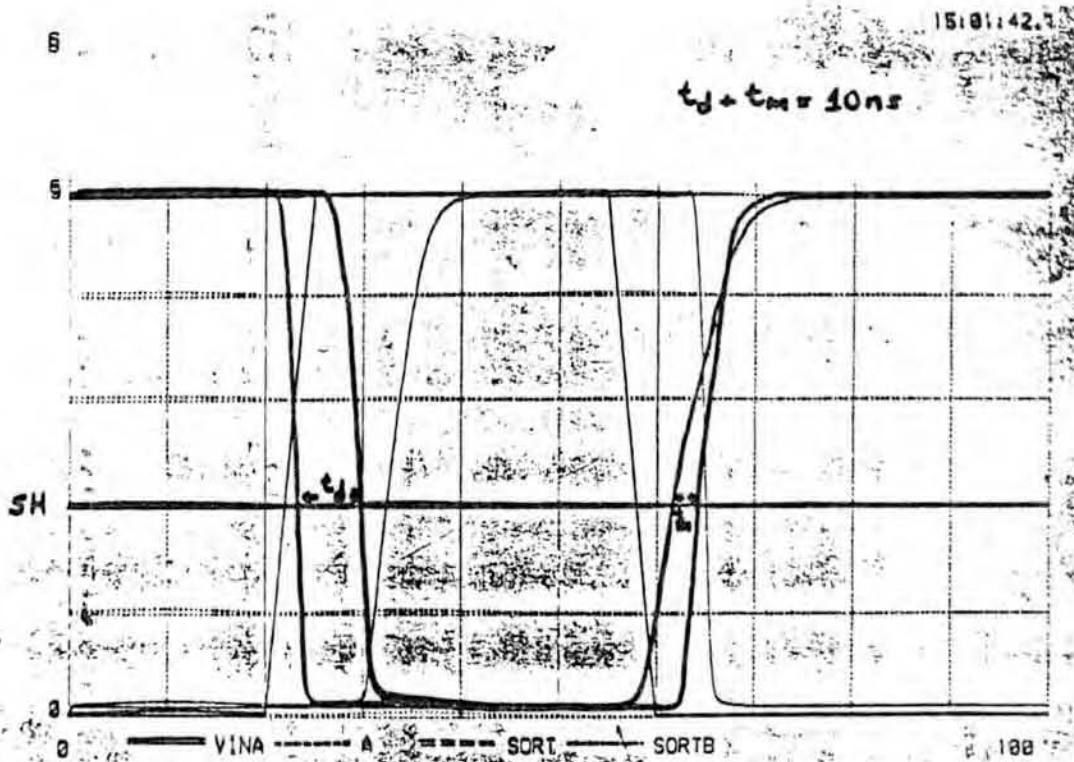


Figure A5.2-h - Résultat de la simulation ELDO.

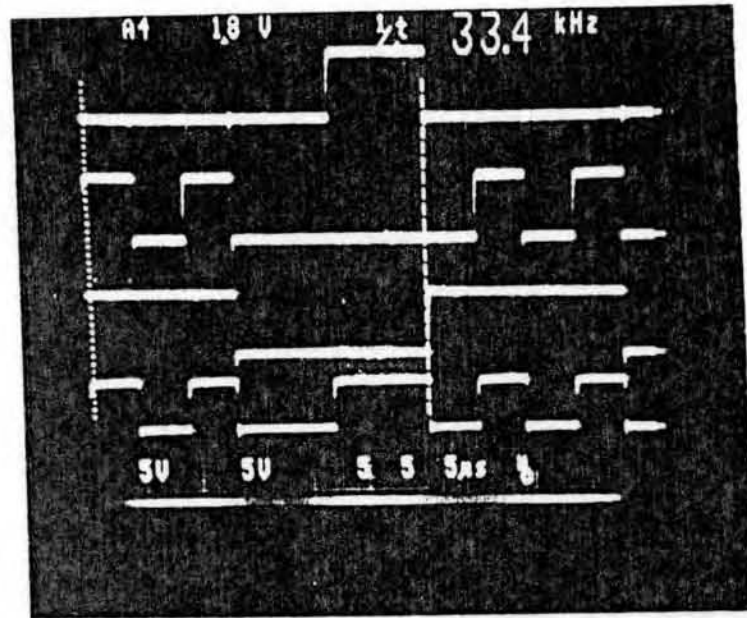


Figure A5.2-i - Test statique sous pointes du multiplexeur.
Sortie = ASEL + BSEL

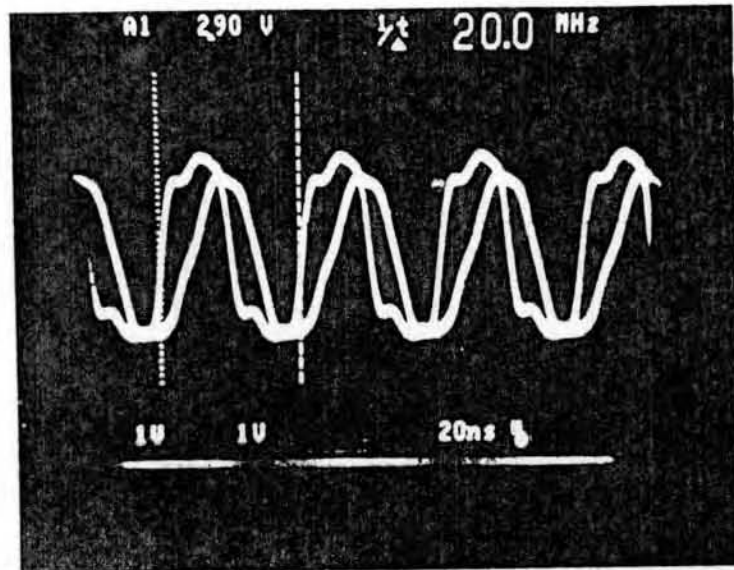


Figure A5.2-j - Réponse dynamique du multiplexeur à 20 MHz, à une variation du signal de sélection (A et B constantes).

5.6. TABLEAUX DE RÉCAPITULATION DES OUTILS DE CAO UTILISÉS DANS LE CADRE DE CE RAPPORT

Comme on l'a dit en A5.1, les outils de CAO peuvent être classés en deux familles.

Dans le tableau A5.1, on décrit où on a utilisé les outils de la première famille : outils de validation.

Dans le tableau A5.2, on montre où on a utilisé les outils de la deuxième famille : outils d'implantation, de vérification et de génération de masques.

TABLEAU A5.1 - OUTILS DE VALIDATION UTILISES DANS CE TRAVAIL : APPROCHES DISCRETE (Ch. 2) ET INTEGREE (Ch. 3)

TYPE	NOM	UTILISATION	REFERENCE AU TEXTE	DISPONIBILITE SUR CASSIOPEE
FONCTIONNEL	FIDEL [HAZ 84]	1- <u>VALIDATION DES ARCHITECTURES DISCRETES centralisées et décentralisées bâties à partir de l'ABC 90 (Ch. 2)</u> . Description du circuit 8289 (MULTIBUS) . " de l'unité de demande d'accès au bus (VME)	§ 1.3.4. § 1.4.3.	OUI*1
		2- <u>VALIDATION DES ARCHITECTURES INTEGrees (ABC-M). (Modes de fonctionnement RWD et ROR)</u> . Validation du mode RWD : description de l'unité de demande d'accès au bus VME. . Validation du mode ROR : description du circuit 8289 (architecture MULTIBUS).	§ 1.3.4. § 1.4.3.	
LOGIQUE	EPILOG [EPI 79]	3- <u>VALIDATION PAR SIMULATION GLOBALE DES ARCHITECTURES DISCRETES</u> . Description de l'architecture centralisée MULTIBUS/ABC 90 . " " " décentralisée " " . " " " centralisée VME/ABC 90 . " " " décentralisée " "	§ 2.3.3.1. § 2.3.3.2. § 2.4.3.1. § 2.4.3.2.	OUI*2
		4- <u>VALIDATION PAR SIMULATION DES MODIFICATIONS APPORTEES A L'ABC 90 LE TRANSFORMANT EN ABC-M</u> . Description et simulation de la logique associée à chaque modification : - modes de fonctionnement du circuit - divulgation systématique du verdict - logique supplémentaire de divulgation du verdict de forme décodée - automate de déclenchement d'arbitrage	§ 3.3. § 3.3.1. § 3.3.2. § 3.3.3. § 3.3.4.	

TABLEAU A5.1 (suite) - OUTILS DE VALIDATION UTILISES DANS CE TRAVAIL : APPROCHES DISCRETE (Ch. 2) ET INTEGREE (Ch. 3)

TYPE	NOM	UTILISATION	REFERENCE AU TEXTE	DISPONIBILITE SUR CASSIOPEE
LOGIQUE	EPILOG [EPI 79]	5- VALIDATION PAR SIMULATION GLOBALE DES ARCHITECTURES INTEGrees DANS LE CADRE DE LA VALIDATION DE L'ABC-M	§ 3.4.	OUI*2
		. Description de l'architecture SM 90 (mode SM 90)	§ 3.4.3.1.	
		. " " " décentralisée VME (mode RWD)	§ 3.4.3.2.	
		. " " " MULTIBUS (mode ROR)	§ 3.4.3.3.	
ELECTRIQUE	ELDO [HEN 84]	6- SIMULATION DU MULTIPLEXEUR UTILISE DANS LA LOGIQUE SUPPLEMENTAIRE DE DIVULGATION DU VERDICT SPECIFIQUE A L'ABC-M	§ 3.3.3.	OUI

*1 On a utilisé une version en développement laquelle n'appartenait pas à la base de données.

*2 L'interface FIDEL-EPILOG a imposé l'utilisation d'une version d'EPILOG en-dehors de la base de données.

TABLEAU A5.2 - OUTILS D'IMPLANTATION, DE VERIFICATION ET DE GENERATION DES MASQUES, UTILISES DANS CE TRAVAIL : APPROCHE INTEGREE (ABC-M)

TYPE	NOM	UTILISATION	REFERENCE AU TEXTE	DISPONIBILITE SUR CASSIOPEE
IMPLANTATION	CRASH [ROS 82]	- IMPLANTATION DE MODIFICATIONS APPORTEES A L'ABC 90 LE TRANSFORMANT EN ARBITRE MULTIPROCOLES : . logique associée aux modes de fonctionnement . " " à la divulgation systématique du verdict . " " " " " du verdict de forme décodée	§ 3.3.1. § 3.3.2. § 3.3.3.	OUI
	MDMOS [CON82]	- La plus grande partie de l'ABC 90 a été décrite sous MD-MOS		OUI
	Méthode* de synthèse de GRAF CET [OLI 83-2]	- IMPLANTATION DE L'AUTOMATE DE DECLENCHEMENT D'ARBITRAGE	§ 3.3.4.	NON
VERIFICATION	DRC CASSIOPEE [BON 84]	- Les implantations faites en CRASH ont été toutes vérifiées par ce vérificateur de règles de dessin symbolique		OUI
	DRC CALMA	- Toutes les règles de garde de l'implantation globale de l'ABC-M sur Silicium ont été vérifiées par le programme de DRC disponible à l'ordinateur CALMA du CNET. On a utilisé le DRC-CRASH et le DRC-MICRON		NON
GENERATION DE MASQUES	CALMA	- Les règles de dessin étant vérifiées (DRC-CRASH), on a utilisé la génération automatique de masques disponibles sur CALMA		NON

* Cette méthode est décrite en annexe 7.

ANNEXE 6

RESULTATS DES SIMULATIONS
DES ARCHITECTURES DISCRETES UTILISANT L'ABC 90

6.1. INTRODUCTION

6.1.1. OUTILS DE CAO EMPLOYÉS

Dans cette annexe, on valide les propositions architecturales faites au chapitre 2.

Comme on l'a vu dans ce chapitre, l'ABC 90 est compatible avec le MULTIBUS (plus particulièrement avec le circuit 8289) ou avec le VME (plus particulièrement avec l'unité de demande d'accès au bus VME), au coût d'une logique supplémentaire.

En outre, on a proposé dans ce chapitre, pour chaque type de compatibilité (MULTIBUS-ABC 90, ou VME-ABC 90) deux types d'architecture : centralisée et décentralisée.

Comme on avait vu en § 2.5., la simulation la plus adaptée au problème est la validation logique. Cette validation a besoin de deux outils de CAO décrits dans l'annexe 5 :

Le premier EPILOG (voir A5.3.2.) sert à la description de chaque architecture. Notre travail a été allégé dû à la disponibilité d'une description complète de l'ABC 90 sur ce simulateur.

Toutefois, dû à la non-disponibilité d'un schéma d'implantation logique de circuits qui font partie de ces architectures (circuits 8289 et unité de demande d'accès au bus VME), et à la non-disponibilité d'outils de CAO capables de simuler des unités décrites par des chronogrammes, on a décrit ces circuits en FIDEL (voir A5.3.1.).

En fait, il serait possible de ne pas utiliser FIDEL. Pour le faire, on serait obligé de décrire le fonctionnement du circuit sur le simulateur EPILOG, à partir de ses chronogrammes, par un sous-programme FORTRAN. Toutefois, il est difficile d'aboutir à un modèle assez général à partir d'un modèle fonctionnel.

Malgré des problèmes rencontrés au début de l'utilisation du langage, dû au fait qu'il se trouvait en développement, l'emploi de FIDEL pour la description des modèles qu'on a eus à décrire a porté les avantages suivants :

- efficacité de description,
- possibilité de décrire le parallélisme (modèle MULTIBUS) même sans utiliser une description temporelle,
- temps de simulation ainsi que taille mémoire plus intéressants.

Dans la figure A6.1, on montre la description FIDEL du circuit 8289, et dans la figure A6.2, celle pour l'unité de demande d'accès au bus VME (UDABVME), utilisées pour la validation des architectures discrètes décrites dans le chapitre 2.


```

MODEL #8289(RSB,CROLCKB,RESB,ANYROST,IOBB,INITB,BCLKB,BPRNB,BUSYBE,
          CBROBE,BREQB,BUSYBS,CBROBS);
DECLARE INPUT RSB(0:3),CROLCKB,RESB,ANYROST,IOBB,INITB,BCLKB,BPRNB,BUSYBE,CBROBE;
DECLARE OUTPUT BREQH,BUSYBS,CBROBS;
DECLARE STATE SBM,VER,INT,A,B,D,E,F,G;
INITIALIZE BREQB,BUSYBS,CBROBS TO 1;
INITIALIZE SBM,VER TO 0;
WHEN INITB FALLS
  MAKE BREQB=1
  MAKE BUSYBS=1
  MAKE CBROBS=1;
WHEN RSB CHANGES
  IF RSB=7
    THEN IF CBROBE=0 AND CROLCKB=1
      THEN MAKE D=0
      MAKE E=0
      MAKE VER=1
      MAKE SBM=0
    ENDIF
  ENDIF
  IF RSB=3
    THEN MAKE CBROBS=1
    MAKE BUSYBS=1
    MAKE BREQB=1
    MAKE SBM=0
  ENDIF
  IF RSB=0 OR RSB=1 OR RSB=2 OR RSB=4 OR RSB=5 OR RSB=6
    THEN IF IOBB=1 AND RESB=0
      THEN MAKE SBM=1
      MAKE A=1
      MAKE B=1
    ENDIF
  ENDIF;
WHEN BCLKB FALLS AND SBM=1
  IF BPRNB=1
    THEN MAKE BREQB=0
    MAKE CBROBS=0
  ENDIF
  PAR
    MAKE A=0
    MAKE B=A
  ENDPAR
  IF B=0 AND BPRNB=0
    THEN IF BUSYBE=1
      THEN MAKE BUSYBS=0
      MAKE CBROBS=1
      MAKE F=0
      MAKE G=0
      MAKE INT=1
    ENDIF
  ENDIF;
WHEN BCLKB FALLS AND VER=1
  IF BPRNB=0
    THEN MAKE BREQB=1
    ELSE MAKE CBROBS=1
    MAKE BREQB=1
  ENDIF
  PAR
    MAKE D=1
    MAKE E=0
  ENDPAR
  IF E=1
    THEN MAKE BUSYBS=1
    MAKE VER=0
  ENDIF;
WHEN BCLKB FALLS AND INT=1
  PAR
    MAKE F=1
    MAKE G=F
  IF G=1 AND BPRNB=1
    THEN MAKE BUSYBS=1
    MAKE INT=0
  ENDIF
  ENDPAR;
ENDMODEL

```

Figure A6.1 - Description FIDEL du circuit 8289.

```
MODEL UDABVME(NBECHI,DEM,DEC,NBRO,NBBSY);
DECLARE INPUT NBECHI,DEM(0:8),DEC(0:8);
DECLARE OUTPUT NBRO(0:8),NBBSY(0:8);
DECLARE STATE VER(0:8),INT(0:8);
DECLARE INTEGER I,J,K;
INITIALIZE NBRO TO #B111111110;
INITIALIZE NBBSY TO #B111111110;
INITIALIZE INT TO #B111111110;
WHEN DEM CHANGES
  MAKE I=0
  WHILE I<8 DO
    IF DEM(I)=0 AND INT(I)=1 THEN
      MAKE NBRO(I)=0
      MAKE VER(I)=1
      MAKE INT(I)=0
    ENDIF
    IF DEM(I)=1 AND INT(I)=0 THEN
      MAKE INT(I)=1
    ENDIF
    MAKE I=I+1
  ENDWHILE;
WHEN DEC CHANGES
  MAKE J=0
  WHILE J<8 DO
    IF DEC(J)=0 AND VER(J)=1
      THEN MAKE NBBSY(J)=0
      MAKE NBRO(J)=1
    ENDIF
    MAKE J=J+1
  ENDWHILE;
WHEN NBECHI CHANGES
  MAKE K=0
  WHILE K<8 DO
    IF NBECHI=1 AND NBBSY(K)=0
      THEN MAKE NBBSY(K)=1
    ENDIF
    MAKE K=K+1
  ENDWHILE;
ENDMODEL
```

Figure A6.2 - Description FIDEL de l'unité de demande d'accès au bus VME.

6.1.2. RAPPEL DES ARCHITECTURES

Rappelons d'après l'analyse qu'on a faite aux § 2.3.3. et 2.4.3. les caractéristiques des architectures proposées :

- + Bus MULTIBUS - arbitre de bus ABC 90
- * ARCHITECTURE CENTRALISEE, permettant :
 - des arbitrages cachés,
 - l'algorithme rotatif de priorités,
 - la supervision des arbitrages rendus.

Remarque : cette architecture a été montrée à la figure 20.

- * ARCHITECTURE DÉCENTRALISEE, permettant :
 - des arbitrages cachés,
 - l'algorithme rotatif de priorités (possibles) dès qu'un ABC 90 délivre le verdict ou algorithme fixe si c'est à l'encodeur (74 LS 148) de délivrer le verdict,
 - la supervision des arbitrages rendus, qu'ils soient basés sur l'algorithme de l'ABC 90 ou sur l'algorithme fixe de l'encodeur.

Remarque : cette architecture a été montrée à la figure 21.

+ Bus VME - arbitre de bus ABC 90

* ARCHITECTURE CENTRALISEE, permettant :

- des arbitrages visibles,
- l'algorithme rotatif de priorités,
- la supervision des arbitrages rendus.

Remarque : cette architecture a été montrée à la figure 24.

* ARCHITECTURE DECENTRALISEE, permettant :

- des arbitrages visibles,
- l'algorithme rotatif de priorités possibles (verdict délivré par l'ABC 90), ou fixe (verdict délivré par l'encodeur),
- la supervision de tous les arbitrages rendus.

Remarque : cette architecture a été montrée à la figure 25.

6.2. RÉSULTATS POUR L'ARCHITECTURE CENTRALISÉE MULTIBUS-ABC 90

6.2.1. CARACTÉRISTIQUES DE LA SIMULATION

Pour cette architecture, les simulations qu'on a faites ont les caractéristiques suivantes :

. trois unités de traitement, chacune contenant un circuit 8289.

. trois arbitres de bus sur la carte centrale contrôleur du système, configurés aux adresses 0, 1 et 2 selon le même ordre des demandes BA 17, BA 18 et BA 19.

. la première unité de traitement est configurée en priorité fixe, tandis que les deux autres sont configurées en rotative ; paramètre PF câblé à 001 sur les trois arbitres.

. on permet des arbitrages cachés, broche BPAC câblée à 1.

. au début de la séquence de simulation, tous les arbitres et circuits 8289 ont été correctement initialisés (signal INIT actif bas pendant 400 ns au minimum).

Les séquences de demande de bus (registre d'état RSB de chaque 8289) sont les suivantes (figure A6.3) :

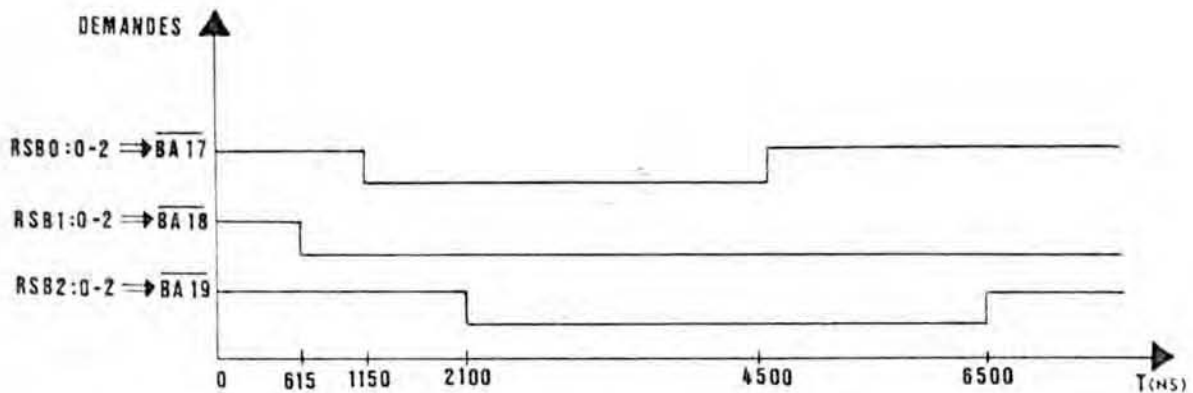


Figure A6.3 - Séquences de demande de bus.

. les circuits 8289 ont été configurés dans le mode bus unique (voir § 1.2.4.) :

IOB câblé à 1
RESB câblé à 0

. pour cette configuration, en-dehors des états 011 (HALT) et 111 (REPOS) du registre d'état, tous les autres états indiquent une demande de bus.

. dès que le bus est pris, il ne sera relâché que si une des conditions suivantes est vraie :

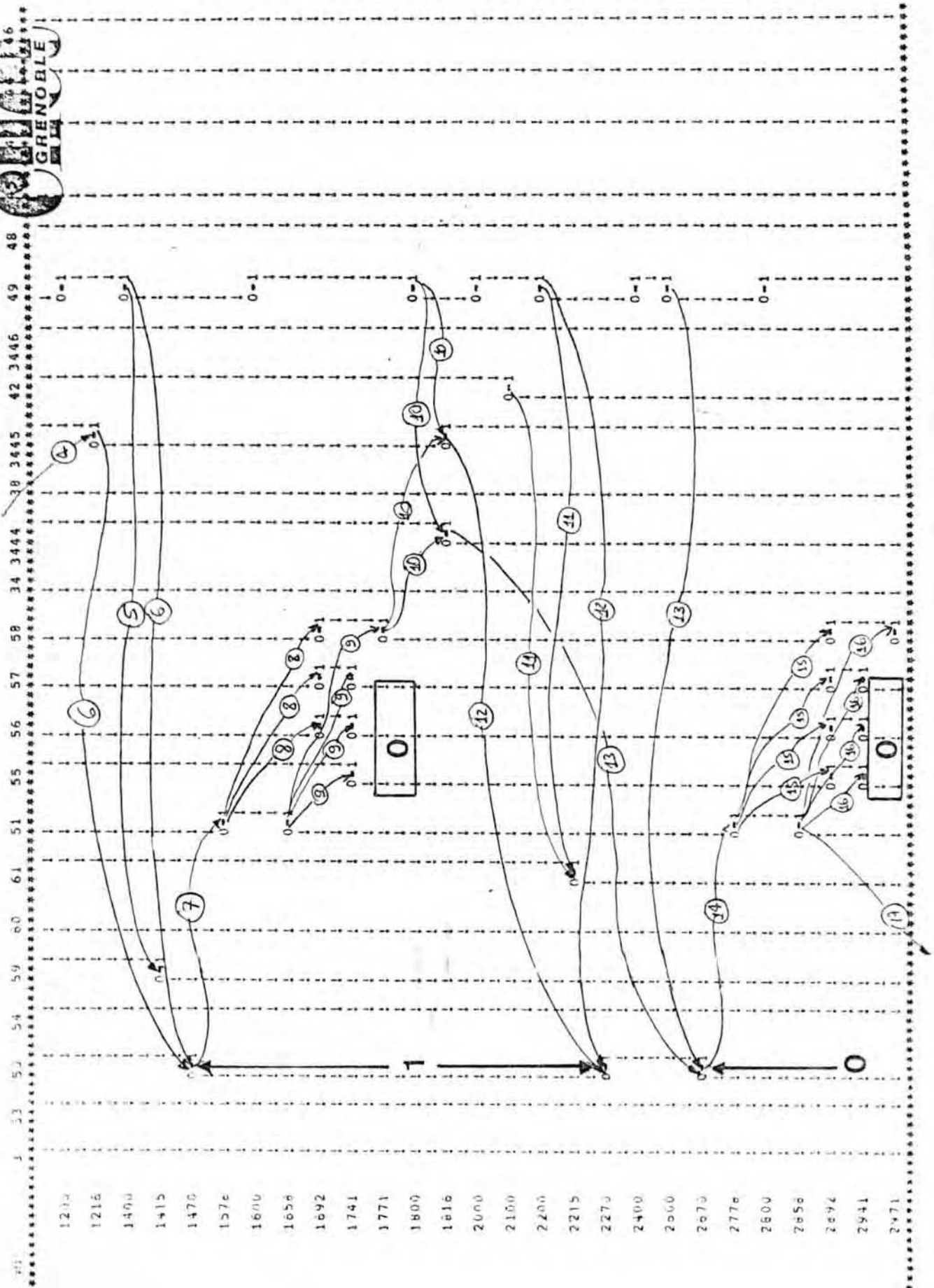
- état HALT,
- état au REPOS et demande d'une unité moins prioritaire (CBRQ) active,
- demande d'une unité de plus forte priorité (BPRN passé à 1 chez l'unité qui était en train d'utiliser le bus),
- prise en considération (pour relâcher le bus, voir § 1.2.4.) des demandes moins prioritaires à travers la broche CBRQ :

ANYRQST câblé à 1
CRQLCK câblé à 1

6.2.2.A. (SUITE)

LE 19/84/ 6 PAGE N 5

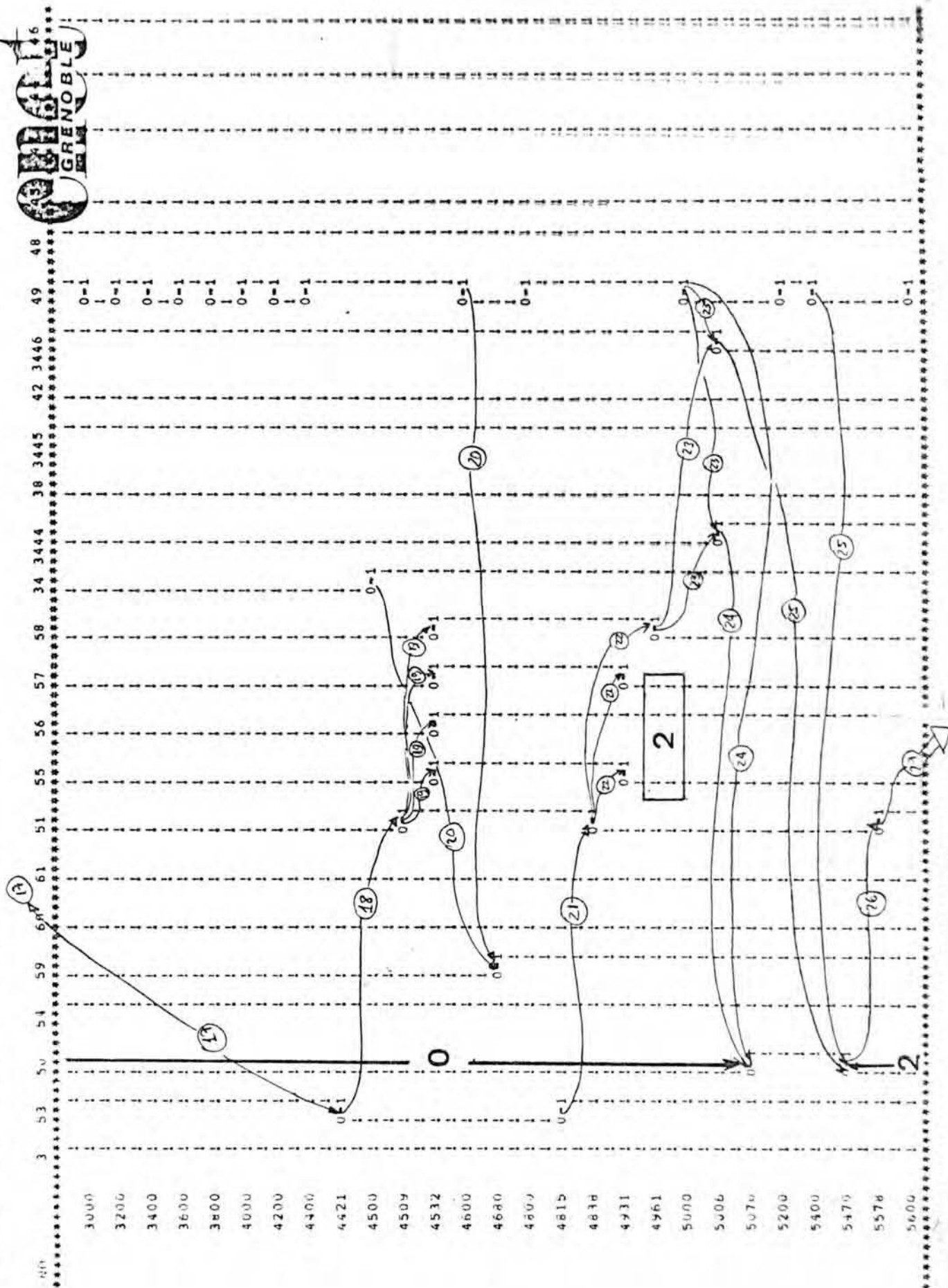
EPIRODE SIMULATION DE LA PROPOSITION CINAQUETTE MULTIBUS

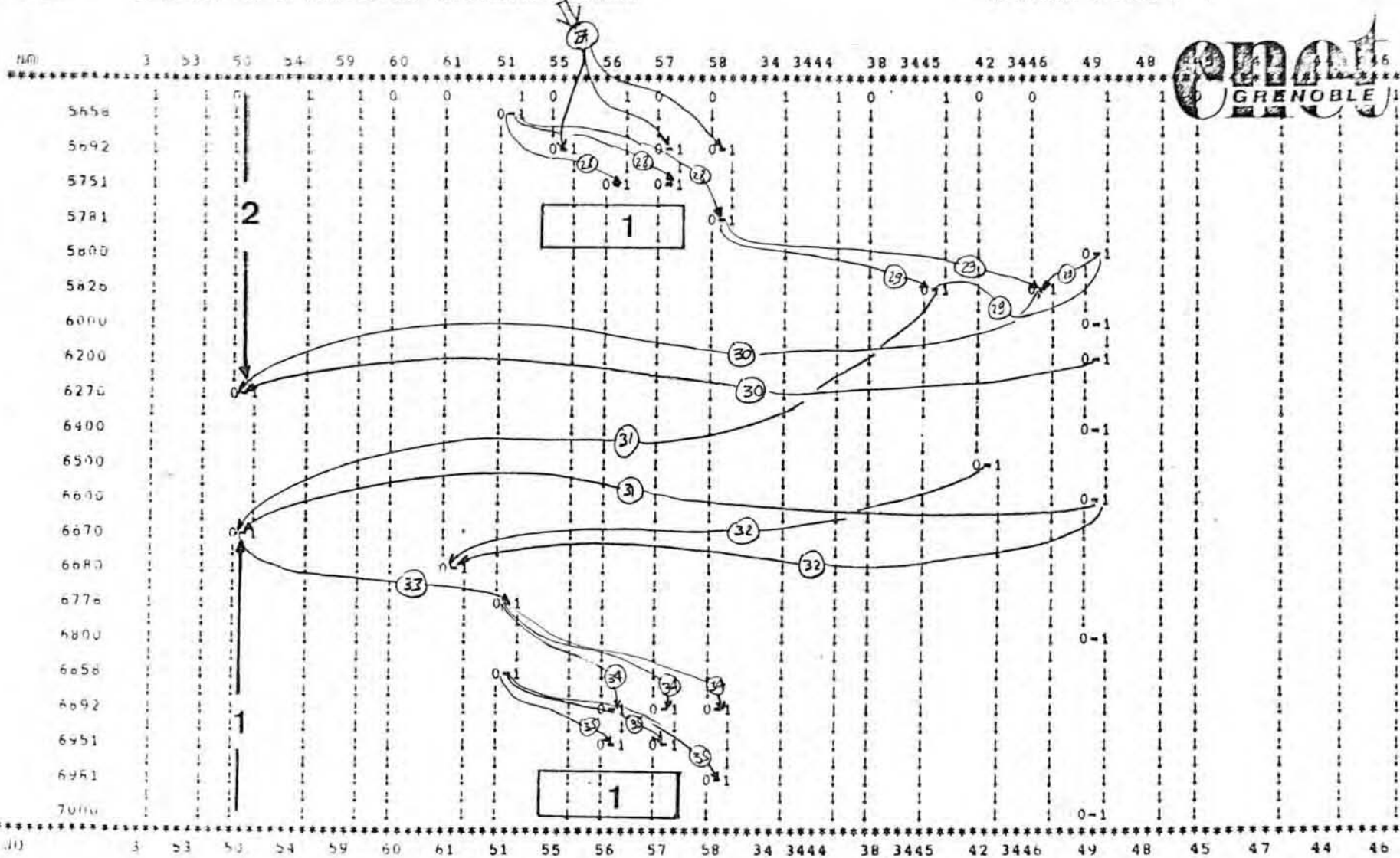


6.2.2.A. (SUITE)

LE 19/84/ 6 PAGE N 6

EPISODE SIMULATION DE LA PROPOSITION CINAQUETTE MULTIBUS





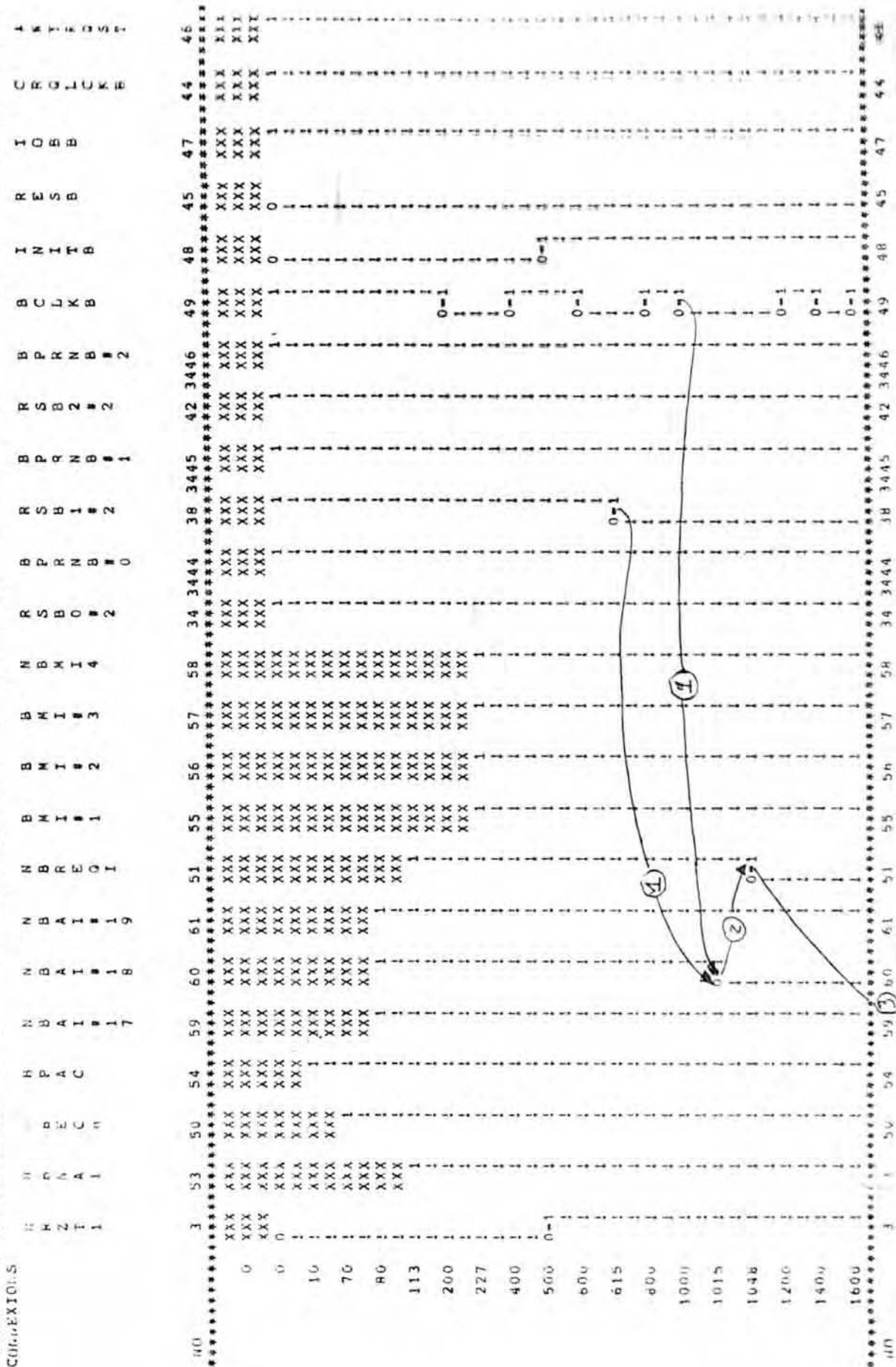
PHASE **LIT** 1-PS CPU : 341,00 SECONDES

6.2.2.B. EXEMPLE D'INTERVENTION DE LA FONCTIONNALITÉ DE SUPERVISION : PANNE OU INEXISTENCE D'UN ARBITRE ABC 90 À L'ADRESSE 0

LE 19/84/ 6 PAGE N 4

EPISODE SIMULATION DE LA PROPOSITION CIMAQUETTE MULTIBUS

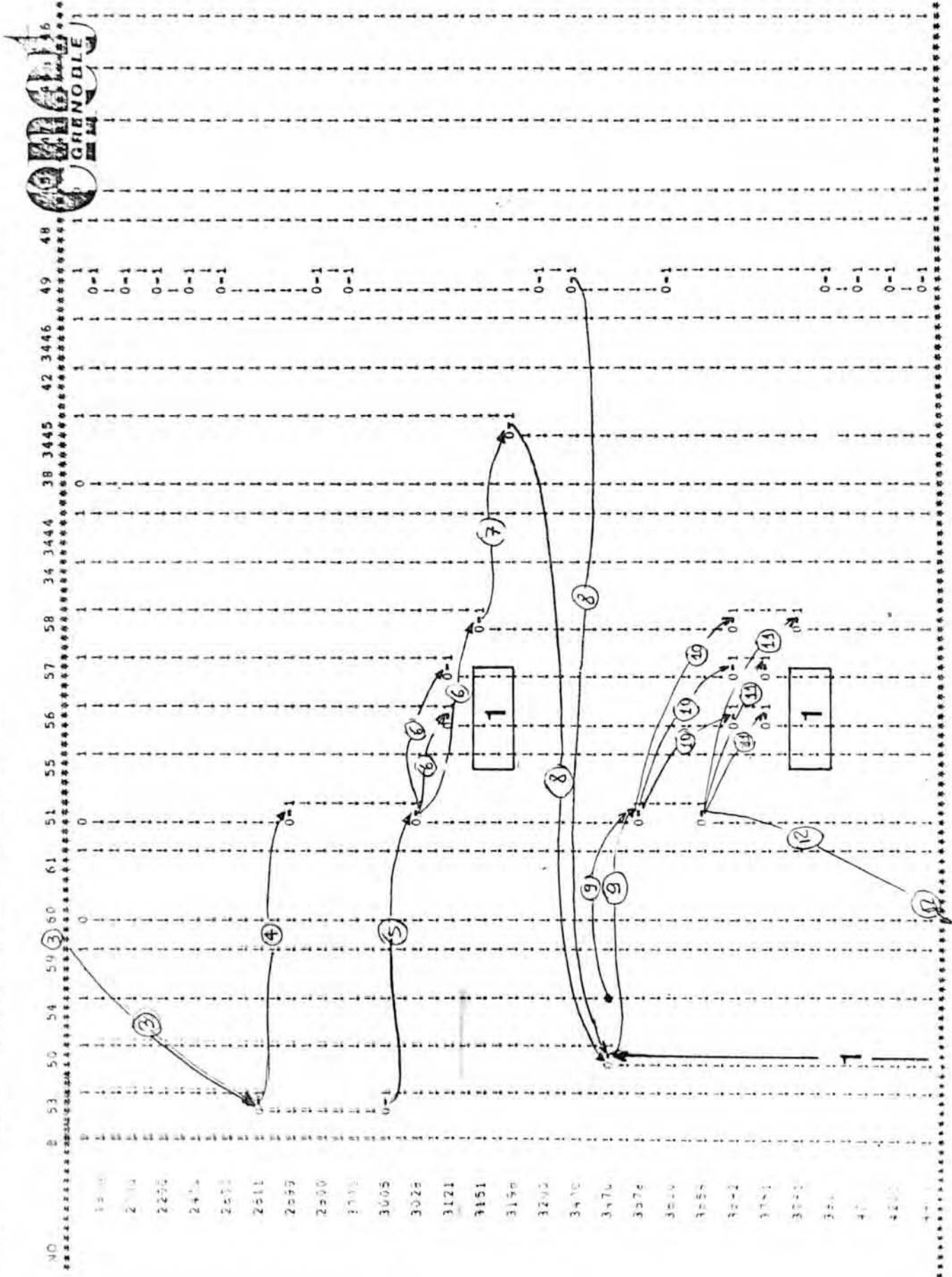
CHRONOGRAMME DU CIRCUIT MCSM90



6.2.2.B. (SUITE)

LE 19/84/ 6 PAGE N 5

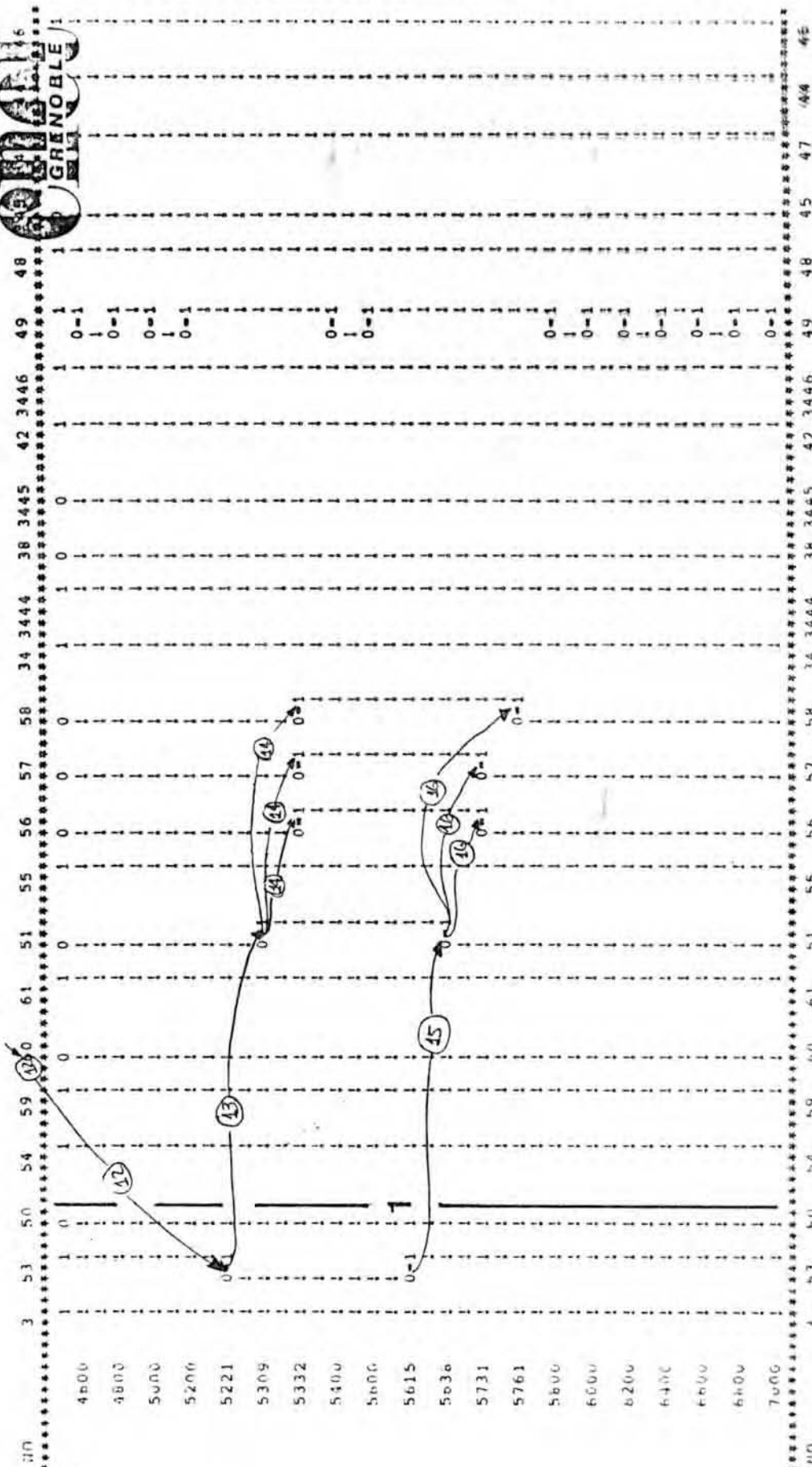
APIS:12 SIMULATION DE LA PROPOSITION CIMA UETTE MULTIBUS



6.2.2.B. (SUITE)

LE 19/84/ 6 PAGE N 6

EPISEUNE SIMULATION DE LA PROPOSITION CIMACQUETTE MULTIBUS



PHASE EDITION TEMPS CPU : 283.00 SECONDES

6.2.3. REMARQUES

D'après les chronogrammes montrés au § 6.2.2., on peut les comparer à celui montré en annexe 2, on peut dégager les caractéristiques suivantes :

. on respecte les contraintes MULTIBUS ce qui valide notre proposition.

. comme le temps d'arbitrage de la structure se situe autour de 200 ns, ce qui est largement supérieur au temps pris par les circuits encodeur-décodeur (structure MULTIBUS classique), on doit augmenter la période d'horloge à 400 ns (2 MHz). Normalement, la fréquence maximale d'horloge BCLK est de 10 MHz, et peut être utilisée dès qu'on utilise l'encodeur-décodeur.

. la fonctionnalité de supervision est nécessaire pour le bon fonctionnement de l'architecture proposée (alors qu'elle est optionnelle dans la structure SM 90), car dès qu'une unité de traitement configurée en priorité fixe supérieure à toutes les autres (dans notre cas 17 (FIXE), 18 et 19 (ROTATIVES)) prend le bus, l'arbitrage caché engendré par le front descendant de BECH lui sera favorable ; ainsi le signal de demande d'arbitrage ne sera plus activé. De cette façon, dès que le temps surveillé de 1 500 ns (réglé par la capacité KP) s'écoule, les ABC 90 génèrent BNA, lequel par la suite à travers BREQ va générer un nouvel arbitrage.

6.3. RÉSULTATS POUR L'ARCHITECTURE DÉCENTRALISÉE MULTIBUS-ABC 90

6.3.1. CARACTÉRISTIQUES DE LA SIMULATION

Pour cette architecture, les simulations qu'on a faites ont les caractéristiques suivantes :

. trois cartes SBC standards, chacune contenant un circuit 8289 décrit en FIDEL (adresses 2, 3 et 4).

. deux cartes spéciales chacune contenant un arbitre de bus ABC 90 (adresses 0 et 1).

. plus logique supplémentaire décrite au § 2.3.3.2.

. toutes les cartes sont configurées en priorité rotative (paramètre PF câblé à 000 sur les trois arbitres).

. 0H permet des arbitrages cachés, broche BPAC câblée à 1.

. la séquence d'initialisation est pareille à celle décrite au § 6.2.1.

. les circuits 8289 ont été configurés de la même façon qu'au § 6.2.1.

Les séquences de demande de bus sont les suivantes (figure A6.4) :

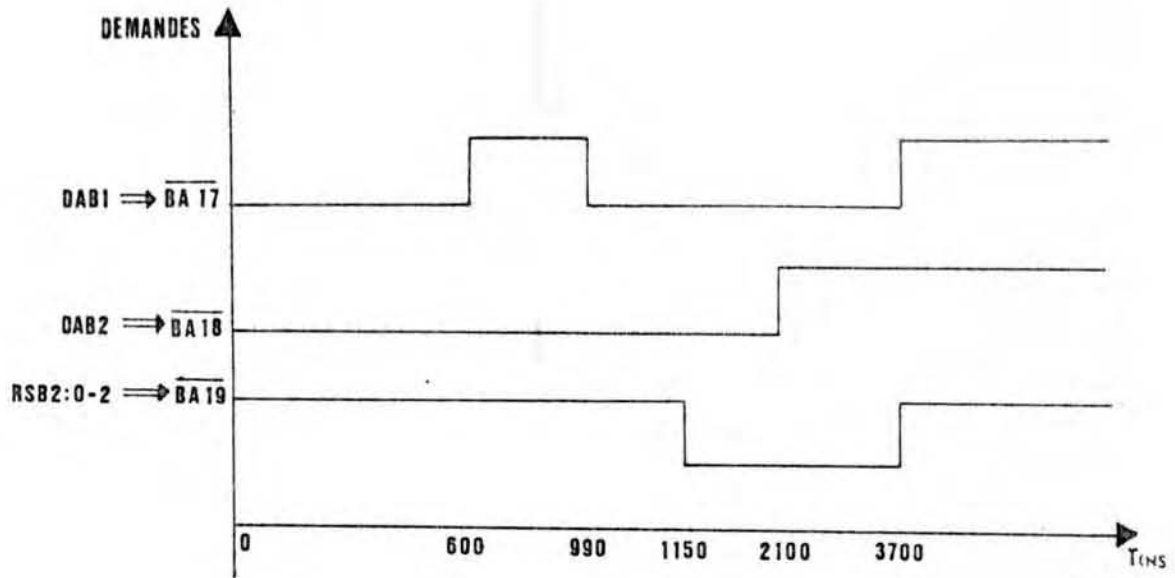


Figure A6.4 - Séquences de demande de bus.

6.3.2. CHRONOGRAMME EPILOG
DE VALIDATION DE L'ARCHITECTURE
DÉCENTRALISÉE MULTIBUS-ABC 90.

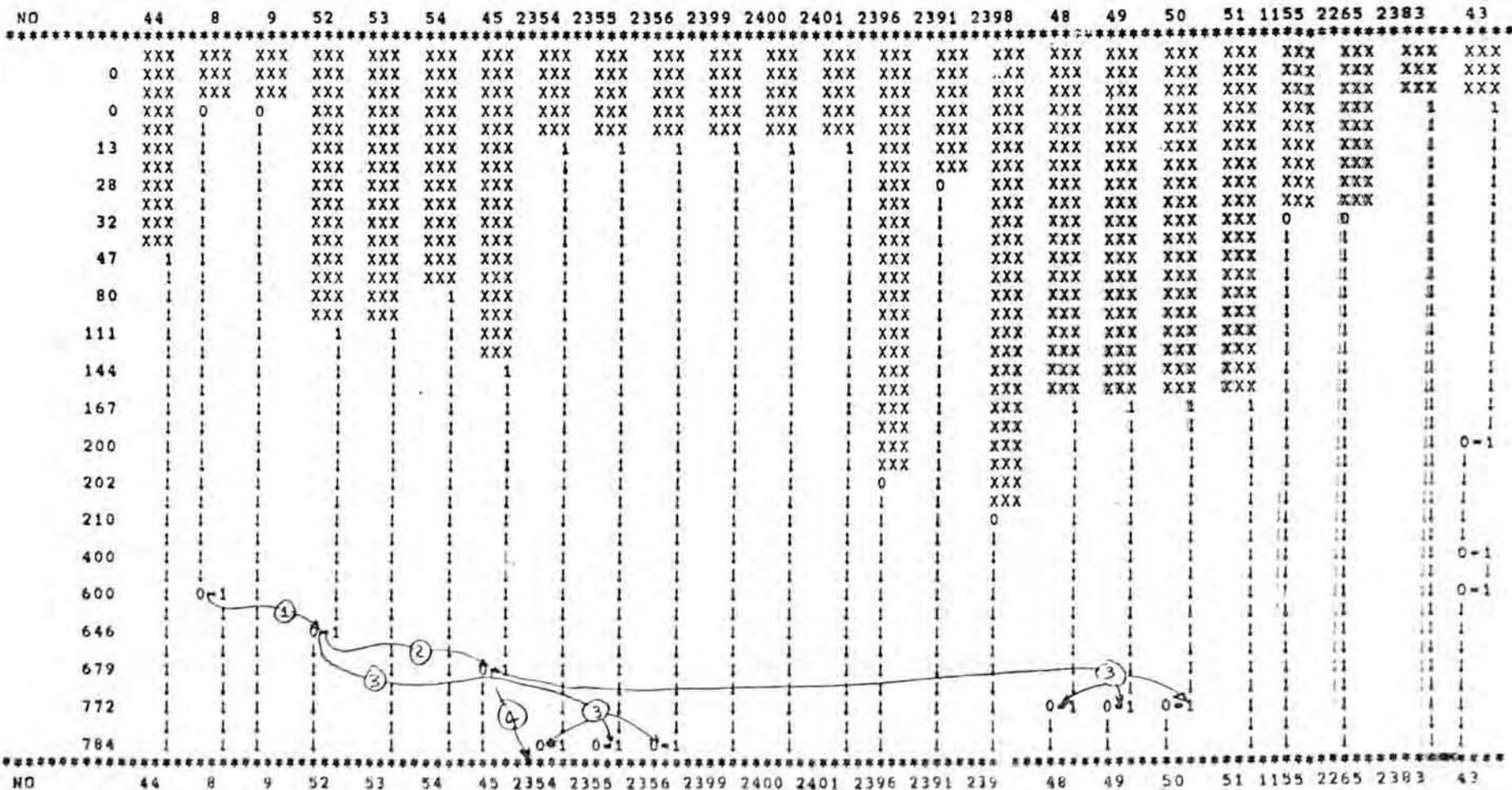


CHRONOGRAMME DU CIRCUIT MASH90

CONNEXIONS

N	D	D	N	N	N	N	A	A	A	S	S	S	V	S	S	B	B	B	N	D	D	B	B
B	A	A	B	B	B	B	O	1	2	E	E	E	L	B	L	M	M	M	B	B	B	P	C
A	B	B	A	A	A	R				#	#	#	D			I	I	I	M	A	A	R	L
C	1	2	I	I	I	E				1	2	3				1	2	3	I	1	2	N	K
H			#	#	#	Q													I	1	2	B	B
			1	1	1	I													I	1	2	B	B
			7	8	9	I													I	1	2	B	B

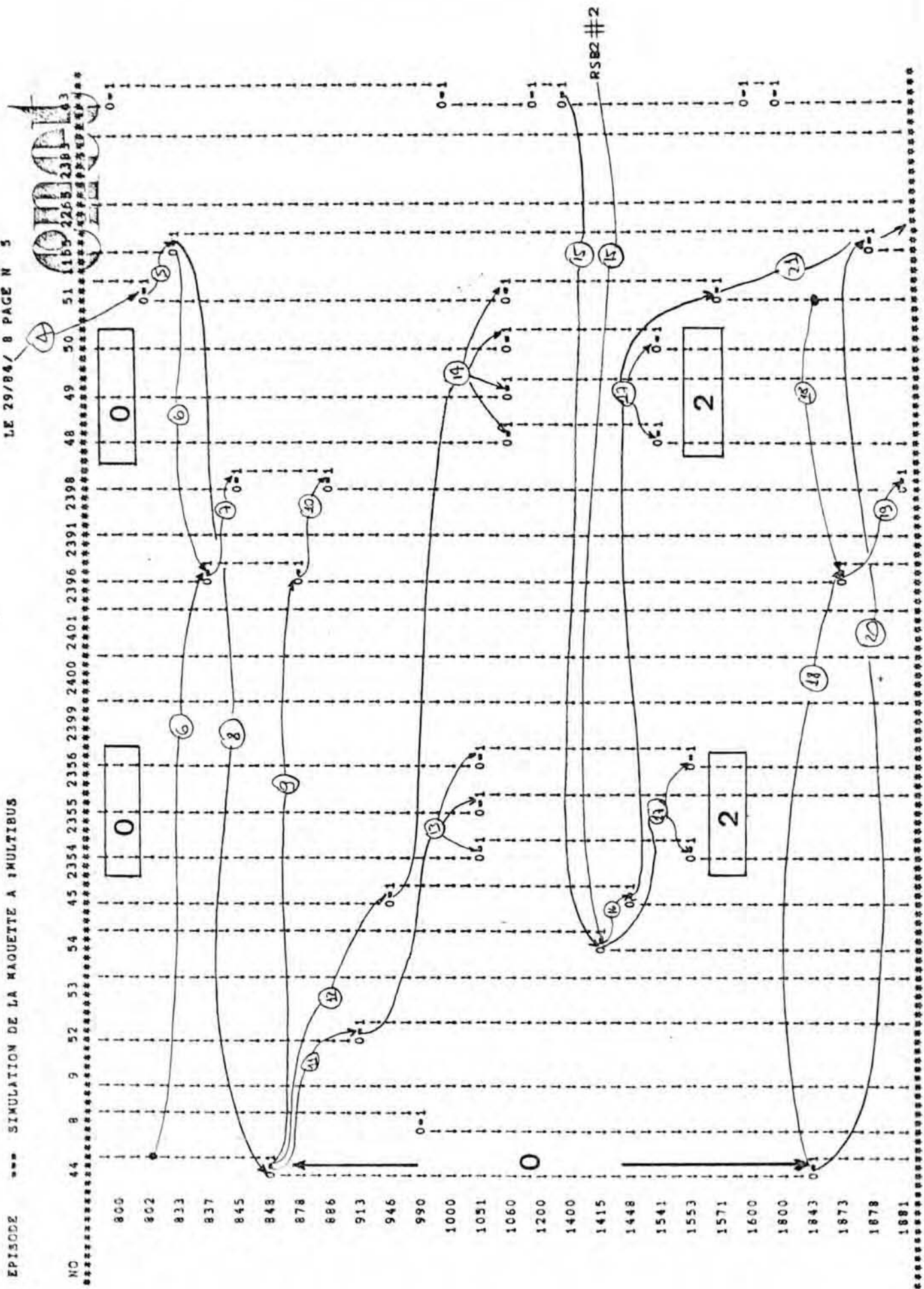
6.3.2.



6.3.2. (SUITE)

LE 29/84 / 8 PAGE N 5

EPISODE *** SIMULATION DE LA MAQUETTE A MULTIBUS

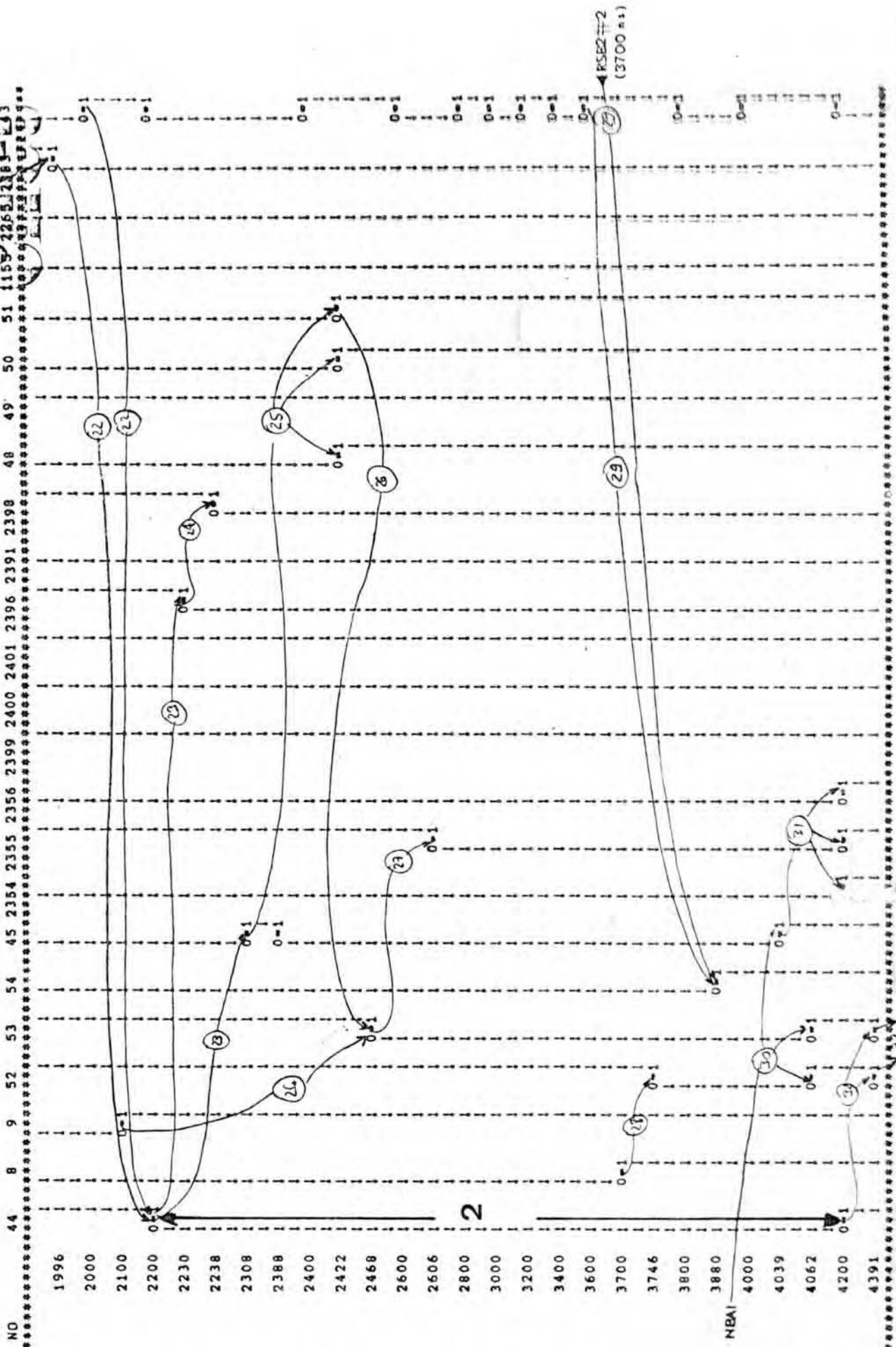


15 16 2265 2383 43

6.3.2. (SUITE)

LE 29/84/ 8 PAGE N 6

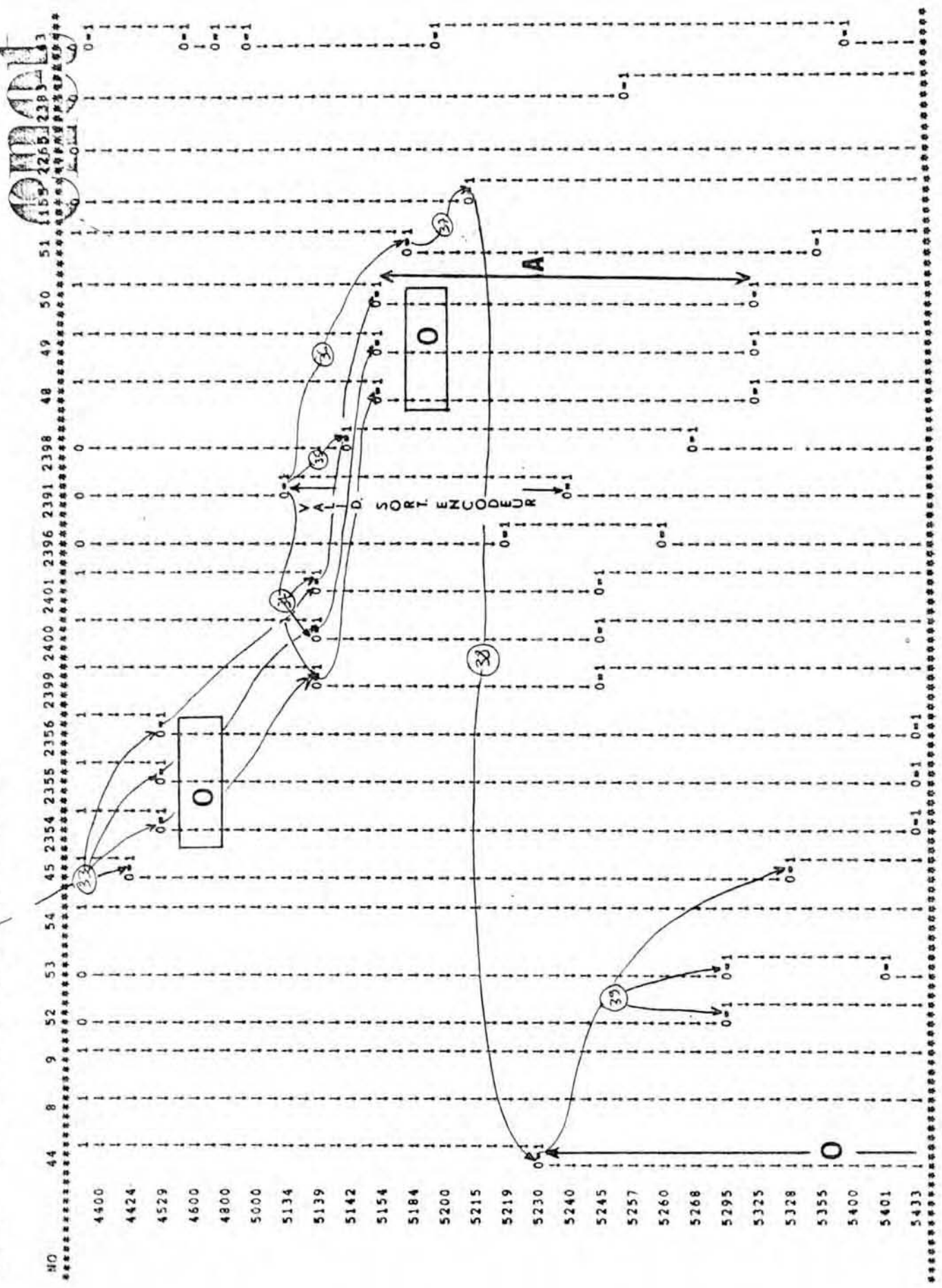
EPISODE --- SIMULATION DE LA MAQUETTE A MULTIBUS



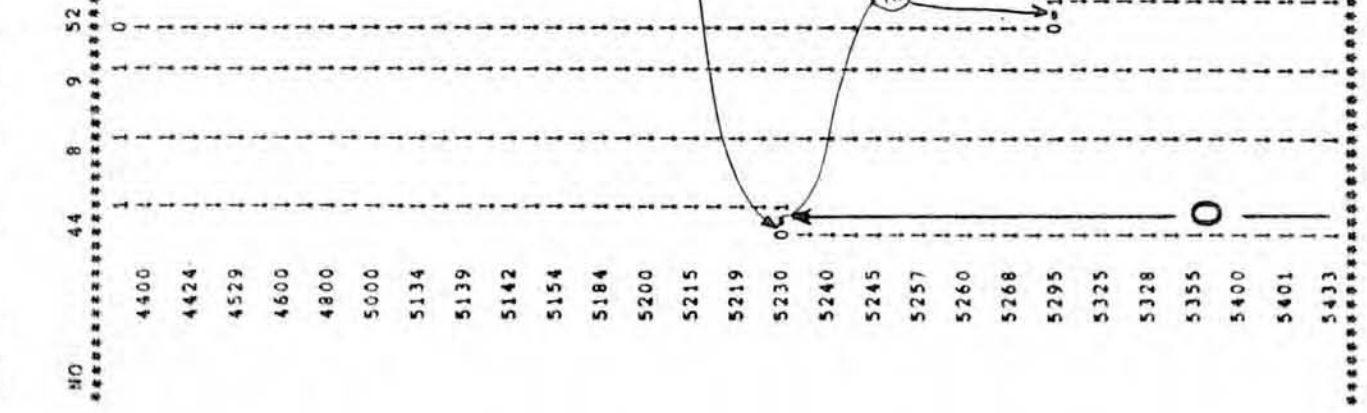
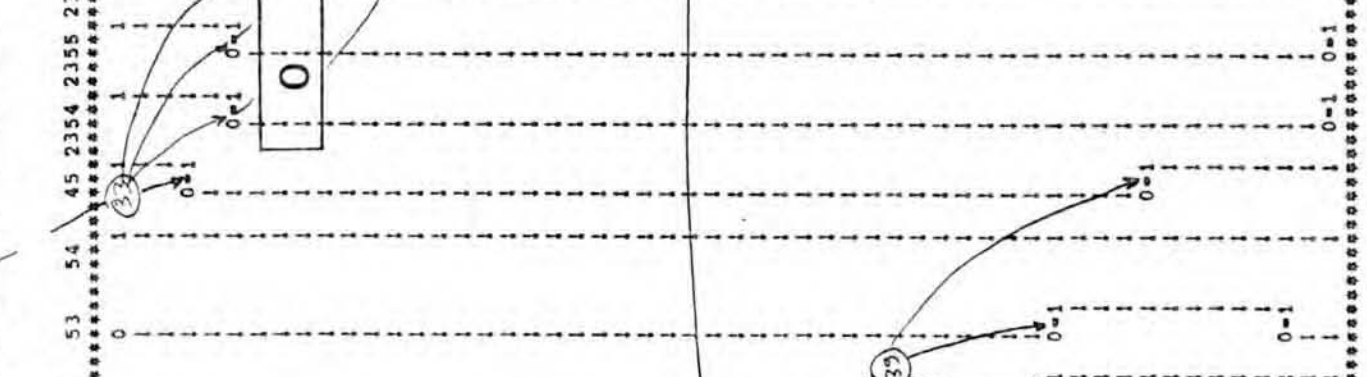
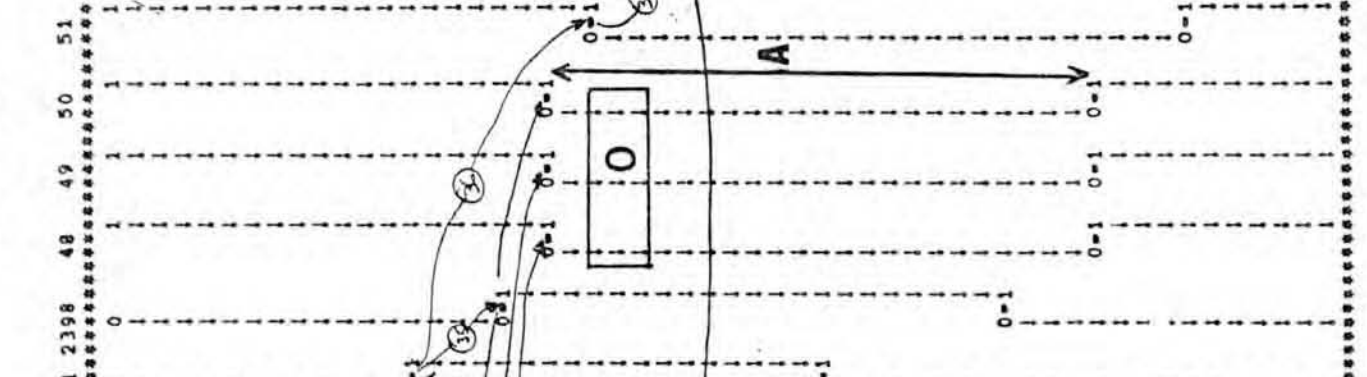
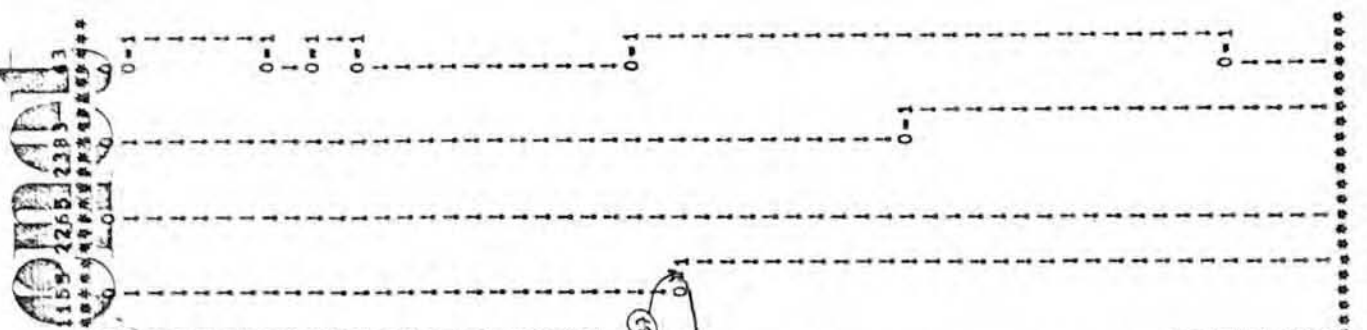
6.3.2. (SUITE)

LE 29/84/ 8 PAGE N 7

EPISEDE - SIMULATION DE LA MAQUETTE A MULTIBUS



LE 29/84/ 8 PAGE N 7



6.3.3. REMARQUES

De même qu'au § 6.2.3., on remarque :

- . période BCLK de 400 ns
- . la fonctionnalité de supervision est aussi mise à profit,
- . les contraintes MULTIBUS sont respectées ; soit le verdict est délivré par les ABC 90, soit par les circuits encodeur-décodeur.

6.4. RÉSULTATS POUR L'ARCHITECTURE CENTRALISÉE VME-ABC 90

6.4.1. CARACTÉRISTIQUES DE LA SIMULATION

Pour cette architecture, les simulations qu'on a faites ont les caractéristiques suivantes :

- . trois unités de traitement, chacune contenant une unité de demande d'accès au bus VME.
- . trois arbitres de bus sur la carte centrale contrôleur du système configurés aux adresses 0, 1 et 2 selon le même ordre des demandes BA 17, BA 18 et BA 19. Ces arbitres ne font pas de demandes (DAB câblé à 0).
- . la première UT est configurée en priorité fixe, tandis que les deux autres sont en rotative (PF-001).
- . on ne permet que des arbitrages visibles, broche BPAC câblée à 0 sur chaque ABC 90.
- . au début de la séquence de simulation, tous les ABC 90 ont été initialisés convenablement.
- . les séquences de demande de bus sont décrites dans la figure A6.5 (voir page suivante).

Pour le deuxième jet de chronogrammes, on simule l'intervention de la fonction de supervision, pour rendre des arbitrages, même s'il n'y a pas d'ABC 90 sur la carte contrôleur du système configuré à l'adresse 0. On simule des adresses UC1 = 001, UC2 = 010, et UC3 = 011 ; PF restant encore à 001.

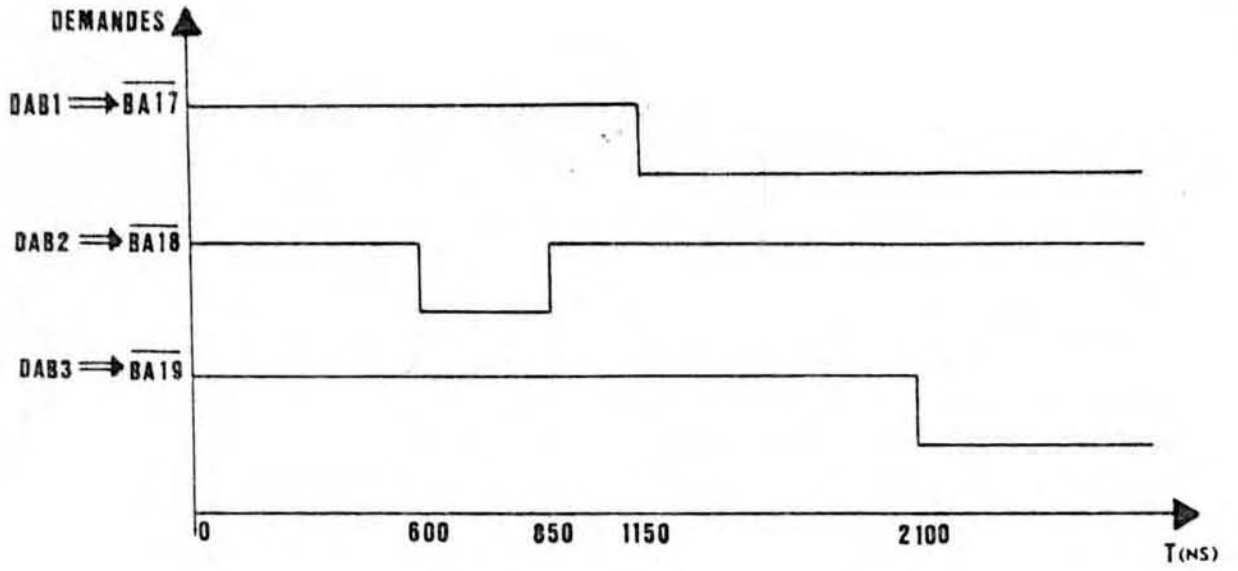


Figure A6.5 - Séquences de demande de bus.

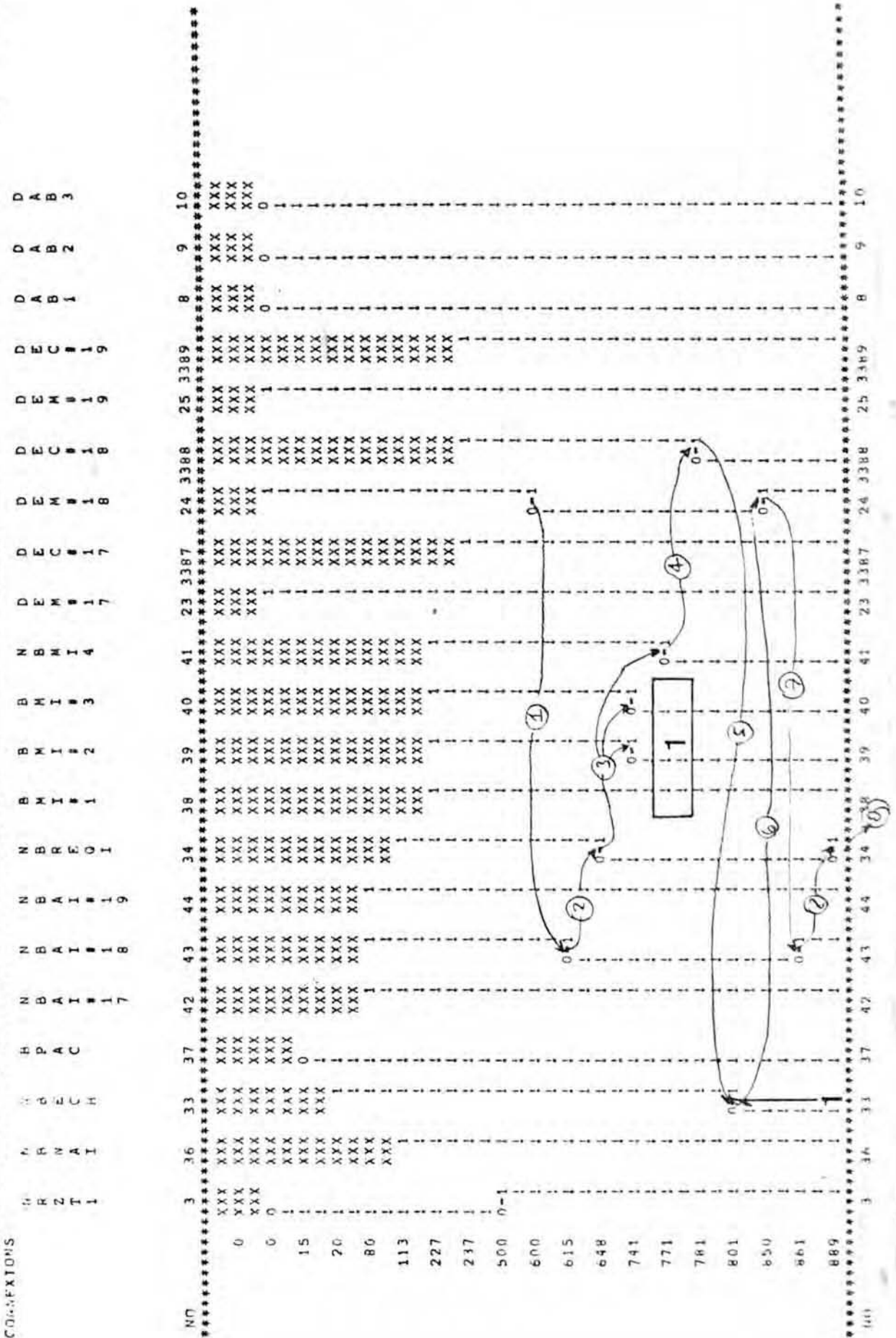
6.4.2. CHRONOGRAMMES EPIL DE VALIDATION DE L'ARCHITECTURE CENTRALISÉE VME-ABC 90

6.4.2.A. CHRONOGRAMMES CORRESPONDANT AUX STIMULIS DÉCRITS EN 6.4.1.

LE 10/04/ 6 PAGE N 4

EPISODE SIMULATION DE LA MAQUETTE C AVEC VME

CHRONOGRAMME DU CIRCUIT VCSM90

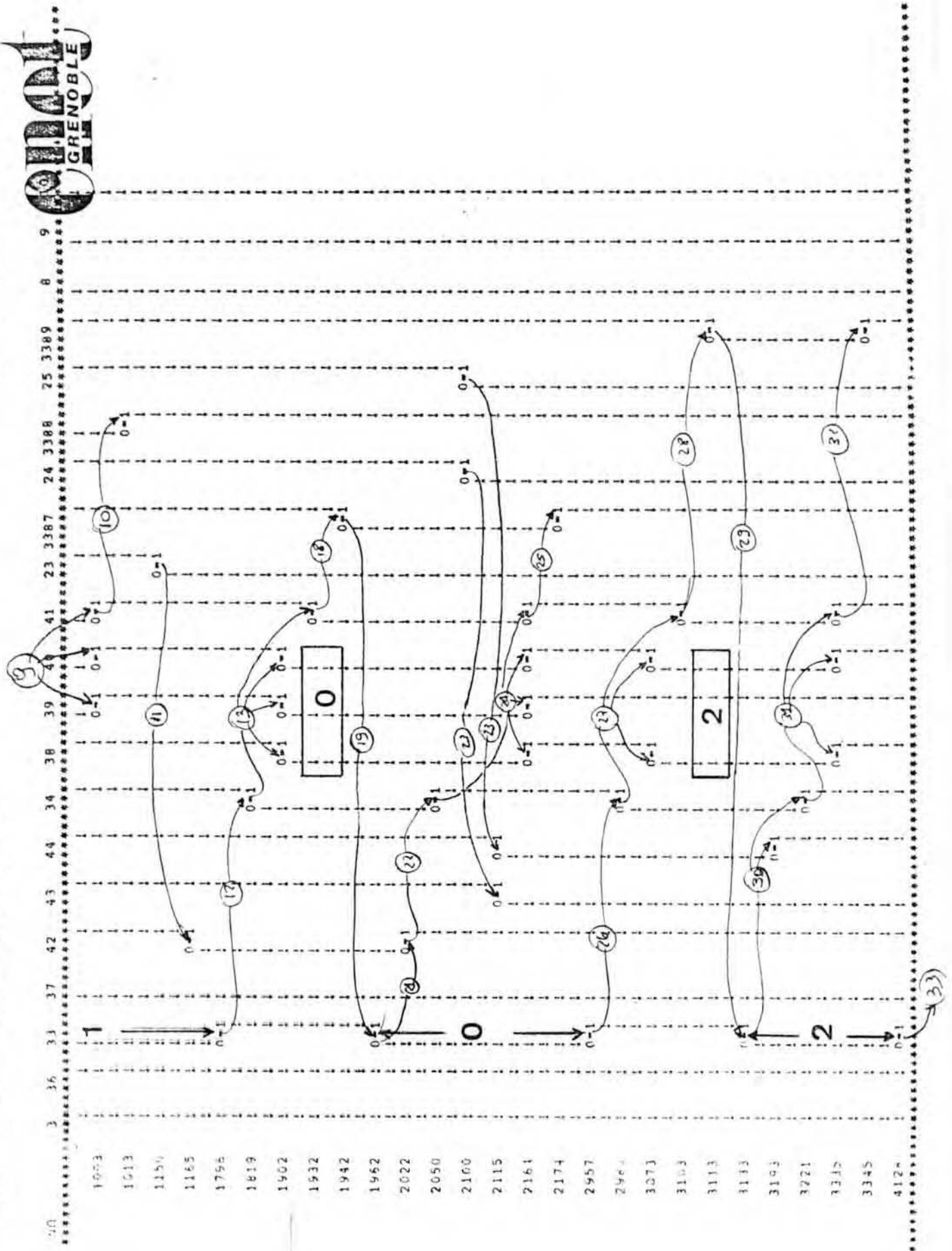


6.4.2.A. (SUITE)

LE 11/84/ 6 PAGE N 5

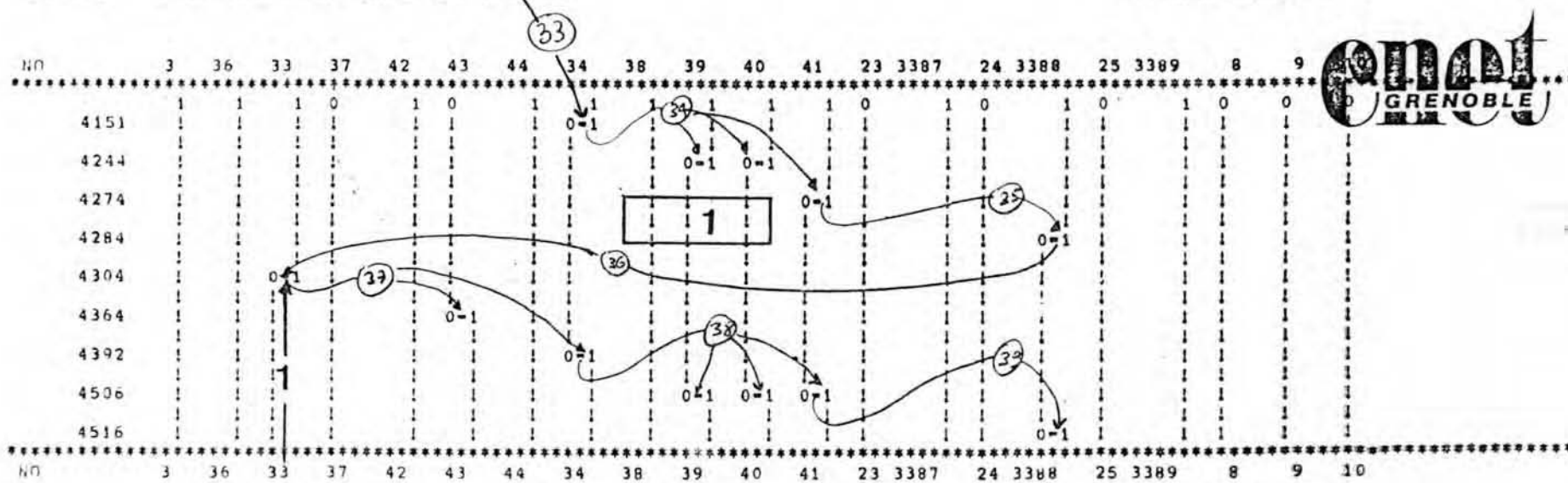


EPISODE SIMULATION DE LA MAQUETTE C AVEC VME



EPISODE SIMULATION DE LA MAQUETTE C AVEC VME

LE 18/84/ 6 PAGE N 6



PHASE **FUTURU TEMPS CPU : 211,00 SECONDES

6.4.2.A. (SUITE)

6.4.2.B. EXEMPLE D'INTERVENTION DE LA
FONCTIONNALITÉ DE SUPERVISION : PANNE OU INEXISTENCE
D'UN ARBITRE ABC 90 À L'ADRESSE 0

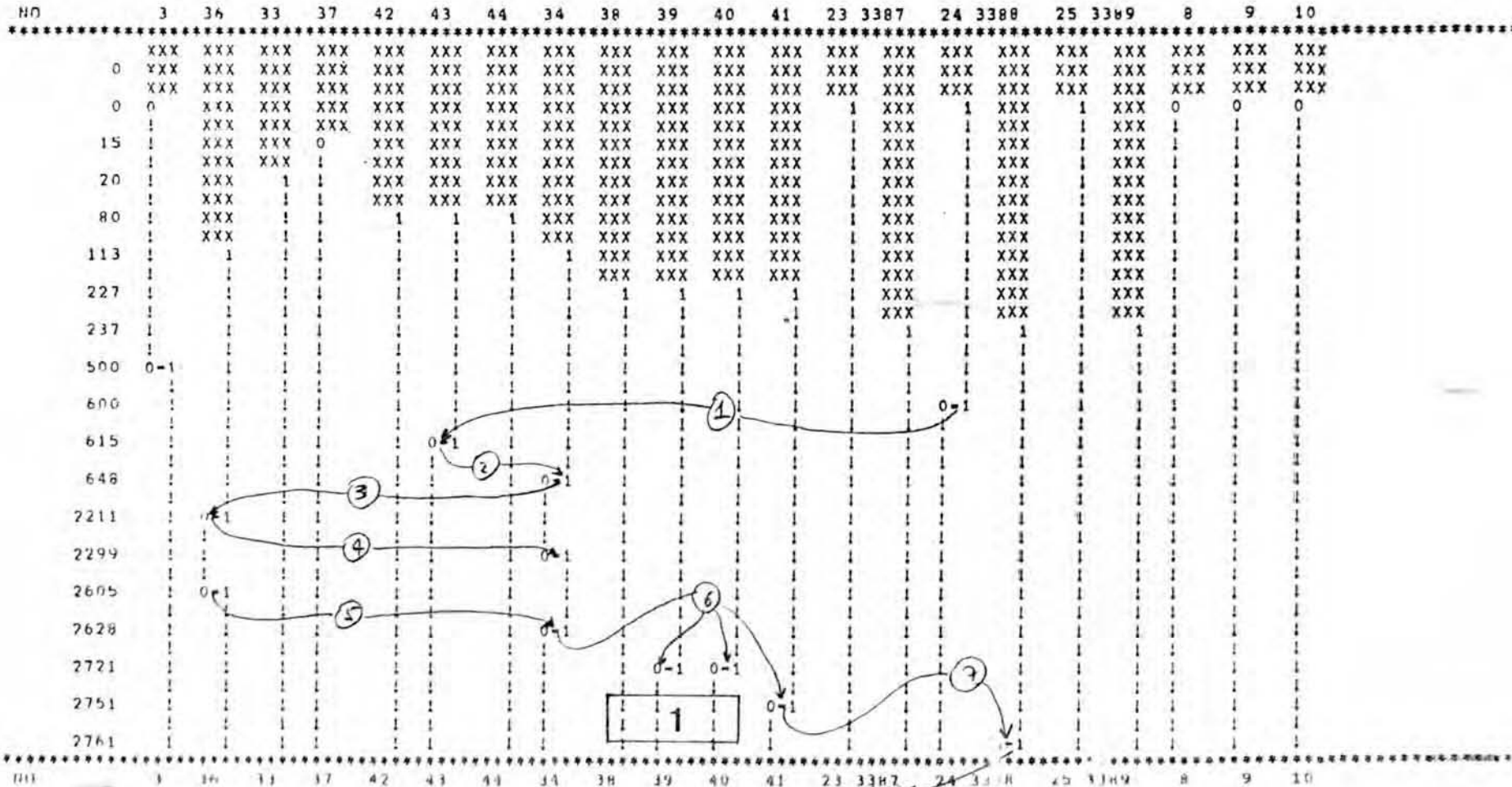
CHRONOGRAMME DU CIRCUIT VCSH90



CONNEXIONS

H	N	H	B	N	N	N	N	B	B	B	N	D	D	D	D	D	D	D	D	D
R	R	B	P	B	B	B	B	M	M	M	B	E	E	E	E	E	E	A	A	A
Z	N	E	A	A	A	A	R	I	I	I	M	M	C	M	C	H	C	B	B	B
T	A	C	C	I	I	I	E	#	#	#	I	#	#	#	#	#	#	1	2	3
1	I	H		#	#	#	Q	1	2	3	4	1	1	1	1	1	1	1	1	1
				1	1	1	I					7	7	8	8	9	9			
				7	8	9														

6.4.2.B.



EPISODE 5 SIMULATION DE LA MAQUETTE C AVEC VME

LE 13/84/ 6 PAGE N 5

NO	3	36	33	37	42	43	44	34	38	39	40	41	23	3387	24	3388	25	3389	8	9
2781	1	1	1	0	1	0	1	0	1	0	0	0	1	1	0	0	1	1	0	0
2841						0-1														
2869							0-1													
2983			1							0-1	0-1	0-1								
2993																				
3776			0-1																	

PHASE **EDITTO TEMPS CPU : 132,00 SECONDES



6.4.2.B. (SUITE)

6.4.3. REMARQUES

Comme on peut constater à travers le chronogramme VME montré en annexe 2, les chronogrammes EPILOG montrés révèlent que l'architecture proposée est valide.

Le temps d'arbitrage (entre \overline{BAT} et DEC) se situe de l'ordre de 160 ns.

On remarque aussi que tous les arbitrages ont été visibles.

6.5. RÉSULTATS POUR L'ARCHITECTURE DÉCENTRALISÉE VME - ABC 90

6.5.1. CARACTÉRISTIQUES DE LA SIMULATION

Pour cette architecture, les simulations qu'on a faites ont les caractéristiques suivantes :

- . deux unités de traitement configurées en cartes spéciales, contenant chacune un ABC 90. Les UT ont les adresses 0 et 1. Les demande formulées par ces UT correspondent à DAB1 et DAB2.

- . l'unité de traitement configurée en carte standard VME contenant une unité de demande d'accès au bus. L'adresse de cette UT correspond à 2. Leur demande d'accès au bus (BR1) correspond à DEM# 19 sur les chronogrammes EPILOG.

- . l'unité de traitement 0 est configurée en priorité fixe, tandis que les deux autres sont configurées en rotative (PF = 001).

- . on ne permet que des arbitrages visibles, broche BPAC câblée à 0 sur les ABC 90 des UT 0 et 1.

- . au début de la séquence de simulation, toutes les UT ont été convenablement initialisées.

Les séquences de demande de bus sont décrites dans la figure A6.6 (voir page suivante).

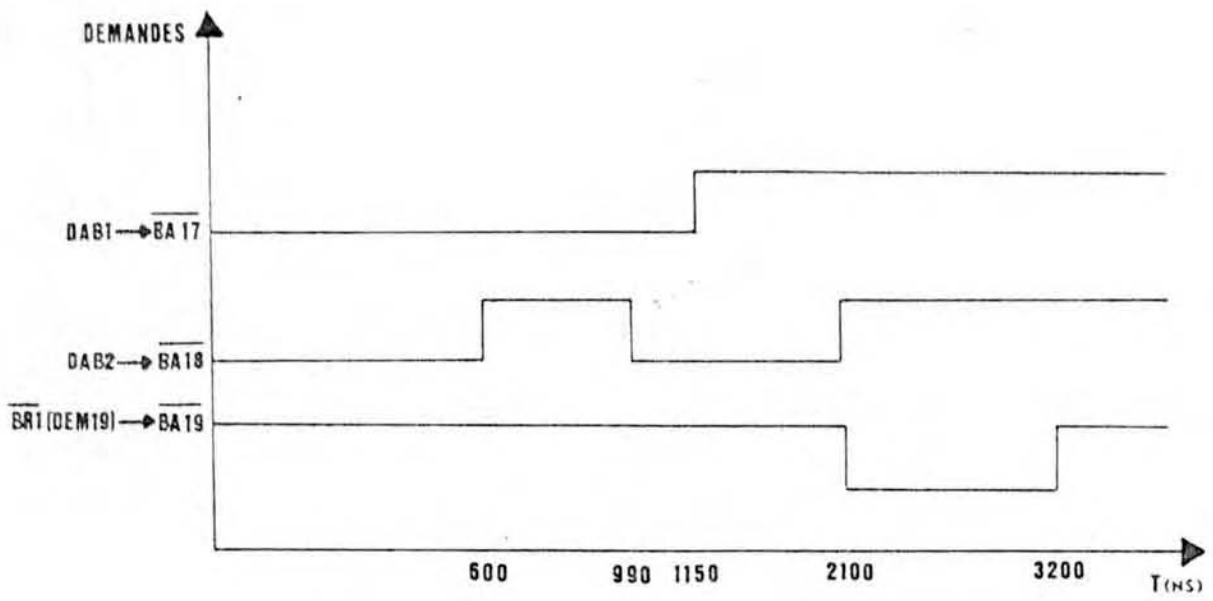


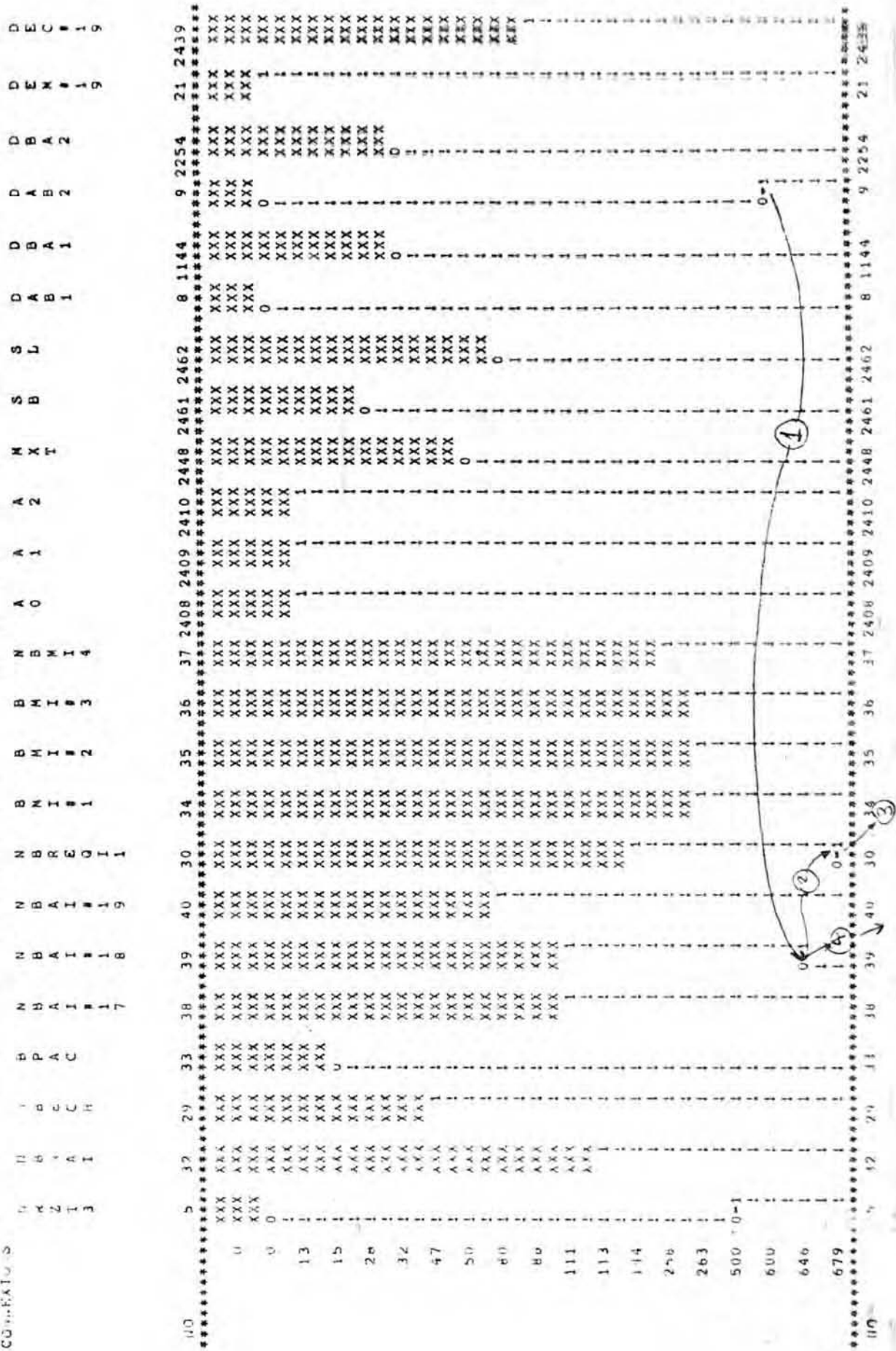
Figure A6.6 - Séquences de demande de bus.

6.5.2. CHRONOGRAMME EPILOG DE VALIDATION DE L'ARCHITECTURE DECENTRALISEE VME-ABC 90

LE 6/84/ 7 PAGE N 4

EPISODE 5 SIMULATION DE LA MAQUETTE A AVEC VME

CHRONOGRAMME DU CIRCUIT VASM90

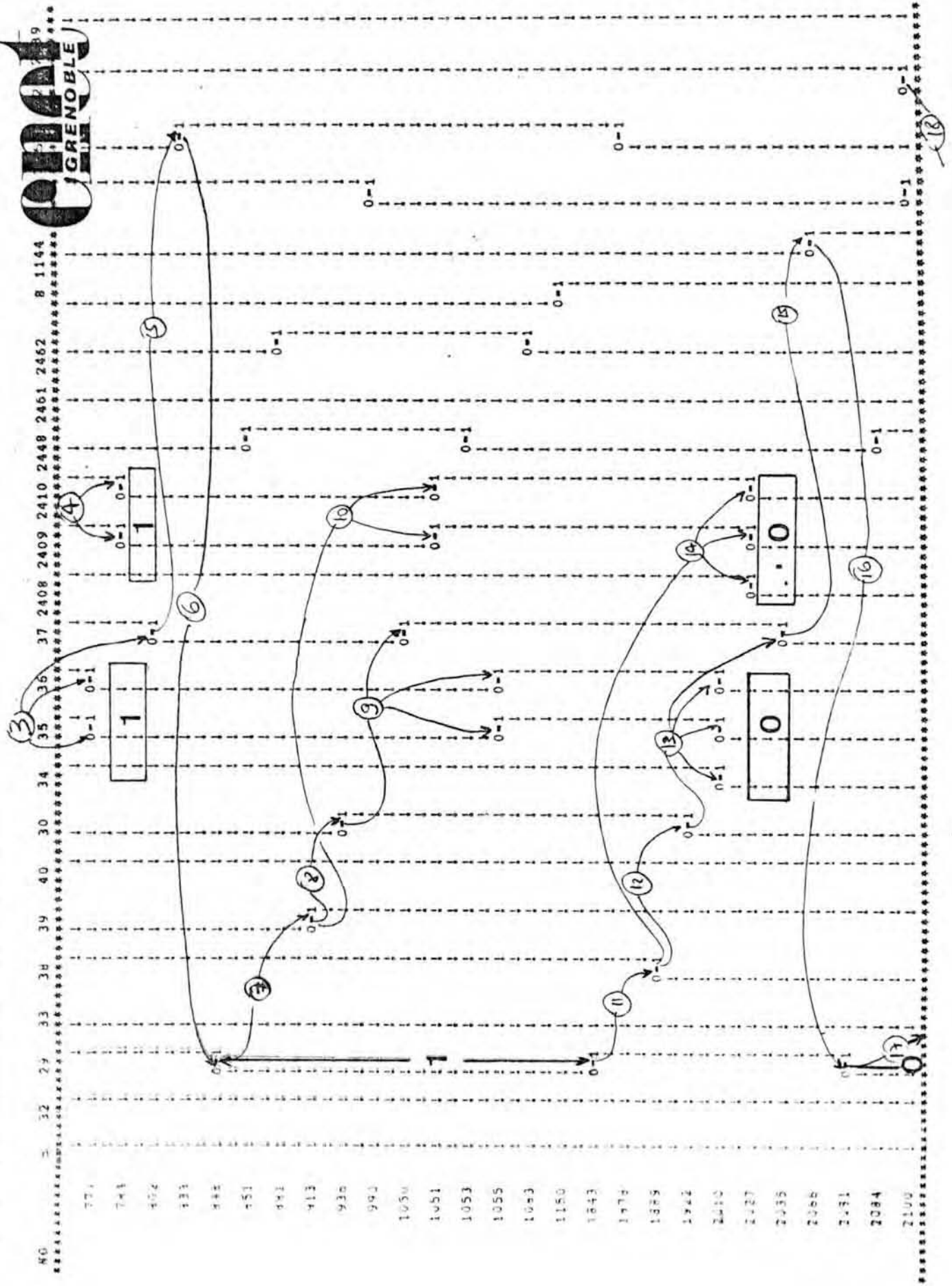


6.5.2. (SUITE)

LE 6/84/ 7 PAGE N 5



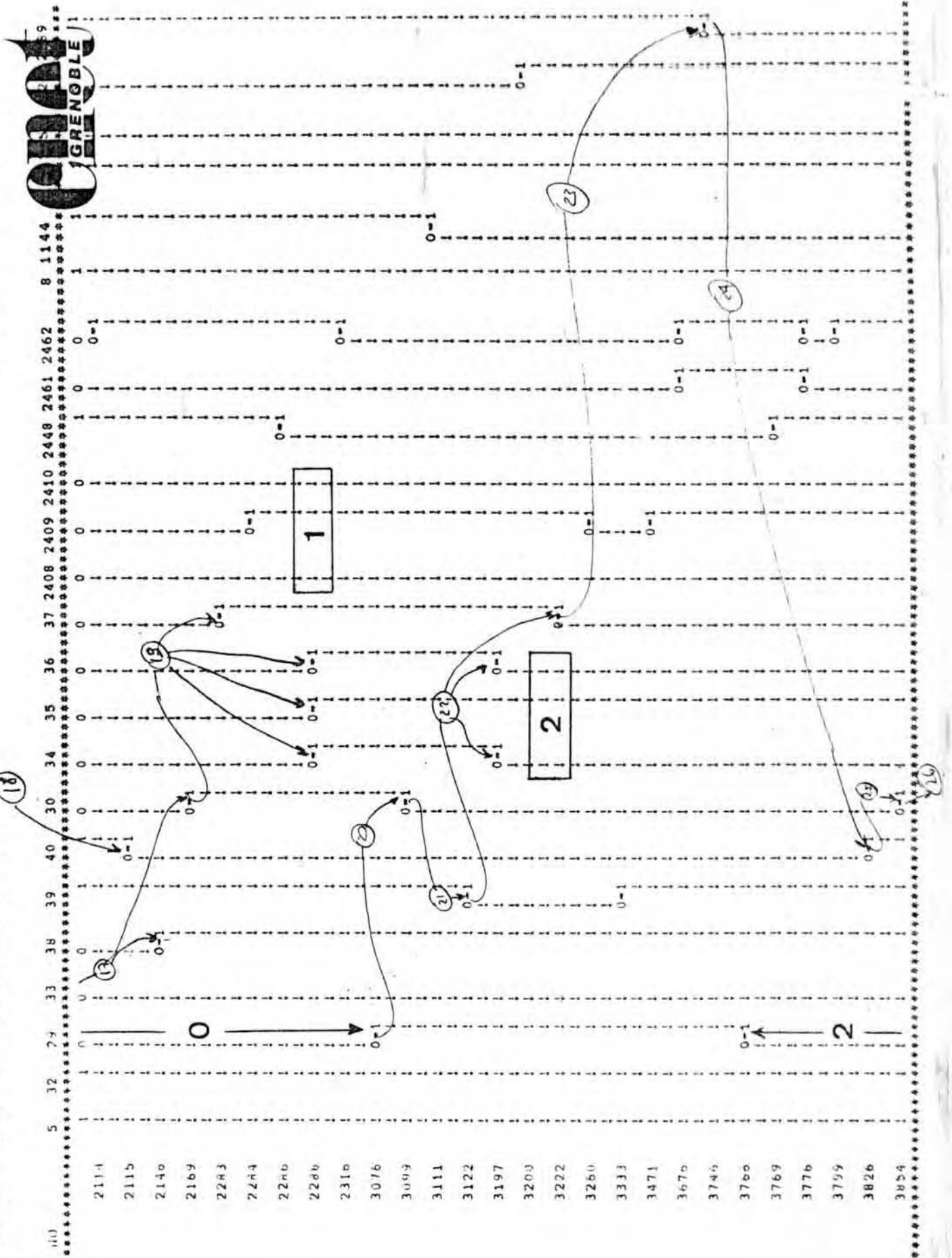
EPISODE 5 SIMULATION DE LA MAQUETTE A AVEC VME



6.5.2. (SUITE)

LE 6/04/7 PAGE N 6

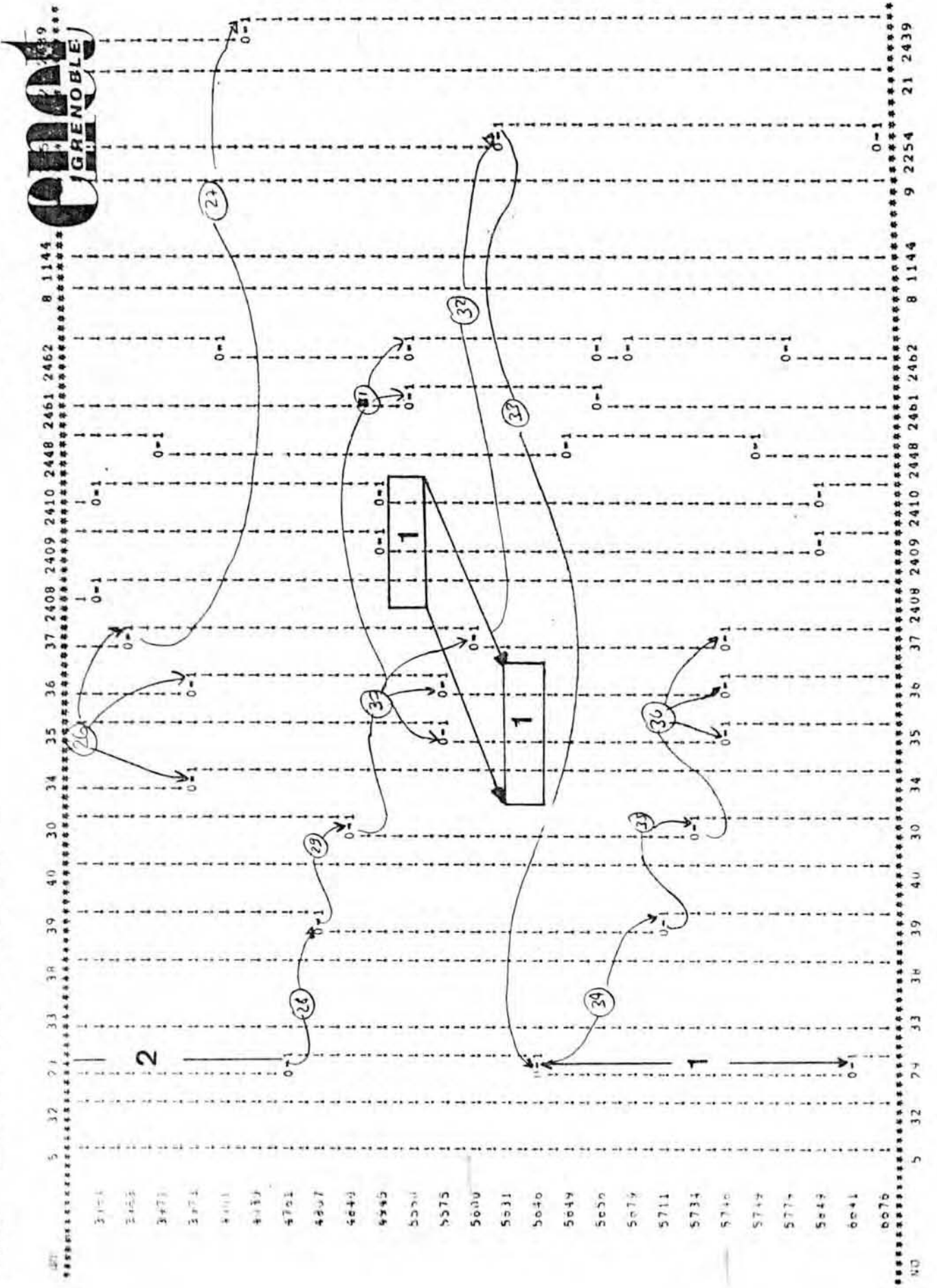
EPISODE 5 SIMULATION DE LA MAQUETTE A AVEC VME



6.5.2. (SUITE)

LE 6/84/ 7 PAGE N 7

EPISODE 5 SIMULATION DE LA MAQUETTE A AVEC VME



6.5.3. REMARQUES

D'après les chronogrammes EPILOG, on constate aussi que l'architecture présentée au § 2.4.3.2. a été validée. Les contraintes VME sont respectées pour les arbitrages rendus par les ABC 90 ou par les circuits encodeur-décodeur.

ANNEXE 7

RESULTATS DES SIMULATIONS ET DE L'IMPLANTATION
DE L'ABC-M

7.1. VALIDATION DES MODIFICATIONS APPORTÉES À L'ABC 90 LE TRANSFORMANT EN CIRCUIT MULTIPROTOCOLES (ABC-M)

Dans cette annexe on valide le circuit ABC-M, à travers différentes architectures multi-processeurs (SM 90 MULTIBUS et VME) décrites dans le chapitre 3.

Comme on l'a vu au § 3.4., l'outil de CAO le plus adapté à cette simulation, c'est EPILOG. De même que pour la validation des architectures discrètes (voir A6.1.1.), l'utilisation du simulateur logico-fonctionnel FIDEL se fait nécessaire pour la description du circuit 8289 (MULTIBUS) et unité de demande d'accès au bus VME (UDABVME) (vérification des modes de fonctionnement RWD et ROR).

Les modifications de l'ABC-M étant complexes, on a dû les valider séparément (voir § 3.4.2.). Chacune des nouvelles logiques introduites (voir § 3.3.1., 3.3.2., 3.3.3. et 3.3.4.) ont été directement décrites sous EPILOG.

Ces modifications, décrites sous la forme de circuits logiques dans la terminologie EPILOG (voir § A5.3.2.), constituent des blocs qui feront partie de la description de l'ABC-M.

Dans la figure A7.1, on montre la description EPILOG de l'ABC-M (on marque avec un croix les parties qui ont été modifiées ou introduites par rapport à la description de l'ABC 90).

```

TOPOLOGIE DU CIRCUIT ABC-M ( AVEC LES PF )
** BIBLIOTHEQUE
* EFFACER
  ABCM
** DESCRIPTION
* TOPOLOGIE
ABC (F, VRAI, NRZT, NBREQI, NBREQ2, DAB, MBNAI, BPAC, BMT#1:3, NBMI4, NBAI#17:24)
  , UC#1:3, PF#1:3, PE1, PE2-DIA, NBAO#17:24, NDAC#17:24;
  , NBREQ0, MBNAI, BMO#1:3, BMO#4;
  , S20, S21, S22, S23, S24;
  , S30, S40, S11, S12, S13, S10;
  , S50, S51, S52;
  , S02, S03, S04, S05, S06, S07;
  , S71, S72, S73)
S
PC1=MAFFI (F, NBREQ, NBMI4, BMO, MBNAI, NRZT, DAB, BPAC, DIA;
  -NVALDEM1, S20, S21, S22, S23, S24)
PC2=ARRI (F, NBREQ, DIA, MBNAI, NRZT, BPAC, NBREQI;
  -CLPAR, PLUS1, NVAL, LOAD, S30, S40, S11, S12, S13, S10)
PC3=VERD (F, NBREQ, DIA, BMO, MBNAI, NRZT-DIA, S50, S51, S52)
PC4=SUPE (F, NBREQ, NRZT, MBREQ1, KPO, KPI, MBNAI-RZKP, NBNAO, DCKPXX,
  S02, S03, S04, S05, S06, S07)
PC5=DECL (F, DIA, MBNAI, NBREQ, BPAC, MBNAI, NRZT-NVALDEM2, S71, S72, S73)
S
OP1=NAFOP (NVALDEM1, NVALDEM2, UC#1:3-NBREQ0, NBAO#17:24)
OP2=NAROP (PE1, PE2, PLUS1, LOAD, CLEAR, NVAL, BMT#1:3, UC#1:3, PF#1:3,
  MBNAI#17:24, VRAI, F-BMO#1:3, NBMO4, NDAC#17:24)
OP3=VROP (BMT#1:3, UC#1:3-BMO)
#####
RAMX=RAMTE (DCKPXX-DCKP)
OP4=SPOP (DCKP, RZKP-KPO, KPI)
OP5=DECLOP (NBAI#17:24-BMO)
M12
**FIN

```

PARTIE CONTROLE

PC1: AFFICHAGE
PC2: ARBITRAGE
PC3: RETRANSM. VERDICT
PC4: SUPERVISION

PARTIE OPERATIVE

OP1: AFFICHAGE
OP2: ARBITRAGE
OP3: RETRANSM. VERDICT
RAMX: EQUIP. SUPERV.
OP5: DECLENCHEMENT D'ARBITRAGE

Figure A7.1 - Description EPILOG de l'ABC-M
On note avec une croix les "blocs" EPILOG qui ont été introduits ou modifiés par rapport à la description de l'ABC 90.

Dans la suite, on montre la validation de chacune des modifications introduites.

7.1.1. MODES DE FONCTIONNEMENT DU CIRCUIT

Cette modification a été décrite au § 3.1. du chapitre 3.

Elle a été décrite sous EPILOG, sous le nom DECOD(sous-partie de la partie opérative d'arbitrage nommée NAROP (voir figure A7.1) dans le sens EPILOG).

Au niveau de la description, on a décrit un décodeur 2 x 4, tandis qu'au niveau de l'implantation (voir figure 32) on n'a besoin que de deux de ces états en sortie (NM1 et NM4).

Le résultat de simulation EPILOG est montré à la figure A7.2 :

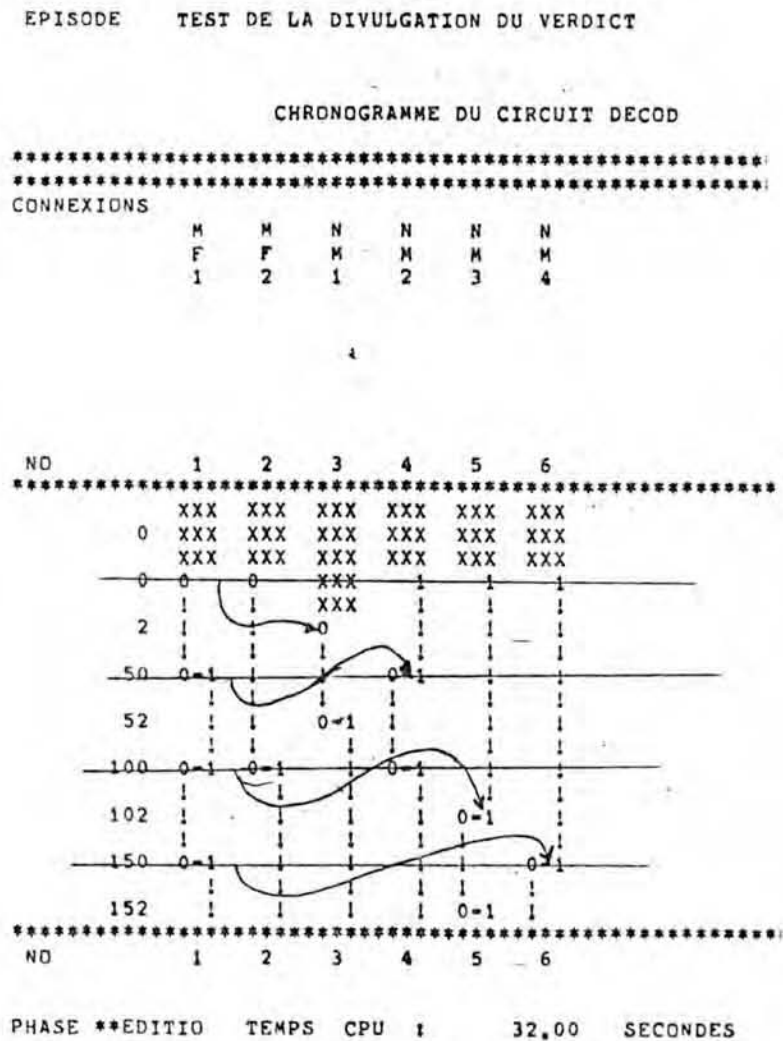


Figure A7.2 - Chronogramme EPILOG de validation liée aux modes de fonctionnement de l'ABC-M (voir § 3.3.1.)

7.1.2. DIVULGATION SYSTÉMATIQUE DU VERDICT

Cette modification a été décrite au § 3.2. du chapitre 3.

Elle a été décrite sous EPILOG, sous le nom DIVL (sous-partie de la partie opérative d'arbitrage nommée NAROP (voir figure A7.1.) dans le sens EPILOG).

Le résultat de la simulation EPILOG est montré à la figure A7.3 :

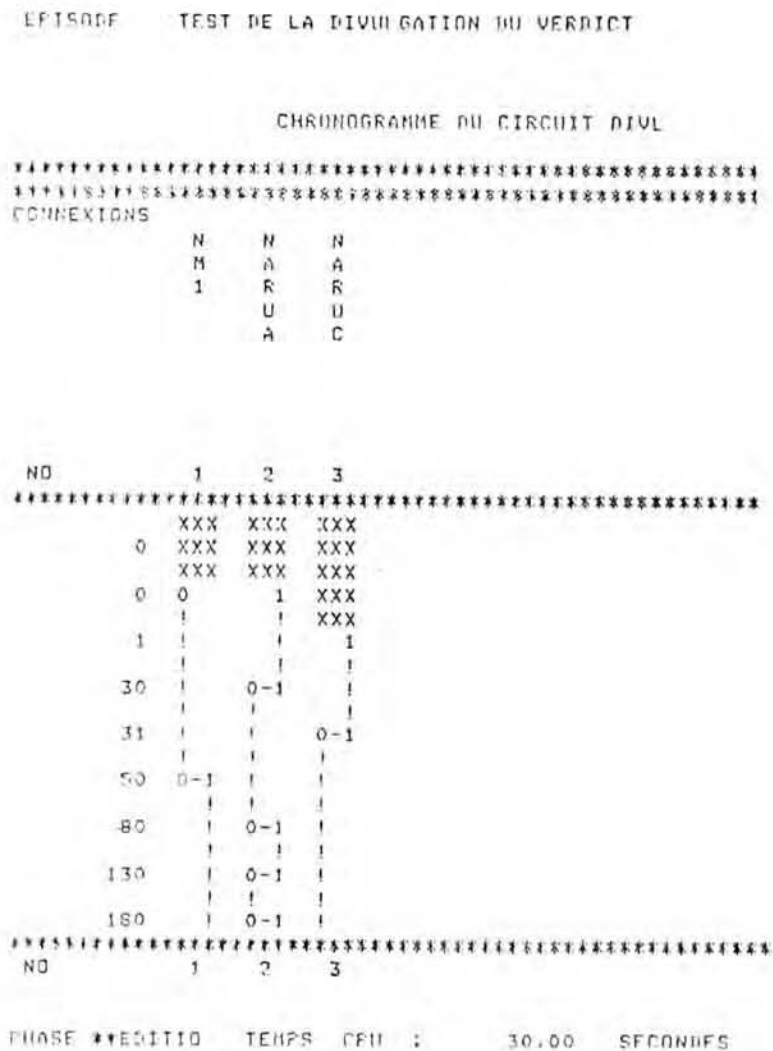


Figure A7.3 - Simulation EPILOG de la divulgation systématique du verdict.

D'après cette figure, on constate que pour le mode de fonctionnement SM 90 (NMI actif bas), le signal de validation de divulgation du verdict NARUC suit le signal correspondant NARUA délivré par l'ABC 90. Pour tous les autres modes de fonctionnement (NMI inactif haut) NARUC reste actif (bas).

7.1.3. LOGIQUE SUPPLÉMENTAIRE DE DIVULGATION DU VERDICT DE FORME DÉCODÉE

Cette modification a été la plus difficile à mettre au point, dû à notre décision de ne pas rajouter d'autres retards au circuit.

Comme on peut constater d'après la figure 34 (chapitre 3), cette modification se constitue en un assemblage de plusieurs parties logiques distinctes :

- * décodeur 3 x 8 (bloc DECOD sur EPILOG) :
 - . entrées BM01, BM02, BM03
 - . sorties:ND 0-7 } (figure A7.4)
- * générateur de deux phases d'horloge non-recouvrantes (bloc GEN2PH) sur EPILOG) :
 - . entrée NBM04
 - . sorties NH, H
- * latch RS (à partir de portes ETNON). Ce latch, dont la description EPILOG est montrée à la figure A7.4, a le nom LTCH :
 - . entrées NCLEAR, H
 - . sorties Q (NMULT) et \bar{Q} (NMULTB)
- * décodeur 2 x 4, dont le chronogramme de validation a été montré à la figure A7.2. Ce décodeur des modes de fonctionnement du circuit a été mis aussi sur la partie opérative d'arbitrage :
 - . entrées MF1, MF2
 - . sorties NM1, NM2, NM3, NM4
- * sélecteur (SORTIE \overline{ASEL} + BSEL) lequel a été montré en A.5.5. Ce multiplexeur est décrit sous EPILOG sous le nom SLCT :
 - . entrées A(NMULTB), B (NVME) et SEL (NM4)
 - . sortie NVAL
- * registre maître/esclave 8 bits, lequel a été décrit et simulé sous EPILOG sous le nom RME8 :
 - . entrées ND : 0-7, H, NH
 - . sorties NS : 0-7
- * ensemble de portes OUNON et d'inverseurs, pour la divulgation vers l'extérieur (broches \underline{NS} de l'ABC-M) du verdict décodé :
 - . entrées NS : 0-7
 - . sorties NDAC : 0-7

Remarque :

C'est NDAC : 0-7 (sous EPILOG) qui correspond aux broches NS de l'ABC-M.

Toutes ces parties ont été validées séparément ; pour ne pas charger ce document, on ne montre que le résultat de simulation du circuit EPILOG SRTVM2 (qui regroupe tous ces blocs) lequel décrit la logique supplémentaire de divulgation du verdict de forme décodée spécifique à l'ABC-M (voir § 3.3 du chapitre 3, figure 34).

SRTVM2 devient un bloc pour le circuit NAROP (partie opérative d'arbitrage) de l'ABC-M.

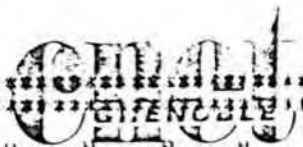
```
DESCRIPTION DE LA SORTIE
**RÉRIOTHÉQUE
**EFFACER
LTCM
SRTVM2
*
**DESCRIPTION
**TOPOLOGIE
LTCM(R,S-Q,OR)
Z1-ETHON(R,S2-S1)
Z2-ETHON(S,S1-S2)
R1-RAMIF(S1-Q)
R2-RAMIF(S2-QR)
FIN
*
**SYNTHÈSE
SRTVM2(CF1,CF2,CLEAR,RM01,RM02,RM03,NRM04-NDAC10;7)
Z1-ETHON(CLEAR-HCLEAR)
Z2-DECODR(RM01,RM02,RM03-ND10;7)
Z3-DECODPH(RM04-NH,H)
L4-ORNON(CLEAR,H-NVHF)
L5-LTCM(CLEAR,H-NMULT,NMULTB)
L6-DECOD(CF1,CF2-NM1,NM2,NM3,NM4)
L7-SLCT(NMULT,NVHF,NM4-NVAL)
L8-RSEH(ND10;7,H,NH-NS10;7)
L9-ETHON(NVAL-VAL)
R170;7/ONON(NS1,VAL-HDC1)
R270;7/ETHON(HDC1-NDAC1)
FIN
**FIN
```

Figure A7.4 - Description EPILOG de la logique supplémentaire de divulgation du verdict de forme décodée spécifique à l'ABC-M (circuit SRTVM2)

Les résultats de simulation de ce circuit sont montrés à la figure A7.5. Dans cette simulation, on valide cette logique dans le contexte où l'ABC-M marche dans le mode RELEASE ON REQUEST (ROR) car NM4 est à 0 (voir § 3.3).

CHRONOGRAMME DU CIRCUIT SRTVH2

 CONNEXIONS



C R B P N
 L H M H B
 E O O O M
 A 1 2 3 0
 R 4

SIGNAUX DELIVRES PAR
 L'ABC 90

SIGNAUX SPECIFIQUE DE L'ABC - M

NO	3	4	5	6	7	28	29	36	34	42	45	17	18	19	20	49	50	51	52	8	9	10	11	
0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
0	1	1	1	1	1	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
1						XXX	XXX	XXX	0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
2						XXX	XXX	XXX		0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
6						0	XXX	XXX			XXX	1	1	1	1	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
10							XXX	XXX			XXX					XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
12								0			XXX					XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
14											0					XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
21																XXX	XXX	XXX	XXX	XXX	1	1	1	1
50	0-1															XXX	XXX	XXX	XXX	XXX				
80		0-1	0-1	0-1												XXX	XXX	XXX	XXX	XXX				
86																XXX	XXX	XXX	XXX	XXX				
130																XXX	XXX	XXX	XXX	XXX				
137																XXX	XXX	XXX	XXX	XXX				
138																XXX	XXX	XXX	XXX	XXX				
139																XXX	XXX	XXX	XXX	XXX				
141																XXX	XXX	XXX	XXX	XXX				
142																XXX	XXX	XXX	XXX	XXX				
143																1	1	1	1	1				

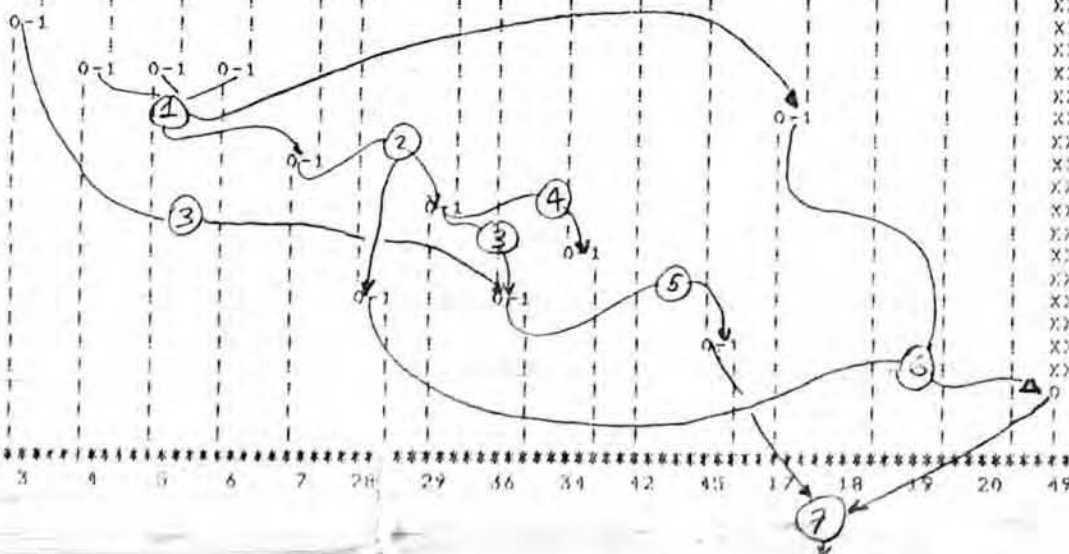


FIGURE A.7.5. - CHRONOGRAMME EPILOG DE VALIDATION
 DE LA DIVULGATION DU VERDICT DE
 FORME DÉCODÉE

7

NO	3	4	5	6	7	28	29	36	34	42	45	17	18	19	20	49	50	51	52	8	9	10	11	
198																								
200		0-1	0-1	0-1	0-1																			
202													0-1	0-1	0-1									
208						0-1						0-1	0-1	0-1	0-1									
210							0-1																	
211									0-1															
280		0-1	0-1																					
330					0-1																			
337							0-1																	
338									0-1															
339						0-1																		
343																0-1								
344																					0-1			



PHASE #MEDICID TEMPS CPU : 123.00 SECONDES

FIGURE A.7.5. - (SUITE)

7.1.4. LOGIQUE ASSOCIÉE À L'AUTOMATE DE DÉCLENCHEMENT D'ARBITRAGE

7.1.4.1. VALIDATION DE LA GÉNÉRATION D'UN SIGNAL DE DEMANDE GÉNÉRALE DE BUS

La modification associée à l'automate de déclenchement d'arbitrage, entraîne - comme on a vu au § 3.4. du chapitre 3 - le besoin d'introduire une logique de demande générale de bus (DBUS) (DECLOP sur EPILOG) montrée à la figure 35 du chapitre 3, ainsi que l'introduction d'une logique pour combiner une sortie de l'automate de déclenchement et de l'automate d'affichage (NVALDEM2 et NVALDEM1, respectivement) pour former le signal NVALDEM nécessaire à l'ABC-M.

A l'égard de la validation de la partie liée au déclenchement d'arbitrage, la première validation qu'on a faite a été du circuit DECLOP (partie opérative de déclenchement) (voir figure A7.1).

Sur la figure A7.6, on montre le chronogramme EPILOG obtenu. On remarque que, dès qu'il y a une demande d'accès active présente sur le bus (signaux NE : 0-7, actifs bas), le signal de demande de demande générale de bus DBUS devient actif (haut).

Remarque :

On a choisi pour DBUS un signal de même polarité que DAB (demande locale d'accès au bus).

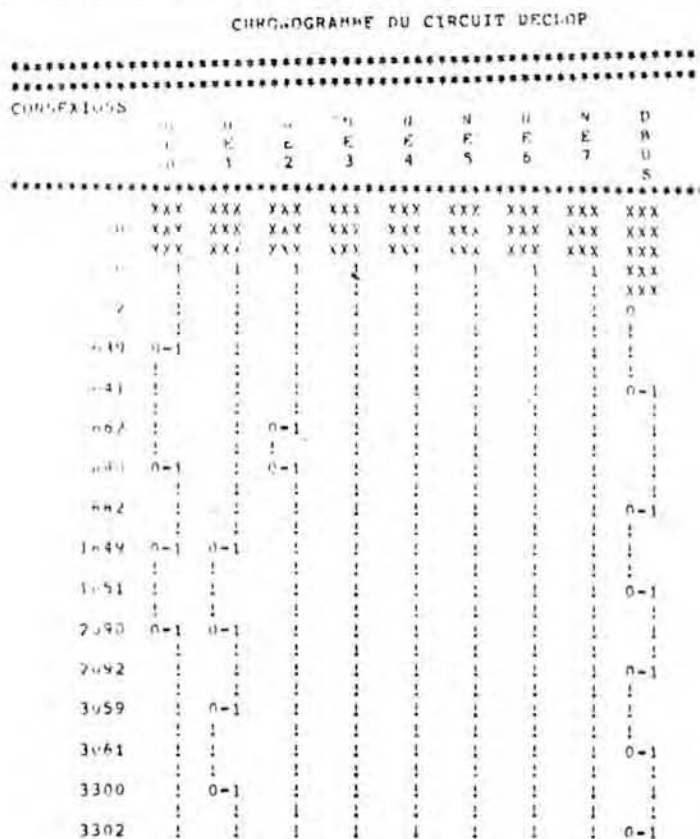


Figure A7.6 - Chronogramme EPILOG pour la validation de la génération d'un signal de demande générale de bus.

7.1.4.2. VALIDATION DE L'AUTOMATE DE DÉCLENCHEMENT D'ARBITRAGE

La description EPILOG de l'automate tient compte de la méthode de synthèse sur silicium (voir § 7.2.).

Le résultat de la simulation de l'automate tout seul est montré à la figure A7.7.

Bien entendu, la vraie validation (parallélisme entre automates, interface, partie opérative-partie contrôle) n'est obtenue que lorsqu'on simule l'ABC-M entier.

CHRONOGRAMME DU CIRCUIT DECL

CONNEXIONS

D	H	M	H	H	R	N	S	S	S
U	M	F	P	N	Z	V	7	7	7
S	4	C	A	A	T	A	1	2	3
		H	C			L			
						D			
						E			
						M			
						2			

NO	2	13	14	5	16	17	8	9	10	11
0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
0	0	XXX	XXX	1	XXX	XXX	XXX	XXX	XXX	XXX
2	!	u	0	!	XXX	XXX	XXX	XXX	XXX	XXX
2	!	!	!	!	XXX	XXX	XXX	XXX	XXX	XXX
3	!	!	!	!	0	XXX	XXX	XXX	XXX	XXX
6	!	!	!	!	!	!	XXX	XXX	XXX	XXX
6	!	!	!	!	!	!	XXX	XXX	XXX	XXX
7	!	!	!	!	!	!	XXX	0	XXX	XXX
7	!	!	!	!	!	!	XXX	!	XXX	XXX
8	!	!	!	!	!	!	XXX	!	XXX	!
13	!	!	!	!	!	!	XXX	!	XXX	!
13	!	!	!	!	!	!	XXX	!	!	!
14	!	!	!	!	!	!	!	!	!	!
14	!	!	!	!	!	!	!	!	!	!
506	!	!	!	!	!	0-1	!	!	!	!
506	!	!	!	!	!	!	!	!	!	!
639	0-1	!	!	!	!	!	!	!	!	!
639	!	!	!	!	!	!	!	!	!	!
651	!	!	!	!	!	!	!	0-1	!	!
651	!	!	!	!	!	!	!	!	!	!
652	!	!	!	!	!	!	!	!	0-1	!
652	!	!	!	!	!	!	!	!	!	!
653	!	!	!	!	!	!	0-1	!	!	!
653	!	!	!	!	!	!	!	!	!	!
778	!	0-1	!	!	!	!	!	!	!	!
778	!	!	!	!	!	!	!	!	!	!
782	!	!	!	!	!	!	!	!	!	0-1
782	!	!	!	!	!	!	!	!	!	!
785	!	!	!	!	!	!	!	!	0-1	!
785	!	!	!	!	!	!	!	!	!	!
786	!	!	!	!	!	!	0-1	!	!	!
786	!	!	!	!	!	!	!	!	!	!
817	!	!	0-1	!	!	!	!	!	!	!
817	!	!	!	!	!	!	!	!	!	!

FIGURE A.7.7. - CHRONOGRAMME EPILOG DE VALIDATION DE L'AUTOMATE DE DECLENCHEMENT D'ARBITRAGE

EPISODE TEST DU NOUVEAU AUTOMATE : DECLENCHEMENT D'ARBITRAGE

NO	2	13	14	5	16	17	8	9	10	11
1821	1	1	1	1	1	1	1	0-1	1	1
1822	1	1	1	1	1	1	1	1	1	0-1
1842	0-1	1	1	1	1	1	1	1	1	1
1848	1	0-1	1	1	1	1	1	1	1	1
1812	1	1	0-1	1	1	1	1	1	1	1
1851	0-1	1	1	1	1	1	1	1	1	1
1863	1	1	1	1	1	1	1	0-1	1	1
1864	1	1	1	1	1	1	1	1	0-1	1
1865	1	1	1	1	1	1	0-1	1	1	1
1988	1	0-1	1	1	1	1	1	1	1	1
1992	1	1	1	1	1	1	1	1	1	0-1
1995	1	1	1	1	1	1	1	1	0-1	1
1998	1	1	1	1	1	1	0-1	1	1	1

PHASE **EDITTU TEMPS CPU : 82,00 SECONDES

FIGURE A.7.7. - (SUITE)

7.1.4.3. VALIDATION DE LA LOGIQUE ASSOCIÉE À LA GÉNÉRATION DU SIGNAL DE DEMANDE D'ARBITRAGE BREQ

Cette logique résulte, comme on vient de le dire, de la combinaison de la sortie NVALDEM2 de l'automate de déclenchement d'arbitrage (introduit dans l'ABC-M) avec la sortie NVALDEM1 de l'automate d'affichage (constant dans l'ABC 90 et reprise tel quel dans l'ABC 90) à travers une logique décrite au chapitre 3 (§ 3.3.4.).

Cette logique a été introduite dans la partie opérative de l'affichage de demandes (NAFOP sur EPILOG, voir figure A7.1.).

Le résultat de la simulation EPILOG est montré ci-après (figure A7.8) :

EPISODE TEST DE LA P.O. AFFI

CHRONOGRAMME DU CIRCUIT NAFOP

```

*****
*****
CONNECTIONS
*****

```

N	N	U	U	U	N	N	N	N	N	N	N	N	N	N	N	N
V	V	C	C	C	B	B	B	B	B	B	B	B	B	V	V	
A	A	1	2	3	R	A	A	A	A	A	A	A	A	L	A	
L	L				E	1	1	1	2	2	2	2	2	D	L	
D	D				Q	7	8	9	0	1	2	3	4	X	D	
E	E														E	
M	M														M	
1	2															

NO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	22	23
0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
0	1	1	0	1	0	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
1	1	1	1	1	1	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	0	XXX
2	1	1	1	1	1	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	1	XXX
5	1	1	1	1	1	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	1	1
6	1	1	1	1	1	XXX	1	1	XXX	XXX	1	1	XXX	XXX	1	1
34	1	1	1	1	1	XXX	1	1	1	1	1	1	1	1	1	1
650	0-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
651	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0-1	1
652	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0-1
656	1	1	1	1	1	1	1	1	0-1	1	1	1	1	1	1	1
664	1	1	1	1	1	0-1	1	1	1	1	1	1	1	1	1	1
750	1	0-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PHASE **EDITIU TEMPS CPU : 50,00 SECONDES

Figure A7.8 - Chronogramme EPILOG de validation de la logique de génération du signal NVALDEM.

7.1.5. REMARQUES

La validation de ces modifications ne se complète que dans une approche globale : un ou plusieurs ABC-M réalisent l'arbitrage dans des structures multi-processeurs différents. C'est ce type de simulation (architectures multi-processeurs intégrant l'ABC-M) qui pourra valider complètement l'ABC-M.

A l'égard du type d'architecture retenue (centralisée ou décentralisée), on a porté notre choix sur la deuxième, car c'est elle qui pourra déceler tous les problèmes de compatibilité (entre ABC-M et modules MULTIBUS ou VME).

On a donc simulé les architectures suivantes :

- * ARCHITECTURE DECENTRALISEE SM 90
→ VALIDATION MODE SM 90
- * ARCHITECTURE DECENTRALISEE VME
→ VALIDATION MODE RWD
(RELEASE WHEN DONE)
- * ARCHITECTURE DECENTRALISEE MULTIBUS
→ VALIDATION MODE ROR
(RELEASE ON REQUEST)

Remarque :

1. On remarque que la validation du mode RWD sur une architecture VME et du mode ROR sur une architecture MULTIBUS a été décidée en fonction de la littérature (voir § 3.4.3.2. et 3.4.3.3.).

2. Les descriptions FIDEL du circuit 8289 et de l'unité de demande d'accès au bus VME, sont les mêmes que celles montrées aux figures A6.1 et A6.2, respectivement.

7.2. VALIDATION DES ARCHITECTURES MULTI-MICROPROCESSEURS

7.2.1. MODE SM 90

Comme on a vu en § 3.4.1., pour que l'ABC-M marche comme un ABC 90 dans une structure SM 90, on doit câbler les broches MF1 et MF2 à zéro.

L'architecture multi-processeurs SM 90 employant l'ABC-M a été montrée à la figure 26 (voir § 3.2.2.).

Cette architecture permet :

- l'algorithme fixe, rotatif ou mixte de priorités,
- la supervision des arbitrages rendus,

- des arbitrages visibles ou cachés selon BPAC,
- d'autre part, la configuration de demandes BAi peut être multiplexée ou non-multiplexée (voir § 1.3.4.).

On rappelle que l'architecture de la structure SM 90 est décentralisée : (un arbitre de bus sur chaque module de traitement (voir § 1.3.4.)).

C'est la validation de cette architecture qu'on montre dans cette section.

Remarques :

- l'ABC-M permet aussi, à la différence de l'ABC 90, de réaliser des arbitrages centralisés de façon plus économique (un seul ABC-M pour toutes les U.T. raccordées au bus).
- si on veut utiliser l'ABC-M dans une architecture SM 90, pour réaliser des arbitrages centralisés, on doit configurer l'arbitre dans le mode RWD.

7.2.1.1. DESCRIPTION DE L'ARCHITECTURE SIMULÉE

Pour cette architecture, les simulations qu'on a faites ont les caractéristiques suivantes :

. trois modules de traitement, contenant chacun un arbitre ABC-M. Ces modules sont configurés aux adresses 0, 1 et 2. Les demandes d'accès au bus, à l'intérieur de chaque module, sont signalées par DAB (de l'unité de traitement vers l'ABC-M).

L'ABC-M d'adresse 0 (UC = 000) reçoit DAB1, celui d'adresse 1, DAB2 et ainsi de suite.

Le module de traitement 0 (adresse 0) est configuré en priorité fixe, tandis que les autres sont configurés en priorité rotative : on a alors un mécanisme mixte de calcul de priorités (PF = 001).

. l'ABC-M est placé dans un contexte SM 90 réel : l'unité de traitement après la prise du bus active DBAC (dès que les fils BA 17-24 de la configuration de demandes sont disponibles) de façon à ce que les arbitrages cachés se produisent.

. les ABC-M sont configurés de la façon suivante :

MF1 et MF2 à VDD (mode SM 90).

Remarque :

On rappelle que l'ABC-M présente ce câblage à VDD en interne.

Les séquences de demande d'accès au bus (DAB) sont les suivantes (figure A7.9) :

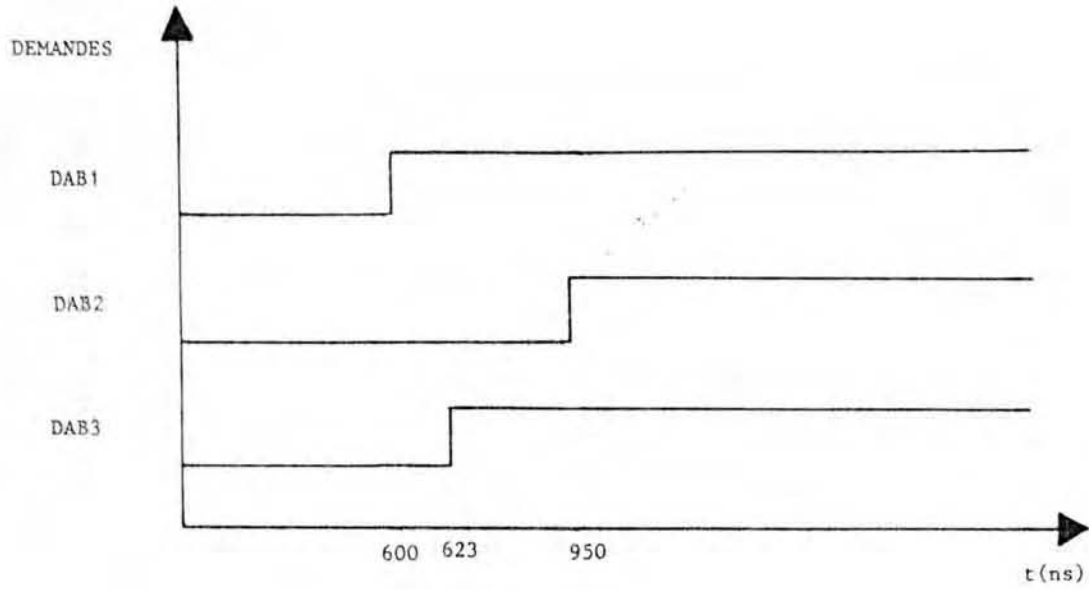
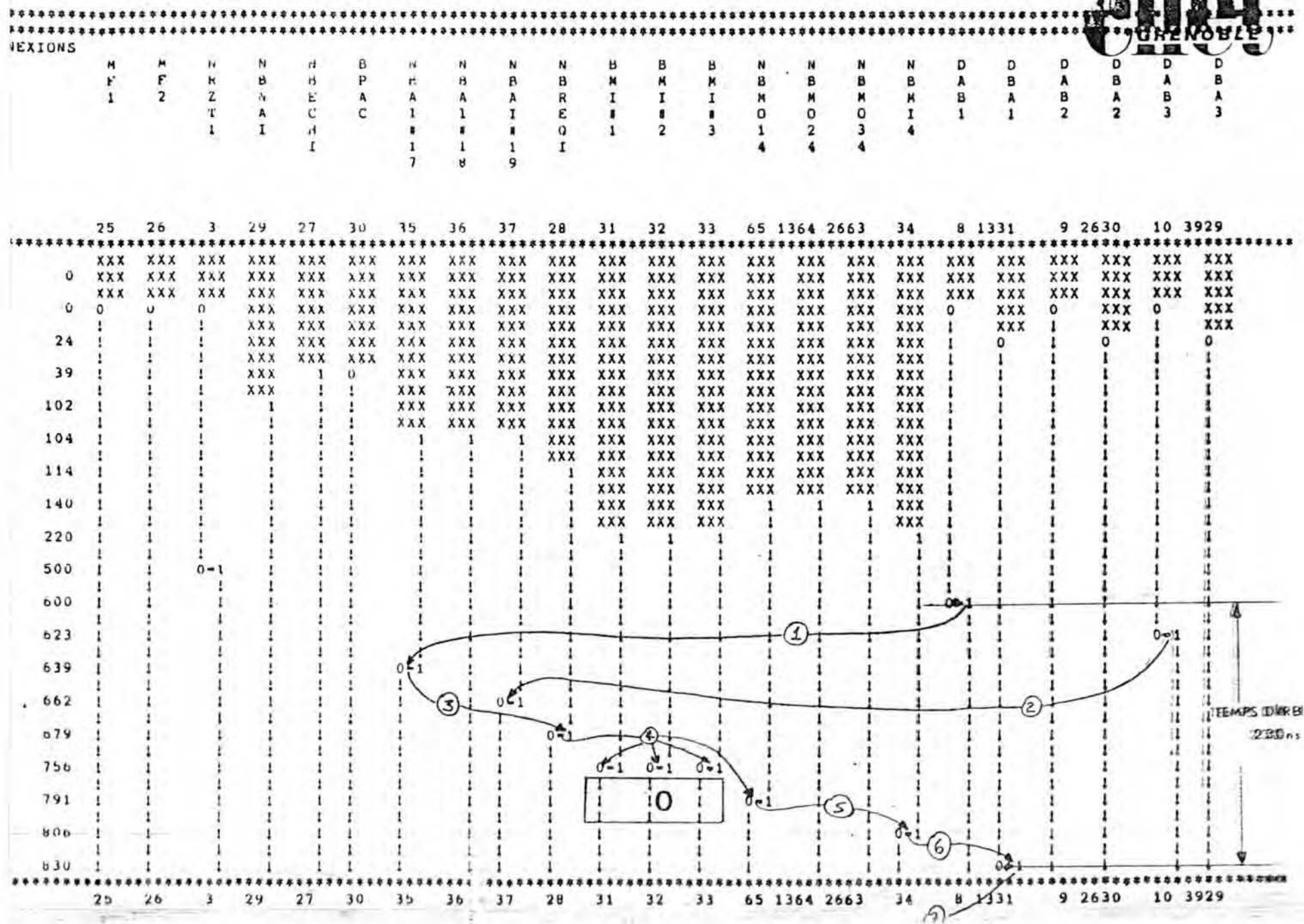


Figure A7.9 - Séquences de demande de bus.



CHRONOGRAMME DU CIRCUIT SM100



7.2.1.2. - CHRONOGRAMME EPILOG DE VALIDATION DU MODE DE FONCTIONNEMENT SM 90 DE L'ABC-M

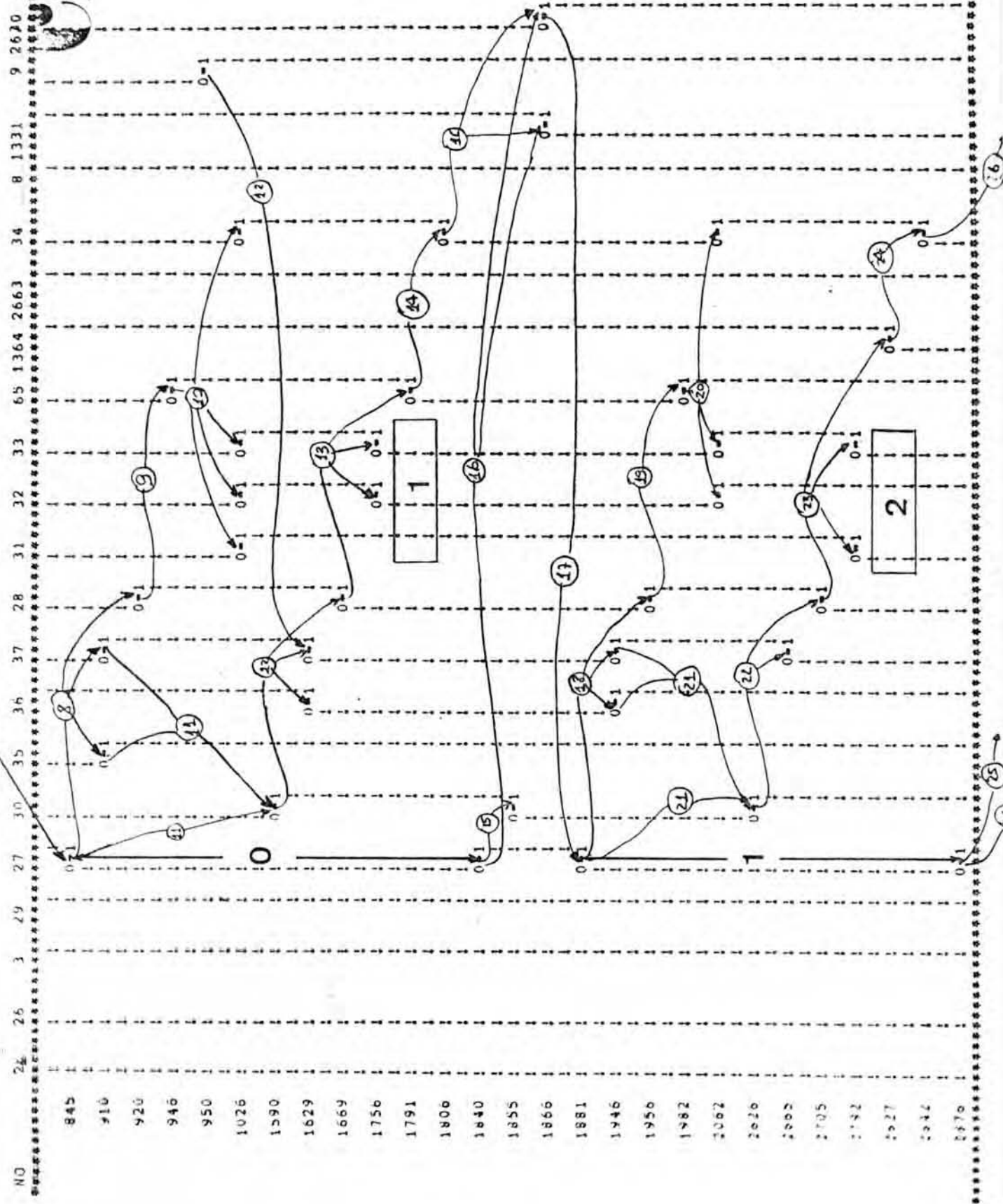
7.2.1.2. (SUITE)

LE 17/84/ B PAGE N 5

EPISODE 7. TITRE DE LA RAQUETTE CONTENU DES ABC-M DANS UNE STRUCTURE SM90

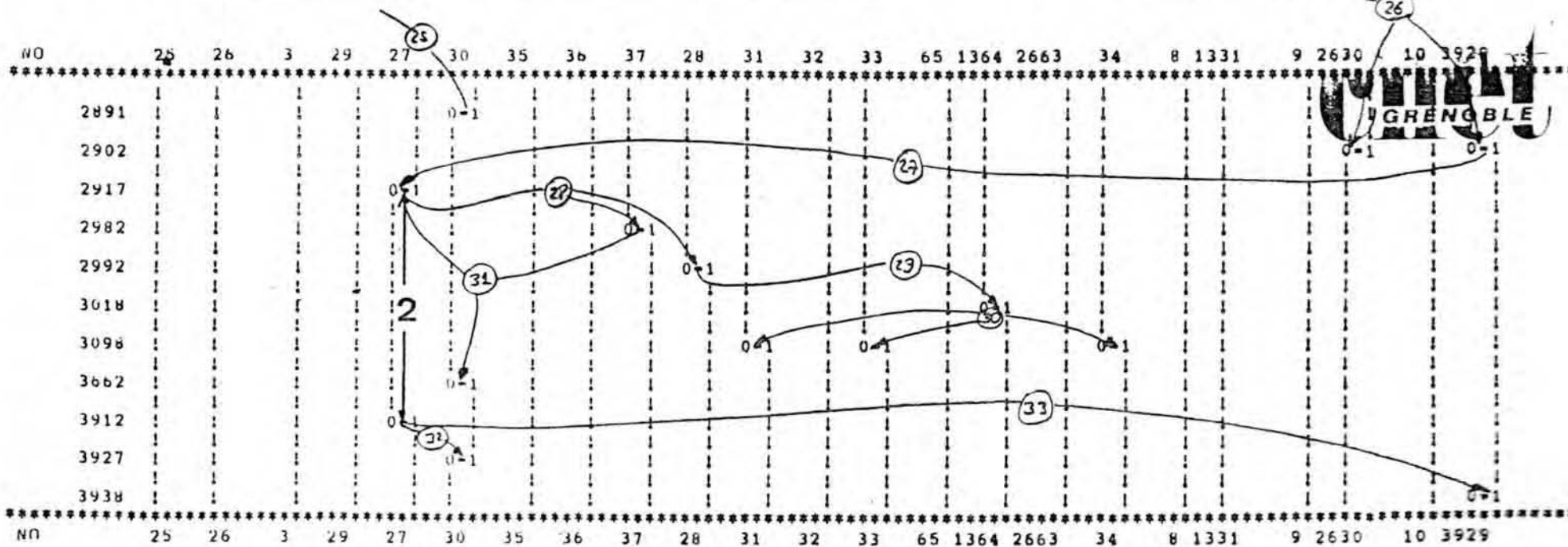


NO 845 910 920 946 950 1026 1590 1629 1669 1756 1791 1806 1840 1855 1866 1881 1946 1956 1982 2062 2070 2002 2105 2192 2227 2344 2470



EPISODE TEST DE LA MAQUETTE CONTENANT DES ABC-M DANS UNE STRUCTURE SM90

LE 17/84/ 8 PAGE N 6



PHASE **EDITIU TEMPS CPU : 396.00 SECONDES

7.2.1.2. - (SUITE)

7.2.1.3. REMARQUES

D'après les chronogrammes qu'on vient de montrer, on peut dégager les caractéristiques suivantes :

- le temps d'arbitrage (entre DAB et DBA) pris par l'ABC-M est le même que celui pris par l'ABC 90 (230 ns).

- toutes les fonctionnalités offertes par l'ABC 90 sont aussi disponibles avec l'ABC-M.

- dans ce mode de fonctionnement, il n'y a qu'un seul arbitre qui divulgue le verdict à la fois (on ne trouve qu'un seul signal de validation du verdict actif : à la fois), soit NBM014 (ABC-M 0), soit NBM024 (ABC-M 1) ou NBM034 (ABC-M 2).

Note : le signal de validation du verdict sur le bus BM4 (NBM4 sur le chronogramme EPILOG) est obtenu à travers le ET CABLE de tous les signaux de validation délivrés par chaque arbitre (NBM014, NBM024 et NBM034).

- on a eu un arbitrage visible avant le premier échange (effectué par le module de traitement 0) et caché pendant tous les échanges qui se sont succédés.

7.2.2. MODE "RELEASE WHEN DONE"

Comme on a vu en § 3.3.1., une unité de traitement configurée dans ce mode de fonctionnement enlève sa demande d'accès au bus dès qu'elle prend le contrôle du bus.

Des unités de traitement MULTIBUS ou VME peuvent être configurées dans ce mode de fonctionnement.

Toutefois, ce mode de fonctionnement est plus fréquemment utilisé pour les systèmes VME. C'est pour cette raison qu'on valide le mode RWD de l'ABC-M avec une structure VME.

L'architecture simulée est décentralisée (voir figure 28).

Cette architecture permet :

- des arbitrages visibles,
- une liberté totale de configuration de la priorité des U:T. (elles peuvent être fixes, rotatives ou mixtes).
- la supervision des arbitrages rendus.

D'autre part, l'architecture décentralisée utilisant l'ABC-M est moins chère que l'architecture analogue utilisant l'ABC 90, et est encore plus intéressante car les ABC-M présentent une redondance de service (divulgaration du verdict par tous les arbitres présents sur la structure).

7.2.2.1. DESCRIPTION DE L'ARCHITECTURE SIMULÉE

Pour cette architecture, les simulations qu'on a faites ont les caractéristiques suivantes :

- deux cartes spéciales VME (contenant chacune un ABC-M) configurées aux adresses 0 et 1 (demandes DAB1 et DAB2, respectivement).
- deux cartes standards VME (contenant chacune une unité de demande d'accès au bus, décrite en FIDEL (§ A.6.1.1.) configurées aux adresses 2 et 3 (demandes DABV3 et DABV4, respectivement).

L'unité de traitement d'adresse 0 est configurée en priorité fixe, tandis que les autres sont configurées toutes en rotative.

La broche BPAC dans tous les ABC-M est câblée à 0 (pas d'arbitrage caché possible).

Les ABC-M sont configurés de la façon suivante :
. MF1 à la masse et MF2 à VDD (mode RWD).

Les séquences de demande de bus sont les suivantes (figure A7.10) : (l'ABC-M a sa demande active haute (DAB1 et DAB2) et l'unité de demande VME a sa demande active basse (DABV3 et DABV4)).

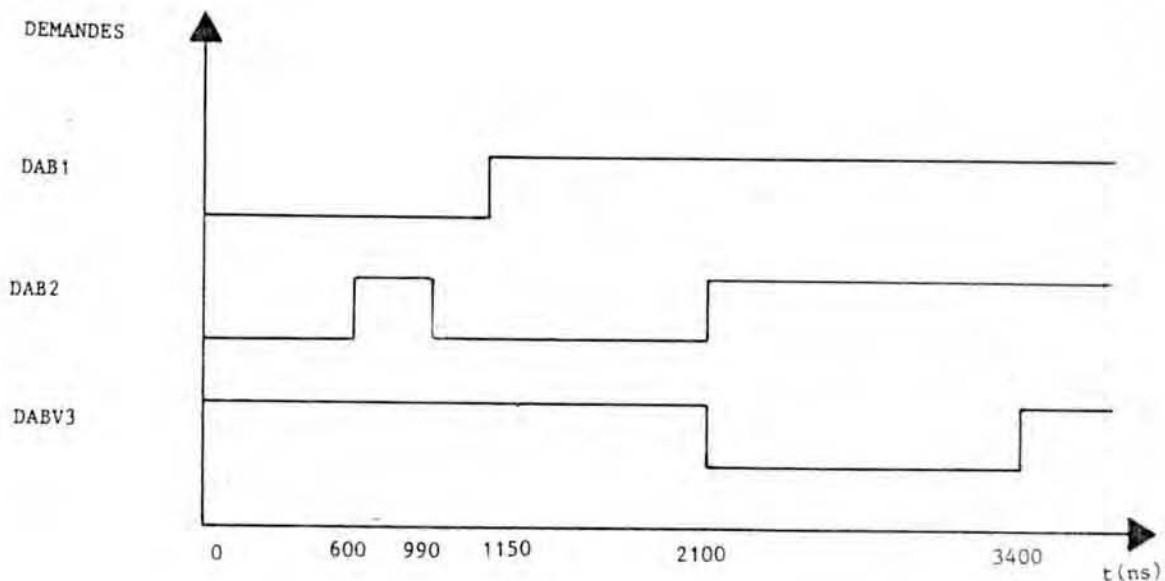


Figure A7.10 - Séquences de demande de bus.

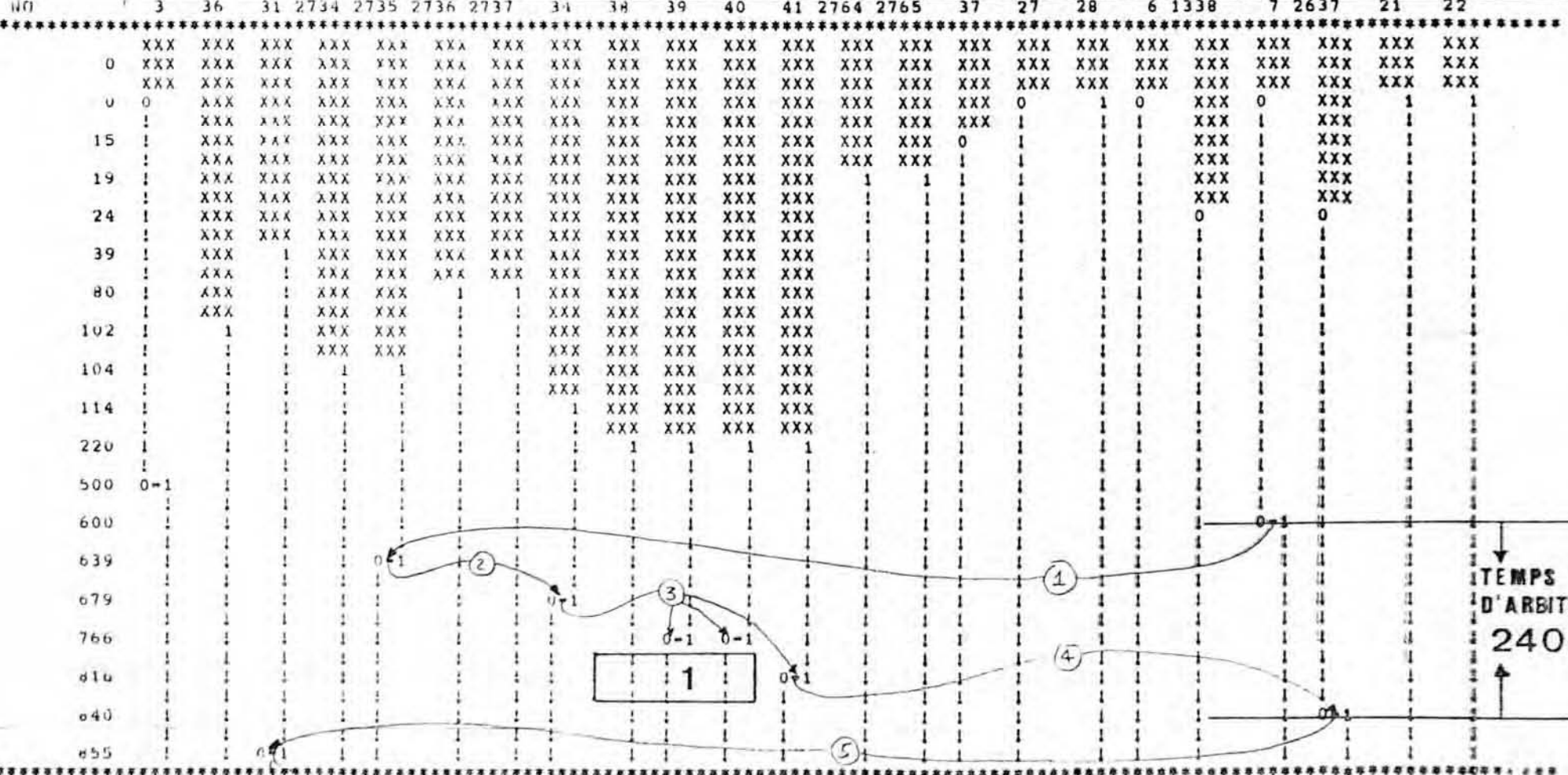
7.2.2.2. CHRONOGRAMME EPILOG
DE VALIDATION DU MODE DE
FONCTIONNEMENT "RELEASE WHEN DONE" (RWD)
DE L'ABC-M (ARCHITECTURE VME)

CHRONOGRAMME DU CIRCUIT VME

CONNECTIONS

S	H	N	H	H	N	A	H	H	B	B	N	N	N	B	M	M	D	D	D	D	D	D
Y	N	B	H	B	J	R	H	M	M	M	B	N	B	A	F	F	A	B	A	B	B	A
S	A	S	3	2	1	V	F	#	#	#	I	1	0	C	1	2	1	1	2	2	2	2
R	I	Y					Q	1	2	3	4	I	I									
S							I					N	N									

SIGNAUX SUR LE BUS													SIGNAUX CARTES SPECIALES (ABC - M)					SIGNAUX CARTES STANDARD					
NO	3	36	31	2734	2735	2736	2737	34	38	39	40	41	2764	2765	37	27	28	6	1338	7	2637	21	22



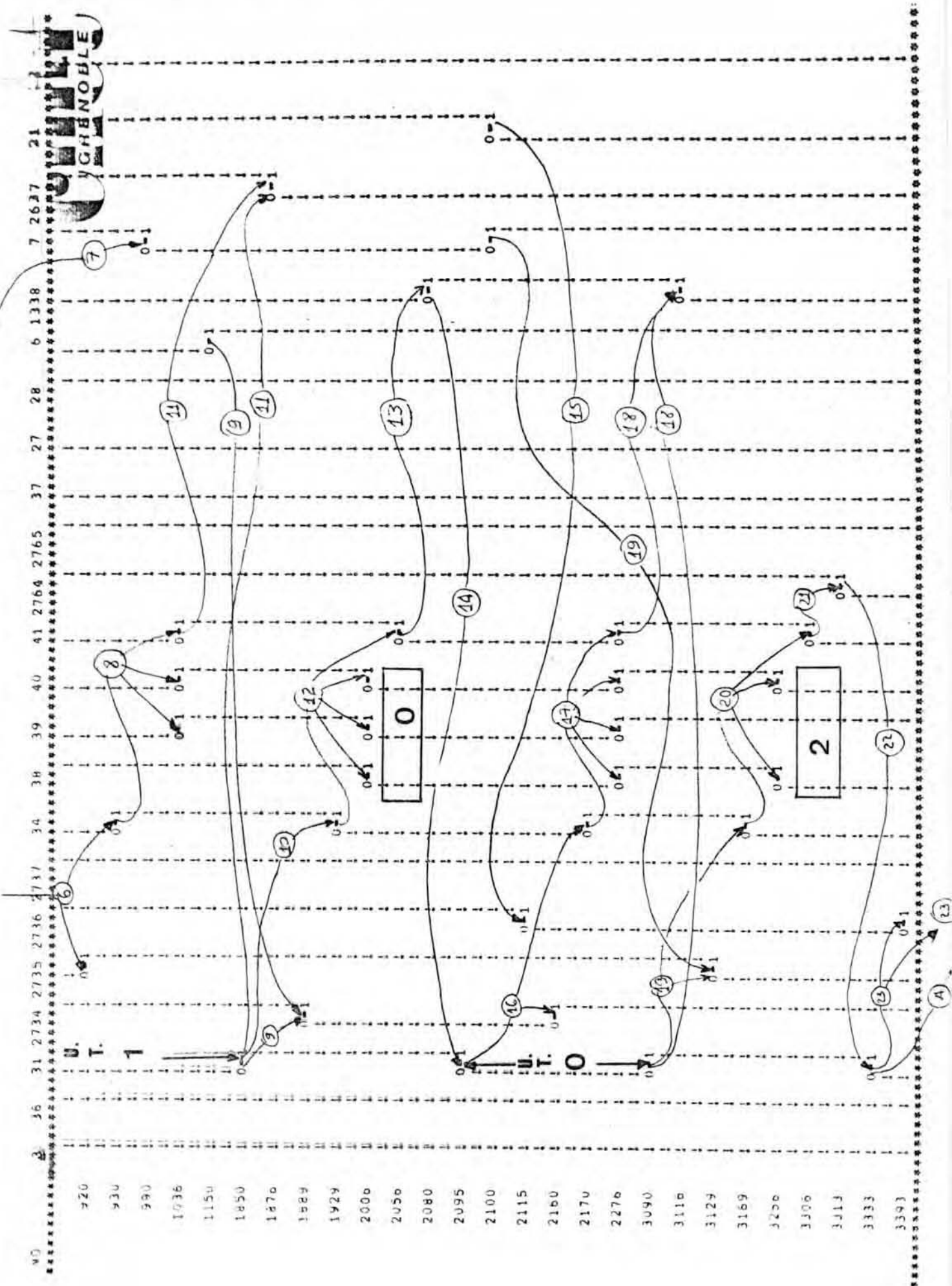
NO 3 36 31 2734 2735 2736 2737 34 38 39 40 41 2764 2765 37 27 28 6 1338 7 2637 21 22

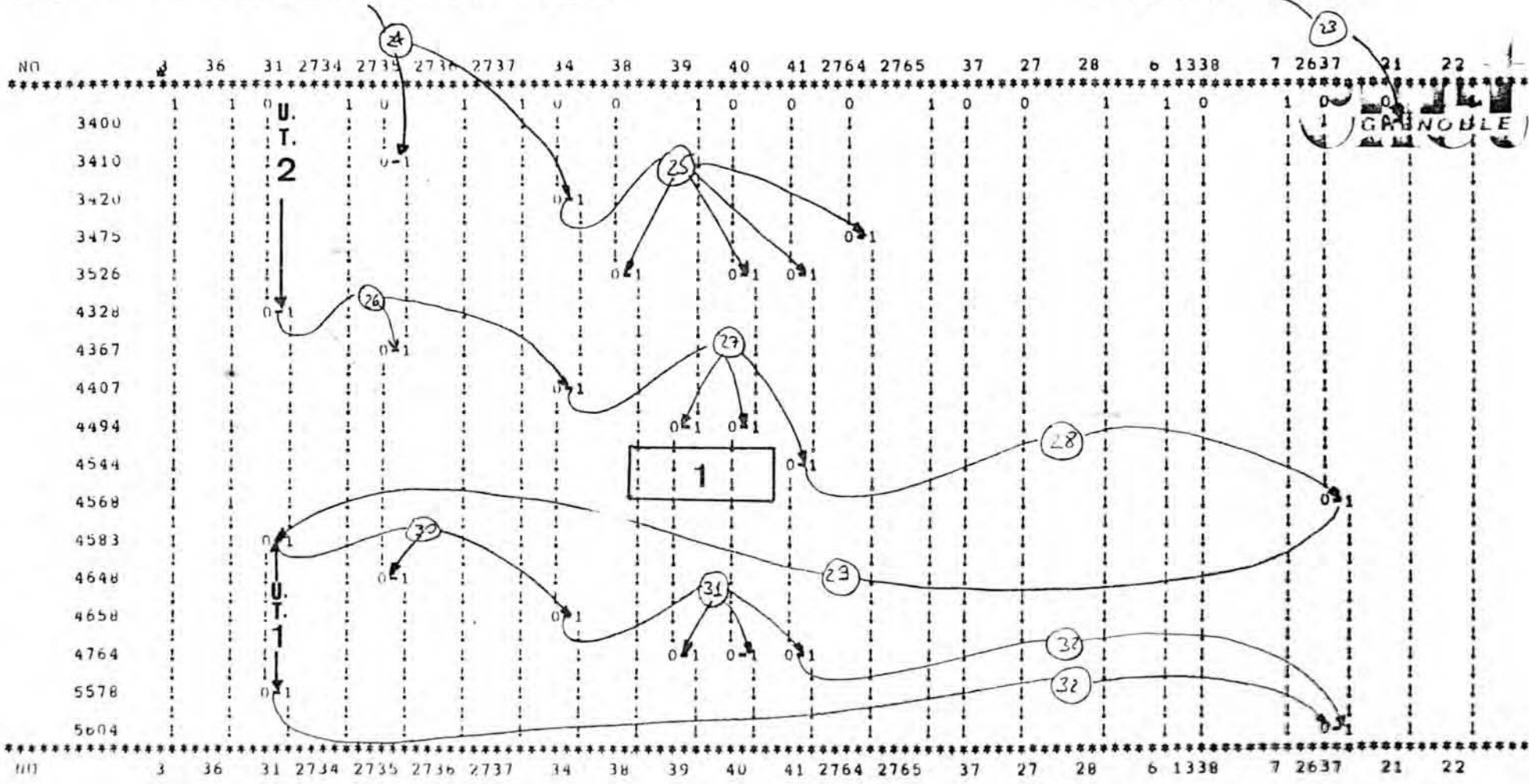
7.2.2.2.

7.2.2.2. (SUITE)

LE 17/84 / 8 PAGE N 4

EPISODE TEST DE LA MAQUETTE CONTENANT DES ABC100





PHASE **EDITIO TEMPS CPU : 1257,00 SECONDES

7.2.2.2. (SUITE)

7.2.2.3. REMARQUES

D'après les chronogrammes qu'on vient de montrer, on peut dégager les caractéristiques suivantes :

- les temps d'arbitrage entre DAB et DBA (cartes spéciales) ou entre DABV et BGXIN (cartes standards) sont inférieurs à 250 ns.

- dans ce mode de fonctionnement, les deux ABC-M présents sur la structure divulguent le verdict simultanément et systématiquement. De ce fait, en outre l'économie matérielle, l'architecture décentralisée avec l'ABC-M permet des débits plus importants par rapport à l'architecture décentralisée utilisant l'ABC 90 (voir § 2.4.3.2.).

Remarques :

- d'après la comparaison des chronogrammes EPILOG montrés en § 6.5.2. et 7.2.2., on constate que quatre transferts effectués sur les bus (effectués respectivement par les U.T. 1, 0, 2 et 1) prennent de l'ordre de 6600 ns (VME-ABC 90) tandis qu'ils prennent environ 5600 ns (VME-ABC-M).

- l'aspect tournant des priorités se réalise aisément :

- . tout d'abord l'U.T. 1 (rotative) prend le bus,
- . ce transfert est suivi de celui effectué par l'U.T. 0 (fixe),
- . deux demandes d'U.T. configurées en priorité rotative se présentent (celles des U.T. 1 (carte spéciale DAB2) et U.T. 2 (carte standard DABV3))

Comme à l'instant de la prise en compte de ces demandes (BREQ à 3169 ns), la dernière U.T. à prendre le bus a été la 1, la demande la plus prioritaire devient elle formulée par l'U.T. 2 (signal BRI sur le bus).

Cette flexibilité de configuration des U.T. peut être mise à profit, apportant des avantages intéressants dans plusieurs systèmes multi-microprocesseurs.

7.2.3. MODE "RELEASE ON REQUEST"

Comme on a vu en § 3.3.1., une unité de traitement configurée dans ce mode de fonctionnement n'enlève sa demande d'accès au bus que si une autre demande plus prioritaire s'est présentée sur le bus.

Des unités de traitement MULTIBUS ou VME peuvent être configurées dans ce mode de fonctionnement.

Néanmoins, ce mode de fonctionnement est très répandu dans les systèmes multi-processeurs MULTIBUS. Pour cette raison, on a décidé de valider ce mode de fonctionnement (ROR) de l'ABC-M avec une structure MULTIBUS.

L'architecture MULTIBUS choisie pour la validation a été la décentralisée (voir figure 30), car c'est elle qui peut mettre en évidence tous les problèmes possibles de compatibilité ABC-M - MULTIBUS.

Cette architecture décentralisée permet :

- une entière liberté de choix de configuration des U.T. (la priorité de chaque U.T. peut être fixe ou rotative).
- la supervision des arbitrages rendus.

En outre, l'architecture décentralisée, utilisant l'ABC-M, est intéressante pour des systèmes cherchant à avoir des mécanismes sûrs de fonctionnement, étant plus économique et fiable (moins de composants) que l'architecture analogue avec l'ABC 90.

7.2.3.1. DESCRIPTION DE L'ARCHITECTURE SIMULÉE

Pour cette architecture, les simulations qu'on a faites ont les caractéristiques suivantes :

- deux cartes spéciales MULTIBUS (contenant chacune un ABC-M) configurées aux adresses 0 et 1 (demandes DAB1 et DAB2, respectivement).
- deux cartes standards MULTIBUS (contenant chacune un circuit 8289 décrit en FIDEL) (voir annexe 6), configurées aux adresses 2 et 3 (demandes RSO # 2 et RS1 # 2, respectivement).

Remarque :

On rappelle (voir § 1.4.4.) qu'à l'intérieur d'une carte standard MULTIBUS, dès que le microprocesseur veut accéder au bus, il active le registre d'état du circuit 8289, en lui envoyant une configuration active (ni 011, ni 111) sur 3 bits RS 0-2.

Dans les simulations qu'on a faites, on a pris tous les bits RS à 0 pour la configuration des demandes (sur les chronogrammes EPILOG, on ne montre que le bit de poids le plus fort (RS # 2) : dès que ce bit est à 0, une demande a été formulée par le microprocesseur).

L'unité de traitement d'adresse 0 est configurée en priorité fixe, tandis que les autres sont configurées toutes en rotative.

Pour que le mode RELEASE ON REQUEST puisse fonctionner convenablement, les arbitrages cachés doivent être validés en permanence (broche BPAC câblée à 1 dans tous les ABC-M).

Les ABC-M sont configurés de la façon suivante :

- . MF1 et MF2 à VSS (mode ROR).

Les séquences de demande de bus sont les suivantes (figure A7.11) :

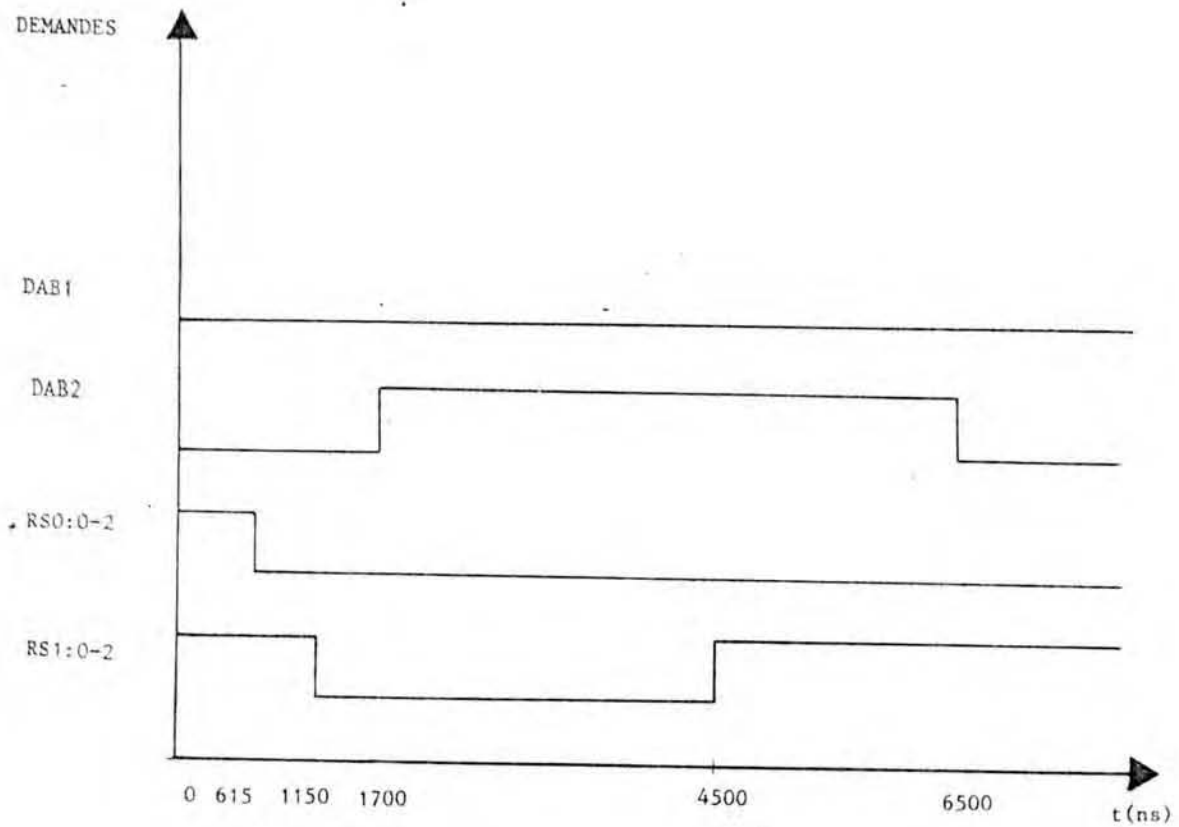
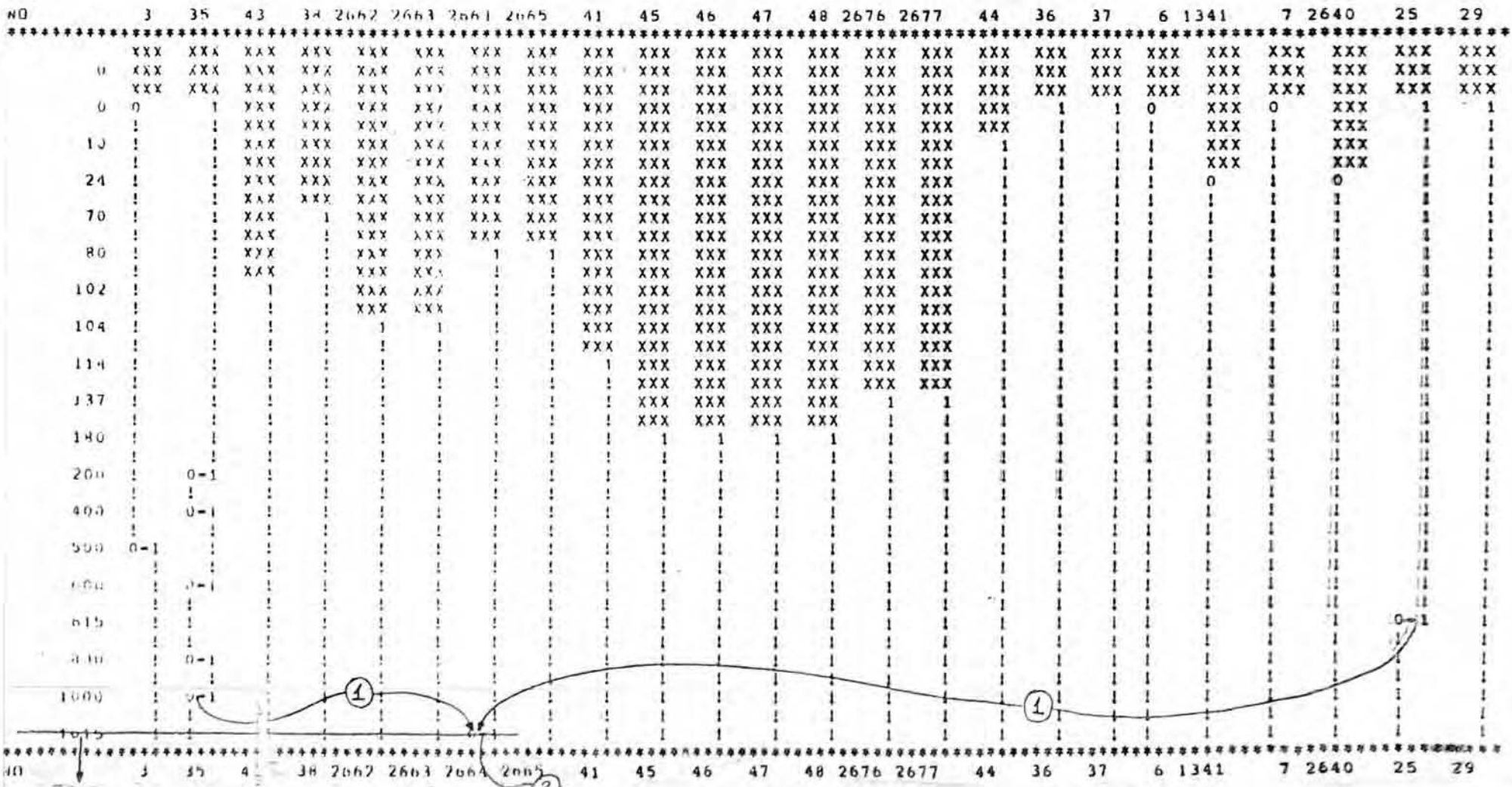


Figure A7.11 - Séquences de demande de bus.

CHRONOGRAMME DU CERCLE MULTIBUS

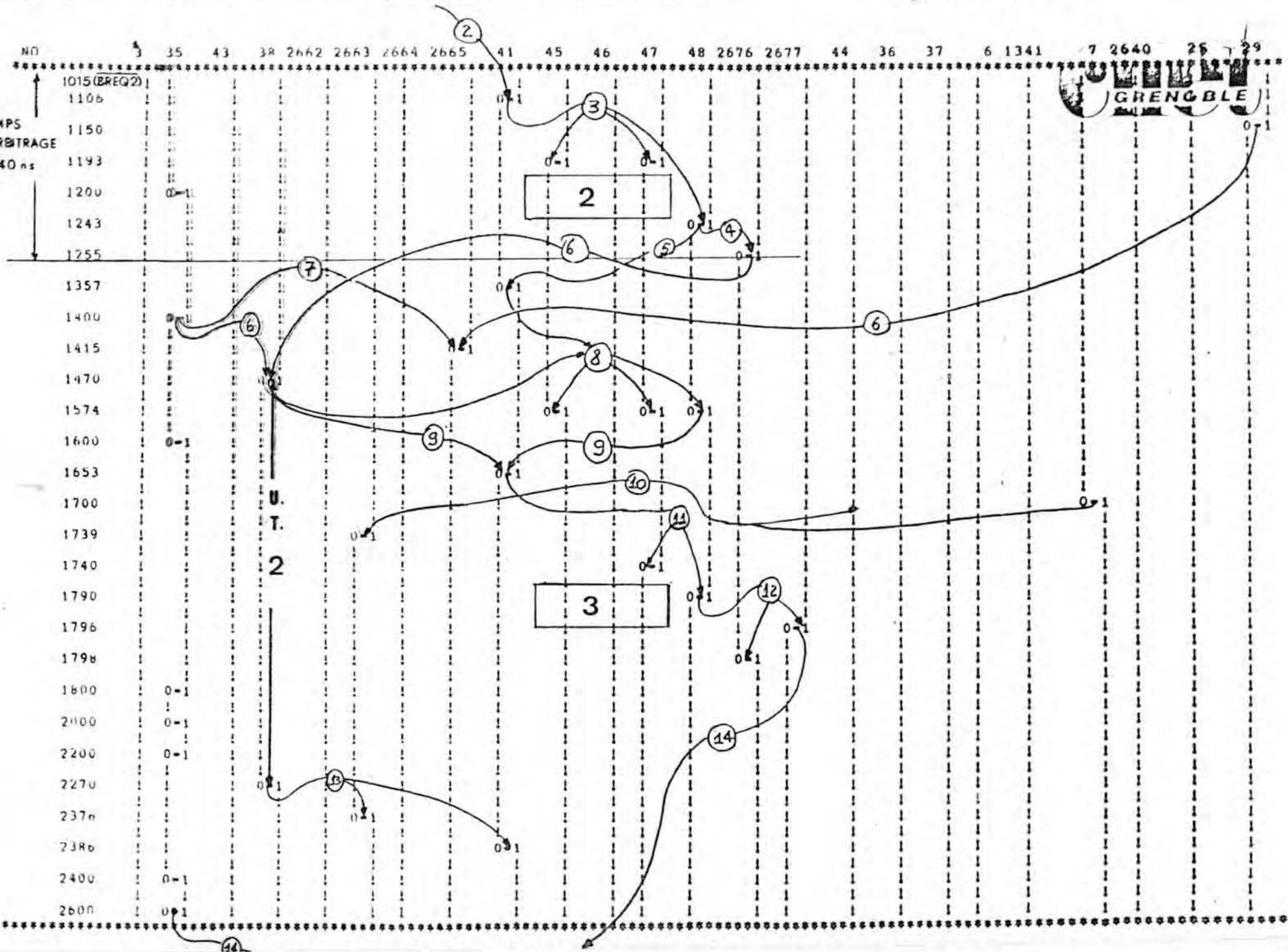
INDEXIUS

N	H	P	N	P	P	R	H	P	H	R	B	N	N	N	B	M	M	D	D	D	D	N	N
1	H	G	U	P	P	R	H	P	H	M	M	H	B	B	P	F	F	A	B	A	B	R	R
U	C	A	U	P	P	R	H	P	H	I	I	M	P	P	A	1	2	B	A	B	A	S	S
I	I	A	S	E	E	E	F	E	E	#	#	I	R	R	C			1	1	2	2	O	O
T	K	L	Y	L	U	U	U	U	U	1	2	3	4	N	N							S	S
				U	1	2	1	1	1				2	3								2	2

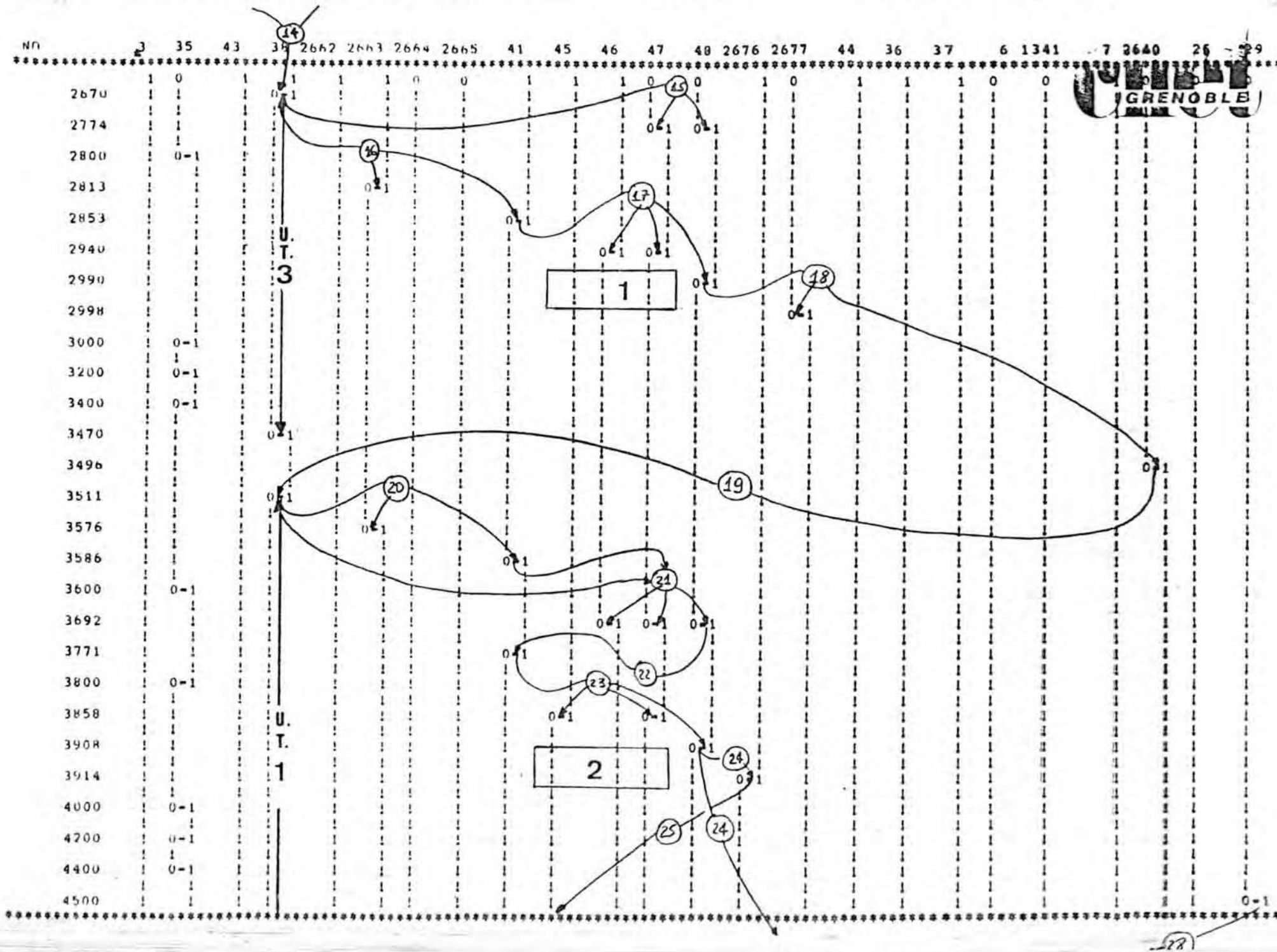


7.2.3.2. CHRONOGRAMME EPILOG DE VALIDATION DU MODE DE FONCTIONNEMENT ROR DE L'ABC-M (ARCHI-TECTURE MULTIBUS)

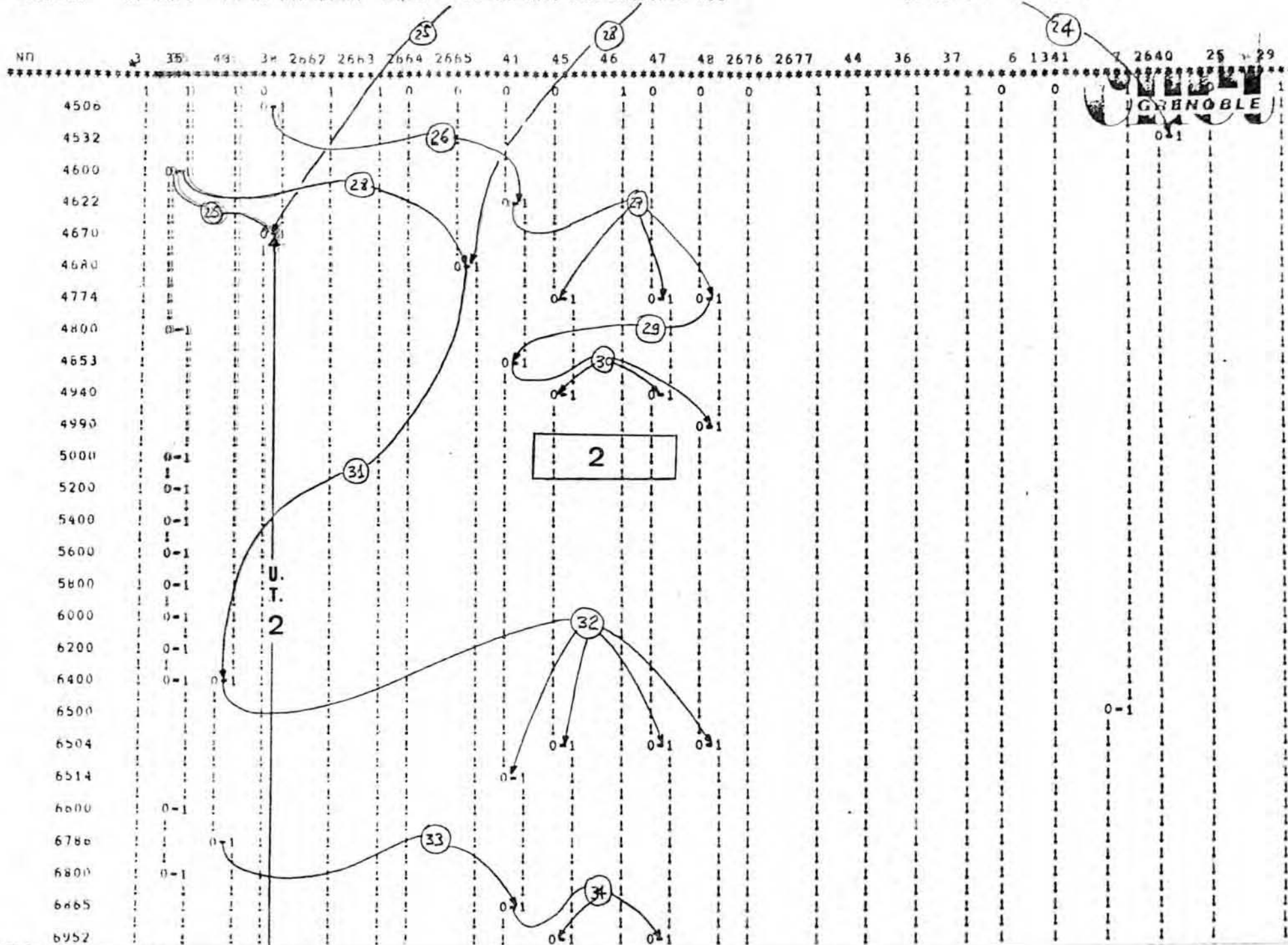
TEMPS D'ARBITRAGE
240 ns



7.2.3.2. (SUITE)



7.2.3.2. (SUITE)



7.2.3.2. (SUITE)

EPISODE SIMULATION DE LA STRUCTURE MULTIBUS CONTENANT DES ARBITRES ABC-M

LE 21/84/ 8 PAGE N 8

```

ND      3      35      43      38 2662 2663 2664 2665      41      45      46      47      48 2676 2677      44      36      37      6 1341      7 2640      25 - 29
*****
7000    !      !      !      !      !      !      !      !      !      !      !      !      !      !      !      !      !      !      !      !
*****
ND      3      35      43      38 2662 2663 2664 2665      41      45      46      47      48 2676 2677      44      36      37      6 1341      7 2640      25  29

```

2

CHENGBLE

PHASE **EDTTU TEMPS CPU : 1267.00 SECONDES

7.2.3.3. REMARQUES

D'après les chronogrammes qu'on vient de montrer, on peut dégager les caractéristiques suivantes :

- comme les temps d'arbitrage de l'ABC-M sont de l'ordre de 250 ns (entre \overline{BREQx} et \overline{BPRNx} : signaux du circuit SCS9, pour la demande d'accès et autorisation d'accès au bus, respectivement) et dus au fait que pour MULTIBUS on doit avoir

$$t \text{ ARBITRAGE} < T_{BCLK}$$

(temps d'arbitrage inférieur à la période de résolution des conflits de priorité (BCLK)), on doit :

. augmenter la période d'horloge à 400 ns (comme on l'a fait d'ailleurs pour l'architecture décentralisée MULTIBUS-ABC 90 (voir 5.2.1.1.c)).

- de même que pour l'architecture décentralisée discrète (voir § 5.2.1.1. c), le fonctionnement de la structure VME en mode ROR exige que l'ABC-M utilise la fonctionnalité de supervision (qui en principe est optionnelle) dès qu'une U.T. prend le bus indéfiniment (cas de l'U.T. 2 à partir de 4670 ns). De cette façon, le signal \overline{BNA} actif, généré par la supervision de l'ABC-M, produit (s'il y a des demandes actives \overline{BAI} sur le bus) un nouveau front actif (descendant) du signal de demande d'arbitrage \overline{BREQ} , ce qui permet à l'ABC-M d'arbitrer les demandes présentes sur le bus à cet instant. (Dans notre exemple, aucune demande en-dehors de l'U.T. qui occupe le bus ($\overline{BREQ2}$) n'était présentée à la nouvelle activation de \overline{BREQ} : 6865 ns)).

On rappelle qu'à chaque prise de bus (\overline{BECH} actif) (dans le cas d'arbitrage caché possible), l'ABC-M désactive le signal \overline{BREQ} , de façon à ce que un nouvel arbitrage puisse se produire s'il y a des demandes sur le bus (cas de la prise de bus par l'U.T. 2 à 1470 ns, de l'U.T. 3 à 2670 ns, de l'U.T. 1 à 3511 ns et de l'U.T.2 à 4670 ns).

Pour que les demandes présentées après le front actif du signal de demande d'arbitrage (\overline{BREQ}) ne soient pas pénalisées, il est intéressant que le temps surveillé pour le déclenchement du signal de supervision (\overline{BNA}) soit de l'ordre de 2 périodes d'horloge BCLK (800 ns).

7.3. SYNTHÈSE SUR SILICIUM DE MODIFICATIONS APPORTÉES À L'ABC 90

Les modifications introduites à l'ABC 90 le transformant en circuit arbitre de bus de commutation multiprotocoles (ABC-M) sont synthétisées sur silicium sur 2 méthodes différentes :

- l'une pour la logique associée à chaque modification (partie opérative). Ces parties logiques ont été décrites par le langage symbolique CRASH décrit en annexe 5.
- l'autre pour la synthèse de partie contrôle définie par un GRAFCET (partie contrôle). Cette méthode originale [OLI 83-2] est décrite ci-après. Elle a été appliquée pour l'obtention de l'automate de déclenchement d'arbitrage (voir § 3.3.4).

7.3.1. MÉTHODE DE SYNTHÈSE DE L'AUTOMATE DE DÉCLENCHEMENT D'ARBITRAGE

Pour traduire la description comportementale d'une partie contrôle liée au déclenchement d'arbitrage (automate correspondant de l'ABC-M), on a utilisé la même méthode employée pour la synthèse de partie contrôle de l'ABC 90 : le GRAFCET [ITI 82].

7.3.1.1. GRAFCET

Le GRAFCET est un symbolisme graphique spécifié par l'AFCEC (Association Française pour la Cybernétique Economique et Technique) pour la description des automatismes séquentiels [ITI 82]. Il est dérivé des réseaux de PETRI [MOA 78] ; toutefois, il se diffère par :

- les états ne peuvent être qu'actifs ou inactifs (il n'existe pas plusieurs jetons dans la même place).
- les transitions sont toujours déterministes.

De plus, le GRAFCET - qui devient d'utilisation de plus en plus courante - permet la description de séquences parallèles et est plus propice à la réalisation sur silicium.

D'autre part, une méthode automatique de synthèse [OLI 83-2] a été développée au CNET/GRENOBLE, laquelle permet la compilation en silicium d'une description sous GRAFCET.

Le GRAFCET de l'automate de déclenchement d'arbitrage est montré à la figure A7.12.

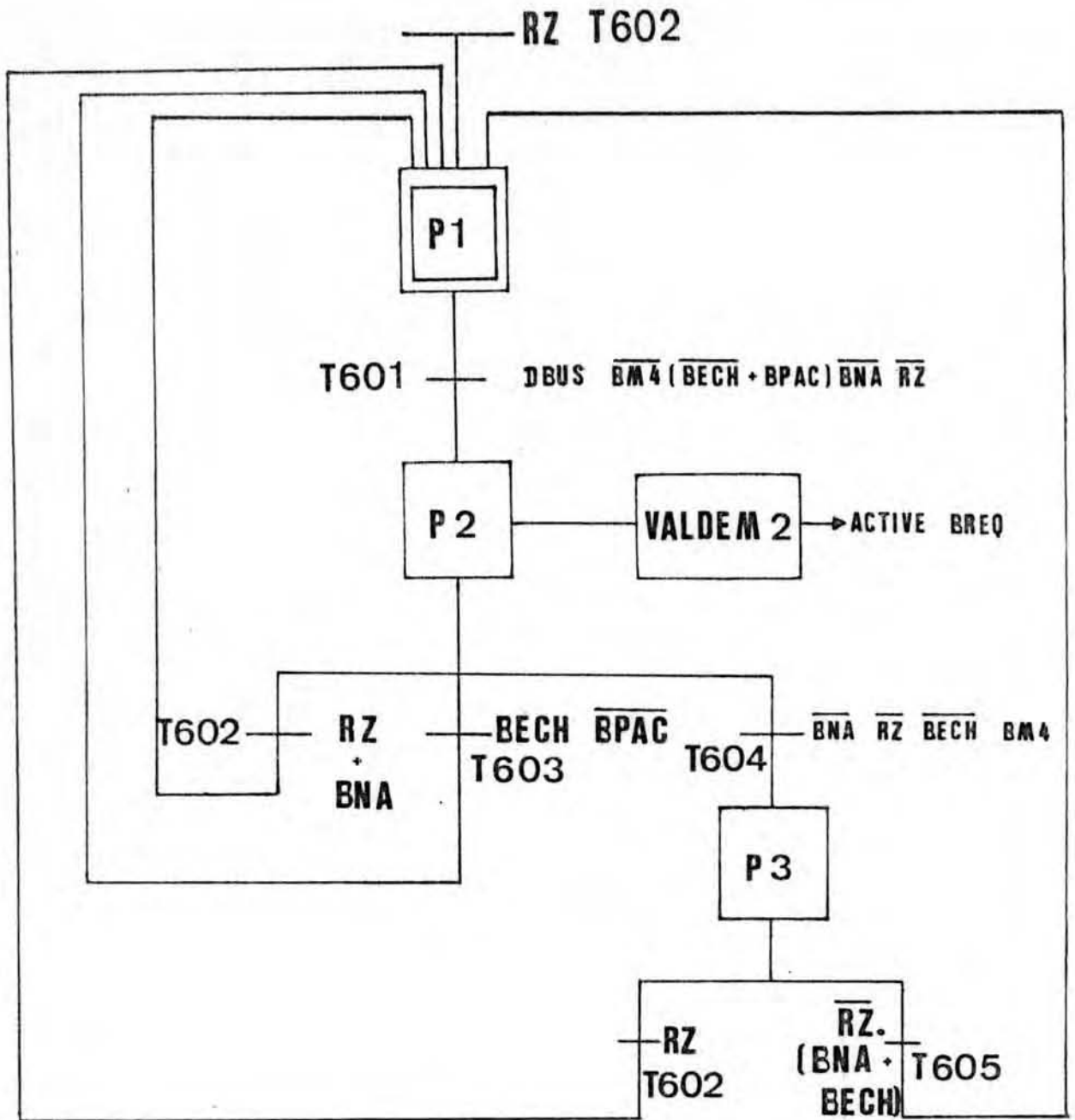


Figure A7.12 - GRAFCET de l'automate de déclenchement d'arbitrage.

7.3.1.2. MÉTHODE DE SYNTHÈSE SUR SILICIUM

Pour l'instant, la méthode de synthèse porte sur des automates asynchrones (cas de l'automate de déclenchement d'arbitrage, car on n'a pas voulu introduire une horloge à l'ABC-M).

La méthode de synthèse consiste à associer à chaque place une simple bascule R-S composée de portes OUNON mémorisant l'état de la place.

De même, chaque transition est réalisée avec une porte OUNON qui reçoit (en inverse) le signal de réceptivité de cette transition et l'état de la place précédente (voir figure A7.13).

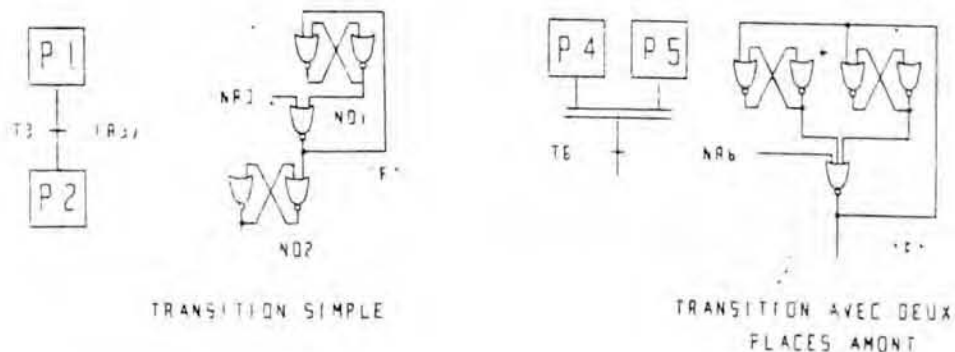


Figure A7.13 - Schéma logique correspondant aux transitions.

La porte OUNON de chaque transition délivre un signal de franchissement F transitoire :

. ce signal est à 1, quand la transition est validée.

A ce moment là, F est utilisé pour mettre à 1 la (les) place(s) en aval et remettre à zéro la (les) place(s) en amont.

Les conditions de bon fonctionnement de cette structure sont montrées en [OLI 83-2].

7.3.1.3. PLAN DE MASSE DE L'AUTOMATE

La disposition obtenue du plan de masse de l'automate décrit sous GRAFCET après la synthèse est montrée à la figure A7.14 :

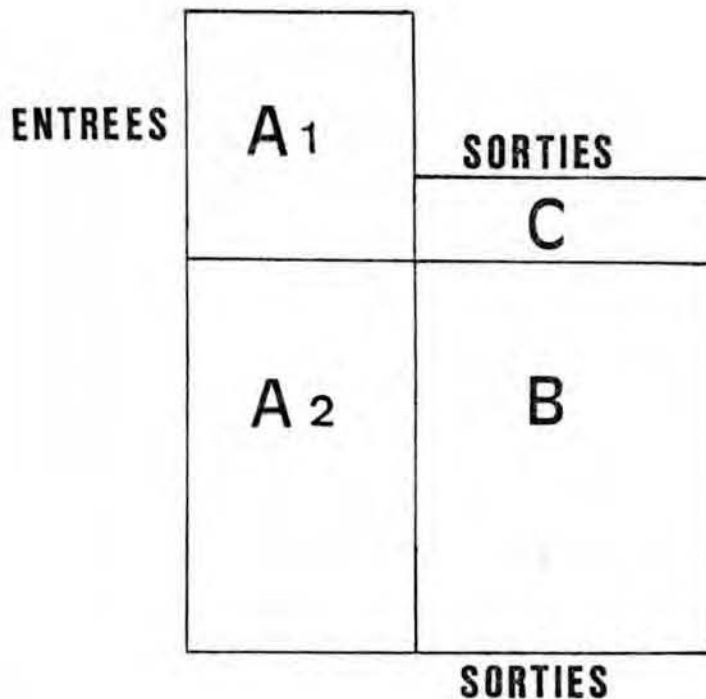


Figure A7.14 - Disposition du plan de masse de l'automate.

Sur cette figure, les différentes parties représentent :

- * A : PLA qui calcule les inverses des réceptivités à partir des signaux d'entrée de l'automate en question (signaux DBUS, $\overline{BM4}$, \overline{BECH} , BPAC, \overline{BNA} et \overline{RZ}). Ces entrées arrivent à gauche sur la partie haute de A (A1).

A partir des inverses des entrées, le demi-PLA A1 produit les combinaisons (Ci) nécessaires pour les réceptivités utilisées dans l'automate.

Dans la matrice A2, on valide les combinaisons Ci dont on a besoin.

- * B : est la matrice qui représente le GRAFCET. Cette matrice contient les cellules suivantes :
 - . "DEP" (départ) si la place Pj précède la transition Ti.
 - . "ARR" (arrivée) si Pj suit Ti.
 - . "XQ" si l'activité de la place Pj valide la transition Ti.
 - . "XNQ" si l'inactivité de la place Pj valide la transition Ti.

S'il n'existe aucun lien entre la place P_j et la transition T_i , aucune cellule n'est placée dans la cellule et leur intersection reste vide.

* C : est la rangée de bascules. Il y a autant de colonnes (1 bascule par colonne) qu'il y a d'états dans l'automate. Dans notre cas, 3.

L'implantation de l'automate de déclenchement d'arbitrage est montrée à la figure A7.15.

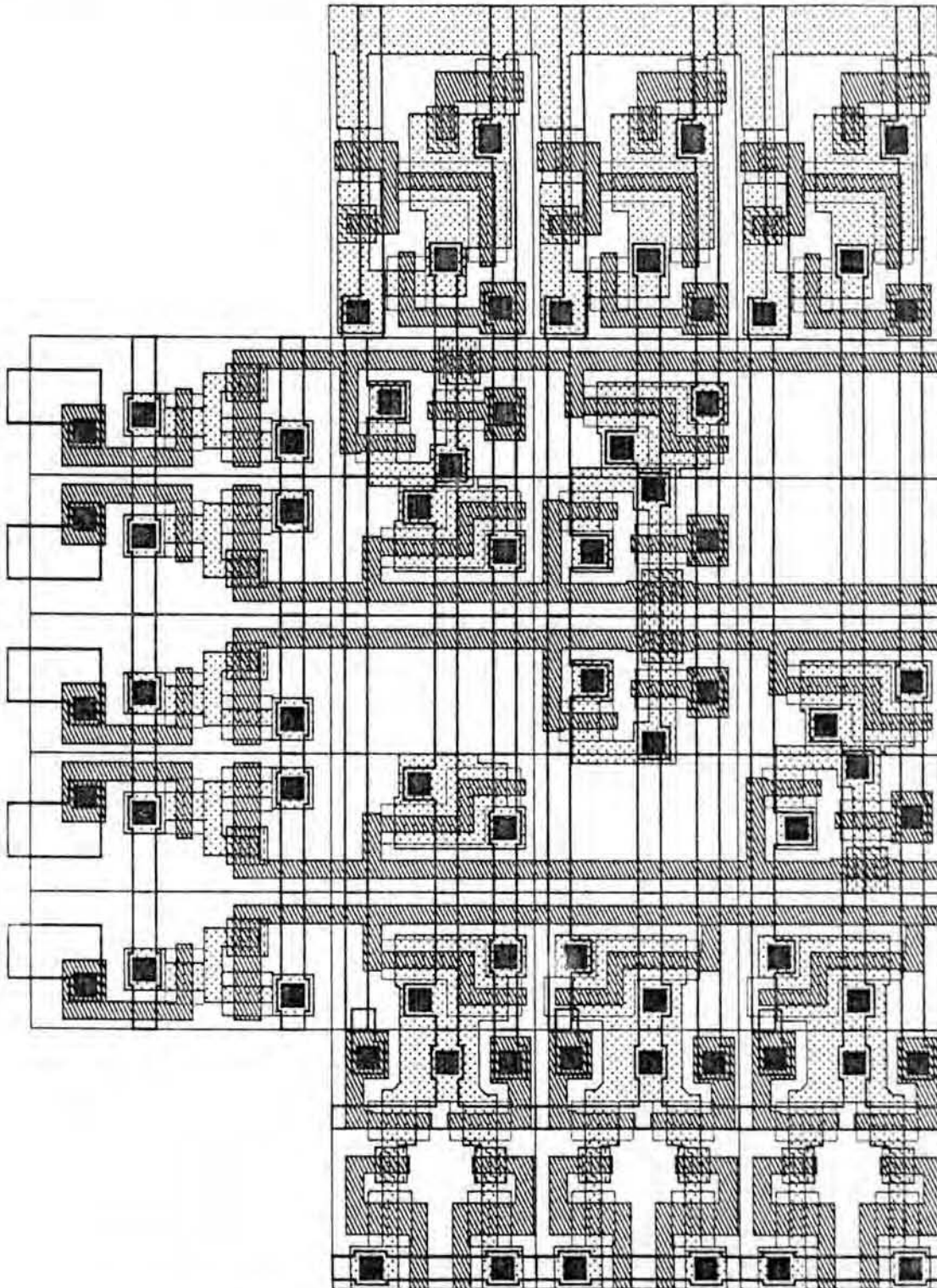


Figure A7.15 - Implantation de l'automate de déclenchement d'arbitrage.

BIBLIOGRAPHIE DES ANNEXES

[AYA 82] AYACHE J.M., COURTIAT J.P. & DIAZ M.

REBUS, a fault tolerant distributed system for industrial real time control.

IEEE Transactions on Computers, vol C31, n° 7, Juillet 1982.

[BAR 83-1] BARONE D.A.B.

Etude des bus de communication série.

Journées d'études internes au département Architecture des Micro-systèmes, CNET-CNS Grenoble, Juin 1982.

[BAR 83-2] BARONE D.A.B.

Etude de l'interface côté application de différents gestionnaires de communication de Réseaux Locaux. Analyse et Proposition selon la norme IEEE 802.

Rapport technique interne à l'équipe de Parallélisme, Communication et Circuits Intégrés de l'IMAG, 1983.

[BEY 82] BEYLS A.M., HENNION B., LECOURVOISIER J., MAZARE G., PUISSOCHET A.

A design methodology based upon symbolic layout and integrated CAD tools.

Design Automation Conference 1982

[BON 84] BONTRON G.

DRC CASSIOPEE APPLIQUE AU SYMBOLISME CRASH.

Rapport technique interne au CNET-CNS, Grenoble 1984.

[CON 82] CONQ B.

Implantation symbolique MDMOS.

Note technique NT/CNS/CCI/03, CNET-CNS Grenoble, Mai 1982.

[COR 80] CORNAFION

Systèmes Informatiques Distribués - Concepts et Techniques.

Edition DUNOD-Informatique, Paris 1980.

[DAN 82] DANG M., MAZARE M. & MICHEL M.

Local Area Networks for Distributed Process Control Systems.

International Seminar on Synchronization control and communication in distributed computing systems. Polytechnic of Central London - Angleterre - 20-24 Septembre 1982.

[DIA 79] DIAZ L. et Coll.

Système de Communication Adapté à la commande coordonnée des processus industriels.

LAAS, rapport interne, Décembre 1979.

[ECM 82] ECMA : EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

Local Area Networks (CSM/CD Baseband, Physical Layer).

Standard ECMA-81, Septembre 1982.

[EPI 79] EPILOG

Manuel de référence, Janvier 1979.

[ETI 83] ETIENNE R.

Etude des méthodologies de conception, outils de synthèse et de génération automatiques de parties contrôles de microprocesseurs.

Thèse de Docteur Ingénieur, INPG, Juin 1983.

[HEN 84] HENNION B. & SENN P.

ELDO : Présentation de ce nouveau simulateur électrique fin, suivie d'exemples d'application sur des cellules à capacités commutées.

Séminaire sur des circuits intégrés analogiques - GCIA Paris, Octobre 1984.

[GIR 81] GIROD D.

Caractéristiques et fonctionnement maître/esclave du MULTIBUS.

Minis et Micros, n° 150.

[GIR 82] GIROD D.

Le bus VME : description du fonctionnement dynamique.

Minis et Micros, n° 162, Avril 1982.

[IEE 81] IEEE Computer Standard Committee

Draft of the proposed IEEE 802 local network standard.

IEEE, Décembre 1981.

[INT 81] INTEL MULTIBUS Specification

Manual Order Number : 98 00 683 03

Copyright 1978, 1979, 1981, Intel Corporation.

[INT 84] INTEL CORPORATION

Microsystem Components Handbook - Volume II, 1984.

[ITI 82] ITICSOHN P.

Représentation des automatismes séquentiels : le GRAFCET.

Electronique Industrielle, n° 34, Mai 1982.

[ITI 84] ITICSOHN P.

FACTOR pour un scénario volontariste en contexte hétérogène.

Electronique Industrielle, n° 66, Mars 1984.

[KEN 82] KENNETH MILLER C & THOMPSON D.M.

Making a case for token passing in local networks.

Date communication, Mars 1982.

[LEF 82] LEFRANC J.P., RENOULIN R. & TAKHEDMIT M.T.

Réalisation d'un système de communication pour paquets à haut débit destiné au réseau local en boucle du CCETT (projet CARTHAGE).

Congrès AFCET Informatique, 1982.

[MAC 79] MACCHI C. & GUILBERT J.F.

TELEINFORMATIQUE.

Collection technique et scientifique des télécommunications - CNET-ENST.

Editions DUNOD, 1979.

[MEA 80] MEAD C. & CONWAY L.

Introduction to VLSI systems.

Addison -Wesley Publishing Company, USA, 1980.

[MET 76] METCALFE R.M. & BOGGS D.R.

ETHERNET : Distributed Packet Switching for Local Computer Networks.

Communication of the ACM, vol. 19, n° 7, Juillet 1976.

[MOA 78] MOALLA M., PULOU J. & SIFAKIS J.

Réseau de PETRI synchronisés.

RAIRO Automatique, vol. 12, n° 2, 1978.

[MOT] MOTOROLA

VME bus : specification manual.

M 68 KVMEB (D1) MICROSYSTEMS.

[OLI 83-1] OLIVE V. & ROUQUIER D.

Arbitre de bus de communication pour la SM 90 : "ABC 90".

Rapport interne, CNET GRENOBLE, 1983.

[OLI 83-2] OLIVE V. & ROUQUIER D.

A systematic method for the synthesis of control part defined by GRAFCET.

IFIP 83, PARIS.

[REN 83] RENOULIN R.

Une troisième voie pour les réseaux locaux d'entreprise : l'exemple de CARTHAGE.

Minis et Micros, n° 185, Avril 1983.

[ROS 82] ROSET A.

CRASH : une méthode de conception symbolique pour
le NMOS.

Note technique NT/CNS/CCI/05, CNET Grenoble,
Juillet 1982.

[ROS 84] ROSENBERG R.

Closing in open systems.

Electronics, Mai 1984.

[RUT 82] RUTLEDGE J.M.

OSI and SNA : A Perspective.

Journal of Telecommunications Networks, Computer
Science Press Inc., 1982.

[SAB 81] SABATIER A.

Les concepts-systèmes de INTEL : le bus iSBX et
les "multimodules".

Electronique Industrielle, n° 14, Avril 1981.

[TOB 80] TOBAGI F.A. & BRUCE HUNT V.

Performance Analysis of Carrier Sense Multiple
Access with Collision Detection.

North-Holland Publishing Company.
Computer Networks 4, 1980.

AUTORISATION de SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU les rapports de présentation de Messieurs

- . G. MAZARE, Professeur et
- M. NGUYEN-XUAN-DANG, Maître Assistant
- . G. MICHEL, Docteur-Ingénieur

Monsieur COUTO BARONE Dante Augusto

est autorisé à présenter une thèse en soutenance en vue de l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Informatique".

Fait à Grenoble, le 18 octobre 1984

Le Président de l'I.N.P.-G

D. BLOCH
Président
de l'Institut National Polytechnique
de Grenoble

P.O. le Vice-Président.

