

SOLAR

BUS D'ENTRÉE/SORTIE

Description et Utilisation

BUS D'ENTREE/SORTIE

MANUEL DE FONCTIONNEMENT

- * en haut de page indique le changement complet de la page par rapport à l'IE précédent
- l en marge indique la partie modifiée par rapport à l'IE précédent

SOMMAIRE	Page
1 - GENERALITES	1-1
2 - REPERTOIRE DES SIGNAUX	2-1
3 - DEFINITION DES LIGNES	3-1
4 - DEROULEMENT DES SEQUENCES	4-1
4.1 - DECOUPAGE DANS LE TEMPS	4-1
4.2 - SYSTEME D'ENTREE-SORTIE	4-1
4.3 - SYSTEME D'INTERRUPTION	4-2
4.3.1 - Interruptions NORMALES	4-3
4.3.2 - Interruptions d'EXCEPTION	4-5
4.3.3 - Utilisation du TRAP WORD	4-6
5 - EXEMPLES DE SEQUENCES E/S	5-1
6 - CODAGE ET DECODAGE D'ADRESSE	6-1
7 - ANNEXES	7-1
7.1 - ORGANISATION GENERALE	7-1
7.2 - LIGNES DE FOND DE BAC	7-2
7.3 - UTILISATION DES LIGNES	7-4
7.4 - LES 3 ETATS DU BUS	7-5
7.5 - ARBRE DE POLLING	7-6
7.6 - INTERRUPTIONS CANAL	7-7
7.7 - HARD LEVEL WORD (H.L.W.)	7-8
7.8 - TRAP WORD	7-9
7.9 - SEQUENCE D'ENTREE-SORTIE - Exemple 1	7-10
7.10- SEQUENCE D'ENTREE-SORTIE - Exemple 2	7-11
7.11- ADRESSES	7-12
7.12- DECODAGE ADRESSE REGISTRE	7-13
7.13- POLLING COUPLEUR STANDARD	7-14
7.14- POLLING COUPLEUR CANAL	7-15
7.15- INTERRUPTIONS ET POLLING	7-16
7.16- LEXIQUE	7-17

AVERTISSEMENT

C'est à dessein que seront utilisés dans les développements ci-après certains termes, groupes de mots, mnémoniques et abréviations en langue anglaise ; nous avons pensé en effet, que cette pratique faciliterait la concordance entre le texte et les noms des signaux, ceux-ci ayant été systématiquement baptisés en anglais.

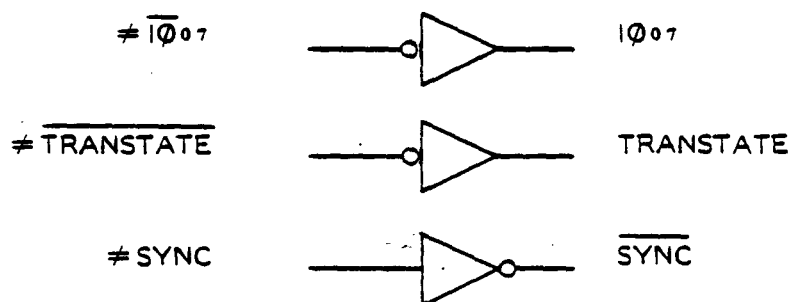
Pour cette raison, le présent manuel comporte en dernière page un petit lexique des termes anglais figurant dans le texte et auquel le lecteur non familiarisé avec l'usage de cette langue pourra se reporter utilement (Annexe 7.16).

Nous traiterons ici le fonctionnement du bus I/O dans sa forme la plus complexe, c'est-à-dire celle où quatre processeurs P0 à P3 peuvent se le disputer. Il faut entendre par processeur Pi indifféremment un CPU 40, un CPU 65 ou un IOP-16, l'un au moins des quatre processeurs étant obligatoirement un CPU. S'il y a plus d'un CPU dans la configuration, nous parlerons alors de multiprocesseur vrai, par opposition à la configuration ne comportant qu'un seul CPU accompagné de trois processeurs IOP-16.

Quant au SOLAR 16-05, rappelons qu'il est par conception monoprocesseur et qu'il utilise le bus I/O pour les échanges avec sa mémoire, laquelle se comporte pour lui comme un coupleur de périphérique.

1 - GENERALITES

- Pour alléger le texte, le bus d'entrée-sortie sera désigné dans ce qui suit sous l'abréviation anglaise bus I/O (bus Input/Output).
- Le bus I/O assure le transfert de tous les signaux nécessaires au dialogue et aux échanges entre processeurs et coupleurs, ainsi qu'entre processeurs dans le cas des multiprocesseurs.
- Les coupleurs n'étant pas reliés au bus mémoire, ils ne peuvent qu'utiliser les lignes du bus I/O et l'intermédiaire du processeur auquel ils sont affectés pour réaliser un échange avec la mémoire (voir annexe 7.1 : organisation générale).
- Les lignes du bus I/O portent le nom du signal qu'elles véhiculent ; ce nom est immuable quelle que soit, à un instant donné, la signification du signal, celle-ci étant susceptible d'évoluer dans le temps.
- Le nom des signaux circulant sur le bus I/O est d'office précédé du signe \neq signifiant que le signal en question est émis ou reçu par les cartes sur leur connecteur arrière (c'est-à-dire côté fond de bac).
- Tous les signaux du bus I/O sont synchronisés sur une horloge symétrique. Ce signal, reçu par toutes les cartes (qu'elles soient on-rack ou off-rack) est appelé # CLOCK.
- A l'exception des signaux # CLOCK. et # $\overline{\text{CLEAR}}$, lesquels m'appartiennent pas en propre au bus I/O et sont dits signaux de service, tous sont émis soit par une carte processeur, soit par une carte coupleur, soit encore par la carte Rack-Extender qui sert de trait d'union entre les coupleurs off-rack et les processeurs.
- De tous les signaux propres au bus I/O, seul # SYNC n'est pas transmis sous forme complémentée (Voir annexe 7.2 : Signaux de Fond de bac). Pour éviter d'émailler le texte d'une surabondance de noms de signaux surmontés d'une barre et précédés d'un # nous les prendrons après leur inversion tant au niveau des coupleurs que des processeurs. C'est ainsi que dans les explications à venir il sera fait mention par exemple de $\overline{\text{IO 07}}$, TRANSTATE et $\overline{\text{SYNC}}$ respectivement obtenus à partir des signaux # IO 07, # TRANSTATE et # SYNC de la manière suivante :



- A l'état repos, toutes les lignes du bus sont au niveau "1" logique, y compris la ligne SYNC, le signal # SYNC étant actif au "0" logique (voir plus loin).

2 – REPERTOIRE DES SIGNAUX

Etiquette	Signification anglaise	Signification Française *	Emis par	Reçu par	Broches
# \overline{IO} 0-15	Input/Output	Entrée-sortie Ces 16 bits peuvent représenter : . des informations . des adresses . des interruptions	Toute carte raccordée au bus I/O	Toute carte raccordée au bus I/O	A 10-25 et E 10-25
# \overline{PPIO} 03	Processor Priority Input - Output	Priorité des processeurs sur le bus I/O sur le 05 # \overline{PPIO} 0 = bit de parité	Tous processeurs Processeur et mémoire	Processeurs Processeur et mémoire	A 5-8 et E5-8 A5 et E5
# $\overline{ADDRESS}$	ADDRESS	Adresse - le bus I/O est dit dans l'état ADR pendant toute la durée de ce signal	Processeur Po (le plus prioritaire)	Toute carte raccordée au bus I/O	A27 et E27
# $\overline{TRAN-STATE}$	TRANSfert STATE	Etat de transfert (des informations ou des interruptions) Le bus I/O est dit dans l'état TRANS pendant toute la durée de ce signal	idem	idem	A9 et E9
# SYNC	SYNChronizer	Synchroniseur-signal permettant de synchroniser certains coupleurs avec le bus.I/O	Certains coupleurs et le Rack Extender	idem	A28 et E28
# $\overline{PROCESSOR INT}$	PROCESSOR INTerrupt	Interruption inter-processeurs	Processeur 40 ou 65	Tous processeurs	A26 et E26
SIGNAUX DE SERVICE # CLOCK	CLOCK	Horloge	Minipupitre	Toutes cartes	A34, E34
# CLEAR	CLEAR	Signal d'initialisation (ou logique entre l'initialisation à la mise sous tension et l'initialisation pupitre)	POWER MONITOR ou INI	Toutes cartes	A33, E33 B32, H32

* Des explications succinctes concernant l'utilisation de ces signaux sont fournies dans le chapitre 3 : Définition des lignes.

3 - DEFINITION DES LIGNES

Au total : 26 lignes (voir annexes 7.2 et 7.3)

- 16 lignes $\overline{IO} 0 - 15$

Ces lignes sont bidirectionnelles et la signification des signaux qu'elles véhiculent dépend et du sens de l'échange et du moment où il intervient.

a) Sens Processeur – Coupleur :

a.1) Pendant la durée du signal $\# \overline{ADDRESS}$, l'état des lignes $\overline{IO} 0 - 15$ code une adresse pouvant concerner soit tous les coupleurs, soit un certain nombre de coupleurs, soit un seul coupleur, ou plus finement encore, un registre déterminé dans le coupleur concerné.

a.2) Pendant la durée du signal $\# \overline{TRANSTATE}$, lequel suit systématiquement le signal $\# \overline{ADDRESS}$, le processeur qui a le bus émet sur les lignes $\overline{IO} 0 - 15$ soit une info à sortir, soit un mot de commande, dans un cas comme dans l'autre, cette émission concerne exclusivement un registre bien précis dans un coupleur déterminé, à savoir le registre de sortie qui vient d'être adressé.

Autrement dit, un processeur émet sur les lignes $\overline{IO} 0 - 15$ pendant $\# \overline{TRANSTATE}$ que s'il vient d'adresser un registre de sortie dans un coupleur défini. Dans tous les autres cas, l'état des lignes $\overline{IO} 0 - 15$ concerne obligatoirement un échange dans le sens Coupleurs-Processeurs (voir ci-dessous).

b) Sens Coupleur – Processeurs

b.1) Pendant $\# \overline{TRANSTATE}$, la signification des 16 signaux circulant sur les lignes $\overline{IO} 0 - 15$ dépendra de ce qu'elle a été pendant $\# \overline{ADDRESS}$. A une adresse de registre d'entrée appartenant à un coupleur déterminé, succédera soit une info à entrer, soit un mot d'état, selon le type du registre adressé.

A une adresse concernant un groupe de coupleurs, succédera sur les lignes $\overline{IO} 0 - 15$ un ensemble de signaux (de 1 à 16), chacun signifiant que parmi les coupleurs adressés, il y en a au moins 1 par ligne activée qui réitère ainsi une demande d'interruption (voir paragraphe 4.3).

b.2) En l'absence des signaux $\# \overline{ADDRESS}$ et $\# \overline{TRANSTATE}$ (rappelons que cela signifie que les lignes $\overline{ADDRESS}$ et $\overline{TRANSTATE}$ sont à 1) l'état des 16 lignes $\overline{IO} 0 - 15$ représente ce que nous désignons sous le vocable TRAP WORD et permet à 16 sous-ensembles de coupleurs (pouvant comprendre de 1 à 240 coupleurs), rattachés chacun à une ligne, de faire savoir qu'un coupleur au moins dans le sous-ensemble désire provoquer une interruption.

- 4 lignes \overline{PPIO}_0 à \overline{PPIO}_3

Elles permettent aux processeurs P0 à P3 de formuler une demande d'attribution du bus I/O, laquelle sera satisfaite selon un ordre de priorité décroissant de 0 à 3. La logique déterminant la priorité est implantée sur chaque processeur, chacun envoyant son $\# \overline{PPIO}$ sur sa ligne propre et recevant les 3 autres.

Le Solar 16-05 étant monoprocesseur, il profite de l'inutilisation des lignes \overline{PPIO} en tant que telles pour la transmission de la parité mémoire sur la ligne \overline{PPIO}_0 .

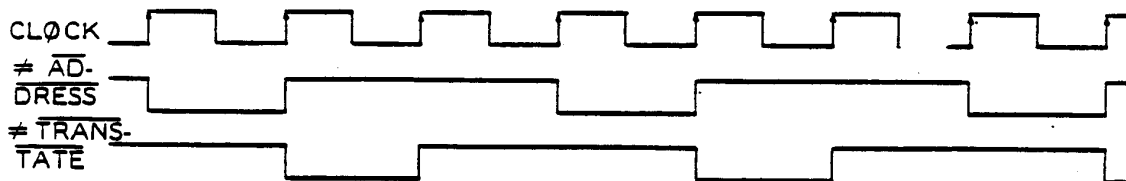


- 1 ligne ADDRESS

Le processeur le plus prioritaire (Po) émet sur cette ligne durant une période d'horloge le signal # ADDRESS qui a pour but de valider les circuits de décodage d'adresse de tous les coupleurs. Il est émis à chaque montée de CLOCK à la seule condition qu'il n'y ait pas # TRANSTATE.

- 1 ligne TRANSTATE

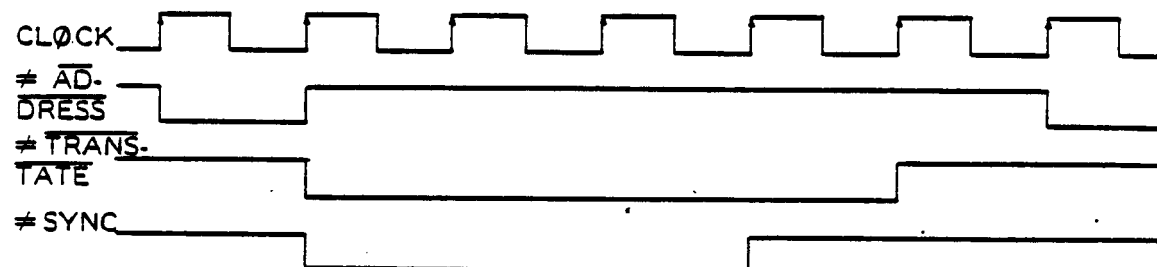
Le processeur le plus prioritaire (Po) fait systématiquement suivre sur cette ligne le signal # TRANSTATE au signal # ADDRESS, c'est-à-dire que la montée de CLOCK qui met fin à l'un lance l'autre sur la ligne TRANSTATE. En l'absence du signal # SYNC la montée suivante de CLOCK fera disparaître le signal # TRANSTATE et une période d'horloge plus tard le processeur Po recommencera à émettre successivement les signaux # ADDRESS et # TRANSTATE. D'où la séquence répétitive suivante que le processeur Po relance en permanence dès l'instant qu'il n'y a pas le signal # CLEAR sur le bus.



- 1 ligne SYNC

Certains coupleurs qui seraient pris de vitesse par la séquence ci-dessus, ainsi que le Rack Extender pour synchroniser le bus du bac d'extension avec celui du bac principal, utilisent cette ligne pour émettre le signal # SYNC. Celui-ci a pour effet de maintenir le signal # TRANSTATE, et par suite de "figer" le bus, puisque nous avons vu qu'il ne peut disparaître sur une montée de CLOCK que s'il n'y a pas # SYNC.

La ligne SYNC étant à l'état haut au repos, on voit que # SYNC = 1 signifie synchronisation et que # SYNC = 0 marque un défaut de synchronisation nécessitant une mise en attente du bus. L'enchaînement des séquences est alors le suivant :



- 1 ligne PROCESSOR INT

Cette ligne concerne tous les processeurs et aucunement les coupleurs. Seuls les processeurs déroulant du programme (c'est-à-dire les CPU40 et 65 et non les IOP-16) peuvent l'utiliser, lorsqu'ils rencontrent l'instruction IPI, pour émettre, en l'absence de # ADDRESS et # TRANSTATE, le signal # PROCESSOR INT qui dure période d'horloge et dont l'effet est le suivant : tous les processeurs (CPU et IOP-16) y compris celui qui l'a émis, vont lire, en mémoire commune, une "boite aux lettres" chargée par le programmeur de telle sorte que l'un d'eux (voire lui-même) se déroute de la tâche qu'il traitait pour en attaquer une nouvelle (voir instruction IPI).

4 - DÉROULEMENT DES SÉQUENCES

Pour les explications qui vont suivre, nous allons, comme annoncé plus haut, considérer les signaux du bus I/O après leur inversion au niveau des processeurs et des coupleurs (voir page 1.1). Nous verrons successivement les séquences relatives au système d'entrée-sortie, puis celles concernant le système d'interruption ; mais toutes ces séquences étant cadencées par la fin des signaux ADDRESS et TRANSTATE, il nous faut auparavant présenter ce découpage dans le temps.

4.1 - DÉCOUPAGE DANS LE TEMPS

Nous avons vu que la signification des 16 signaux IO₀₋₁₅ est différente selon la présence ou l'absence des signaux ADDRESS et TRANSTATE.

Ceux-ci étant synchronisés sur l'horloge de référence (CLOCK), leur évolution crée un découpage précis dans le temps, ce qui permet de considérer trois états distincts pour le bus I/O :

- l'état ADR (Adresse) lorsque ADDRESS = 1
- l'état TRANS (TRANSfert) lorsque TRANSTATE = 1
- l'état INT (INTerruption) lorsque ADDRESS = 0 et TRANSTATE = 0

Les changements d'état se font sur les fronts montants de CLOCK (voir annexe 7.4).

La montée de SYNC, laquelle ne peut se faire que si TRANSTATE = 1. n'entraîne pas un changement d'état, mais "fige" le bus dans l'état TRANS. En effet, TRANSTATE ne peut repasser à 0 que si SYNC = 0 et ADDRESS ne peut monter que si TRANSTATE = 0.

Lorsque SYNC retombe (après n périodes d'horloge) il faut attendre la prochaine montée de CLOCK pour que TRANSTATE redescende, puis la suivante pour que TRANSTATE étant à 0, le signal ADDRESS passe à 1 pour une période d'horloge et relance le découpage ADR, TRANS, INT ...

On voit que les états ADR et INT ne durent jamais plus d'une période d'horloge, alors que l'état TRANS peut s'étaler sur (n+1) périodes lorsque SYNC monte pour une durée de n fois une période d'horloge.

A titre d'exemple, le SYNC émis par les coupleurs MXP04 ou MXP16 est maintenu haut quatre périodes d'horloge ce qui fait que le bus I/O reste à l'état TRANS durant 5 périodes de CLOCK.

Rappelons que SYNC peut également monter du fait du Rack Extender pour synchroniser les I/O du rack d'extension avec le bus du rack principal.

L'utilisation des lignes du bus I/O pendant les trois états ADR, TRANS et INT est différente selon qu'il s'agit d'une séquence d'entrée-sortie ou d'une séquence d'interruption (voir paragraphe 4.3 : Système d'interruption).

4.2 - SYSTEME D'ENTRÉE-SORTIE

Le système d'entrée-sortie est un système qui s'adresse, mieux qu'aux coupleurs, à leurs registres d'entrée ou registres de sortie ; ceux-ci peuvent être de quatre types :

- 1- registres INPUT DATA (entrée d'info)
- 2- registres OUTPUT DATA (sortie d'info)
- 3- registres INPUT STATUS (entrée de mot d'état)
- 4- registres OUTPUT COMMAND (sortie de fonction).

Matériellement, rien n'empêche un coupleur de posséder un nombre quelconque de ces registres, mais le coupleur "classique" n'en compte que 4 : un de chaque type.

Des coupleurs plus complexes en nécessitent 5 ou 6 (par exemple 2 registres de Mot d'état) et certains coupleurs spéciaux (tels les MXPxx) plusieurs de chaque type.

Chaque registre a une adresse unique codée sur 16 bits *et qu'il cherche à reconnaître, à chaque état ADR, parmi les 16 bits IO 0 – 15 . Le registre qui reconnaît son adresse agit alors selon sa nature :

- un registre INPUT DATA ou un registre INPUT STATUS envoie sur le bus l'info ou le mot d'état qu'il contient. Dans un cas comme dans l'autre, la prise en compte par le processeur ayant adressé le registre en question se fera sur le CLOCK qui fait retomber TRANSTATE.
- un registre OUTPUT DATA ou OUTPUT COMMAND se tient prêt à prendre en compte à la fin de TRANS l'info ou le Mot de commande que le processeur mettra sur le bus.

La prise en compte du contenu des 16 lignes IO_{0-15} se fait donc, que ce soit par le processeur en entrée, ou le registre de sortie du coupleur, à la fin de l'état TRANS, c'est-à-dire sur la montée du CLOCK faisant retomber TRANSTATE, ce qui implique que SYNC = 0.

Mais pour adresser un tel registre, le processeur doit auparavant avoir formulé une demande d'attribution du bus en faisant monter son PPIO pendant l'état INT et s'être avéré le plus prioritaire.

En résumé, une séquence d'entrée-sortie commence par l'état INT et l'utilisation du bus I/O durant chacun des 3 états est la suivante :

- 1/ ETAT INT : chaque processeur désirant faire un échange avec un coupleur fait monter son PPIO i. Seul le plus prioritaire (indice i le plus petit) reste à l'état haut et ne redescendra qu'au début du prochain état INT.

Durant cet état les 16 lignes IO 0 – 15 restent à l'état haut, sauf s'il y a des interruptions, leur état formant alors le TRAP WORD (voir plus loin le système d'interruption).

- 2/ ETAT ADR : Seul le processeur ayant son PPIO à 1 peut envoyer une adresse de registre sur les lignes IO 0 – 15 . Tous les coupleurs la reçoivent et leur logique de décodage se trouvant validée par le signal ADDRESS = 1, un seul registre dans un seul coupleur se verra concerné.

- 3/ ETAT TRANS : L'utilisation des 16 lignes IO 0 – 15 (qui rappelons-le sont bidirectionnelles) sera différente selon que le processeur aura adressé un registre INPUT ou OUTPUT.

- Le registre INPUT adressé met sur le bus les 16 bits de l'info (si c'est un registre INPUT DATA) ou du Mot d'état (si c'est un registre INPUT STATUS) dans les meilleurs délais, afin que le processeur puisse les prendre en compte sur le CLOCK suivant. C'est l'échange le plus rapide, celui qui n'a pas recours à SYNC.

Si le coupleur, ou le Rack Extender (pour un coupleur off-rack) fait monter SYNC, le processeur, dont le PPIO est toujours à 1. prend en compte les 16 bits IO_{0-15} à chaque montée de CLOCK tant que TRANSTATE est à 1, chaque mot écrasant le précédent. Et seul le dernier sera "pris pour bon" par le processeur.

- Le registre OUTPUT adressé prend en compte sur le CLOCK qui fait retomber TRANSTATE les 16 bits envoyés par le processeur et qui représentent soit une info à sortir (s'il s'agit d'un registre OUTPUT DATA) soit une fonction (s'il s'agit d'un registre OUTPUT COMMAND).

Bien entendu, le processeur maintient les 16 bits pendant tout l'état TRANS, ce qui peut représenter plusieurs périodes s'il y a montée de SYNC.

4.3 - SYSTEME D'INTERRUPTION

Il faut entendre par "interruption" toute requête formulée par un coupleur à un processeur en vue de lui faire interrompre le programme en cours pour qu'il puisse satisfaire à un besoin quelconque dudit coupleur.

* Voir chapitre 6 : Codage et décodage des adresses.

Nous appellerons "interruptions NORMALES" celles qui sont attendues par le processeur procédant à un échange de caractères et qui ne peuvent signifier que "buffer de sortie vide" ou "info en entrée valide", selon le sens de l'échange. Il n'y a alors pas lieu de lire le registre d'état (registre INPUT STATUS) pour déterminer la nature de la requête formulée par le coupleur.

Toutes les autres interruptions sont dites "interruptions EXCEPTION" et peuvent signifier par exemple "fin d'échange", "défaut périphérique", "erreur de cadence", etc... Le processeur doit alors lire le registre d'état du coupleur pour savoir ce qu'il en est.

Nous distinguons deux classes de coupleurs. Ceux qui travaillent en mode canal (canal HDC ou MDC ou LDC) sont appelés "coupleurs CANAL", alors que ceux qui n'opèrent pas en mode canal sont dits "coupleurs STANDARD".

Parmi les coupleurs CANAL, nous aurons :

- Les coupleurs HDC (Highspeed Data Channel) opérant en canal à haute vitesse,
- Les coupleurs MDC (Mediumspeed Data Channel) opérant en canal à vitesse moyenne,
- Les coupleurs LDC (Lowspeed Data Channel) opérant en canal à basse vitesse.

Nous verrons que le système d'interruption est différent selon la catégorie dont il s'agit.

Mais tous procèdent de la même manière pour adresser une interruption à un processeur : ils activent pendant l'état INT la ligne IO sur laquelle ils sont branchés. Comme il n'y a que 16 lignes IO pour tous les coupleurs, il est bien évident que certaines lignes devront nécessairement regrouper en "OU logique" un nombre plus ou moins grand d'interruptions. Le processeur qui, durant l'état INT trouve un bit IO à 1 sur une telle ligne doit faire une recherche appelée polling pour savoir lequel (ou lesquels) parmi tous les coupleurs pouvant appeler sur cette ligne vient d'envoyer une interruption.

Les 16 bits trouvés sur les lignes IO 0 - 15 pendant l'état INT forment ce que nous appelons le TRAP WORD *.

Celui ci est basé sur le principe suivant :

Plus le nombre des coupleurs susceptibles d'envoyer une interruption sur une même ligne est grand, plus il faut de temps pour déterminer lequel est à satisfaire (voir ci-dessous en se reportant à l'Annexe 7.5 : Arbre de Polling).

En particulier, la ligne IO 0 - 15 regroupe, outre les interruptions NORMALES des coupleurs STANDARD, toutes les interruptions EXCEPTION quelle qu'en soit la provenance. Autrement dit, un coupleur CANAL dispose de deux lignes IO pour envoyer ses interruptions dans le TRAP WORD une ligne canal pour les interruptions NORMALES et la ligne lois pour les interruptions EXCEPTION ; alors que les coupleurs STANDARD envoient les unes et les autres sur la même ligne IO 0 - 15.

Nous nous intéresserons d'abord aux interruptions NORMALES des coupleurs CANAL et des coupleurs STANDARD.

4.3.1 - Interruptions normales

a/ Coupleurs HDC (canal HDC)

Afin d'obtenir un maximum de rapidité dans les échanges effectués en canal HDC, chaque coupleur HDC envoie ses interruptions NORMALES au processeur auquel il est affecté sur une ligne qui lui est propre, choisie parmi les 7 ou 8 lignes (IO 0 à IO 6 ou IO 7)* réservées à cet effet.

Il en résulte que le processeur qui, pendant l'état INT, trouve un de ces bits à 1, n'a pas à faire de polling, le coupleur appelant étant instantanément identifié. Son contexte canal est alors aussitôt sélectionné, directement à partir du bit du Trap Word, par une transposition hardware à grande vitesse. C'est la réponse la plus rapide qui puisse être faite à une interruption NORMALE.

* Voir paragraphe 4.3.3 : Utilisation du TRAP WORD.

* Voir Coupleurs MDC pour l'utilisation de IO 7 .

b/ Coupleurs MDC (canal MDC)

Les lignes IO₇ à IO₁₀ peuvent être attribuées chacune en propre à un parmi quatre processeurs P₀ à P₃ pour recevoir des interruptions NORMALES en provenance pour chaque ligne de 16 coupleurs MDC possibles. A noter que dans le cas d'un multiprocesseur avec un processeur P₀ de type 65, ce dernier n'opérant pas en canal MDC, il n'y aura pas de MDC₀ et la ligne IO₇ pourra être utilisée pour un huitième coupleur HDC, travaillant en canal ultrarapide HDC.

Supposons que, pendant un état INT, le processeur P₂ trouve à 1 le bit IO₉ (appelé MDC₂ dans le Trap Word), il sait que cela signifie qu'un au moins des 16 coupleurs MDC possibles rattachés à la ligne IO₉ envoie une interruption.

Pour identifier lequel (ou lesquels) il lui faudra, pendant un prochain état INT, faire monter son PPIO₀₂ et que celui-ci s'avère prioritaire pour qu'il puisse durant l'état ADR suivant envoyer une adresse de polling que nous baptiserons NORMAL MDC POLLING. (voir annexe 7.6)

Seuls les 16 coupleurs MDC branchés sur la ligne IO₉ se verront concernés par cette adresse et tous répondront, pendant l'état TRANS qui suit, chacun sur une ligne propre parmi les 16 lignes IO₀₋₁₅ en faisant IO_i = 1 s'il avait précédemment mis à 1 le bit IO₉ du TRAP WORD. Ces 16 bits lus par le processeur P₂ pendant l'état TRANS forment un mot appelé NORMAL MDC WORD. La micro-machine fait alors un Discover Bit pour déterminer lequel des 16 coupleurs MDC appelle (ou, s'ils sont plusieurs, quel est le plus prioritaire : bit à 1 le plus à gauche) afin d'adresser, le changement de contexte étant fait, soit son registre INPUT DATA, soit son registre OUTPUT DATA selon le sens de l'échange, puisque, rappelons-le, il s'agit d'une interruption NORMALE.

Cette séquence d'entrée-sortie ne pourra bien sûr se réaliser que lorsque P₂ aura obtenu à nouveau le bus grâce à son PPIO₀₂, comme vu plus haut.

c/ Coupleurs LDC (canal LDC)

Les lignes IO₁₁ à IO₁₄ peuvent être attribuées chacune en propre à un parmi quatre processeurs P₀ à P₃ pour recevoir des interruptions NORMALES en provenance pour chaque ligne de 64 coupleurs LDC possibles, classés en 4 GROUPEs numérotés de 0 à 3 (voir annexe 7.5).

Supposons que, pendant un état INT, le processeur P₂ trouve à 1 le bit IO₁₃ (appelé LDC₂ dans le Trap Word), il sait que cela signifie qu'un au moins des 64 coupleurs LDC possibles connectés à la ligne IO₁₃ envoie une interruption.

Il procédera à une opération de polling tout à fait similaire à celle vue pour le canal MDC, à la différence qu'elle pourra se répéter 4 fois si le coupleur appelant est dans le GROUPE 3.

En effet, à une adresse de polling, baptisée cette fois NORMAL LDC POLLING, et codant sur 2 bits le GROUPE 0, les 16 coupleurs LDC concernés enverront pendant l'état TRANS qui suit leur réponse sur les lignes IO₀₋₁₅ formant ainsi un mot appelé "NORMAL LDC GROUPE 0".

Si aucun bit à 1 n'est découvert dans ce mot, un nouveau polling est lancé par l'envoi d'une adresse NORMAL LDC POLLING, laquelle ne diffère de la précédente que par le fait qu'elle code maintenant le GROUPE 1. Les réponses des 16 coupleurs concernés formeront le "NORMAL LDC GROUPE 1", dans lequel la micromachine fait un Discover Bit, et ainsi de suite jusqu'à la découverte d'un bit à 1. Le processeur saura à ce moment là quel est le coupleur appelant et l'échange en canal LDC pourra se réaliser par une séquence d'entrée-sortie classique.

d/ Coupleurs STANDARD

Tous les coupleurs STANDARD envoient leurs interruptions NORMALES sur la ligne IO₁₅. En se reportant à l'arbre de polling (annexe 7.5) on voit qu'ils peuvent être jusqu'à 240, répartis en 15 niveaux de 16 sous-niveaux chacun, le niveau 0 étant réservé aux interruptions issues du pupitre opérateur.

Le bit IO₁₅ (appelé I/O dans le TRAP WORD) est donc le OU logique des 16 bits du H.L.W. (Hard Level WORD) chacun étant lui-même le OU logique des interruptions de 16 coupleurs STANDARD possibles et de 48 interruptions EXCEPTION.

Lorsqu'un processeur voit dans le Trap Word le bit IO₁₅ à 1, il envoie une adresse baptisée HARD LEVEL POLLING.

Cette adresse est unique et peut être envoyée par n'importe quel processeur. De plus, elle est reconnue par tous les coupleurs, et ceux-ci, pendant l'état TRANS suivant, enverront leur réponse sur la ligne IO_i qui correspond à leur niveau i. Les 16 bits IO₀₋₁₅ formeront alors le H.L.W.

Pour savoir s'il est concerné par un des bits à 1 de ce H.L.W., le processeur qui l'a provoqué va le comparer avec le contenu de son registre IM qui est le masque sélectif des interruptions (bit à 1 = niveau masqué) en faisant $H.L.W. \wedge IM$. Il lui faudra ensuite comparer le résultat avec le contenu de son registre HV qui reflète les tâches en attente sauf celle correspondant au bit à 1 le plus à gauche qui est la tâche en cours (tâche soft si aucun bit à 1).

Si le nouvel appelant sur le niveau i est moins prioritaire, la tâche en cours se poursuit et ce n'est qu'à la fin de son traitement qu'il sera procédé à une nouvelle hiérarchie.

Par contre, si le nouvel appelant s'avère plus prioritaire, la tâche en cours sera interrompue et un changement de contexte sera effectué au profit de la tâche de niveau i , tandis que le bit correspondant de HV sera positionné à 1.

Il reste alors au Processeur à déterminer lequel (ou lesquels) parmi les 16 coupleurs STANDARD possibles a appelé sur le niveau i . Pour ce faire, il envoie pendant ADR une adresse de polling baptisée NORMAL I/O POLLING.

Seuls les 16 coupleurs en question se reconnaîtront et répondront, pendant l'état TRANS suivant, sur les 16 lignes $IO_0 - 15$, formant ainsi le "NORMAL I/O WORD" correspondant au niveau i concerné (c'est-à-dire, dans l'arbre de polling, la colonne directement en dessous du bit i du H.L.W.)

Le processeur fera alors un Discover Bit dans le "NORMAL I/O WORD" pour satisfaire le coupleur le plus prioritaire (par son sous-niveau) en entrée ou en sortie, puisqu'il s'agit, rappelons-le, d'une interruption NORMALE. Si aucun bit à 1 ne devait être trouvé dans le NORMAL I/O WORD cela signifierait qu'il s'agit d'une interruption EXCEPTION (voir plus loin).

Nous venons de voir le cas où un bit à 1 du H.L.W. concernait effectivement le processeur qui avait envoyé l'adresse HARD LEVEL POLLING.

Mais il n'en est pas toujours ainsi : nous avons vu qu'un masquage sélectif dans le H.L.W. permet à chaque processeur d'avoir ses niveaux propres, grâce au registre de masque IM. C'est donc en fait le niveau le plus prioritaire, parmi ceux qui ne sont pas masqués dans le registre IM, que le processeur recherche dans le H.L.W. qu'il a provoqué par son HARD LEVEL POLLING et il peut lui arriver de n'en point trouver, le bit IO_{15} étant monté du fait d'un ou plusieurs bits de niveau concernant d'autres processeurs. Ceux-ci les trouveront lorsque viendra leur tour de lire le Trap Word et de traiter le bit $IO_0 - 15$.

On voit que les 240 coupleurs STANDARD possibles sont partagés entre les processeurs 40 ou 65 possibles (les processeurs IOP-16 n'opérant pas en NORMAL I/O), et que leur niveau d'interruption dépend du processeur auquel ils sont rattachés.

4.3.2 - Interruptions EXCEPTION

a/ Coupleurs STANDARD

Ces coupleurs font monter IO_{15} par le même bit du H.L.W. qu'il s'agisse d'une interruption NORMALE ou d'une interruption EXCEPTION. Le processeur concerné par ce bit procédera comme vu plus haut et s'il ne trouve aucun bit à 1 dans le NORMAL I/O WORD, enverra une nouvelle adresse baptisée EXCEPTION LEVEL 0 (EXL 0) à laquelle répondront les 16 mêmes coupleurs STANDARD, cette réponse formant le mot EXCEPTION LEVEL 0 WORD. Et un Discover Bit sur ce mot révélera quel est le coupleur le plus prioritaire (par son sous-niveau) qui fait une interruption EXCEPTION.

Le programme de service associé appelé par cette interruption devra lire le registre d'état du coupleur (registre INPUT STATUS) et cette lecture aura pour effet de remettre à zéro les bits de ce registre, y compris celui ou ceux qui avaient provoqué l'interruption en question. Il en résultera que ce coupleur ne répondra plus par 1 aux prochaines adresses de polling qu'il pourra recevoir.

En effet, une fois le présent programme de service terminé par l'instruction ΔCK^* , le processeur continuera à explorer la colonne en envoyant des adresses de polling auxquelles répondront les

* Voir cette instruction dans le Manuel de Référence.

coupleurs de ce niveau qui n'auraient pas encore été servis. Si le mot EXCEPTION LEVEL 0 WORD ne comporte pas de 1, le processeur enverra l'adresse EXCEPTION LEVEL 1 (EXL 1), qui provoquera la réponse EXCEPTION LEVEL 1 WORD (paragraphe suivant) et ainsi de suite, afin que la recherche des interruptions de même niveau (c'est-à-dire même colonne de l'arbre de polling) ait un caractère systématique.

b/ Coupleurs CANAL

Quel que soit le type de canal, les interruptions EXCEPTION en provenance des coupleurs CANAL font monter le bit IO₁₅ et la réponse au polling se fait dans le mot EXCEPTION LEVEL (1 ou 2) WORD que provoque l'envoi de l'adresse EXCEPTION LEVEL (1 ou 2) lorsque les mots NORMAL I/O WORD et EXCEPTION LEVEL 0 WORD ont été trouvés vides.

Une interruption EXCEPTION émise par un coupleur CANAL ne sera donc traitée que si aucun coupleur STANDARD de même niveau n'a envoyé d'interruption (NORMALE ou non) et que son sous-niveau est le plus prioritaire.

A noter que si sur un niveau donné (3 par exemple) nous avons moins de 16 coupleurs STANDARD (7 par exemple) les bits IO₇ à IO₁₅ du mot EXCEPTION LEVEL 0 WORD pourront être utilisés pour des interruptions EXCEPTION en provenance de coupleurs CANAL (voir annexe 7.7).

4.3.3 - Utilisation du Trap Word

Le TRAP WORD est l'état des 16 lignes IO₁₅, auquel vient s'ajouter celui de la ligne PROCESSOR INT, ces états étant pris uniquement pendant l'instant INT.

Mais un processeur donné n'est intéressé que par certains de ces 16 bits et il se constitue un TRAP WORD personnel dans lequel il mémorise à chaque état INT l'image des ligne IO qui le concernent (voir annexe 7.8).

Et c'est ce TRAP WORD personnel qu'il viendra lire à chaque "trappe" du microprogramme qu'il déroule.

Tous mémorisent le bit PROC et tous (sauf les processeurs IOP-16) le bit I/O correspondant à la ligne IO₁₅.

Les bits MDC₀ et LDC₀ sont mémorisés par le processeur P₀, les bits MDC₁ et LDC₁ par le processeur P₁ et ainsi de suite.

Toutefois, si le processeur P₀ est de type 65 il n'y a pas de MDC₀ et ce bit (IO₇) devient un huitième bit HDC.

Les bits-HDC (IO₀ à IO₆ ou IO₇) ne concernent que les processeurs de type 65 ou IOP-16 à l'exception du bit IO₀ (ou IO₁) qui peut être attribué en propre à un processeur de type 40.

A cette exception près, les processeurs 65 et IOP-16 reçoivent à chaque état INT tous les bits HDC, mais ils n'en mémorisent que 2 à un moment donné, le choix en ayant été fait par programme en chargeant leurs deux registres CIL-A et CIL-B (Channel Interrupt Line) du numéro des lignes à "surveiller".

Supposons une configuration comprenant 4 processeurs, P₀ de type 65, P₁ de type 40, P₂ et P₃ de type IOP-16 et voyons comment ils peuvent se partager les 8 coupleurs CANAL susceptibles d'être rattachés aux 8 lignes HDC (IO₀ à IO₇).

P₁ pourra être affecté en permanence au coupleur branché à la ligne IO₀, ou IO₁, et uniquement à celui-là.

P₀ pourra être affecté temporairement par le programmeur aux 2 coupleurs branchés sur IO₂ et IO₅, tant que son CIL-A aura été chargé par 2 et son CIL-B par 5.

II en sera de même pour P₂ et P₃ avec par exemple IO₁ et IO₇ pour le premier et IO₄ et IO₆ pour l'autre.

Cette attribution est mouvante au gré du programmeur qui doit prendre garde qu'il n'y ait jamais, à un instant donné, la même valeur dans 2 CIL quelconques. Les CIL sont des registres à 4 bits dont 3 servent à coder le n° de la ligne à "surveiller" et le quatrième de bit d'inhibition.

-Rappel sur les configurations possibles

	16-65	16-40	16-05	IOP-16	n/m signifie :
HDC	2/8	1/1 ⁽¹⁾	/	2/8	n échanges simultanés possibles
MDC	/	6/6 ⁽¹⁾	/	16/16	parmi m périphériques connectés
LDC	64/64	64/64	16/16	64/64	

¹ L'utilisation d'une ligne HDC sur 16-40 ramène à 5/6 les possibilités en MDC.

5 - EXEMPLES DE SEQUENCES E/S

5.1 - EXEMPLE 1 (voir annexe 7.9)

Deux Processeurs P1 et P2 demandent le bus I/O au même instant INT,

- P1 pour entrer une info détenue par le coupleur A (opérant en LDC)
- P2 pour envoyer une info au périphérique géré par le coupleur B (opérant en MDC).

L'un et l'autre savent que le coupleur auquel ils s'adressent est prêt pour l'opération à venir, chacun ayant envoyé, en son temps, une interruption NORMALE signifiant :

- pour A : "INFO VALIDE"
- pour B : "REGISTRE DE SORTIE VIDE"

Bien que branché sur un canal LDC, donc moins rapide que le canal MDC auquel est rattaché le coupleur B, le coupleur A sera satisfait en premier.

Cela tient au fait que P₁ étant plus prioritaire que P₂, son PPIO₀₁ fera retomber le PPIO₀₂ et se maintiendra jusqu'à la retombée de TRANSTATE.

C'est donc P₁ qui pendant l'état ADR adressera le registre INPUT DATA du coupleur A, lequel mettra son info sur les 16 lignes IO_{0 - 15} au cours de l'état TRANS qui suit.

P₂ attend donc le prochain état INT pour renouveler son PPIO₀₂ qui cette fois sera maintenu puisque P₀ ne se manifeste pas. Le bus lui appartient dès lors jusqu'à la fin de son échange, c'est-à-dire tant que TRANSTATE = 1.

Sa retombée étant prorogée par la présence de SYNC, P₂ en est quitte pour maintenir sur les lignes IO_{0 - 15} l'info destinée au registre OUTPUT DATA du coupleur B pendant trois périodes d'horloge.

5.2 - EXEMPLE 2 (voir annexe 7.10)

Trois Processeurs P₀, P₁ et P₃ demandent le bus I/O au même instant INT.

- P₀ (de type 65) pour entrer une info détenue par le coupleur A (opérant en NORMAL I/O).
- P₁ (de type 40) pour lire le mot d'état du coupleur B (opérant en MDC).
- P₃ (de type IOP-16) pour envoyer une info au coupleur C (opérant en HDC).

La situation présente signifie que nous avons eu précédemment les opérations suivantes :

- A a fait monter le bit IO_{0 - 15}, d'où I/O = 1 dans le TRAP WORD, pour signaler qu'il détenait une "INFO VALIDE". II a ensuite répondu à une adresse HARD LEVEL POLLING en mettant à 1 le bit du H.L.W. correspondant à son niveau, puis à une adresse NORMAL I/O POLLING, ce qui a permis à P₀ de l'identifier par son sous-niveau et de préparer l'adresse de registre qu'il lui faut mettre sur le bus. Comme il s'agit d'une interruption NORMALE, donc attendue par P₀, celui-ci sait qu'elle signifie "INFO VALIDE" et que le registre du coupleur A qu'il doit adresser est le registre INPUT DATA.

- B, lui aussi, a appelé sur la ligne IO₁₅ mettant à 1 le bit I/O du TRAP WORD. II a également répondu à toutes les adresses HARD LEVEL POLLING qui ont pu se présenter en mettant à 1 le bit du H.L.W. correspondant à son niveau et ne pouvant concerner que P₁. Lorsque ce dernier a envoyé successivement une adresse NORMAL I/O POLLING puis une adresse EXCEPTION LEVEL 0, B a "gardé le silence", attendant l'adresse EXCEPTION LEVEL 1

(par exemple) pour mettre un 1 sur la ligne IO_i correspondant à son sous-niveau *i*. P₁ connaît donc maintenant ce coupleur B qui a envoyé une interruption EXCEPTION et va chercher à savoir quelle en est la cause en adressant son registre INPUT STATUS.

- C, enfin, a envoyé sur une des 8 lignes HDC (IO₀₋₇) une interruption NORMALE que P₃ a décelée dans son TRAP WORD personnel (concordance entre le *n* de la ligne et le contenu de l'un de ses 2 registres CIL), ce qui du même coup l'a identifié. P₃, se référant à son contexte canal (qu'il détient dans ses registres) sait que le coupleur C lui signale qu'il est prêt à recevoir une info à sortir et qu'il doit donc mettre sur le bus le plus rapidement possible l'adresse de son registre OUTPUT DATA.

On constate qu'avant d'en arriver à "se disputer" le bus I/O sur un même instant INT, nos 3 processeurs ont eu à effectuer un travail très différent, celui de P₃ étant de loin le plus rapide. Le fait que P₃ aura le bus en dernier (son PP IO₀₃ étant "avorté" à deux reprises, par le PPIO₀₀ de P₀, puis par le PPIO₀₁ de P₁ comme le montre le diagramme) n'occasionne qu'une faible perte de temps en comparaison de celle subie par P₀ et P₁ qui ont eu à s'octroyer le bus plusieurs fois pour procéder à leurs diverses opérations de POLLING. Et celles auxquelles a dû se livrer P₁ (bien qu'opérant en MDC) ont été les plus laborieuses du fait qu'il s'agissait d'une interruption EXCEPTION.

Il est donc important de se convaincre que le jeu des priorités pour l'obtention du bus I/O à un moment donné ne laisse en rien préjuger de la rapidité des échanges en cours et que réciproquement la hiérarchie de célérité des canaux HDC, MDC et LDC n'impose pas une priorité sur le bus.

6 - CODAGE ET DECODAGE DES ADRESSES

Nous avons vu que les processeurs alertent les coupleurs en leur envoyant, pendant l'état ADR, des adresses codées sur les 16 bits IO_{0-15} .

Ces adresses concernent soit un registre précis dans un coupleur déterminé (système d'entrée-sortie) soit un sous-ensemble de coupleurs dans les opérations de polling (système d'interruption).

Nous nous intéresserons successivement aux adresses de registres, puis aux adresses de polling. Pour tout ce qui suit, il est conseillé d'avoir en permanence sous les yeux l'annexe 7.11, sur les FORMATS D'ADRESSE.

6.1 - ADRESSES DE REGISTRES

Elles se distinguent des adresses de POLLING par le bit $IO_{1=0}$.

Quatre formats sont possibles : le SHORT ON RACK se reconnaît par le bit $IO_{0=0}$ et les trois autres se codent grâce aux bits IO_3 et IO_4 .

Ces adresses ont pour objet d'alerter un des quatre types de registres (d'entrée ou de sortie) présents sur tous coupleurs. Un coupleur étant déterminé (par les bits "CARD NUMBER" ou "ADDRESS" selon le format utilisé) le choix entre les quatre registres se fait par les bits INST (IO_{14}) et OUTPUT (IO_{15}) de la manière suivante :

INST (IO_{14})	OUTPUT (IO_{15})	
0	0	registre DATA INPUT (DI)
0	1	registre DATA OUTPUT (DO)
1	0	registre STATUS INPUT (SI)
1	1	registre COMMAND OUTPUT (CO)

Le format court (SHORT format) fait appel à un troisième bit, REG (IO_{13}), ce qui permet d'adresser jusqu'à 8 registres (2 de chaque type). On peut par exemple avoir 2 registres d'état (STATUS A et STATUS B) comme ci-dessous :

REG (IO_{13})	INST (IO_{14})	OUTPUT (IO_{15})	
0	0	0	registre DATA INPUT
0	1	0	registre STATUS A INPUT
1	1	0	registre STATUS B INPUT
0	0	1	registre DATA OUTPUT
0	1	1	registre COMMAND OUTPUT

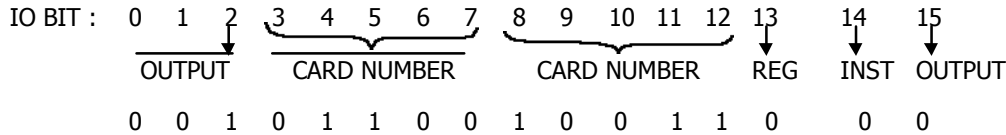
Nous voyons que le format SHORT permet d'adresser jusqu'à 32 coupleurs par rack et le SHORT OFF RACK jusqu'à 8 racks. Pour les coupleurs spéciaux (tels les multiplexeurs ou compteurs)

dont le nombre peut être très important, le format long offre la possibilité de coder jusqu'à 512 groupes de 4 registres (DI, DO, SI, SO) on-rack, et autant off-rack.

Voici deux exemples :

1^{er} exemple :

Un processeur désirant recevoir un caractère détenu par un coupleur on-rack dont le numéro de carte (CARD NUMBER) est 19, envoie sur le bus durant l'état ADR l'adresse suivante :



Le coupleur décodera avec succès cette adresse, validant ainsi une logique qui mettra sur le bus, durant l'état TRANS, le contenu de son registre DATA INPUT.

2^{ème} exemple :

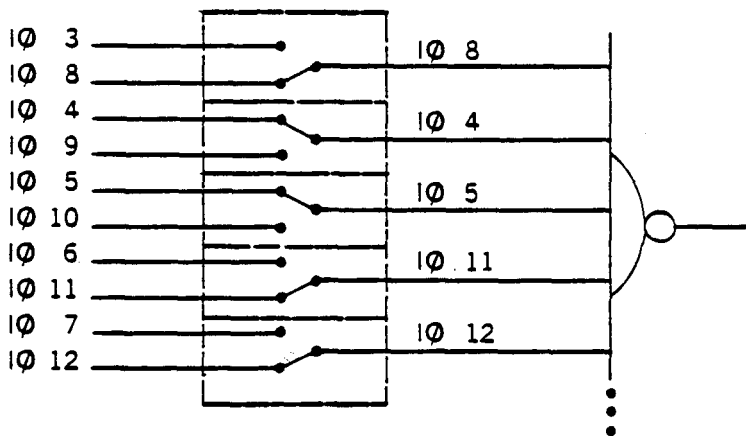
Un processeur désirant envoyer une commande à ce même coupleur 19, mettra sur le bus, durant l'état ADR, l'adresse suivante :



Ici encore, le coupleur décodera avec succès cette adresse, validant ainsi une logique qui permettra la prise en compte par son registre COMMAND OUTPUT des 16 bits envoyés sur le bus par le processeur durant l'état TRANS.

La figure de l'annexe 7.12 illustre le circuit de décodage du coupleur 19 appelé à décodé avec succès les adresses SHORT ON RACK des deux exemples ci-dessus, le 7430 étant validé pendant l'état ADR par le signal ADDRESS = 1, et les trackswitches étant positionnés pour répondre à l'équation logique :

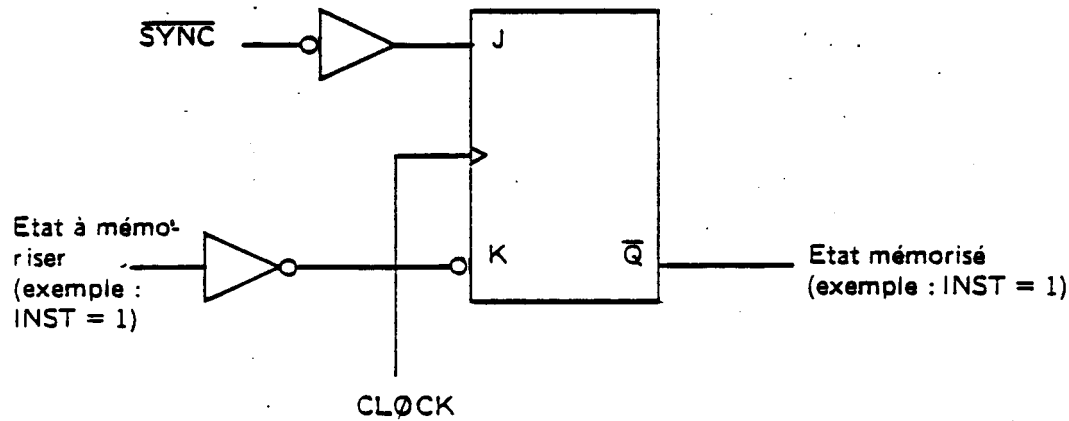
$$\begin{aligned}
 19 &= IO_8 - \overline{IO_9} - \overline{IO_{10}} - IO_{11} - IO_{12} \\
 &= IO_8 - IO_4 - IO_5 - IO_{11} - IO_{12}
 \end{aligned}$$



Grâce à CARD NUMBER, le décodage est extrêmement rapide et deux couches au plus de logique standard sont à traverser pour attaquer les entrées des bascules D.

Aussi est-on assuré qu'au front montant de CLOCK qui fera retomber ADDRESS, elles mémoriseront bien pour une période d'horloge (donc pendant tout l'état TRANS) les valeurs adéquates pour les quatre sorties : INPUT, OUTPUT, INST et REG. Dans le 1^{er} exemple seul INPUT sera à 1 alors que dans le 2^{ème} nous aurons OUTPUT = 1 et INST = 1

Notons que dans le cas présent (coupleur adressable en format SHORT ON RACK) il n'y a pas à envisager de montée de $\overline{\text{SYNC}}$ ce qui n'est pas le cas pour ceux qui sont adressés dans l'un des trois autres formats et pour lesquels les 4 bascules D doivent être remplacées par 4 J.K. ainsi montées :



Ainsi les 4 sorties INPUT, OUTPUT, INST et REG se trouvent-elles figées pendant toute la durée de l'état TRANS.

6.2 - ADRESSES DE POLLING

Elles se distinguent des adresses de registres par le bit $\text{IO}_1 = 1$.

Les formats possibles sont au nombre de 7 (voir annexe 7.11).

Les figures des annexes 7.13 et 7.14 représentent des exemples de décodage d'adresses de POLLING. Durant l'état ADR le signal ADDRESS = 1 valide tous les circuits de décodage. Les coupleurs qui décodent avec succès une adresse de polling mémorisent son type grâce à des bascules J.K. sur le CLOCK qui fait retomber le signal ADDRESS et envoient la réponse appropriée durant l'état TRANS.

Pour un coupleur donné une adresse de polling ne peut être que de 3 types :

- Type HARD LEVEL (reconnue par tous les coupleurs) donnant naissance au signal HIP (Hard Interrupt Polling).
- Type NORMAL (I/O ou MDC ou LDC) donnant naissance au signal NIP (Normal Interrupt Polling).
- Type EXCEPTION (LEVEL 0, 1 ou 2) donnant naissance au signal EIP (Exception Interrupt Polling).

C'est, à chaque polling, un de ces 3 signaux qui validera pendant TRANS la réponse appropriée (0 ou 1) sur une des 16 lignes IO_{0-15} .

Mais il se peut qu'un Rack Extender ait à collecter des réponses de polling de coupleurs off-rack. Il fait alors monter le signal $\overline{\text{SYNC}}$, ce qui a pour effet d'une part de prolonger l'état TRANS et d'autre part d'obliger les coupleurs à répéter leur réponse à chaque montée de CLOCK tant que $\text{TRANSTATE} = 1$. On voit en effet que sur les bascules J.K. le fait que $\overline{\text{SYNC}} = 1$ inhibe la remise à zéro des sorties $\overline{\text{Q}}$. Nous aurons donc un des 3 signaux NIP, EIP ou HIP qui sera maintenu à 1 jusqu'à la montée de CLOCK faisant suite à la retombée de $\overline{\text{SYNC}}$.

La figure de l'annexe 7.13 représente le circuit d'un coupleur STANDARD de niveau 9 qui répond aux adresses NORMAL I/O POLLING et EXCEPTION LEVEL 0 codant sur les bits IO_{12} à IO_{15} un H.L. BIT N0 égal à 9 et sur les bits IO_4 à IO_7 son complément H.L. BIT N0.

La figure de l'annexe 7.14 représente le circuit de décodage d'un coupleur LDC rattaché au processeur P_1 sur GROUPE 2.

Nous allons suivre en détail les opérations d'interruption NORMALE et d'interruption EXCEPTION en nous référant également à l'annexe 7.11 pour le codage des adresses de polling et à l'annexe 7.15 pour l'arbre de polling.

Lorsqu'il est prêt à effectuer un échange de caractère, notre coupleur LDC envoie une interruption sur le bit IO_{12} pendant l'état INT. Le processeur P_1 qui "surveille" ce bit IO_{12} en venant périodiquement regarder son TRAP WORD personnel, va devoir identifier le coupleur.

Pour ce faire, il enverra, lorsqu'il aura obtenu le bus et pendant l'état ADR, une adresse NORMAL LDC POLLING avec LEVEL NO = 0 (GROUPE 0) et PROC. NO = 1. Le switch A du coupleur étant positionné pour décoder PROC. NO = 1 et LEVEL NO = 2 il n'y aura pas de montée du signal NIP et par suite pas de réponse durant l'état TRANS qui suit.

Le processeur P_1 devra redemander le bus pour envoyer cette fois une adresse NORMAL LDC POLLING avec LEVEL NO = 1 (GROUPE 1). Toujours pas de réponse. Il recommence l'opération avec maintenant LEVEL NO = 2 (GROUPE 2), ce qui permet au switch A de décoder l'adresse avec succès et de faire monter NIP, lequel restera haut pendant toute la durée de l'état TRANS, validant l'envoi d'un 1 sur la ligne IO_{0-15} appropriée (IO_3 dans notre exemple).

Le coupleur étant alors parfaitement identifié par cette réponse sur IO_3 , le processeur P_1 fait une transposition du contexte canal de notre coupleur et redemande le bus pour envoyer enfin une adresse de registre (n'oublions pas qu'il s'agissait d'une interruption NORMALE) et procéder à l'échange du caractère.

Voyons maintenant comment se déroule une interruption EXCEPTION.

Notre coupleur fait monter IO_{15} durant l'état INT, ce qui provoquera l'envoi d'une adresse HARD LEVEL POLLING. Tous les coupleurs décodent cette adresse et feront monter HIP qui validera une réponse (0 ou 1), pendant l'état TRANS suivant, sur la ligne IO_{0-15} appropriée, soit IO_{11} dans notre exemple.

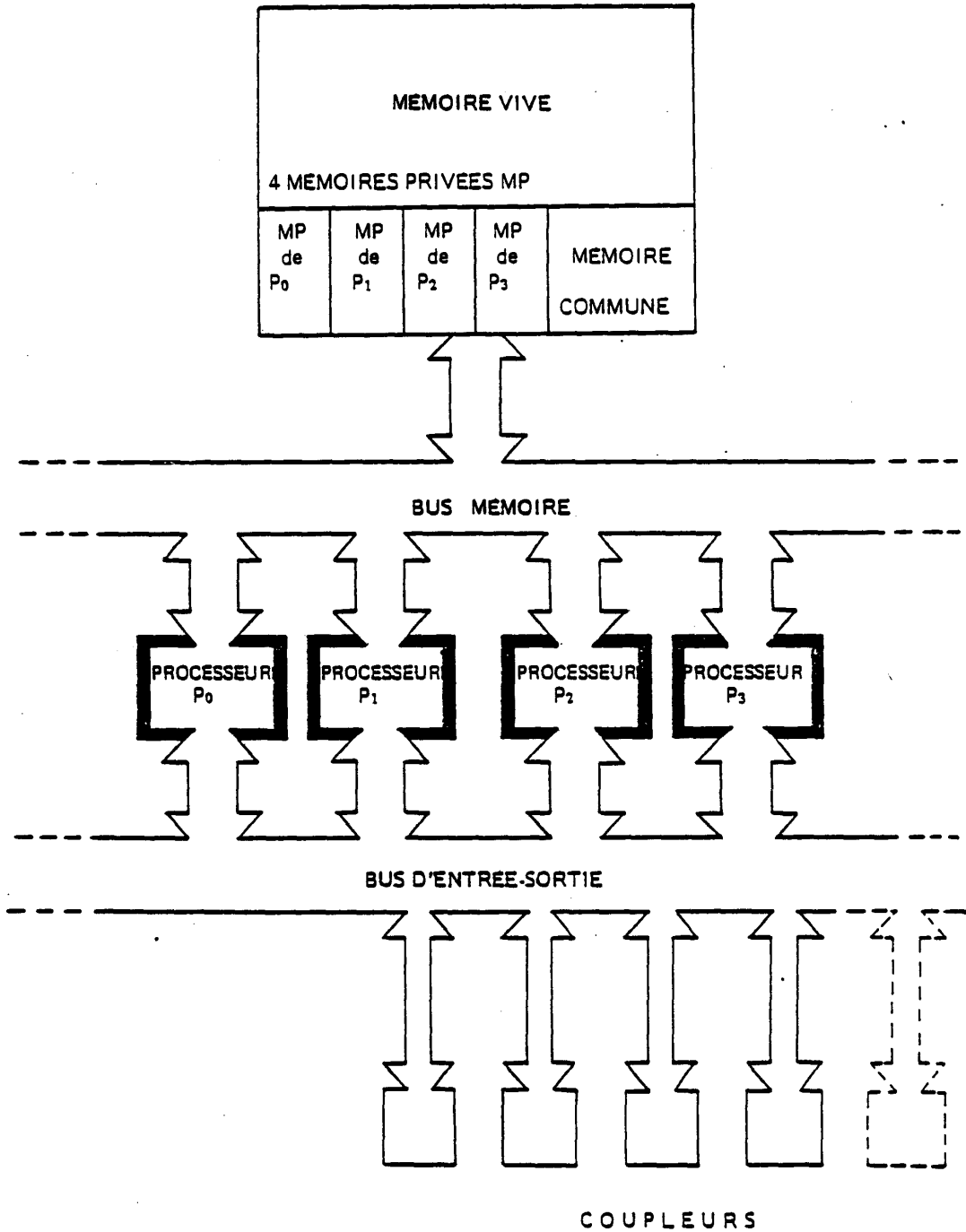
Le processeur P_1 , trouvant à 1 le bit IO_{11} du H.L.W. enverra une adresse NORMAL I/O POLLING codant 11 dans le H.L. BIT NO. Le switch B est bien positionné pour décoder 11, mais le switch C ne laissant passer que IO_{10} (EXL 1) nous n'aurons pas de EIP et donc pas de réponse à cette première adresse de polling, pas plus qu'à la suivante qui codera $IO_9 = 1$ à la place de $IO_8 = 1$. Et ce n'est qu'à la troisième adresse de polling, qui, elle, aura le bit IO_{10} à 1, que notre coupleur fera monter EIP, validant ainsi un 1 sur la ligne IO_3 du bus pendant tout l'état TRANS. Le processeur P_1 , connaissant maintenant le coupleur responsable de l'interruption EXCEPTION qui nous préoccupe, redemande le bus pour envoyer cette fois l'adresse de son registre INPUT STATUS. Ceci aura pour effet non seulement de mettre sur les lignes IO_{0-15} le contenu dudit registre pendant l'état TRANS qui suit, mais également de faire en sorte que le coupleur ne réponde plus par 1 sur IO_{11} aux adresses HARD LEVEL POLLING, pas plus que sur IO_3 aux adresses EXCEPTION LEVEL 1 (EXL1) qu'il pourrait être amené à décoder.

Il est à remarquer qu'un coupleur qui répond sur la ligne IO_i à une adresse de polling consécutive à une interruption NORMALE, répond sur cette même ligne IO_i (dans notre exemple IO_3) à une adresse de polling consécutive à une interruption EXCEPTION.

ANNEXES

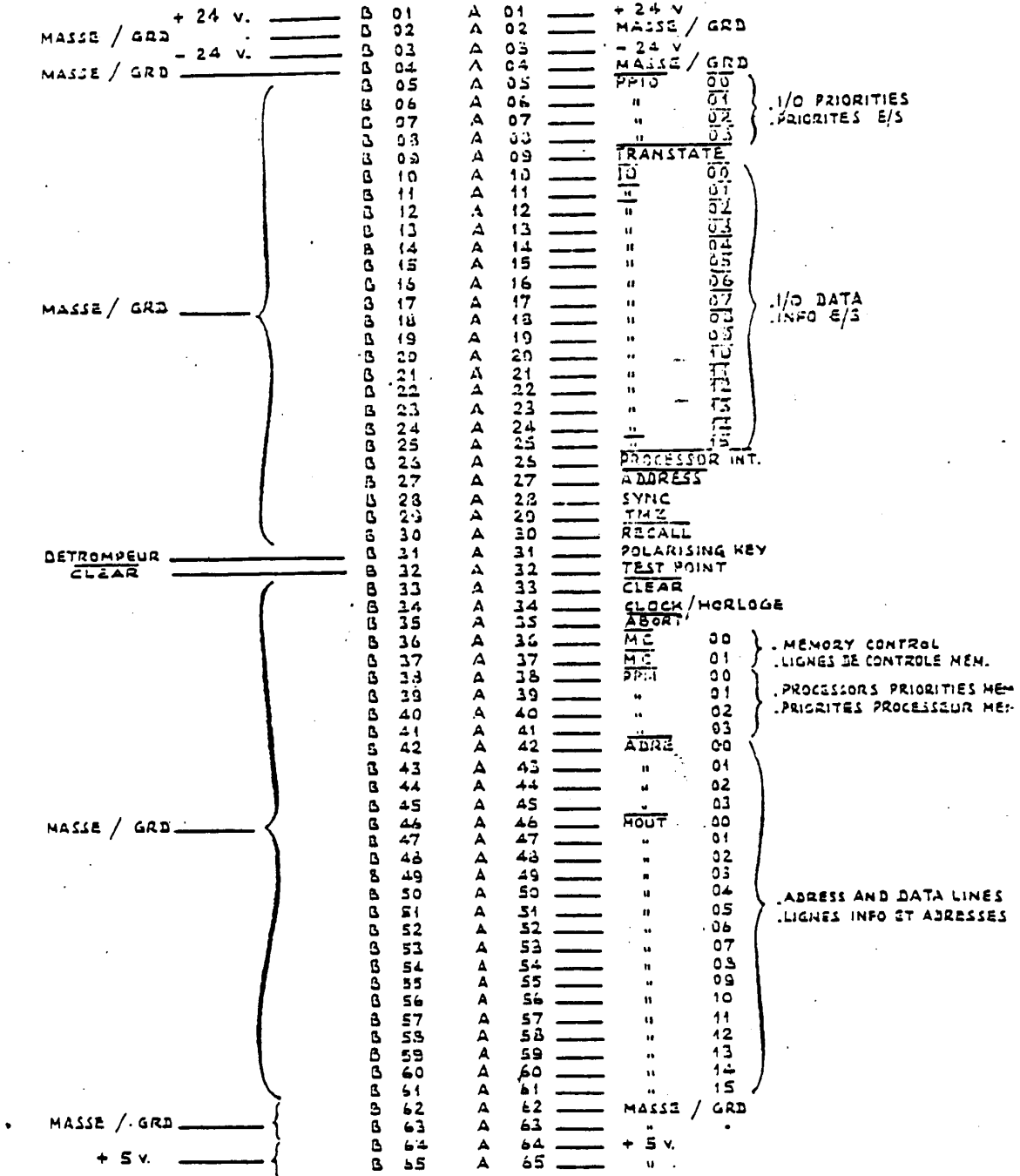
ANNEXE 7.1

ORGANISATION GENERALE



ANNEXE 7.2

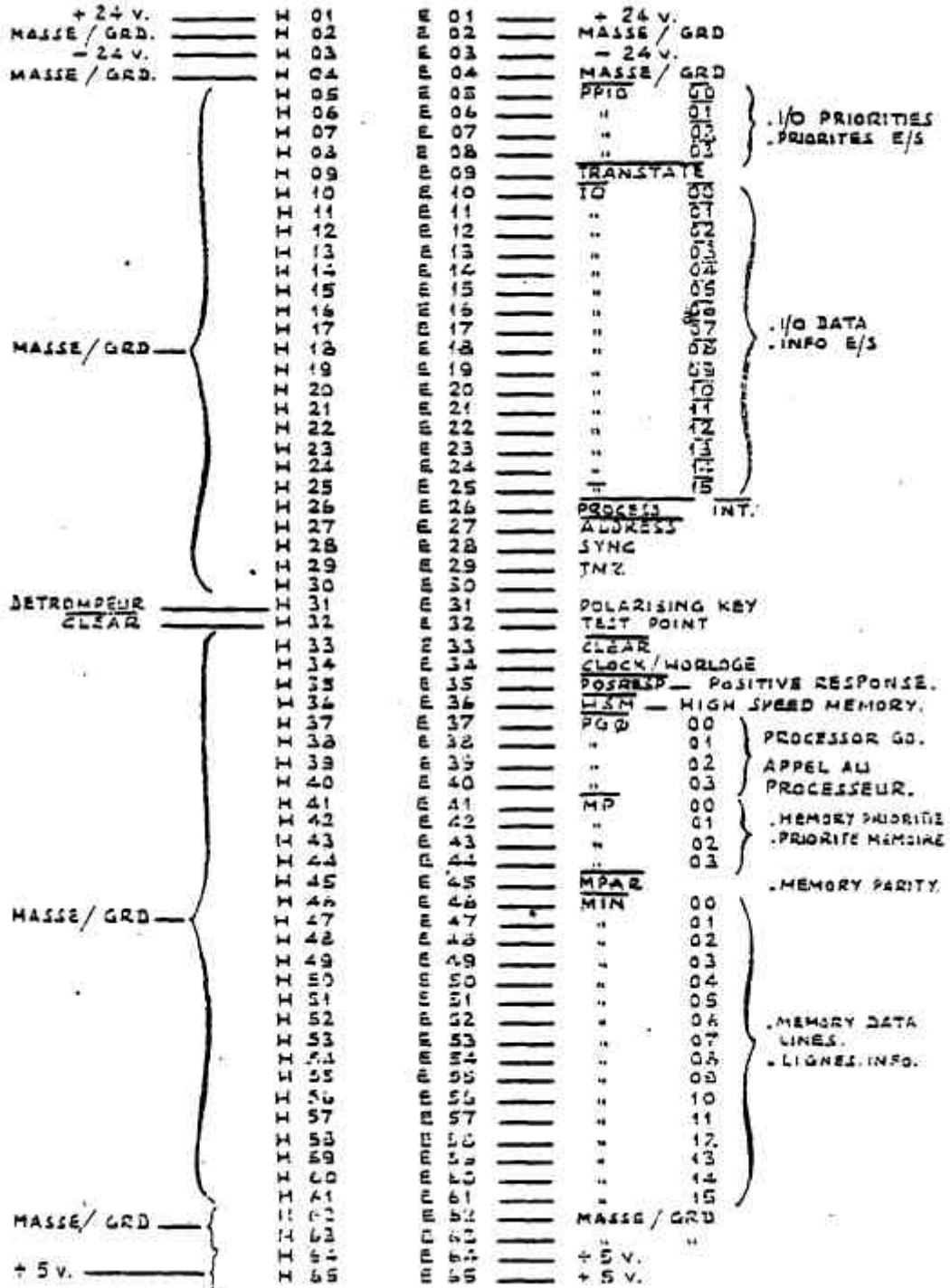
LIGNES DU FOND DE BAC (PARTIE HAUTE)



T.S.V.P.

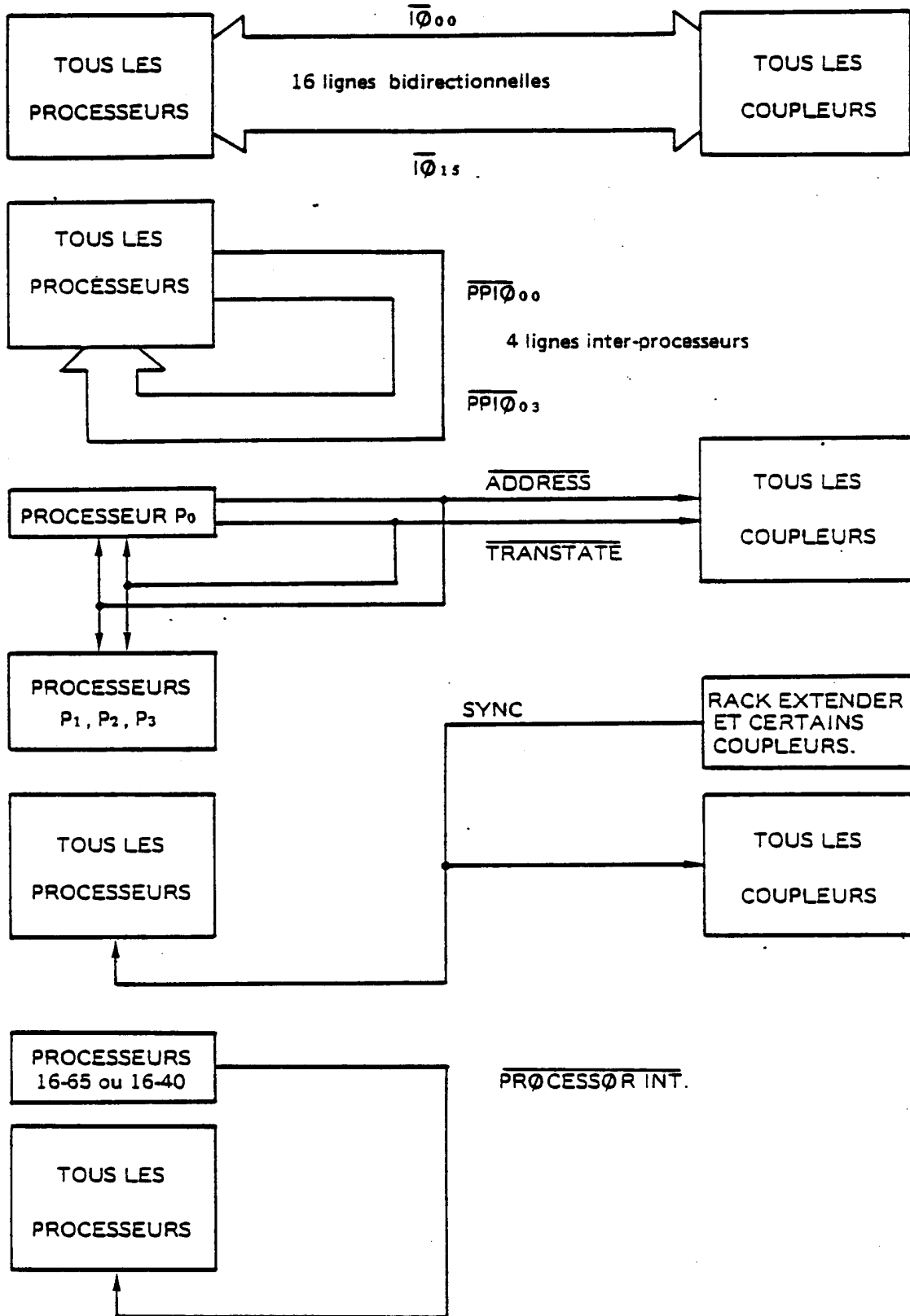
ANNEXE 7.2 (suite)

LIGNES DU FOND DE BAC (PARTIE BASSE)



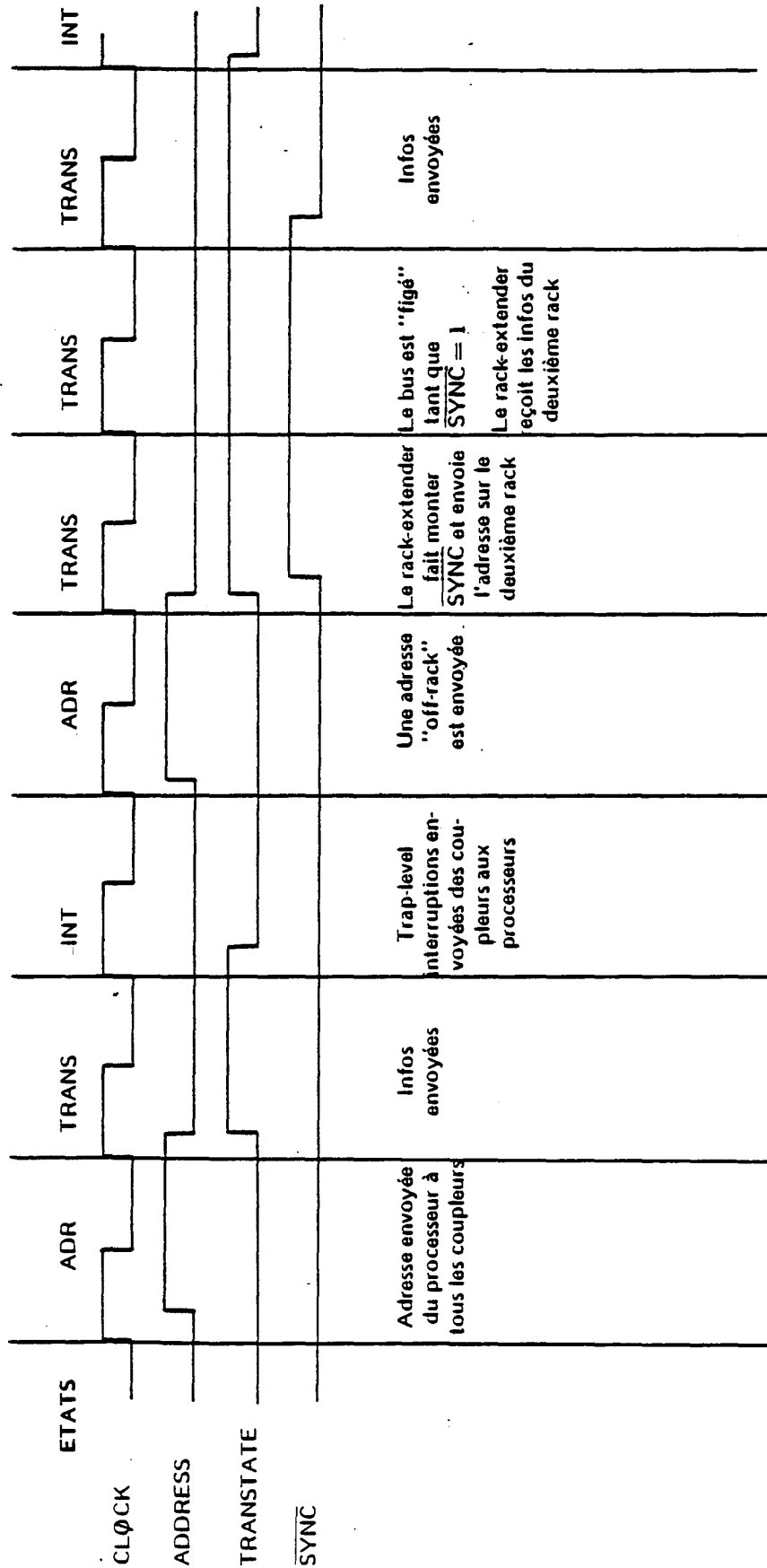
ANNEXE 7.3

BUS I/O – UTILISATION DES LIGNES

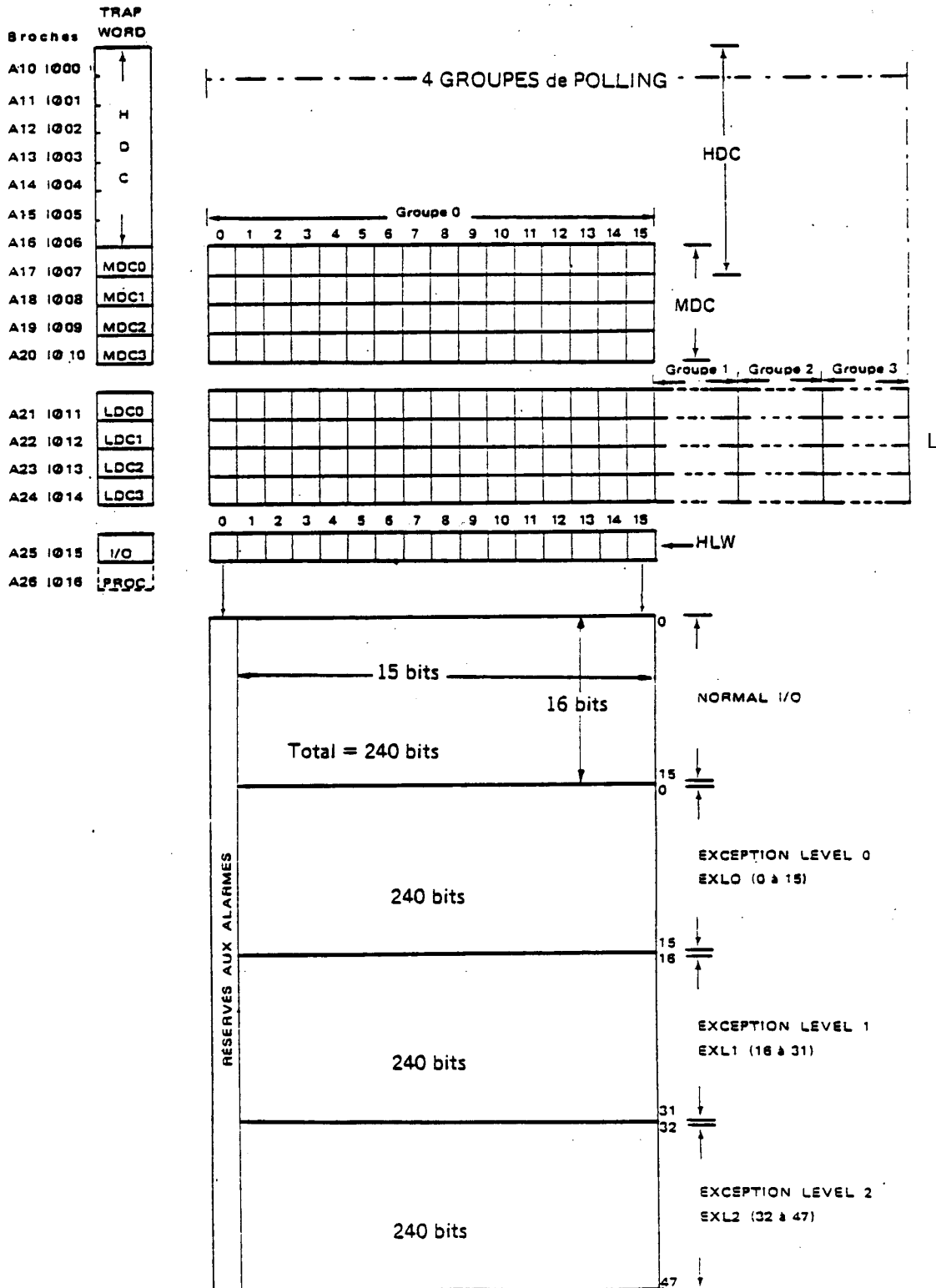


ANNEXE 7.4

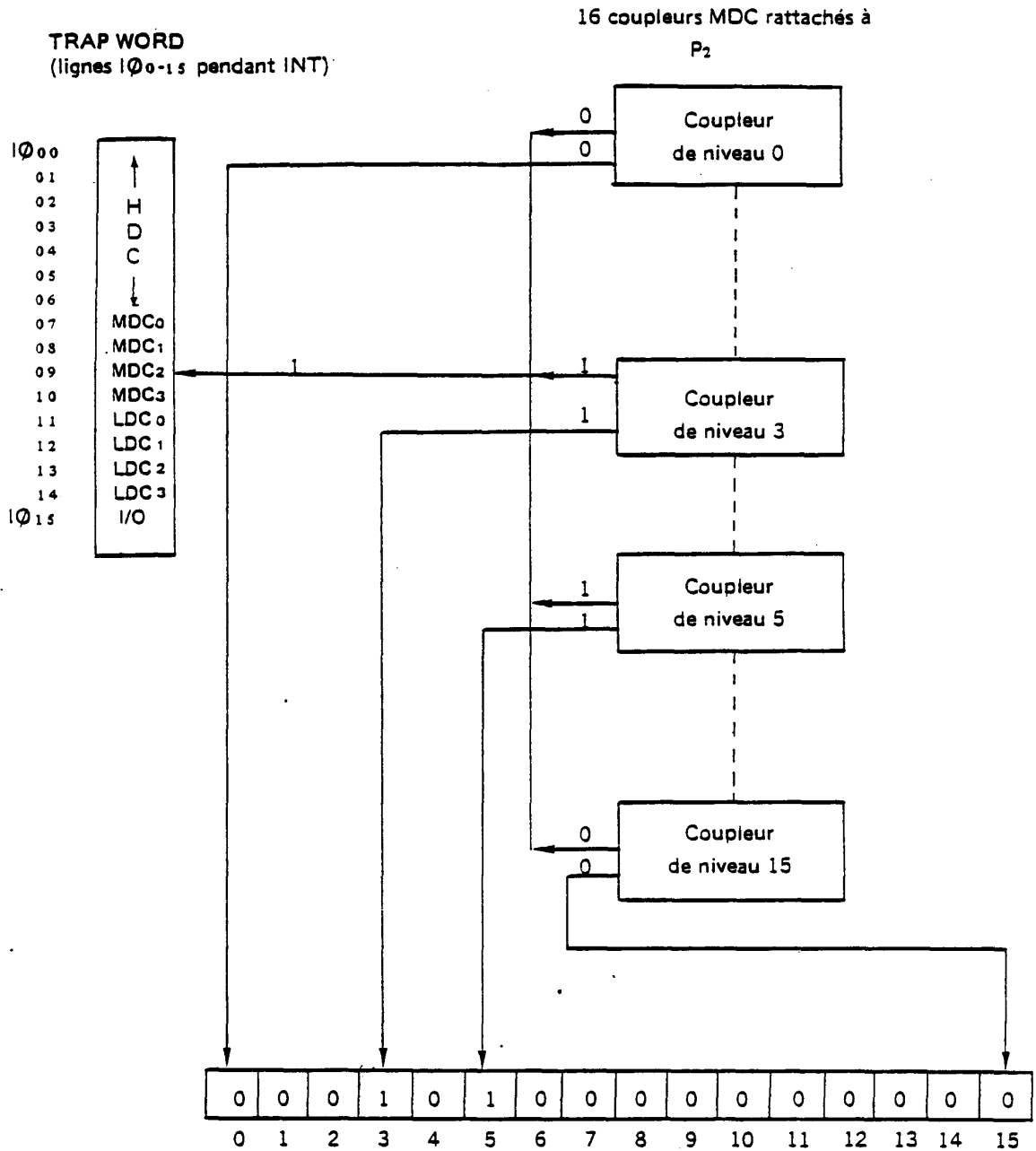
TROIS ETATS DE BUS I/O



ANNEXE 7.5
ARBRE DE POLLING



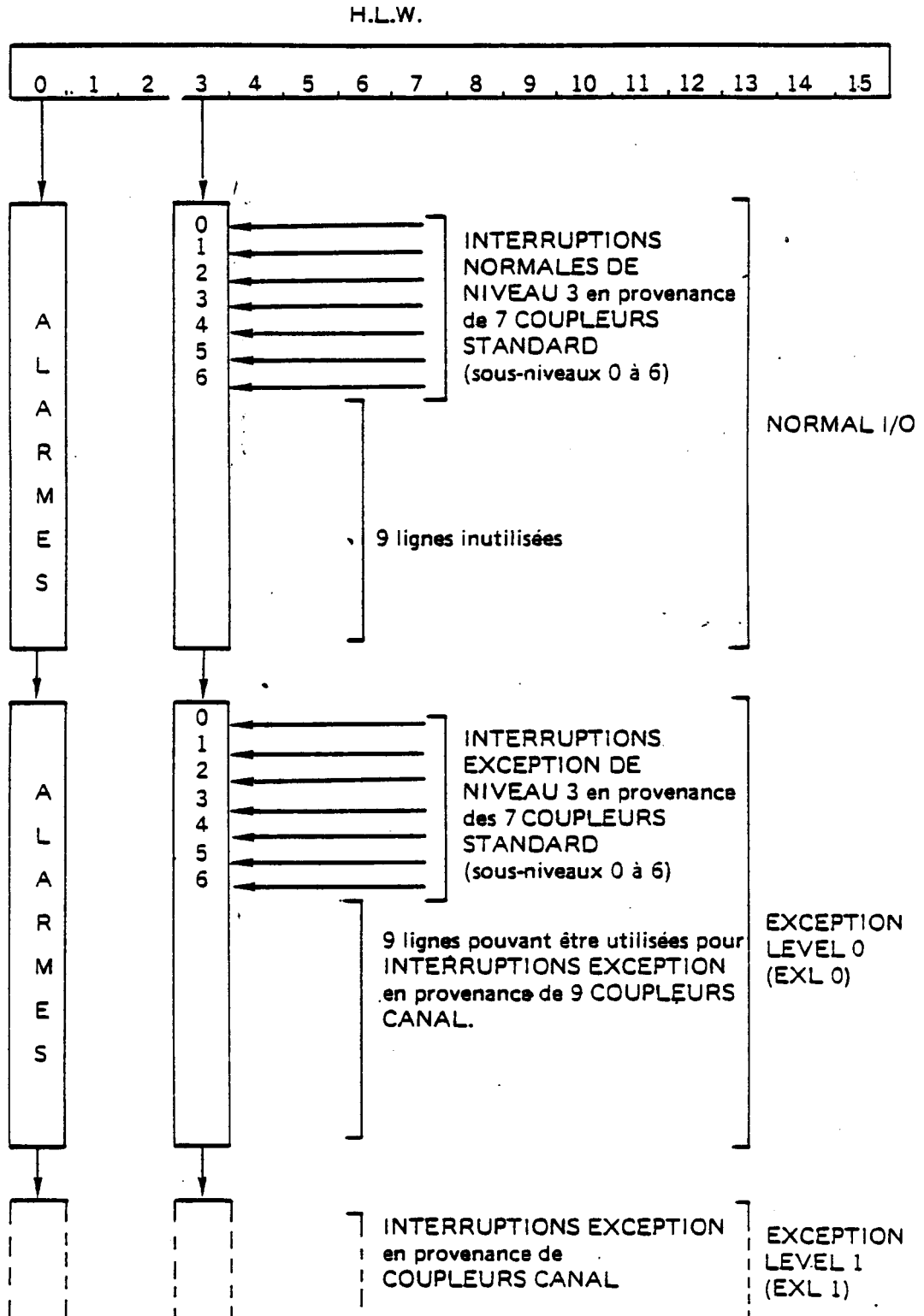
INTERRUPTIONS CANAL



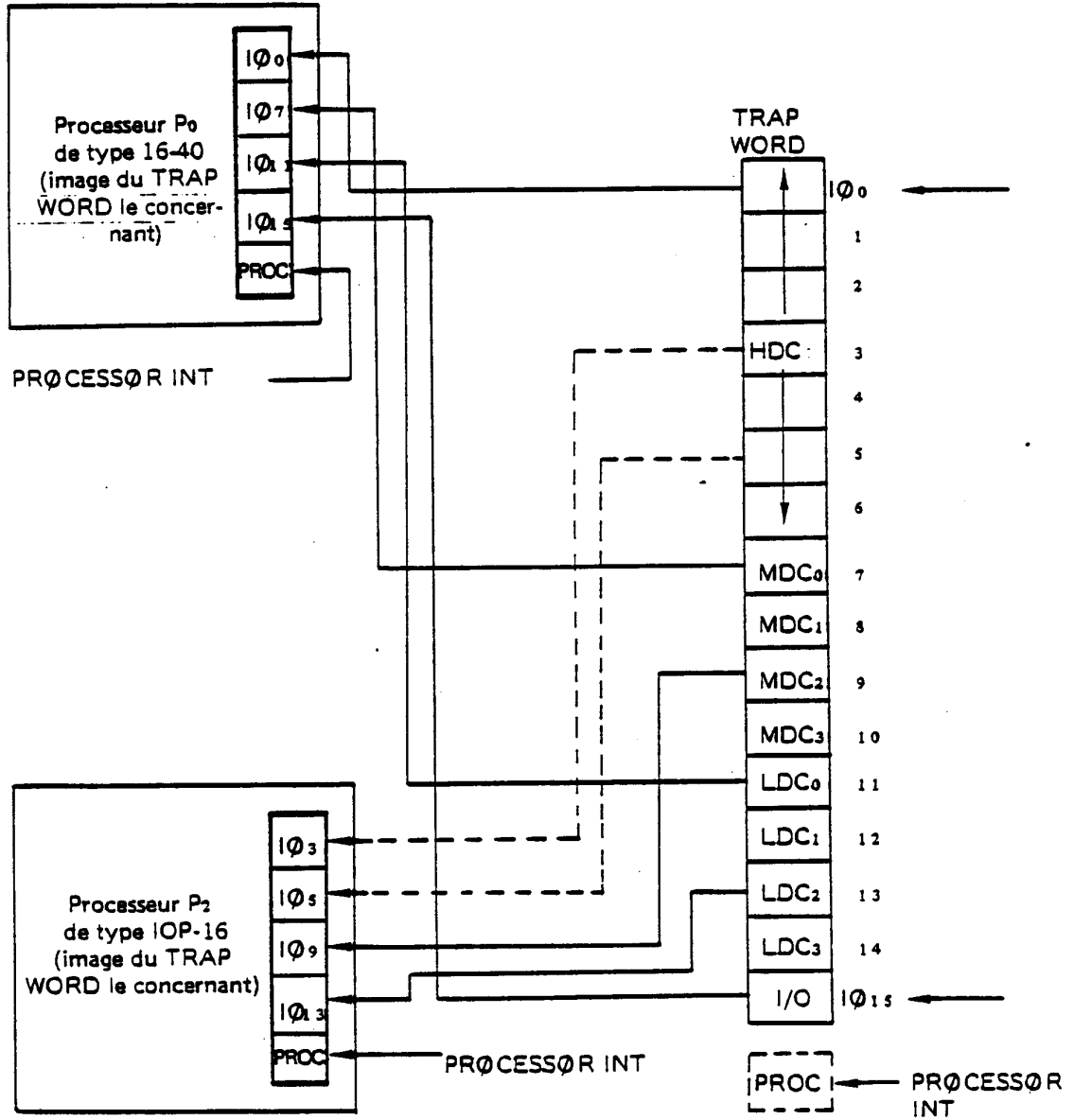
Mot de réponse à un NORMAL MDC POLLING
(lignes I Φ _{0 - 15} pendant TRANS)
Les coupleurs 3 et 5 envoient une interruption.

ANNEXE 7.7

H.L.W. (HARD LEVEL WORD)



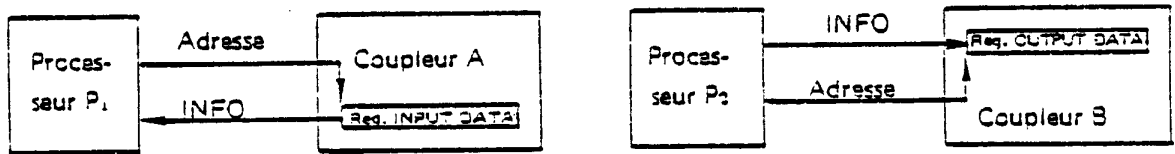
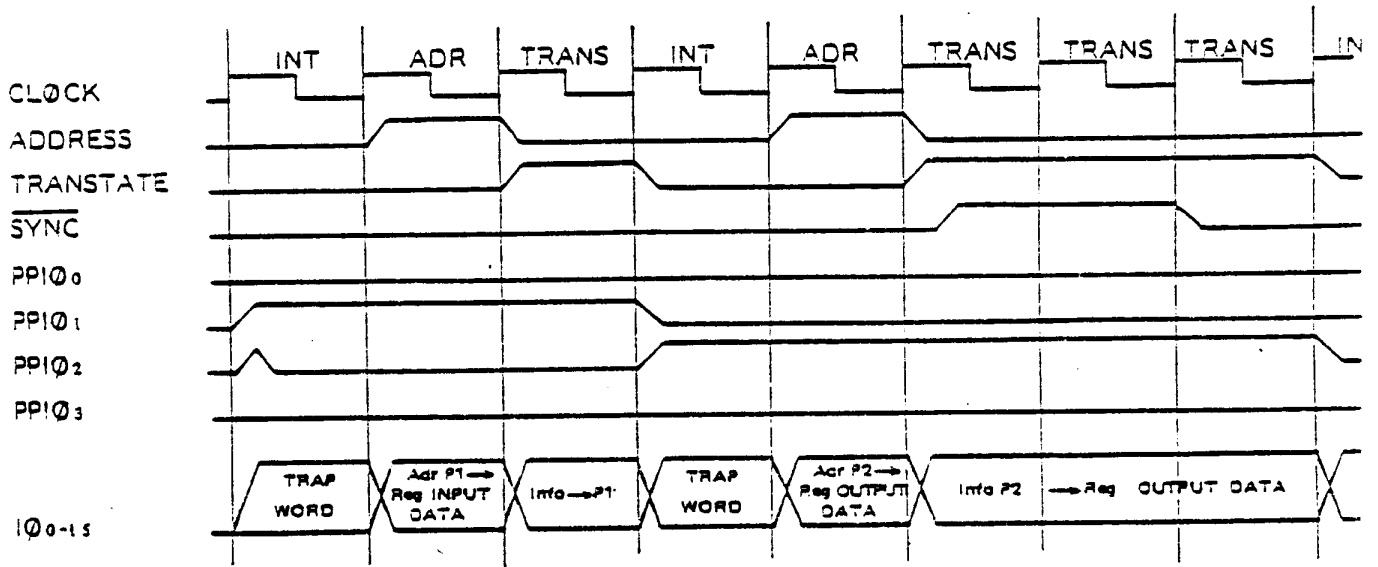
ANNEXE 7.8
TRAP WORD



Le processeur P₂ (de type IOP-16) a été programmé ici pour "surveiller" dans le TRAP WORD les interruptions NORMALES des coupleurs CANAL branchés sur les lignes (HDC) IO₃ et IO₅, et ce, à chaque état INT.

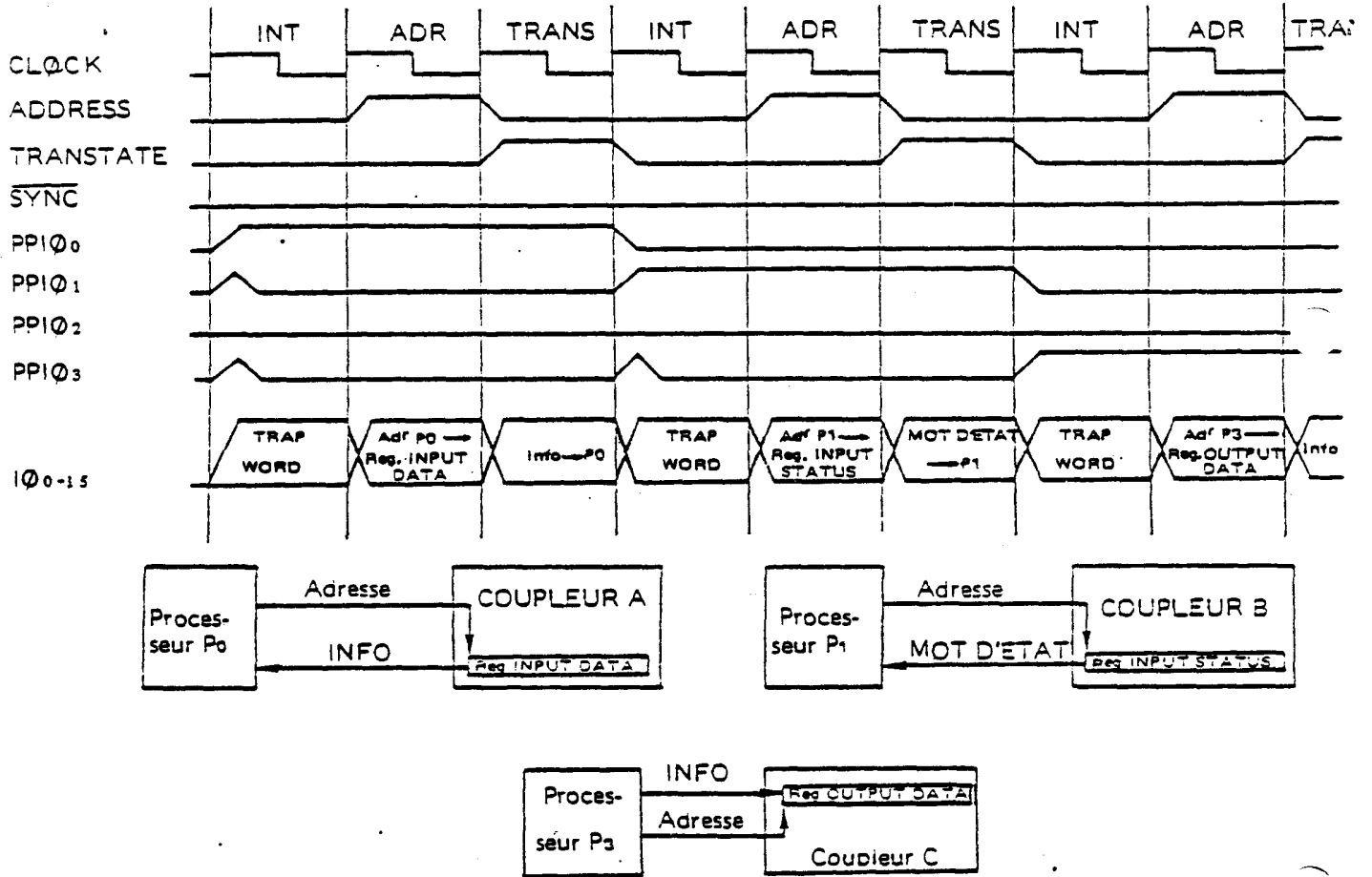
ANNEXE 7.9 SEQUENCE D'ENTREE-SORTIE

EXEMPLE 1.



ANNEXE 7.10
SEQUENCE D'ENTREE-SORTIE

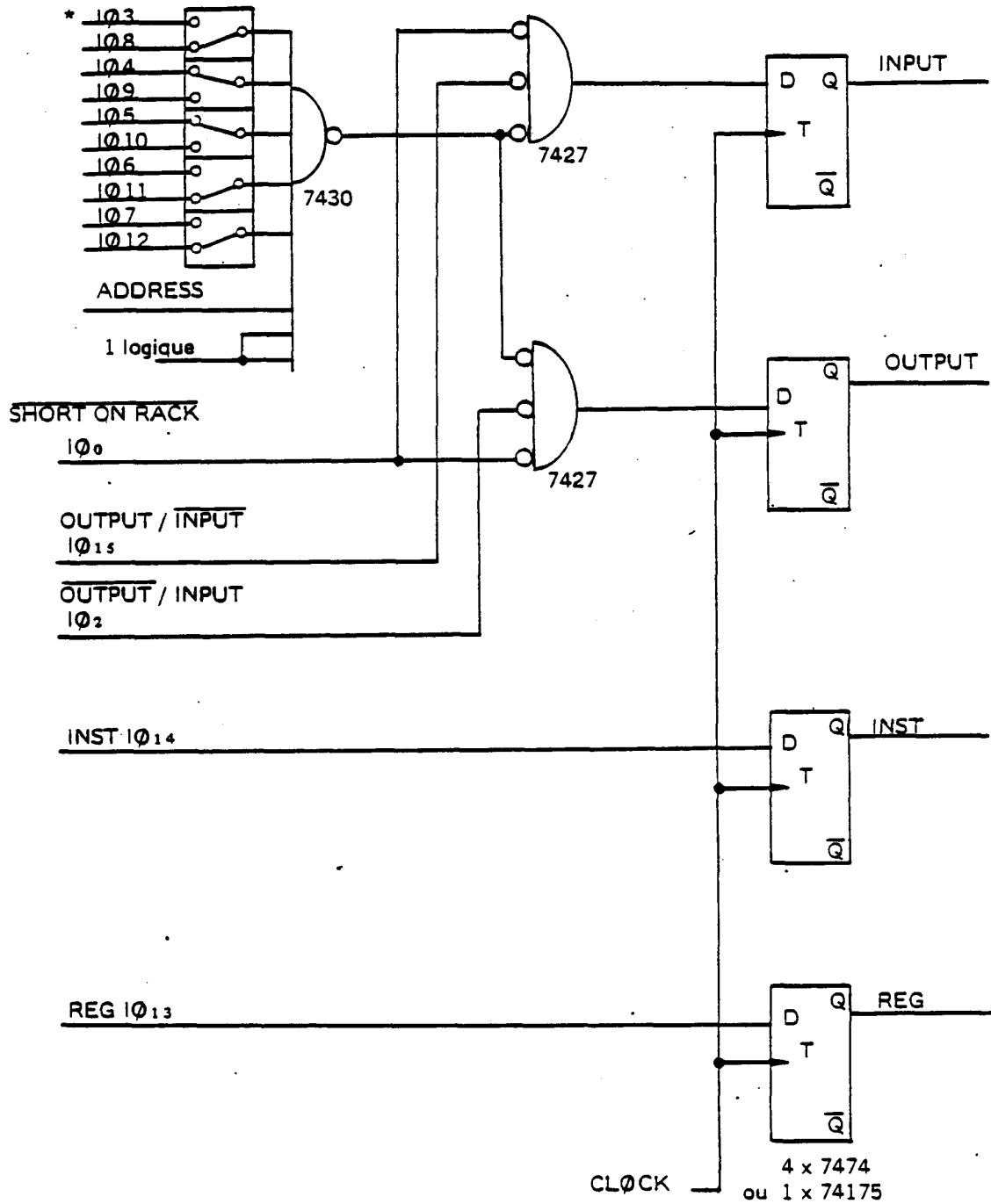
EXEMPLE 2.



ANNEXE 7.11
ADRESSES

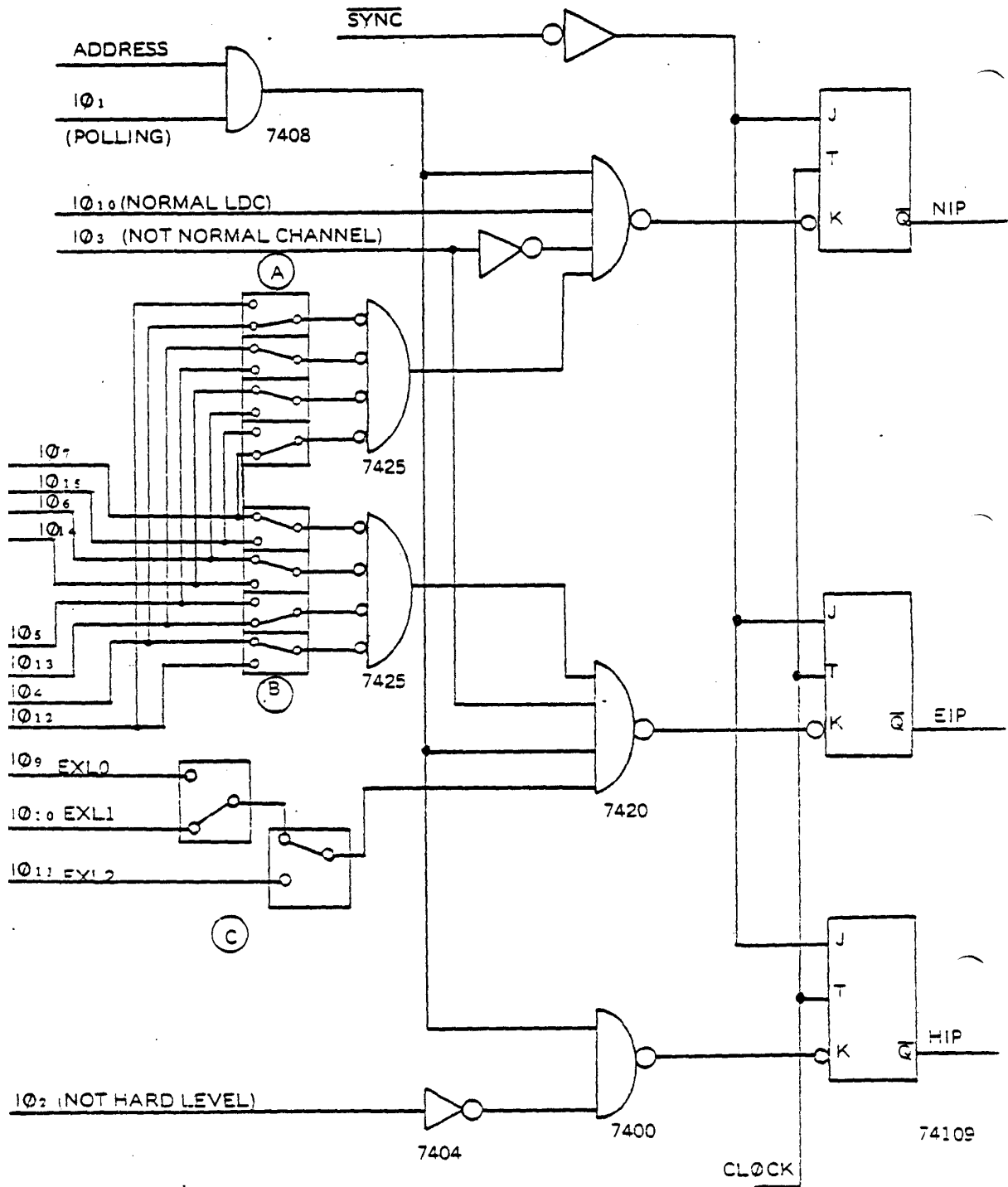
ADRESSES DE REGISTRES		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
SHORT ON RACK		0	0	Out-put	CARD NUMBER				CARD NUMBER				REG	INST	Out-put			
OFF RACK		1	0	0	0	1	RACK NO			CARD NUMBER				REG	INST	Out-put		
LONG ON RACK		1	0	0	1	0	ADDRESS								INST	Out-put		
OFF RACK		1	0	0	1	1	ADDRESS								INST	Out-put		
ADRESSES DE POLLING																		
HARD LEVEL POLLING		1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
NORMAL I/O POLLING		1	1	1	1	H.L. BIT NO				1	0	0	0	H.L. BIT NO				
EXCEPTION LEVEL 0 (EXL 0)		1	1	1	1	H.L. BIT NO				0	1	0	0	H.L. BIT NO				
EXCEPTION LEVEL 1 (EXL 1)		1	1	1	1	H.L. BIT NO				0	0	1	0	H.L. BIT NO				
EXCEPTION LEVEL 2 (EXL 2)		1	1	1	1	H.L. BIT NO				0	0	0	1	H.L. BIT NO				
NORMAL LDC POLLING		1	1	1	0	LEVEL NO	PROC. NO		0	0	1	0	LEVEL NO	PROC. NO				
NORMAL MDC POLLING		1	1	1	0	1	1	PROC. NO		0	0	0	1	0	0	PROC. NO		
NOT SHORT ON RACK POLLING																		
NOT HARD LEVEL NOT NORMAL CHANNEL																		
NORMAL I/O EXL 0 EXL 1 ou NORMAL LDC EXL 2 ou NORMAL MDC																		

ANNEXE 7.12
DECODAGE ADRESSE REGISTRE



* CARD NUMBER : IO 8-12
 CARD NUMBER : IO 3-7
 EXEMPLE DE DECODAGE ADRESSE
 "SHORT ON RACK".

ANNEXE 7.14
POLLING COUPLEUR CANAL

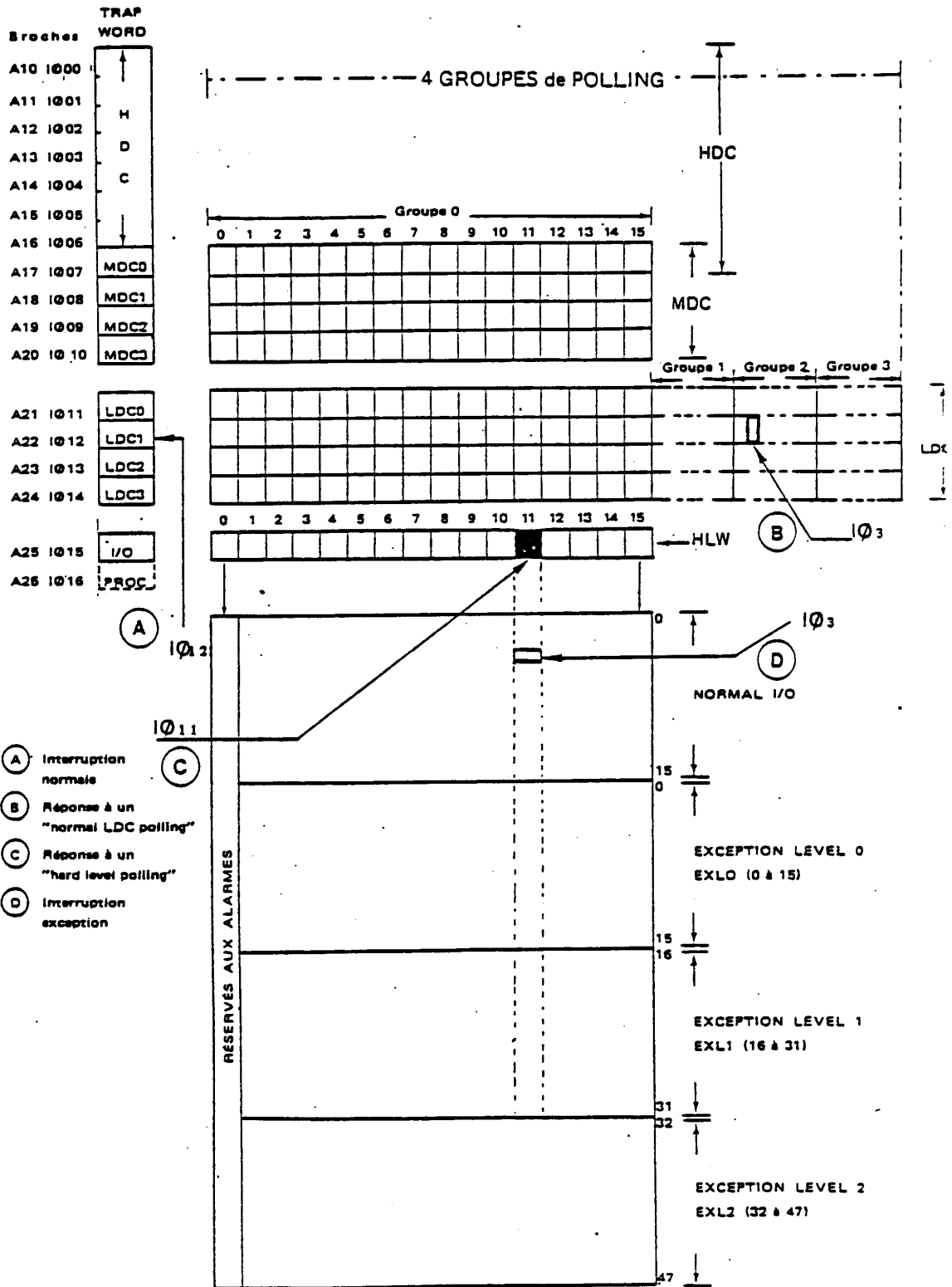


EXEMPLE DE
DECODAGE
D'ADRESSE DE POLLING
POUR UNE CARTE
TRAVAILLANT EN CANAL.

ANNEXE 7.15

INTERRUPTIONS ET POLLING

ARBRE DE POLLING



ANNEXE 7.16

LEXIQUE DES TERMES ET ABREGES ANGLAIS

<p style="text-align: center;">A</p> <p>ADDRESS : adresse</p> <p>ACK : abréviation de ACKNOWLEDGE signifiant reconnaissance. Instruction qui permet de reconnaître les sous-niveaux d'interruption (voir Manuel de Référence).</p> <p>ACQ : abréviation de ACQUIT signifiant acquittement. Instruction qui permet d'acquitter une interruption (voir Manuel de Référence).</p>	<p style="text-align: center;">H</p> <p>HDC: abréviation de Highspeed Data Channel. Canal à haute rapidité.</p> <p>HIP : abréviation de Hard level Interrupt Polling. Signal validant une réponse à une adresse HARD LEVEL POLLING.</p> <p>H.L. BIT N0: numéro de.bit du H.L.W.</p> <p>H.L.W. : abréviation de Hard Level Word.</p> <p>HV : Hardware Vector. Registre vecteur des tâches hardware en cours de traitement.</p>
<p style="text-align: center;">B</p> <p>BIT : élément d'information binaire</p> <p>BUS : groupe de lignes distribuant des signaux électriques à un ensemble de cartes.</p>	<p style="text-align: center;">I</p> <p>IM : Interrupt Mask. Masque des interruptions.</p> <p>INITIALIZE : initialiser.</p> <p>INPUT :entrée.</p> <p>INST :abréviation pour instruction.</p> <p>INTERRUPT : interruption (en abr. INT)</p> <p>I/O : abréviation pour Input/Output, comme E/S pour Entrée/Sortie.</p> <p>IOP : Input Output Processor. Processeur d'entrée-sortie.</p> <p>IPI : Inter Processor Interrupt. Interruption inter-processeurs. Instruction servant en particulier à l'initialisation des échanges CANAL (voir Manuel de Référence).</p>
<p style="text-align: center;">C</p> <p>CARD : carte</p> <p>CARD NUMBER : numéro de carte</p> <p>CHANNEL : canal</p> <p>CIL : abréviation de Channel Interrupt Line. Registre se trouvant sur les processeurs CPU 65 et IOP 16 pour y coder le numéro de la LIGNE CANAL dont on désire voir traiter les INTERRUPTIONS.</p> <p>CLEAR : remise à zéro</p> <p>CLOCK : horloge</p> <p>COMMAND : commande</p> <p>CPU : Control Processor Unit c'est-à-dire unité de traitement.</p>	<p style="text-align: center;">L</p> <p>LDC : abréviation de Lowspeed Data Channel. Canal le moins rapide.</p> <p>LEVEL : niveau (SUBLEVEL : sous-niveau)</p> <p>LEVEL N0 : numéro de niveau.</p> <p>LINE: ligne.</p>
<p style="text-align: center;">D</p> <p>DATA : information</p> <p>DATA CHANNEL ou DC: canal</p> <p>DISCOVER BIT recherche du 1^{er} bit à gauche.</p>	<p style="text-align: center;">M</p> <p>MDC : abréviation de Médiumspeed Data Channel. Canal de rapidité intermédiaire.</p>
<p style="text-align: center;">E</p> <p>EIP : abréviation de Exception Interrupt Polling. Signal validant une réponse à une adresse EXCEPTION LEVEL 0, 1 ou 2.</p>	<p style="text-align: center;">N</p> <p>NIP : abréviation de Normal Interrupt Polling. Signal validant une réponse à une adresse NORMAL I/O (ou LDC ou MDC) POLLING.</p>

ANNEXE 7.16 (suite)
LEXIQUE DES TERMES ET ABREGES ANGLAIS

<p style="text-align: center;">O</p> <p>OFF-RACK: dans un bac différent de celui des processeurs. ON-RACK : dans le même bac que les processeurs. OUTPUT : sortie.</p>	<p style="text-align: center;">S</p> <p>SHORT : court. SPEED : vitesse. STATE . état. STATUS : situation, c'est-à-dire MOT D'ETAT. SWITCH : commutateur.</p>
<p style="text-align: center;">P</p> <p>POLLING : recherche des coupleurs appelant et élection des plus prioritaires (to poll : élire] PRIORITY : priorité.</p>	<p style="text-align: center;">T</p> <p>TRAP WORD : mot que les processeurs lisent à chaque "trappe" de leur micro-programme. II tient à jour les demandes d'interruptions à chaque état INT en donnant l'image des lignes IO. TRACKSWITCH : commutateur d'aiguillage.</p>
<p style="text-align: center;">R</p> <p>RACK : bac . RACK EXTENDER : carte du bac principal servant d'interface entre celui-ci et les bacs d'extension.</p>	