Réseaux
et systèmes
d'information

Bull

MANUEL DE FONCTIONNEMENT

SOLAR

BUS MÉMOIRE

Description du transfert des données

Bull

1 160 902 00 022

Réseaux
et systèmes
d'information

Bull

BUS MÉMOIRE

MANUEL DE FONCTIONNEMENT

Δ en haut de page indique le changement complet de la page par rapport à l'IE précédent

I en marge indique la partie modifiée par rapport à l'IE précédent

DÉCEMBRE 1980

1 160 902 00/__ 22 03

Réseaux et systèmes d'information

Bull

AVERTISSEMENT

Ce document ne concerne que les SOLAR 16-40 et 16-65 à l'exclusion du SOLAR 16-05 qui, lui, n'utilise pas le bus mémoire.

Le terme PROCESSEUR (Pi) représentera donc indifféremment un CPU40 ou un CPU65 (Central Processing Unit), voire même un IOP-M (Input Output Processor).

Par ailleurs, il faut entendre par STACK MEMOIRE un des modules suivants : SCMOB, SCM16, SCM32 (Semi-Conductor Memory) ou FCM16 (Ferrit Core Memory) et par STACK HSM (High Speed Memory) le module SCS04.

Le fonctionnement du bus mémoire sera décrit dans son utilisation la plus large, à savoir celle où il assure le dialogue et le transfert de données (Processeur-Mémoire et/ou Mémoire-Processeur) entre 4 Processeurs (P_0 à P_3) présents dans le même rack et un nombre quelconque de stacks mémoire jusqu'à la capacité maximale de 1 méga-mot de 16 bits plus parité.

Réseaux
et systèmes
d'information

Bull

LE BUS MEMOIRE

LE BUS MEMOIRE

I. GENERALITES

I.1 Le bus mémoire assure le transfert de tous les signaux nécessaires au dialogue et aux échanges entre les stocks mémoire et les processeurs, et ce, entre ces deux types de modules exclusivement.

I.2 Il est divisé en deux demi-bus selon que les signaux sont émis par un processeur (groupe processeur) ou par une carte mémoire (groupe mémoire). Voir Doc. 11 et 12.

I.3 Les lignes portent le nom des signaux* qu'elles véhiculent. Celles du groupe processeur sont situées dans la partie haute du fond de panier et celles du groupe mémoire dans la partie basse.

I.4 Tous les signaux circulant sur le bus mémoire sont transmis sous forme complémentée et synchronisés sur une horloge symétrique (T = 125 ns). Celle-ci, CLOCK, est commune à toutes les cartes présentes dans le bac.

* A noter que tous les noms de signaux du bus mémoire sont précédés du signe # signifient qu'ils sortent ou entrent sur les connecteurs arrières des cartes, c'est-à-dire côté fond de panier.

II. DEFINITIONS DES LIGNES

(voir Doc. n° 11 et 12)

II.1 Groupe Processeur : (28 lignes)

- 16 lignes $\overline{\text{MOUT}}$ 0-15 servant à véhiculer, soit les infos à écrire, soit les adresses mémoire jusqu'à 64 K.

- 4 lignes $\overline{\text{ADRE}}$ 0-3 pour les adresses supérieures à 64 K. On peut ainsi adresser 1 mégamot.

  En outre, lors de l'envoi d'infos vers la mémoire, la ligne $\overline{\text{ADRE}}$ 3 transmet le bit de parité.

- 4 lignes $\overline{\text{PPM}}$ 0 à $\overline{\text{PPM}}$ 3 permettant aux processeurs P0 à P3 de formuler une demande d'attribution du bus, laquelle sera satisfaite selon un ordre de priorité décroissant de 0 à 3.

- 2 lignes $\overline{\text{MC}}$ 0 et $\overline{\text{MC}}$ 1 (Memory Control) qui codent la fonction à réaliser en mémoire et répondent à la table de vérité ci-dessous :

	$\overline{\text{MC}}$ 1	$\overline{\text{MC}}$ 0
Ecriture	1	0
*Echange	0	0
Lecture	0	1
Ineffectif	1	1

  On voit qu'au repos les deux signaux $\overline{\text{MC}}$ 0 et $\overline{\text{MC}}$ 1 sont maintenus au niveau 1 logique.

- 1 ligne $\overline{\text{ABORT}}$ pouvant le cas échéant transmettre une commande d'annulation du cycle mémoire sollicité par un $\overline{\text{PPM}}$ (voir plus loin).

- 1 ligne $\overline{\text{RECALL}}$ pour rappeler la mémoire lorsque celle-ci a été trouvée occupée lors d'un précédent appel.

* C'est une écriture avec, en retour, lecture du précédent contenu de l'adresse à laquelle on écrit.

II.2 Groupe mémoire : (27 lignes)

- 16 lignes $\overline{\text{MIN}}$ 0-15 pour le transfert vers les processeurs des infos lues en mémoire.

- 1 ligne $\overline{\text{MPAR}}$ réservée au bit de parité des infos lues.

- 4 lignes $\overline{\text{MP}}$ 0 à $\overline{\text{MP}}$ 3 permettant à un stack mémoire de formuler une demande d'attribution des lignes $\overline{\text{MIN}}$0-15 et $\overline{\text{MPAR}}$ pour envoyer les infos lues au processeur concerné (de même indice). Celle-ci sera satisfaite selon l'ordre de priorité des processeurs, qui va décroissant de 0 à 3, et les infos seront transmises au coup d'horloge suivant.

- 4 lignes $\overline{\text{PGO}}$ 0-3 pour indiquer au processeur concerné (même indice) qu'il doit réitérer sa demande de cycle mémoire, celle-ci n'ayant pu être satisfaite en raison d'une occupation du stack adressé.

- 1 ligne $\overline{\text{POSRESP}}$ qui indique au processeur venant d'envoyer ses signaux de commande que la demande de cycle mémoire formulée est acceptée.

- 1 ligne $\overline{\text{HSM}}$ sur laquelle la mémoire rapide, et elle seule, émet un signal lui permettant de s'octroyer les lignes $\overline{\text{MIN}}$ 0-15 et $\overline{\text{MPAR}}$ en superpriorité par inhibition des infos en provenance de tout autre stack.

II.3 Signaux de service : (2 lignes)

Ces deux lignes, CLOCK et $\overline{\text{CLEAR}}$, n'appartiennent pas en propre au bus mémoire, les signaux qui les parcourent n'étant émis ni par la mémoire, ni par les processeurs. On les trouve à la fois dans la partie haute et la partie basse du fond de panier.

- ligne CLOCK : elle distribue à toutes les cartes présentes dans le bac un signal d'horloge symétrique et de période égale à 125 ns, sur lequel sont synchronisés tous les autres signaux du bus mémoire.
- ligne $\overline{\text{CLEAR}}$: le signal qu'elle transmet a pour but d'initialiser toutes les cartes du bac.

II.4 Répertoire des signaux

ÉTIQUETTE	SIGNIFICATION ANGLAISE	SIGNIFICATION FRANÇAISE	ÉMIS PAR	REÇU PAR	BROCHES
GROUPE PROCESSEUR					
#MOUT 0-15	Memory bus OUTput	Signaux du bus mémoire SORTANT d'un processeur et représentant. sur le top d'horloge consécutif à un #PPM, les poids faibles de l'adresse et sur le top qui suit, l'information à écrire dans 1e cas d'une écriture ou d'un échange.	Processeurs	Mémoires	A46-51
#ADRE 0-3	ADRess Extension	Signaux du bus mémoire représentant sur le top d'horloge consécutif à un #PPM, les poids forts de l'adresse. Sur le top qui suit, pour une écriture ou un échange, seul #ADRE3 a une signification et représente la parité de l'info à écrire.	"	"	A42-45
#PPM 0-3	Processor Priority Memory	#PPM1 = signal de requête d'accès au bus mémoire (groupe processeur) émis par le processeur P1 (Priorité d'ordre 1 décroissante de 0 à 3).	"	Mémoires et Proc.	A38-41
#MC 0-1	Memory Control	Bits de codage de la fonction mémoire demandée.	"	Mémoires	A36-37
#ABORT	ABORTion	Signal da commande d'avortement d'un cycle mémoire demandé. Il doit suivre immédiatement le #PPM.	"	"	A35
#RECALL	RECALL	Signal émis par un processeur P1 lors du renvoi de l'adresse mémoire en réponse à un #PG01.	"	"	A30
GROUPE MÉMOIRE					
#MIN 0-15	Memory bus INput	Information lue en mémoire, ENTRANT dans un processeur	Mémoires	Processeur	E46-51
#MPAR	Memary PARity	Parité lue en mémoire	"	"	E45
#MP 0-3	Memory Priority	#MP1 " signal de requête d'accès au bus mémoire (groupe mémoire) pour envoi d'une info lue vers le processeur P1 (Priorité d'ordre 1 décroissante de 0 à 3).	"	Mémoires et Proc.	E41-44
#PG0 0-3	Processor G0	PGO1 = signal demandant au processeur P1 de renouveler sa demande de cycle précédemment refusée.	"	Processeurs	E37-40
#P0SRESP	POSitive RESPonse	Réponse d'acceptation d'une mémoire à une demande de cycle qui vient d'être formulée par un processeur P1 (#P0SRESP retardé d'un coup d'horloge par rapport au #MC émis par P1).	"	"	E35
#HSM	High Speed Memory	Signal émis par les stacks rapides (125 ns de temps d'accès). I1 leur confère une superpriorité en leur attribuant d'office les lignes MIN et MPAR pour l'envoi de l'information lue.	Mémoire rapide (module SCS 04)	Mém. Et Proc.	E35
SIGNAUX DE SERVICE					
#CLOCK	CLOCK	Horloge	Mini pupitre	Toutes les 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. au pupitre	du bac	A33, E33 B32, H32

III. DIALOGUE PROCESSEURS - MEMOIRE

III.1 Principe

Tout processeur désirant faire appel à un stack mémoire (que ce soit pour lire, écrire, ou faire un échange) doit. avant toute chose, émettre, sur la ligne PPM qui lui est propre, une demande d'attribution du bus.

La logique déterminant la priorité est implantée sur chaque processeur, si bien que les stacks mémoire ne voient, au plus, qu'un 0 parmi des 1 sur les quatre lignes PPM.

Cet état bas est maintenu durant une période d'horloge et au top suivant le processeur correspondant envoie sur le bus l'adresse mémoire (lignes MOUT 0-15 et ADRE 0-3) et la fonction à réaliser (lignes MC 0 et MC 1). Si cette dernière est une ECRITURE, ou un ECHANGE, au coup d'horloge suivant, l'info à écrire sera envoyée sur les lignes MOUT 0-15 et la parité sur la ligne ADRE 3.

Dès la réception de l'adresse et de la fonction, le stack mémoire concerné émet, s'il n'est pas occupé, un signal baptisé #P0SRESP (Positive Response) signifiant que la demande de cycle est prise en compte.

Dans le cas contraire, le processeur appelant Pi devra attendre de recevoir, sur sa ligne propre, le signal #PG0i lui enjoignant de réitérer sa demande de cycle du début. Le déroulement de cette demande sera identique au précédent, avec la seule différence que Pi émettra, en même temps que les #MC, le signal #RECALL spécifiant qu'il s'agit là d'une "redemande".

Pour une lecture ou un échange, un #MPi prévient le processeur Pi que l'info attendue sera sur le bus au clock suivant (sauf s'il y a #HSM comme nous le verrons plus loin).

L'intervalle entre un #P0SRESP et un #MP (toute question de priorité mise à part) dépend du type de stack auquel on s'adresse ; de plus, avec certaines mémoires, il y a risque de "tomber" pendant un cycle de rafraîchissement pouvant provoquer un retard de plusieurs clocks. C'est pourquoi nous avons fait figurer sur les diagrammes illustrant les exemples qui suivent une coupure entre #P0SRESP et #MP. Mais dans l'absolu, rien n'empêche une mémoire, suffisamment rapide, d'envoyer son #MP en même temps que le #P0SRESP.

Quant à la High Speed Memory, elle n'émet pas de #MP, mais le signal #HSM qui accompagne l'info lue sur le même clock que #POSRESP.

Les dix exemples qui suivent ne font pas le tour de tous les cas envisageables, mais les explications et les diagrammes qui les accompagnent permettent de concevoir toutes les possibilités de dialogue offertes par le bus mémoire.

Pour indiquer sur les diagrammes le sens des signaux, nous avons adopté la convention suivante


signal émis par un processeur


signal émis par un stack mémoire

III.2 Un seul processeur demande le bus :

III.2.1 Écriture (W = WRITE)

DIAGRAMME N° 1

Vu du processeur, le cycle ne dure que 3 coups d'horloge, (375 ns) qu'il s'agisse d'un monoprocesseur ou, dans un système multiprocesseur, d'un processeur prioritaire s'adressant à un stack mémoire non occupé. En effet, pour gagner du temps, les infos à écrire suivent systématiquement les adresses, sans attendre le #POSRESP (Positive Response) confirmant que le stack sollicité est en mesure de satisfaire la demande et mettant fin au dialogue.

III.2.2 Lecture (R = READ)

DIAGRAMME N° 2

Comme pour l'écriture, sur le top d'horloge suivant l'adresse, le stack concerné envoie le signal #POSRESP pour prévenir le processeur Pi que sa demande est prise en compte. Puis, après un délai dépendant, comme vu plus haut, du type de mémoire utilisé, il lance sur la ligne MPi, propre à Pi, un signal avertissant ce dernier qu'au prochain clock l'info lue se trouvera sur les lignes MIN et MPAR.

III.2.3 Échange (R.W = READ and WRITE)

DIAGRAMME N° 3

C'est la somme d'une lecture et d'une écriture et le diagramme du dialogue est la superposition des deux précédents. On envoie une adresse et une info comme pour une écriture, à ceci près que #MC 0 et #MC 1 passent tous deux par zéro pour coder la fonction échange, et on reçoit "en échange", comme dans une lecture, l'info qui se trouvait à l'adresse donnée.

III.3 Deux processeurs vers deux stacks distincts

111.3.1 Le plus prioritaire demande une LECTURE

DIAGRAMME N° 4

P2 et P3 appellent en même temps 2 stacks différents, le premier pour lire (R2) et l'autre pour écrire (W3).

Les deux cycles se déroulent normalement ; tout au plus W3 est-il retardé d'un clock du fait que P3 est moins prioritaire que P2.

111.3.2 Le plus prioritaire demande une ÉCRITURE

DIAGRAMME N° 5

P2 voit son cycle se dérouler tout à fait normalement. Par contre, pour P3 la lecture est retardée de 2 coups d'horloge, ce qui correspond au temps d'occupation par P2 des lignes MOUT et ADRE : 1 clock pour l'adresse et un autre pour l'info à écrire. I1 en serait de même pour un échange (puisqu'il implique une écriture), autrement dit chaque fois que #MCO = 0. En bref, un processeur ne peut s'attribuer le bus mémoire que si, au moment où il envoie son #PPM, aucun autre plus prioritaire n'en fait la demande et que la ligne MCO est à l'état haut.

III.4 Deux processeurs vers le même stack

DIAGRAMME N° 6

P3 pour une ECRITURE (W3) et au coup d'horloge suivant P2 pour une LECTURE (R2).

Tout se passe comme dans le cas précédent si ce n'est que P2 ne reçoit pas de #POSRESP, le stack qu'il sollicite étant occupé avec P3. Lorsque le processeur P2 va recevoir, sur sa ligne particulière, le signal #PG02 il saura qu'il doit renouveler sa demande de cycle en renvoyant un #PPM 2 ; au clock d'après, adresse et fonction devront être accompagnées du signal #RECALL pour bien spécifier qu'il s'agit d'un rappel. Le stack répondra alors #POSRESP dans la majorité des cas ; toutefois s'il n'était toujours pas en mesure de prendre en compte la demande de cycle de P2, il enverrait un nouveau #PGO 2 auquel P2 n'aurait qu'à se soumettre en renouvelant la séquence complète.

D'une manière générale, un processeur se tient toujours prêt à réitérer sa demande de cycle dès l'instant qu'elle n'a pas été immédiatement suivie d'un #POSRESP.

NOTE IMPORTANTE

Un stack sollicité par plusieurs processeurs les met dans une file d'attente et ils seront servis dans l'ordre selon lequel ils se sont manifestés pour la première fois.

C'est ainsi que, dans le cas ci-dessus, le #PPM 2 renouvelé après le #PGO 2 aurait très bien pu être repoussé de 2 clocks, à savoir par un #PPM 1, puis un #PPM 0 ; ce qui n'aurait pas empêché P2 d'être servi le premier des trois, suivi de P1 rappelé par #PGO 1 et enfin de PO rappelé par #PG0 0.

III.5 Un des processeurs s'adresse au stack HSM :

La différence entre un stack normal et un stack rapide HSM réside dans le fait que ce dernier envoie l'info lue sur le bus en même temps que le #POSRESP, c'est-à-dire au coup d'horloge qui suit la réception de l'adresse.

De plus, il lance simultanément sur la ligne réservée à cet effet le signal #HSM qui empêche tout autre stack d'émettre sur les lignes MIN et MPAR.

III.5.1 Le stack HSM ne retarde pas les autres :

DIAGRAMME N° 7

Les processeurs P2, puis Pi veulent lire, le premier dans un stack normal, le second dans le stack HSM.

On voit que, si #HSM se présente en même temps que #MP2 (et a fortiori si c'est avant), l'envoi de l'info lue pour P2 n'est en rien retardé.

III.5.2 Le stack HSM fait jouer sa priorité :

DIAGRAMME N° 8

Cette fois le signal #HSM se présente au moment où l'info lue pour P2 va être envoyée sur le bus. Elle est stoppée et c'est celle émanant du stack rapide qui passe sur les lignes MIN et MPAR. Ici le signal #MP 2 qui remontait est remis à l'état bas pour deux clocks de plus et de ce fait, P2 est servi avec 250 ns de retard, ce qui est le meilleur des cas. En effet, tous les stacks laissent passer un coup d'horloge avant de renouveler leur #MP, lequel, qui plus est, peut être repoussé par 1 ou 2 autres #MP plus prioritaires.

III.6 Un processeur envoie un #ABORT :

DIAGRAMME N° 9

P2 et P3 s'adressent simultanément à un même stack pour une lecture. P2, plus prioritaire, obtient le bus, mais travaillant en mode esclave, il lui faut comparer MA + SLO à SLE. Pour gagner du temps il lance le cycle sans attendre le résultat du test ; si celui-ci, comme dans le cas présent, donne MA + SLO > SLE, il génère le signal #ABORT qui annule la demande avant que le stack n'ait émis le #POSRESP. Et le cycle demandé par P3 va se dérouler normalement à la suite du #PRM 3.

III.7 Trois processeurs demandent le bus simultanément

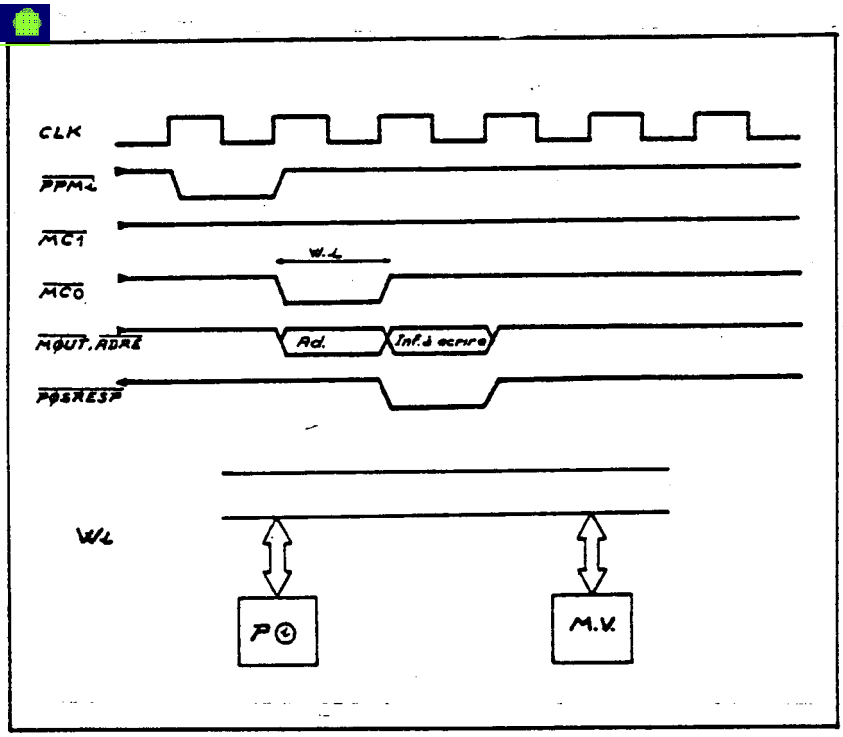
DIAGRAMME N° 10

On peut bien sûr concevoir un grand nombre de cas de figure et il n'est pas question de les passer tous en revue. Voici à titre d'exemple le cas de 3 processeurs P1, P2 et P3 s'adressant à 3 stacks différents pour faire respectivement une lecture (R1), un échange (R.W2) et une lecture (R3).

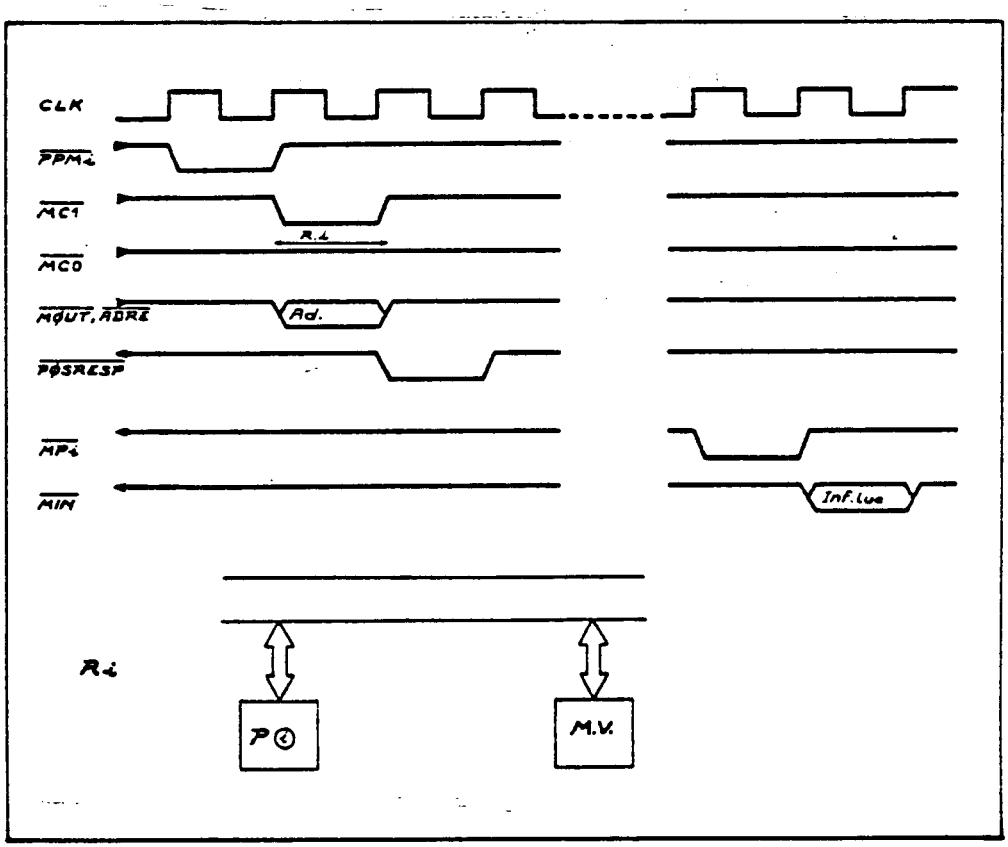
P1 va être servi très rapidement : il est le plus prioritaire et s'adresse qui plus est au stack rapide. Au coup d'horloge suivant l'envoi de l'adresse il reçoit l'info lue, en même temps que les signaux #POSRESP et #HSM.

Pour P2 et P3 le partage du bus se fait comme décrit plus haut et nous supposons ici que les 2 stacks auxquels ils s'adressent veulent envoyer l'info demandée simultanément. Le conflit se règle au niveau du #MP. Les stacks n'ont pas de rang de priorité établi à l'avance ; il est celui du processeur pour lequel il travaille. C'est ainsi que #MP2 repousse #MP3 d'un clock et que les lignes MIN et MPAR verront se succéder les infos lues destinées à P2 puis à P3.

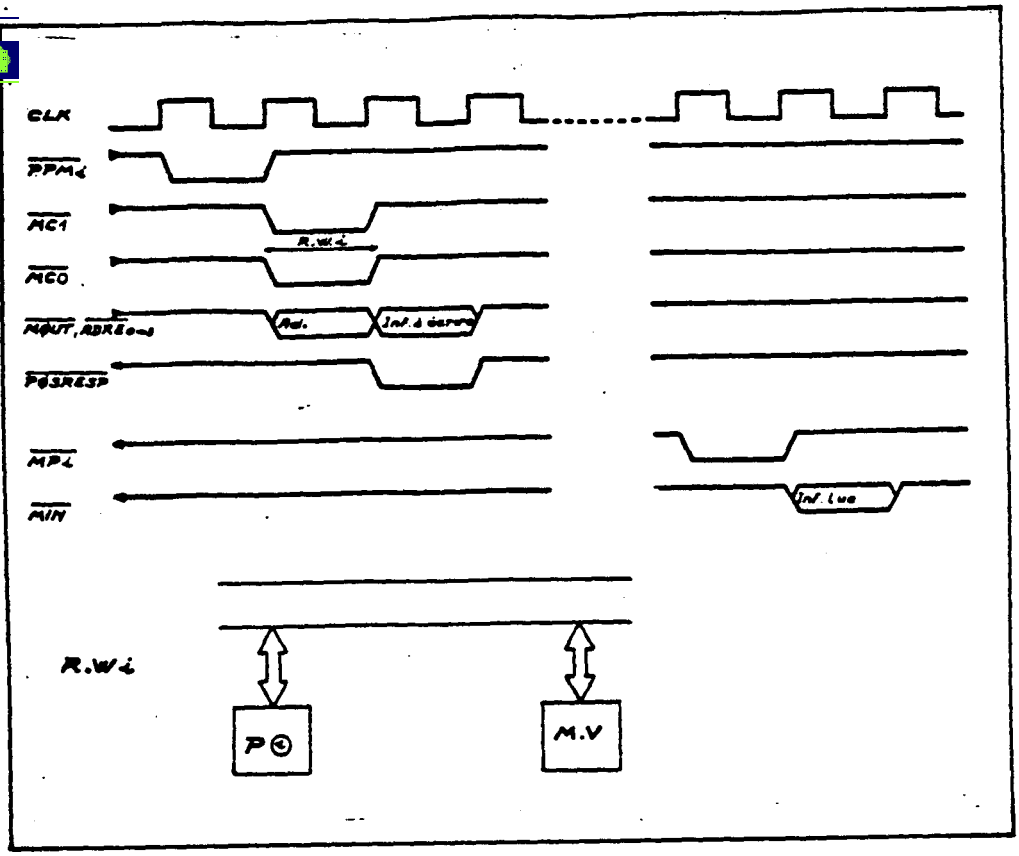
①


CLK
PPM
MC1
MC0
W.L
MOUT. ADRE
Ad.
Inf. à ecrire
POSRESP
W/L
P
M.V.

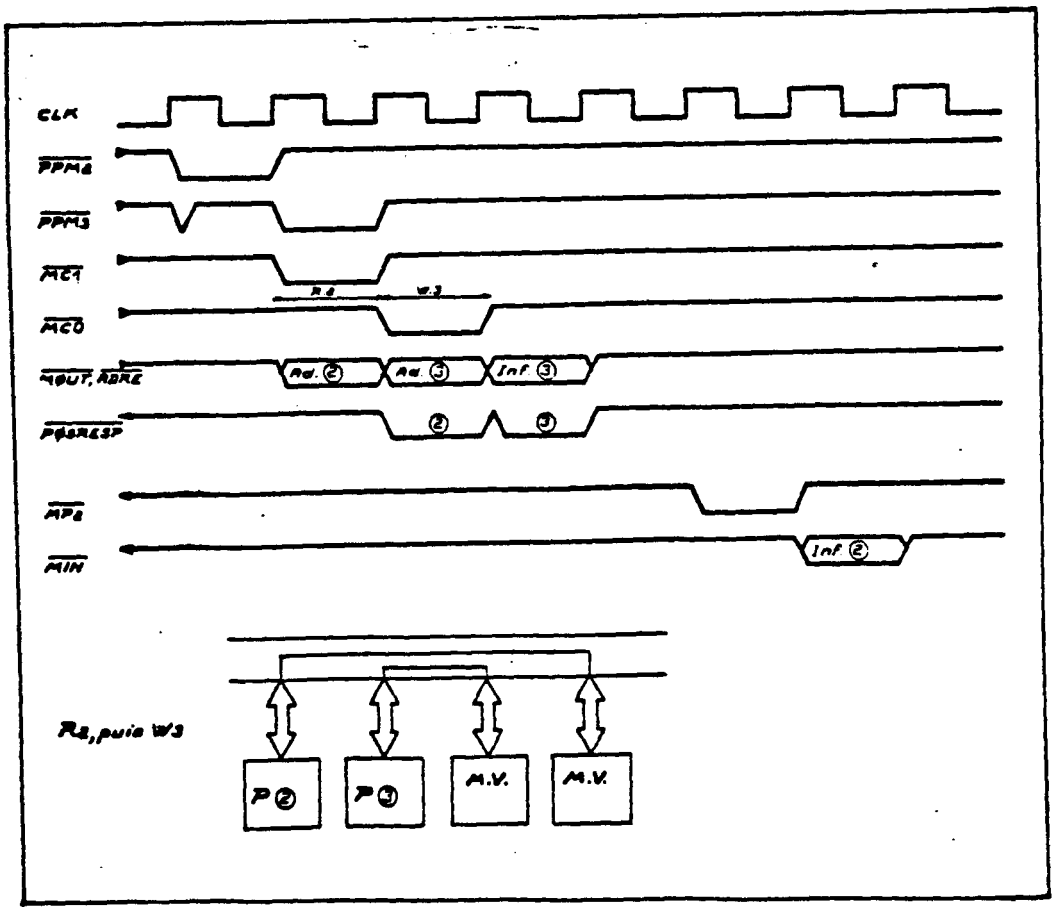
②


CLK
PPM
MC1
R.L
MC0
MOUT. ADRE
Ad.
POSRESP
MP
MIN
Inf. lue
R/
P
M.V.

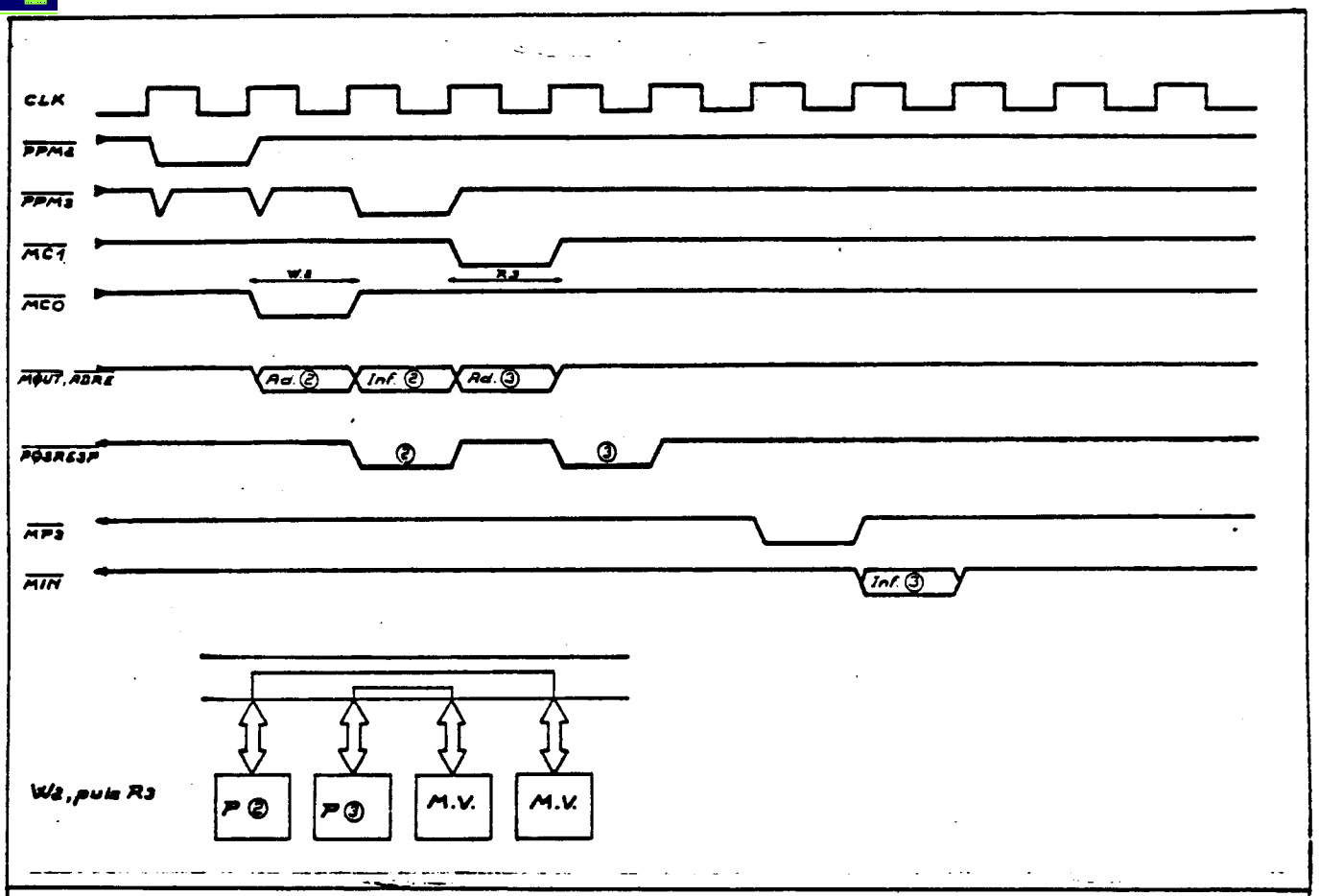
3


CLK
PPM4
MC1
MC0
R.W4
ADOUT, ADRES
Ad.
PGSRESP
MPL
MIN
Inf. lue
R.W4
P④
M.V

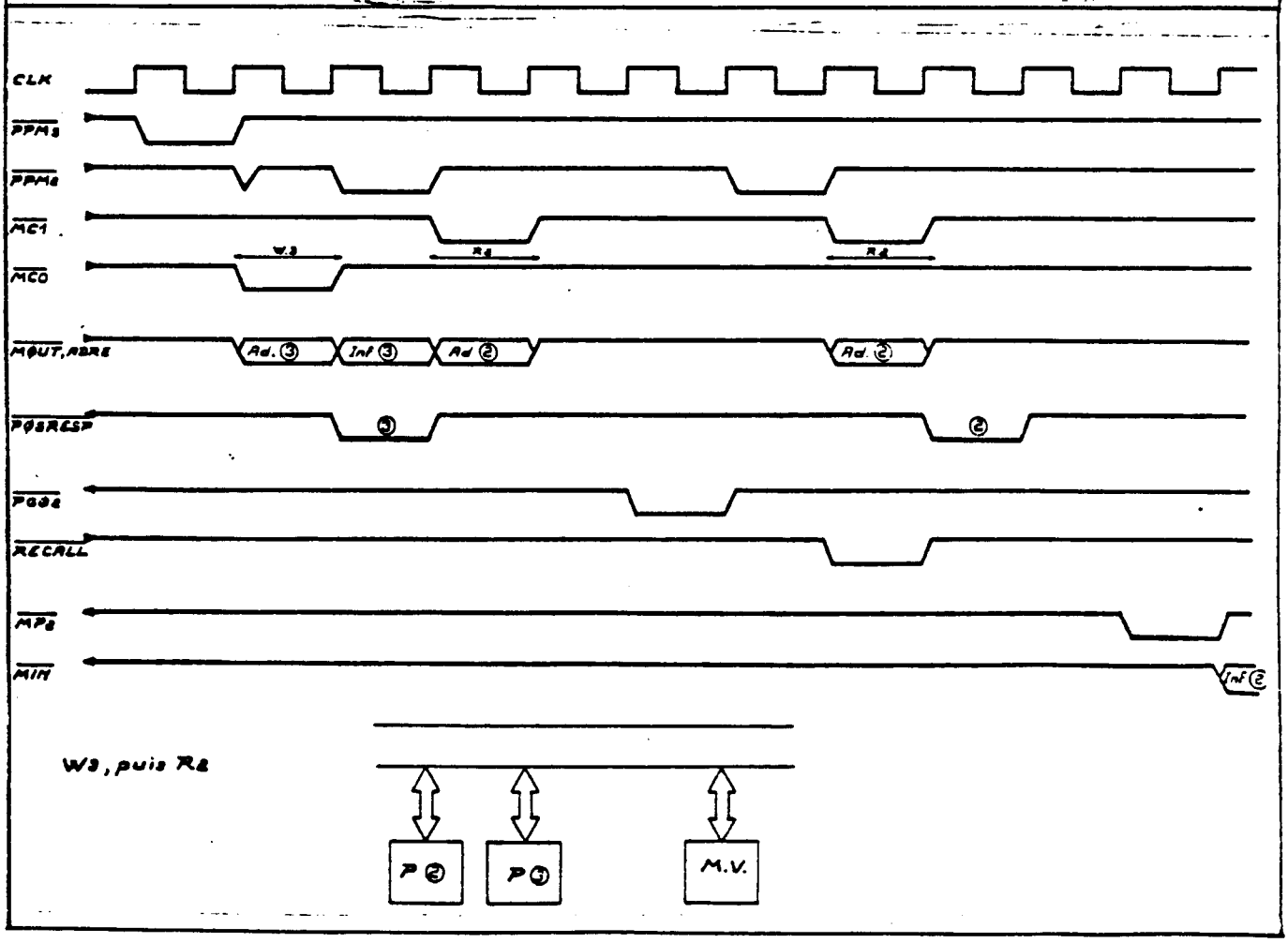
4


CLK
PPM4
PPM3
MC1
MC0
ADOUT, ADRE
Ad. ②
Ad. ③
Inf. ③
PGSRESP
②
③
MPL
MIN
Inf. ②
R2, puis W3
P②
P③
M.V.
M.V.

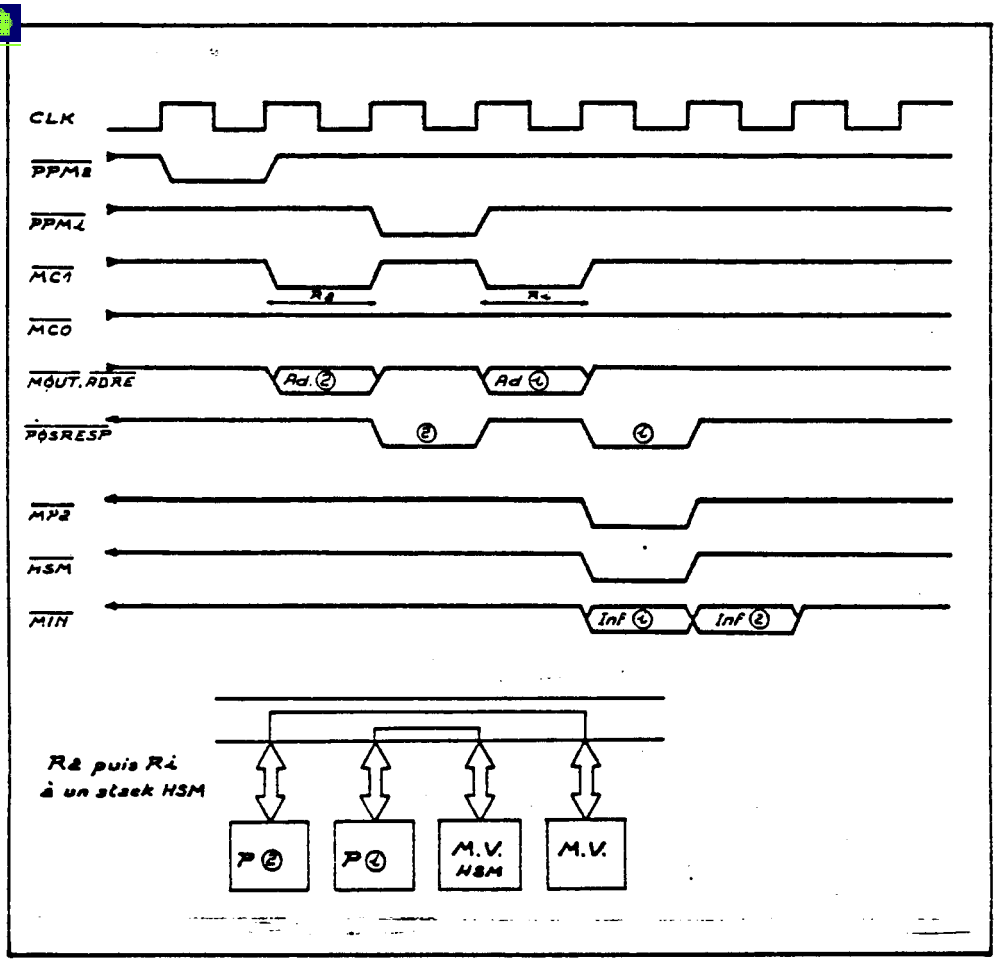
⑤


CLK
PPM4
PPM3
MC1
MC0
W4
R3
MOUT, ADRE
Ad. ②
Inf. ②
Ad. ③
POSREGP
②
③
MP3
MIN
Inf. ③
W2, puis R3
P ②
P ③
M.V.
M.V.

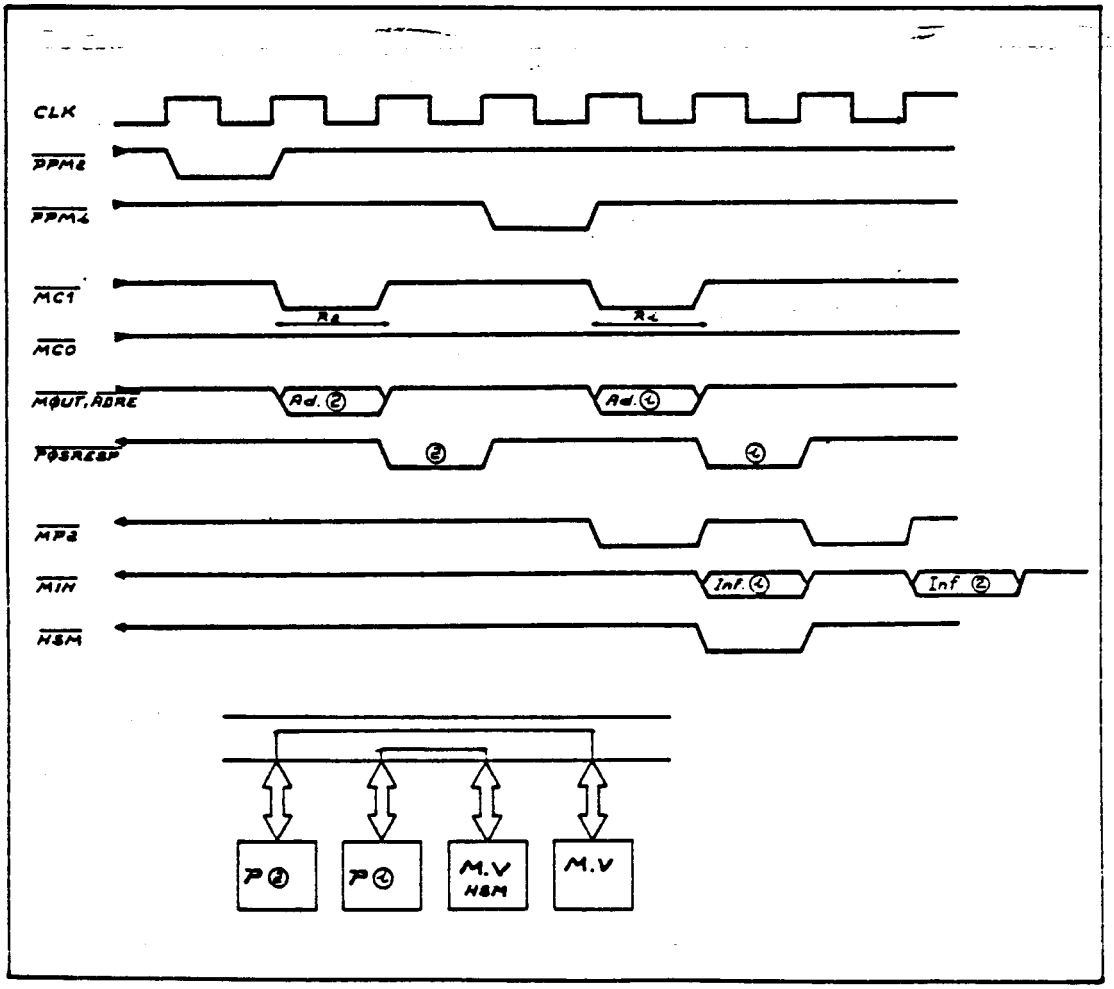
⑥


CLK
PPM3
PPM4
MC1
MC0
W3
R4
R4
MOUT, ADRE
Ad. ③
Inf ③
Ad ②
Ad. ②
POSREGP
③
②
POS4
RECALL
MP2
MIN
Inf ②
W3, puis R2
P ②
P ③
M.V.

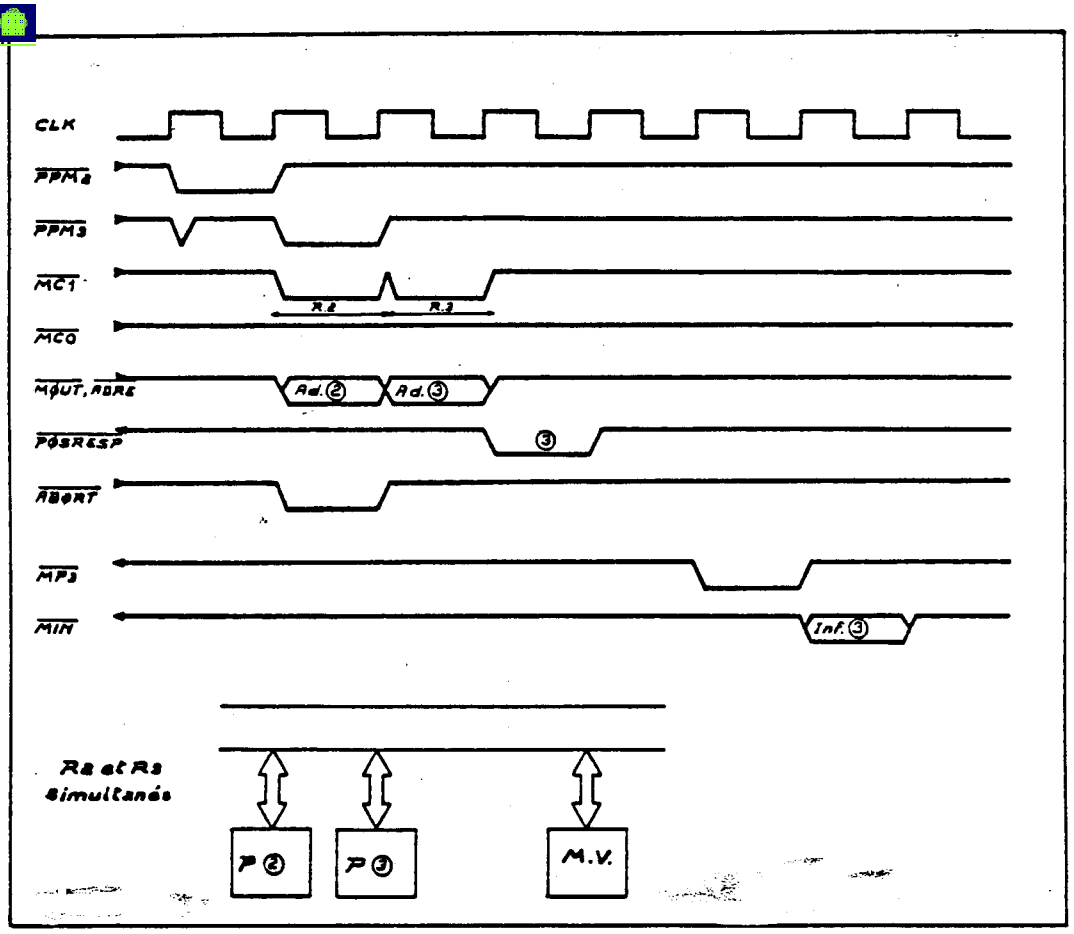
7


CLK
PPM2
PPM4
MC1
MC0
MOUT. ADRE
POSRESP
MP2
HSM
MIN
Ad. 2
Ad 4
Inf 4
Inf 2
R2 puis R4
à un stack HSM
P 2
P 4
M.V. HSM
M.V.

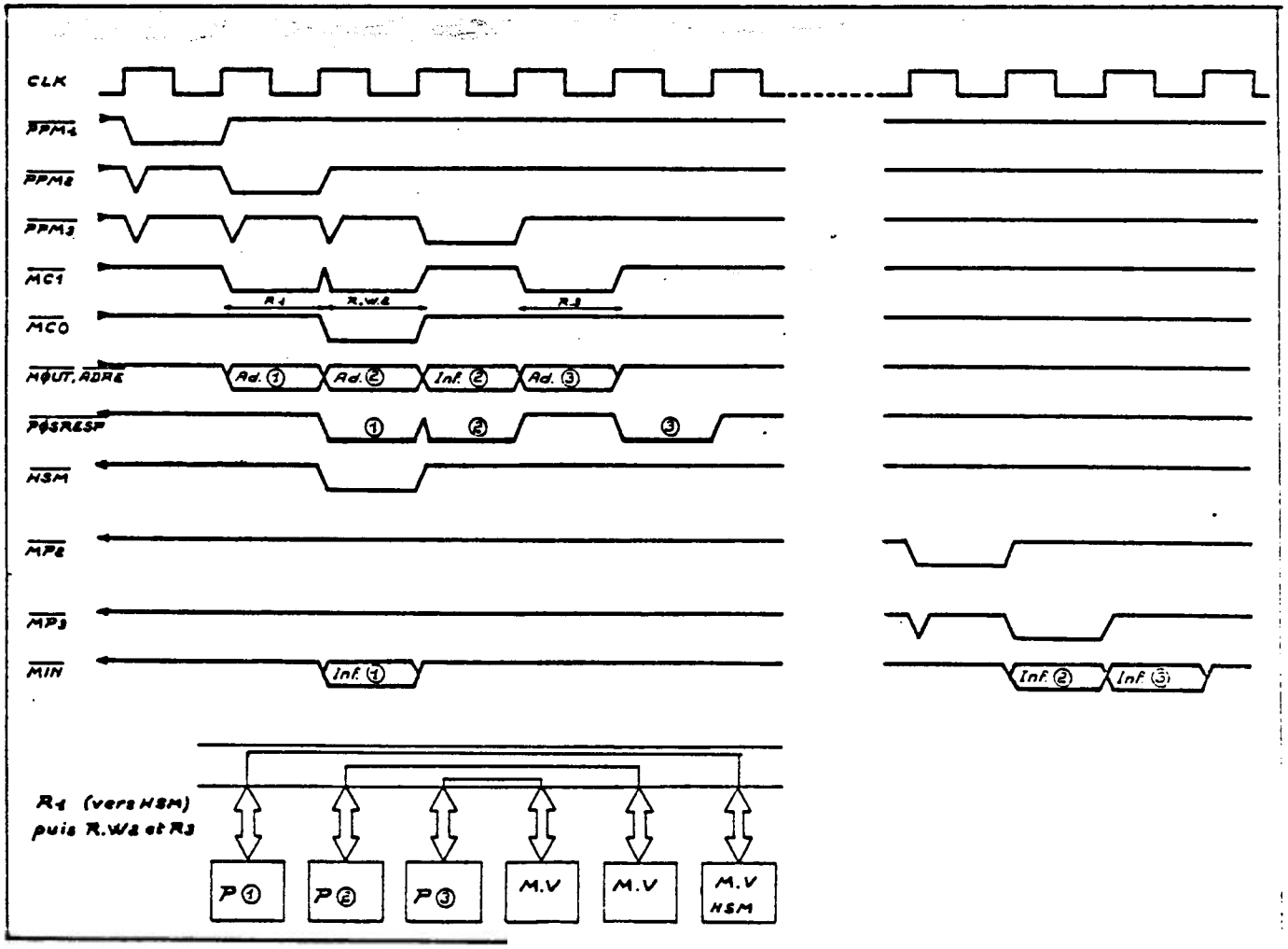
8

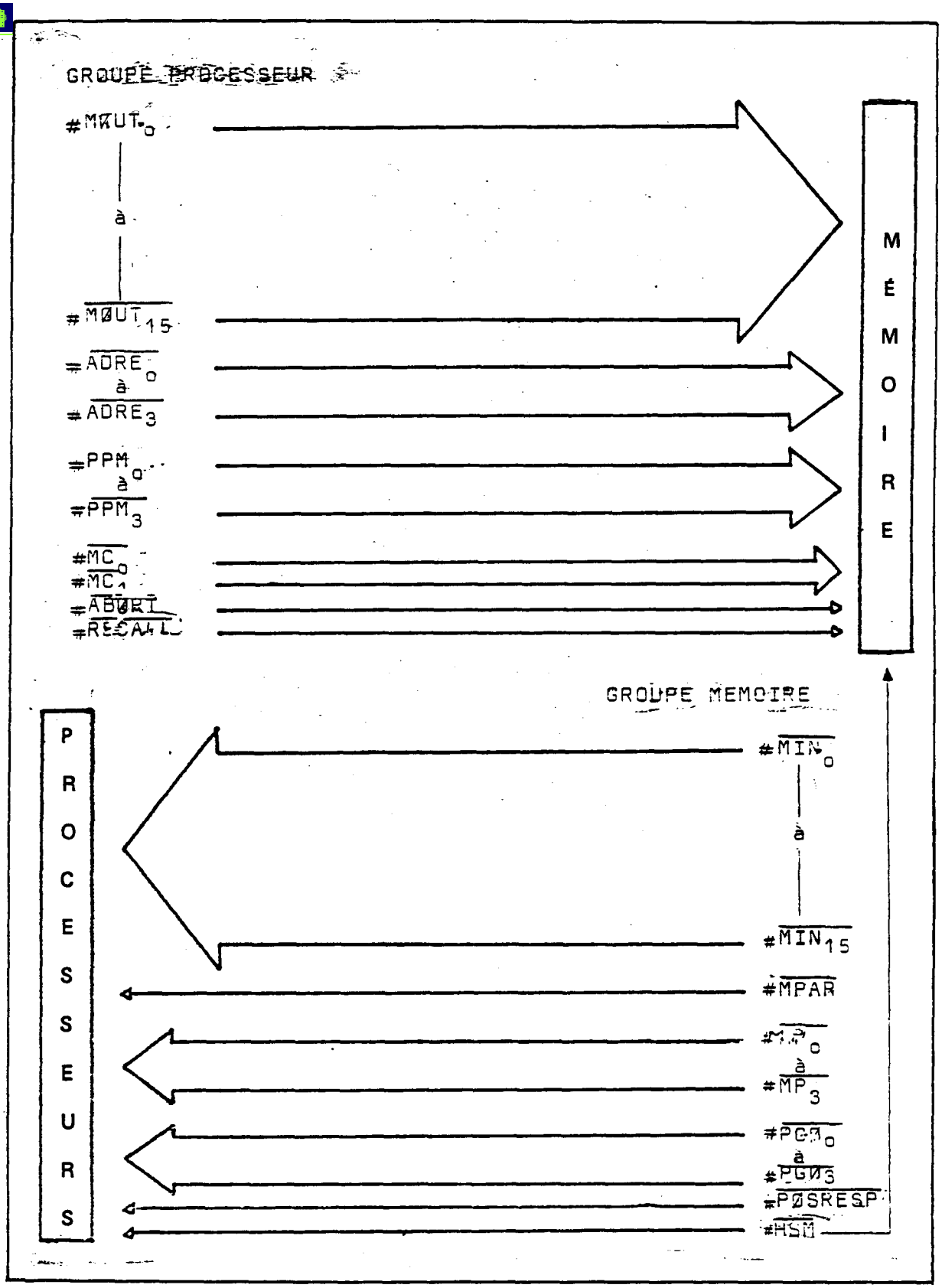

CLK
PPM2
PPM4
MC1
MC0
MOUT. ADRE
POSRESP
MP2
MIN
HSM
R2
R4
Ad. 2
Ad. 4
Inf. 4
Inf 2
P 2
P 4
M.V HSM
M.V

⑨

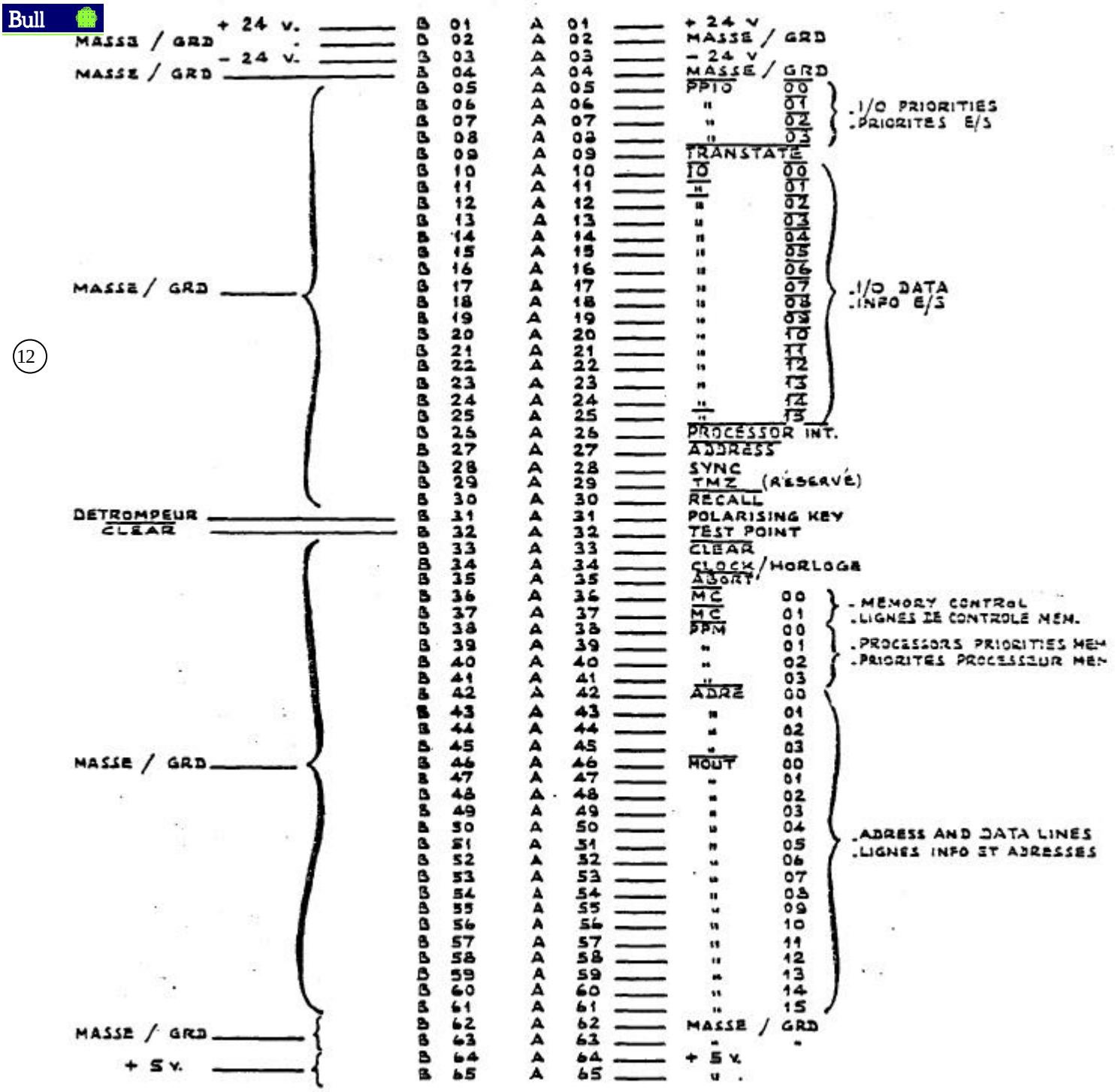

CLK
PPM2
PPM3
MC1
MC0
MOUT, ADRE
POSRESP
ABORT
MP3
MIN
R2
R3
Ad. ②
Ad. ③
③
Inf. ③
R2 et R3 simultanés
P②
P③
M.V.

⑩

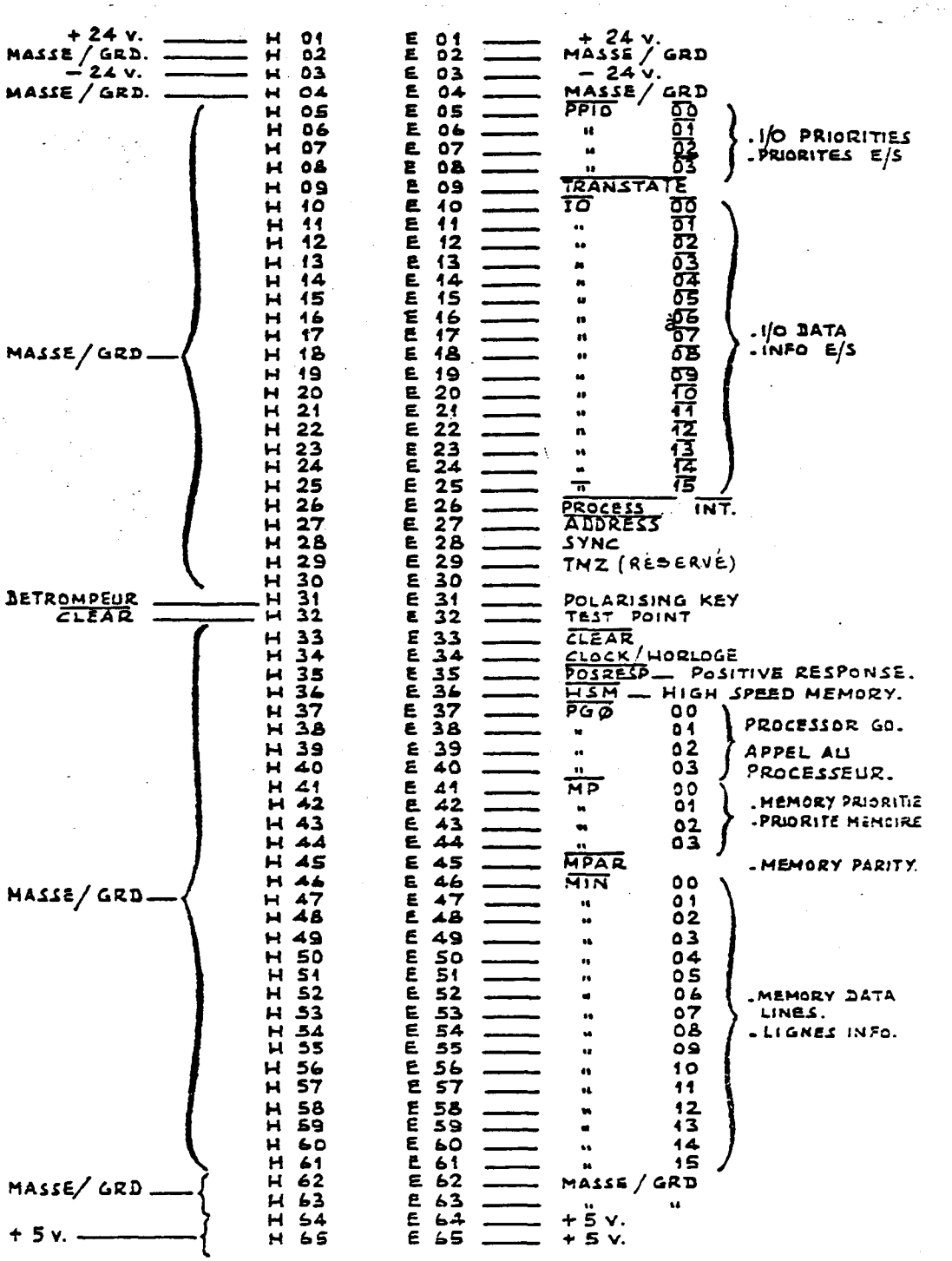

CLK
PPM1
PPM2
PPM3
MC1
MC0
MOUT, ADRE
POSRESP
MSM
MP2
MP3
MIN
R1
R.W2
R3
Ad. ①
Ad. ②
Inf. ②
Ad. ③
①
②
③
Inf. ①
Inf. ②
Inf. ③
R1 (vers MSM) puis R.W2 et R3
P①
P②
P③
M.V
M.V
M.V MSM


GROUPE PROCESSEUR
#MOUT0
à
#MOUT15
#ADRE0
à
#ADRE3
#PPM0
à
#PPM3
#MC0
#MC1
#ABORT
#RESALL
MÉMOIRE
GROUPE MEMOIRE
PROCESSEURS
#MIN0
à
#MIN15
#MPAR
#MP0
à
#MP3
#PGM0
à
#PGM3
#POSRESP
#HSM

11


+ 24 V.
MASSE / GRD
- 24 V.
MASSE / GRD
MASSE / GRD
DETROMPEUR
CLEAR
MASSE / GRD
MASSE / GRD
+ 5 V.
B 01 A 01 + 24 V
B 02 A 02 MASSE / GRD
B 03 A 03 - 24 V
B 04 A 04 MASSE / GRD
B 05 A 05 PPIO 00
B 06 A 06 " 01
B 07 A 07 " 02
B 08 A 08 " 03
.I/O PRIORITIES
.PRIORITES E/S
B 09 A 09 TRANSTATE
B 10 A 10 IO 00
B 11 A 11 " 01
B 12 A 12 " 02
B 13 A 13 " 03
B 14 A 14 " 04
B 15 A 15 " 05
B 16 A 16 " 06
B 17 A 17 " 07
B 18 A 18 " 08
B 19 A 19 " 09
B 20 A 20 " 10
B 21 A 21 " 11
B 22 A 22 " 12
B 23 A 23 " 13
B 24 A 24 " 14
B 25 A 25 " 15
.I/O DATA
.INFO E/S
B 26 A 26 PROCESSOR INT.
B 27 A 27 ADDRESS
B 28 A 28 SYNC
B 29 A 29 TMZ (RÉSERVÉ)
B 30 A 30 RECALL
B 31 A 31 POLARISING KEY
B 32 A 32 TEST POINT
B 33 A 33 CLEAR
B 34 A 34 CLOCK / HORLOGE
B 35 A 35 ABORT
B 36 A 36 MC 00
B 37 A 37 MC 01
B 38 A 38 PPM 00
B 39 A 39 " 01
B 40 A 40 " 02
B 41 A 41 " 03
- MEMORY CONTROL
.LIGNES DE CONTROLE MEM.
.PROCESSORS PRIORITIES MEM
.PRIORITES PROCESSEUR MEM
B 42 A 42 ADRE 00
B 43 A 43 " 01
B 44 A 44 " 02
B 45 A 45 " 03
B 46 A 46 MOUT 00
B 47 A 47 " 01
B 48 A 48 " 02
B 49 A 49 " 03
B 50 A 50 " 04
B 51 A 51 " 05
B 52 A 52 " 06
B 53 A 53 " 07
B 54 A 54 " 08
B 55 A 55 " 09
B 56 A 56 " 10
B 57 A 57 " 11
B 58 A 58 " 12
B 59 A 59 " 13
B 60 A 60 " 14
B 61 A 61 " 15
.ADRESS AND DATA LINES
.LIGNES INFO ET ADRESSES
B 62 A 62 MASSE / GRD
B 63 A 63 " "
B 64 A 64 + 5 V.
B 65 A 65 " "

12


+ 24 V.
MASSE / GRD.
— 24 V.
MASSE / GRD.
MASSE / GRD
DETROMPEUR
CLEAR
MASSE / GRD
MASSE / GRD
+ 5 V.
H 01 E 01 + 24 V.
H 02 E 02 MASSE / GRD
H 03 E 03 — 24 V.
H 04 E 04 MASSE / GRD
H 05 E 05 PPID 00
H 06 E 06 " 01
H 07 E 07 " 02
H 08 E 08 " 03
. I/O PRIORITIES
. PRIORITES E/S
H 09 E 09 TRANSLATE
H 10 E 10 IO 00
H 11 E 11 " 01
H 12 E 12 " 02
H 13 E 13 " 03
H 14 E 14 " 04
H 15 E 15 " 05
H 16 E 16 " 06
H 17 E 17 " 07
H 18 E 18 " 08
H 19 E 19 " 09
H 20 E 20 " 10
H 21 E 21 " 11
H 22 E 22 " 12
H 23 E 23 " 13
H 24 E 24 " 14
H 25 E 25 " 15
. I/O DATA
. INFO E/S
H 26 E 26 PROCESS INT.
H 27 E 27 ADDRESS
H 28 E 28 SYNC
H 29 E 29 TNZ (RESERVÉ)
H 30 E 30
H 31 E 31 POLARISING KEY
H 32 E 32 TEST POINT
H 33 E 33 CLEAR
H 34 E 34 CLOCK / HORLOGE
H 35 E 35 POSRESP — POSITIVE RESPONSE.
H 36 E 36 HSM — HIGH SPEED MEMORY.
H 37 E 37 PGØ 00
H 38 E 38 " 01
H 39 E 39 " 02
H 40 E 40 " 03
PROCESSOR GO.
APPEL AU PROCESSEUR.
H 41 E 41 MP 00
H 42 E 42 " 01
H 43 E 43 " 02
H 44 E 44 " 03
. MEMORY PRIORITIE
. PRIORITE MEMOIRE
H 45 E 45 MPAR
H 46 E 46 MIN 00
H 47 E 47 " 01
H 48 E 48 " 02
H 49 E 49 " 03
H 50 E 50 " 04
H 51 E 51 " 05
H 52 E 52 " 06
H 53 E 53 " 07
H 54 E 54 " 08
H 55 E 55 " 09
H 56 E 56 " 10
H 57 E 57 " 11
H 58 E 58 " 12
H 59 E 59 " 13
H 60 E 60 " 14
H 61 E 61 " 15
. MEMORY PARITY.
. MEMORY DATA LINES.
. LIGNES INFO.
H 62 E 62 MASSE / GRD
H 63 E 63 " "
H 64 E 64 + 5 V.
H 65 E 65 + 5 V.

12

TEMPS A RESPECTER POUR LA GESTION DU BUS MEMOIRE

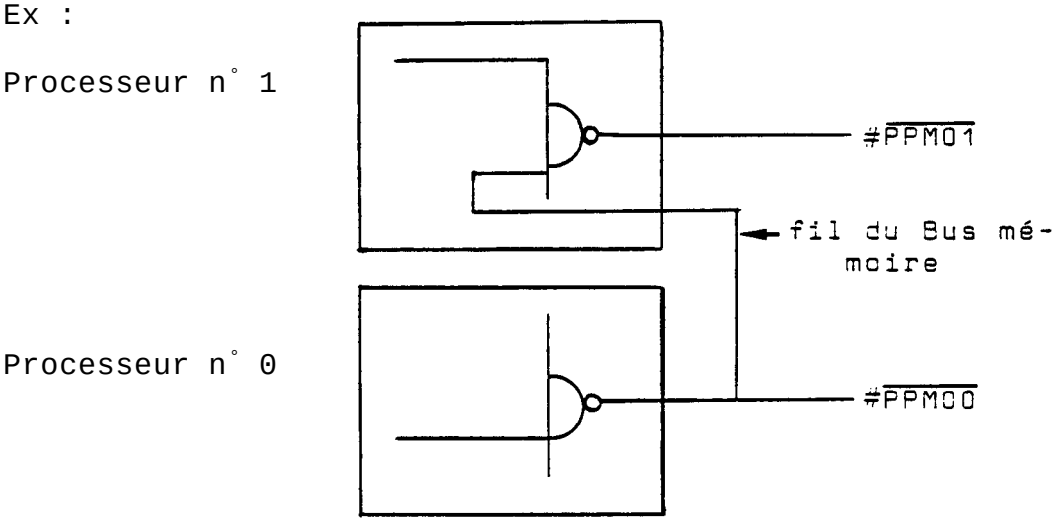
1 - Signaux venant des processeurs (vus du Bus mémoire)

Signaux	Propagation MAXIMUM	Maintien MINIMUM
MOUT et ADRE	70 ns	5 ns
MC	50 ns	5 ns
RECALL	50 ns	5 ns
ABORT	70 ns	5 ns
PPM	70 ns	5 ns
PPM (1)	90 ns	5 ns

(1) En cas de conflit avec une configuration multiprocesseurs le temps de propagation maximum de PPM est égal à : temps de propagation maximum en direct + temps de propagation maximum sur le Bus + temps de retour sur PPM

soit 70 ns + 10 ns + 10 ns = 90 ns

Ex :



2 - Signaux reçus des processeurs (vus du Bus mémoire)

Signaux	SET UP MINIMUM	HOLO MINIMUM
MIN	55 ns	0 ns
PORES	50 ns	0 ns
MP	35 ns	0 ns
PGO	45 ns	0 ns
HSM	40 ns	0 ns
PPM	15 ns	0 ns

NB : Les chaînes de données des processeurs et des mémoires doivent tenir compte des temps présentés ci-dessus et en utilisant une horloge à 120 ns.