

RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) N° de publication :
(A n'utiliser que pour les
commandes de reproduction).

2 513 409

A1

**DEMANDE
DE BREVET D'INVENTION**

(21) **N° 81 17832**

(54) Procédé de synchronisation de deux microprocesseurs.

(51) Classification internationale (Int. Cl. 3). G 06 F 9/00, 13/00.

(22) Date de dépôt..... 22 septembre 1981.

(33) (32) (31) Priorité revendiquée :

(41) Date de la mise à la disposition du
public de la demande B.O.P.I. — « Listes » n° 12 du 25-3-1983.

(71) Déposant : CGEE ALSTHOM, SA. — FR.

(72) Invention de : Daniel Boissard.

(73) Titulaire : *Idem* (71)

(74) Mandataire : Jacques Beylot, SOSPI,
14-16, rue de la Baume, 75008 Paris.

Procédé de synchronisation de deux microprocesseurs

Les microprocesseurs sont de plus en plus souvent utilisés dans des automatismes assurant la conduite d'installations industrielles. Certaines d'entre elles, sur lesquelles l'émission d'un ordre intempestif peut avoir de graves conséquences doivent être protégées contre un défaut éventuel de leur automatisme de commande. Une manière d'assurer cette protection consiste à employer pour la conduite d'une même installation deux microprocesseurs identiques qui fonctionnent en parallèle à partir des mêmes données disponibles sur des entrées communes et à l'aide de programmes distincts mais identiques, et qui sont couplés en sortie sur un comparateur de sécurité en logique câblée ne validant une information qu'en cas d'unanimité des microprocesseurs. Le moyen le plus simple pour obtenir un fonctionnement parallèle des microprocesseurs est de les faire fonctionner à partir d'une horloge commune et de synchroniser leurs cycles machine.

Cette synchronisation n'est ni naturellement établie ni naturellement stable car les phénomènes de départ et de parasites n'ont aucune cohérence avec l'état des microprocesseurs. En outre les microprocesseurs bien qu'identiques n'ont pas rigoureusement les mêmes seuils logiques et reçoivent des signaux d'horloge légèrement déphasés du fait des trajets différents suivis par ces derniers depuis l'horloge. Il en résulte qu'une même information n'est jamais échantillonnée aux mêmes instants ni comparée aux mêmes seuils par les deux microprocesseurs et peut de ce fait, surtout si elle présente des transitions asynchrones par rapport à l'horloge, être vue au cours du même microcycle à une certaine valeur par l'un des microprocesseurs et à une autre valeur par l'autre microprocesseur et entraîner une désynchronisation du fonctionnement de ces derniers.

La présente invention a pour but d'établir une synchronisation au cycle machine près entre deux microprocesseurs et de la rétablir lorsqu'elle disparaît, cela, notamment, en vue de permettre à deux microprocesseurs identiques d'exécuter une même tâche de manière indépendante mais en synchronisme total.

Elle a pour objet un procédé de synchronisation de deux microprocesseurs ayant chacun son programme et fonctionnant à partir des microcycles d'une horloge commune. Ce procédé consiste en :

- à munir chaque microprocesseur d'un registre dit de synchronisation qu'il peut armer par l'inscription d'une première valeur et désarmer par l'inscription d'une deuxième valeur et qui peut être lu par l'autre microprocesseur,
- 5 - à commander simultanément les microprocesseurs d'une part pour que l'un d'entre eux dit premier microprocesseur
 - 10 . effectue deux lectures successives du registre de synchronisation de l'autre microprocesseur dit deuxième microprocesseur, ces deux lectures étant espacées d'un nombre entier N_1 de microcycles comptés au moyen de ses cycles machine
 - . arme son registre de synchronisation au bout d'un nombre entier N_2 de microcycles comptés au moyen de ses cycles machine à partir de sa première lecture du registre de synchronisation du deuxième microprocesseur,
 - 15 . désarme son registre de synchronisation au bout d'un nombre entier $N_3 + 1$ de microcycles comptés au moyen de ses cycles machine depuis l'armement effectif de son registre de synchronisation,
 - 20 . répète l'ensemble des trois opérations précédentes au bout d'un nombre entier N_4 de microcycles comptés au moyen de ses cycles machine depuis l'armement effectif de son registre de synchronisation, tant que le registre de synchronisation du deuxième microprocesseur ne reflète pas un état de synchronisation en lui apparaissant constamment armé au cours des deux lectures successives qu'il en effectue
 - 25 . et, une fois que le registre de synchronisation du deuxième microprocesseur lui apparaît constamment armé, effectue son programme au bout d'un nombre entier N_5 de microcycles comptés au moyen de ses cycles machine depuis le désarmement de son registre de synchronisation,
 - 30 et d'autre part pour que l'autre microprocesseur dit deuxième microprocesseur
 - 35 . arme son registre de synchronisation

1. désarme son registre de synchronisation au bout du nombre entier $N_1 + 1$ de microcycles comptés au moyen de ses cycles machine depuis l'armement de son registre de synchronisation,
- 5 2. effectue deux lectures successives du registre de synchronisation du premier microprocesseur au bout du nombre entier N_2 de microcycles comptés au moyen de ses cycles machine depuis l'armement effectif de son registre de synchronisation, ces deux lectures étant espacées du nombre entier N_3 de microcycles comptés au moyen de ces cycles machine
- 10 3. répète l'ensemble de ces trois opérations au bout d'un nombre N_6 de microcycles différent du nombre N_4 , les microcycles étant comptés au moyen de ses cycles machine à partir de sa première lecture, tant que le registre de synchronisation du premier microprocesseur ne reflète pas un état de synchronisation en lui apparaissant constamment armé au cours de ses deux lectures successives
- 15 4. et une fois que le registre de synchronisation du premier microprocesseur lui est apparu constamment armé, effectue son programme au bout d'un nombre entier $N_5 + 1$ de microcycles comptés au moyen de ses cycles machine depuis sa deuxième lecture.
- 20 5. Le nombre N_6 de microcycles au bout duquel le deuxième microprocesseur répète l'armement et le désarmement de son registre de synchronisation ainsi que les deux lectures successives du registre de synchronisation du premier microprocesseur diffère avantageusement d'un microcycle du nombre N_4 de microcycles au bout duquel le premier microprocesseur répète des opérations analogues. Il est choisi de manière préférentielle égal :
- 25 6. - au nombre N_4 diminué d'une unité si le deuxième microprocesseur a vu le registre de synchronisation du premier microprocesseur armé au cours de sa première lecture et désarmé au cours de sa deuxième lecture,
- 30 7. - au nombre N_4 augmenté d'une unité si le deuxième microprocesseur a vu le registre de synchronisation du premier microprocesseur désarmé au cours de sa première lecture et armé au cours de sa deuxième lecture
- 35 8. - et à la somme du nombre N_4 et du nombre N_1 augmentée d'une .

unité si le deuxième microprocesseur a vu le registre de synchronisation du premier microprocesseur désarmé au cours des deux lectures successives qu'il en effectue.

5 Ce procédé de synchronisation peut comporter en outre un test de synchronisation consistant à commander à chaque microprocesseur, après qu'il ait vu le registre de synchronisation de l'autre constamment armé et avant qu'il exécute son programme, une troisième lecture du registre de synchronisation de l'autre microprocesseur au cours de laquelle ce registre doit apparaître désarmé, un résultat 10 inverse conduisant à une reprise du procédé de synchronisation depuis l'origine.

15 Lorsqu'il est utilisé avec des microprocesseurs identiques qui doivent effectuer en parallèle une même tâche à partir de données identiques disponibles sur des entrées communes et à l'aide de programmes distincts mais identiques et qui sont munis de registres de vérification dans lesquelles ils font transiter leurs informations de sortie pour ne les valider à l'aide d'une logique de comparaison câblée qu'en cas d'identité, il est avantageux de compléter le procédé de synchronisation précité en commandant parallèlement à 20 chacun des microprocesseurs, au cours de l'exécution de leur programme, de transférer toutes les données en provenance de l'extérieur dans les registres de vérification avant leur utilisation et de recommencer cette opération tant que la logique câblée de comparaison détecte une différence entre les données contenues dans les 25 registres de vérification.

D'autres caractéristiques et avantages de l'invention ressortiront des revendications jointes et de la description ci-après d'un exemple d'application. Cette description sera faite en regard du dessin dans lequel :

- 30 - la figure 1 représente schématiquement un automatisme à microprocesseur dédoublé utilisant le procédé de synchronisation selon l'invention,
- la figure 2 est un diagramme de temps illustrant un cas de divergence des deux microprocesseurs
- 35 - et les figures 3, 4 et 5 sont des diagrammes illustrant la

chronologie des opérations effectuées par les microprocesseurs en cours de synchronisation.

La figure 1 représente, de manière schématique un automatisme d'action autotesté destiné à la commande d'une installation industrielle. Deux microprocesseurs 100 et 200 identiques reçoivent en parallèle les signaux d'une horloge commune 300 et des informations 1 relatives à la marche de l'installation industrielle et délivre des commandes en direction d'une logique de sécurité 400 qui ne valide une modification des commandes de l'installation qu'en cas d'unanimité des deux microprocesseurs 100, 200.

Chaque microprocesseur 100, 200 comporte, de manière bien connue, une interface d'entrée II par lequel il a accès aux informations relatives à la marche de l'installation industrielle et de l'autre microprocesseur, une interface de sortie OI par lequel il délivre des commandes, une mémoire morte de programme ROM, une mémoire vive de calcul RAM et une unité centrale CPU interconnectés par un bus de données DB et un bus d'adresses AD.

En plus de ces éléments, chaque microprocesseur 100, 200 est muni d'une logique câblée lui permettant de détecter à tout instant une différence entre les commandes qu'il délivre et celles de l'autre microprocesseur, et d'inhiber les commandes de l'autre microprocesseur. Cette logique comprend :

- un comparateur câblé 101, 201 avec deux entrées de comparaison connectées aux sorties des interfaces de sortie OI des deux microprocesseurs 100, 200 et une sortie qui passe au niveau logique 1 en cas de différence entre les commandes délivrées par les deux microprocesseurs 100, 200.

- une mémoire de défaut 102, 202 constituée d'une bascule bistable dont la commande de remise à un est connectée par l'intermédiaire d'une porte logique de type "ou" 103, 203 à trois entrées, à la sortie du comparateur câblé 101, 201, à celle d'un circuit de surveillance d'horloge 500 et à un interface de positionnement 104, 204 du microprocesseur lui-même 100 respectivement 200, dont la commande de remise à zéro est connectée audit interface de positionnement 104, 204 et dont la sortie est connectée d'une part à la

commande de remise à zéro de l'interface de sortie OI de l'autre microprocesseur 200 respectivement 100 de manière à annuler les commandes de ce dernier lorsque la mémoire de défaut est armée au niveau logique 1 et d'autre part à l'interface d'entrée II de 5 l'autre microprocesseur 200 respectivement 100 de manière à lui signifier une remise à zéro éventuelle de son interface de sortie OI,

- ladite porte logique de type "ou" à trois entrées 103, 203,
- ladite interface de positionnement 104, 204 qui permet 10 d'armer ou de désarmer la mémoire de défaut 102, 202 par programmation de l'unité centrale CPU du microprocesseur 100, 200
- et un registre de synchronisation 105, 205 connecté en 15 sortie à l'interface d'entrée II de l'autre microprocesseur 200 respectivement 100 et constitué d'un registre qui peut être adressé et inscrit par le microprocesseur 100 respectivement 200 et lu par l'autre 200 respectivement 100.

La logique de sécurité 400 est formée d'un ensemble de registres de transfert 401 recevant en entrée les commandes fournies par l'interface de sortie OI du microprocesseur 100, délivrant en sortie 20 les commandes 2 nécessaires à la marche de l'installation industrielle et ayant une commande de transfert connectée par l'intermédiaire d'une porte logique de type "ou" 402 à deux entrées connectées aux sorties des mémoires de défaut 102, 202 des microprocesseurs 100, 200 et bloquée par l'armement de l'une quelconque des mémoires 25 de défaut 102, 202. La porte logique de type "ou" 402 commande également une alarme 403.

Les microprocesseurs 100, 200, sont programmés de manière identique en ce qui concerne la conduite de l'installation industrielle, conduite que chacun d'entre eux pourrait assurer seul si ce 30 n'était l'exigence de sécurité. La sécurité de l'automatisme repose sur le fait que deux microprocesseurs identiques programmés de façon identique doivent avoir des réponses identiques à des sollicitations identiques. Toute différence de comportement entre les deux microprocesseurs est, dans cette optique de sécurité, caractéristique 35 d'un état de défaut de l'automatisme et doit aboutir au gel des

5 commandes dans leurs états antérieurs à l'apparition du défaut. Elle est détectée au niveau des interfaces de sorties OI des microprocesseurs 100, 200 par les comparateurs câblés 101, 201. Dès que le comparateur câblé d'un premier microprocesseur détecte une différence entre les commandes du premier et du deuxième microprocesseurs il bloque la logique de sécurité 400 qui maintient les commandes dans leurs états antérieurs, remet à zéro l'interface de sortie OI du deuxième microprocesseur et entraîne, par réaction, la mise à zéro de l'interface de sortie OI du premier microprocesseur ce qui 10 parfait le blocage de l'automatisme.

15 Pour obtenir un fonctionnement correct de l'automatisme en l'absence d'une panne de l'un des microprocesseurs 100, 200, il est nécessaire que les instructions de modification de contexte des deux microprocesseurs 100, 200 se déroulent au même instant ce qui implique une exécution parfaitement synchrone de leurs programmes.

20 Les instructions s'exécuteront simultanément si, d'une part les programmes des deux microprocesseurs sont compatibles en temps d'exécution et si, d'autre part les microprocesseurs démarrent au même instant et interprètent de la même manière les données à traiter.

25 La compatibilité des programmes en temps d'exécution est facile à obtenir en utilisant une horloge 300 commune aux deux microprocesseurs et en donnant à ces derniers des programmes identiques pour la conduite de l'installation industrielle. Le manque de sûreté du à la non redondance de l'horloge 300 est compensé par la présence d'un circuit de surveillance d'horloge 500 qui peut être une bascule monostable constamment armée dont le déclenchement bloque la logique de sécurité 400 par l'intermédiaire des mémoires de défaut 101, 201 et qui est constamment remise à zéro avant son 30 basculement par le signal d'horloge.

35 Le démarrage des microprocesseurs au même microcycle d'horloge et leur interprétation identique des données à traiter ne peuvent par contre pas être assurés dans toutes les éventualités, cela pour diverses raisons, notamment les parasites qui peuvent affecter différemment les deux microprocesseurs 100, 200, les écarts entre les

5 seuils par rapport auxquels les microprocesseurs comparent une même donnée pour en déterminer le niveau logique et les écarts entre les instants d'échantillonnage d'une même donnée au cours d'un même microcycle d'horloge dus aux différences entre les trajets suivis par le signal d'horloge pour parvenir aux unités centrales CPU des deux microprocesseurs.

10 La figure 2 illustre l'une des nombreuses éventualités où une même donnée peut être vue différemment par deux microprocesseurs cadencés par une horloge commune. Une donnée d'asynchrone par rapport à l'horloge passe de la valeur logique 0 à la valeur logique 1 alors qu'elle est en cours de lecture par les deux microprocesseurs. Le premier microprocesseur qui l'échantillonne à un instant t_1 par rapport à son seuil S_1 lit une valeur logique 0 tandis que le deuxième microprocesseur qui l'échantillonne à un instant t_2 par rapport à son seuil S_2 lit une valeur logique 1. Avec des microprocesseurs identiques cadencés par une horloge commune les seuils S_1 et S_2 sont naturellement très proches de même que les temps t_1 et t_2 de sorte qu'une telle éventualité n'a que peu de chance de se produire. Il faut néanmoins s'en prémunir pour ne pas aboutir à des 15 blocages injustifiés de l'automatisme.

20

Il est donc nécessaire de pouvoir récupérer le synchronisme de marche des deux microprocesseurs 100, 200 soit à la mise en route, soit lorsqu'une mémoire de défaut 102, 202 est armée de façon non attendue. Cela s'obtient en amenant les deux microprocesseurs 100, 200 à effectuer des tâches déterminées au cours des mêmes cycles d'horloge et plus précisément en amenant chaque microprocesseur 100, 200 à armer son registre de synchronisation le microcycle d'horloge précédent celui où ce registre est lu une première fois par l'autre microprocesseur et à le désarmer le microcycle suivant celui où il est lu une deuxième fois par l'autre microprocesseur et ensuite, au bout d'un certain nombre de microcycles, en relançant dans chaque microprocesseur le programme de conduite de l'installation industrielle depuis son origine.

35 Pour ce faire, on entame une procédure de synchronisation des deux microprocesseurs au cours de laquelle on commande périodiquement

quement à chaque microprocesseur d'armer son registre de synchronisation pendant un certain nombre de cycles machine puis d'effectuer deux lectures successives du registre de synchronisation de l'autre microprocesseur espacées d'un nombre convenable de cycles machine,
5 l'un des microprocesseurs opérant avec un programme bouclé de longueur fixe, l'autre avec un autre programme bouclé de longueur différente éventuellement variable et plus ou moins long que le programme de longueur fixe selon que le résultat de ses lectures lui montre qu'il est en avance ou en retard sur l'autre microprocesseur.
10 Les déroulements des programmes des deux microprocesseurs 100, 200 lors de leur synchronisation peuvent s'assimiler aux mouvements de deux mobiles parcourant une circonférence, l'un à vitesse constante, l'autre le poursuivant.

La chronologie des opérations en cours de recherche de synchronisme est définie par des nombres de cycles machine c'est-à-dire de cycles élémentaires d'opérations des microprocesseurs qui correspondent en durée aux microcycles de l'horloge mais qui n'ont pas de référence absolue par rapport à ces derniers mais par rapport au programme exécuté, toute instruction prenant pour son exécution un 15 nombre défini de cycles machine.
20

25 L'un des microprocesseurs, par exemple le microprocesseur 100 est alors programmé pour que :

30 - il effectue successivement
25 . deux lectures successives du registre de synchronisation du deuxième microprocesseur 200 espacées d'un nombre entier N_1 de cycles machine,
35 . un armement de son registre de synchronisation 105 (inscription d'un niveau logique 1) au bout d'un nombre entier N_2 de cycles machine comptés depuis sa première lecture du registre de synchronisation 205 du deuxième microprocesseur 200,

35 . un désarmement de son registre de synchronisation 105 (inscription d'un niveau logique 0) au bout d'un nombre entier $N_3 + 1$ de cycles machine comptés depuis l'armement effectif de son registre de synchronisation 105,
35 - il répète l'ensemble des opérations précédentes au bout d'un

nombre entier N_4 de cycles machine comptés depuis l'armement effectif de son registre de synchronisation 105 tant que le registre de synchronisation 205 du deuxième microprocesseur 200 ne lui est pas apparu constamment armé au cours des deux lectures qu'il en effectue

- et, une fois que le registre de synchronisation 205 du deuxième microprocesseur 200 lui est apparu constamment armé, débute, au bout d'un nombre entier N_5 de cycles machine comptés depuis le désarmement de son registre de synchronisation 105, le programme de conduite de l'installation industrielle tandis que l'autre, le microprocesseur 200 dans l'exemple considéré est programmé pour que :

- il effectue successivement

- . un armement de son registre de synchronisation 205 (inscription d'un niveau logique 1),
- . un désarmement de son registre de synchronisation 205 (inscription d'un niveau logique 0) au bout du nombre entier $N_1 + 1$ de cycles machine comptés depuis l'armement effectif de son registre de synchronisation 205,

. deux lectures successives du registre de synchronisation 105 du premier microprocesseur 100 au bout d'un nombre entier N_2 de cycles machine comptés depuis l'armement effectif de son registre de synchronisation, les deux lectures successives étant espacées du nombre entier N_3 de cycles machine,

- il répète l'ensemble des opérations précédentes au bout d'un nombre entier N_6 de cycles machine, différent du nombre entier N_4 , les cycles machine étant comptés à partir de la première lecture effectuée, tant que le registre de synchronisation 105 du premier microprocesseur 100 ne lui apparaît pas constamment armé au cours des deux lectures successives qu'il en effectue

- et, une fois que le registre de synchronisation 105 du premier microprocesseur 100 lui apparaît constamment armé, débute au bout d'un nombre entier $N_5 + 1$ de cycles machine comptés depuis sa deuxième lecture, le programme de conduite de l'installation industrielle.

Le nombre entier N_6 de cycles machine au bout duquel le deuxième microprocesseur 200 répète l'armement et le désarmement de son registre de synchronisation 205 et la double lecture du registre de synchronisation 105 du premier microprocesseur 100 est avantageusement pris égal :

- au nombre entier N_4 diminué d'une unité si le deuxième microprocesseur 200 a vu le registre de synchronisation 105 du premier microprocesseur 100 armé au cours de sa première lecture et désarmé au cours de sa deuxième lecture,

- au nombre entier N_4 augmenté d'une unité si le deuxième microprocesseur 200 a vu le registre de synchronisation 105 du deuxième microprocesseur 100 désarmé au cours de sa première lecture et armé au cours de sa deuxième lecture

- et à la somme des nombres entiers N_1 et N_4 augmentée d'une unité si le deuxième microprocesseur 200 a vu le registre de synchronisation 105 du premier microprocesseur 100 constamment désarmé au cours de ses deux lectures.

Cette dernière solution est représentée dans les figures 3, 4 et 5 qui sont des diagrammes de temps illustrant la chronologie des inscriptions et lectures des registres de synchronisation 105, 205 des microprocesseurs 100, 200 lors de la procédure de synchronisation. Dans ces figures les courbes a représentent le signal de l'horloge 300, les courbes b les instants de lecture par le premier microprocesseur 100 du registre de synchronisation 205 du deuxième microprocesseur 200, les courbes c les états logiques du registre de synchronisation 105 du premier microprocesseur 100, les courbes f les instants de lecture par le deuxième microprocesseur 200 du registre de synchronisation 105 du premier microprocesseur 100 et les courbes g les états logiques du registre de synchronisation 205 du deuxième microprocesseur 200. L'origine des temps est prise, arbitrairement, dans toutes ces figures par rapport au début des opérations de synchronisation effectuées par le premier microprocesseur. L'ensemble des nombres entiers N_1 , N_2 , N_3 , N_4 et N_6 de cycles machine définissant la chronologie des opérations en cours de recherche du synchronisme ont été repris sur ces figures.

La figure 3 est relative au cas où le premier microprocesseur 100 est en avance de deux microcycles d'horloge sur le deuxième microprocesseur 200. Le premier microprocesseur 100 commence par deux lectures successives du registre de synchronisation 205 du deuxième microprocesseur qui sont espacées d'un nombre N_1 de cycles machine et au cours desquelles il ne voit pas ce registre constamment armé. Il arme ensuite son registre de synchronisation 105, N_2 cycles machine après sa première lecture, pendant $N_3 + 1$ cycles machine et réitère ces opérations N_4 cycles machine après le début de l'armement de son registre de synchronisation 105 car il n'a pas vu le registre de synchronisation 205 du deuxième microprocesseur 200 constamment armé. Le deuxième microprocesseur 200 commence par armer son registre de synchronisation 205 avec un retard de deux microcycles d'horloge par rapport à la première lecture faite par le premier microprocesseur et maintient ce registre 205 armé pendant $N_1 + 1$ cycles machine. Il entame ensuite, au bout de N_2 cycles machine après l'armement de son registre de synchronisation 205, deux lectures du registre de synchronisation 105 du premier microprocesseur espacées de N_3 cycles machine et voit ce registre une première fois armé et une seconde fois désarmé ce qui le conduit à réitérer ces opérations un nombre N_6 de cycles machine après sa première lecture pris égal au nombre $N_4 - 1$ afin de diminuer son retard d'un microcycle d'horloge. Au bout de deux itérations les deux microprocesseurs se retrouvent en synchronisme, cas représenté à la figure 5.

La figure 4 est relative au cas où le premier microprocesseur 100 est en retard de deux microcycles d'horloge sur le deuxième microprocesseur 200. Le comportement du premier microprocesseur 100 reste inchangé par rapport à la figure 4. Celui du deuxième est modifié avant la première itération car il a vu au cours des deux premières lectures effectuées, le registre de synchronisation 105 une première fois désarmé et une deuxième fois armé ce qui le conduit à effectuer sa première itération un nombre N_6 de cycles machine après sa première lecture pris égal au nombre $N_4 + 1$ afin de diminuer son avance d'un microcycle d'horloge. Au bout de deux itérations

rations les deux microprocesseurs se retrouvent en synchronisme, cas représenté à la figure 5.

La figure 5 montre que la chronologie définie par l'ensemble des nombres de cycles machines N_1 , N_2 , N_3 implique que lorsque l'un des microprocesseurs voit, au cours de sa double lecture, le registre de synchronisation de l'autre constamment armé, il en est nécessairement de même pour l'autre. Ces nombres de cycles machine sont obtenus en tenant compte des nombres de cycles machine nécessaires aux instructions d'armement, de désarmement et de lecture des registres de synchronisation ainsi qu'aux instructions nécessaires à l'interprétation des lectures des registres et, éventuellement, en programmant des instructions inopérantes dans le cadre des opérations effectuées en cours de recherche du synchronisme.

L'interprétation d'une double lecture d'un registre de synchronisation s'effectue en faisant calculer, par le microprocesseur concerné la somme de la deuxième lecture avec le double de la première lecture et l'analyse de cette somme, un résultat nul impliquant que le microprocesseur a vu le registre de synchronisation constamment désarmé, un résultat égal à un impliquant que le microprocesseur a vu le registre de synchronisation une première fois désarmé et une seconde fois armé, un résultat égal à deux impliquant que le microprocesseur a vu le registre de synchronisation une première fois armé et une seconde fois désarmé et un résultat égal à trois impliquant que le microprocesseur a vu le registre de synchronisation constamment armé.

Pour davantage de sécurité, on peut procéder à un test de synchronisation consistant à commander, à chaque microprocesseur, après qu'il ait vu le registre de synchronisation de l'autre constamment armé au cours des deux lectures successives qu'il en effectue et avant qu'il ne débute le programme de conduite de l'installation industrielle, une troisième lecture du registre de synchronisation de l'autre microprocesseur au cours de laquelle ce dernier registre doit apparaître désarmé, un résultat inverse conduisant pour le microprocesseur concerné à une reprise du programme de synchronisation depuis l'origine. Cette troisième lecture

constitue un test du bon fonctionnement des registres de synchronisation.

Comme on l'a vu précédemment, la désynchronisation de deux microprocesseurs 100, 200 peut se produire à la suite d'un désaccord sur l'interprétation d'une donnée binaire en cours de variation. Comme ce désaccord disparaît dès que la donnée a fini de changer d'état il est avantageux, d'attendre la disparition de ce désaccord pour ne pas avoir à reprendre depuis son origine le programme de conduite de l'installation industrielle. Dans ce but on complète le procédé de synchronisation en imposant, à chaque microprocesseur 100, 200, au cours de l'exécution du programme de conduite de l'installation industrielle, de transférer à son interface de sortie sur des sorties spécifiques ne délivrant pas de commande mais testées par les comparateurs 101, 201, toutes les données en provenance de l'extérieur avant de les utiliser et de recommencer plusieurs fois cette opération tant que les comparateurs 101, 201 détectent une différence avant d'entamer la procédure de synchronisation précédemment décrite. L'interface de positionnement 104, 204 de la mémoire de défaut 102, 202 de chaque microprocesseur 100, 200 est alors utilisé pour désarmer la mémoire de défaut 102, 202 entre chaque tentative. En dehors de ces tentatives l'interface de positionnement 104, 204 sert à des tests de bon fonctionnement de la mémoire de défaut 102, 202.

REVENDICATIONS

- 1/ Procédé de synchronisation de deux microprocesseurs (100, 200) ayant chacun son programme et fonctionnant à partir des microcycles d'une horloge commune (300) définissant la durée de leurs cycles machine caractérisé en ce qu'il consiste :
- 5 - à munir chaque microprocesseur (100, 200) d'un registre dit de synchronisation (105? 205) qu'il peut armer par l'inscription d'une première valeur et désarmé par l'inscription d'une deuxième valeur et qui peut être lu par l'autre microprocesseur (200, 200),
- 10 - à commander simultanément les microprocesseurs (100, 200) d'une part pour que l'un d'entre eux dit premier microprocesseur (100) :
- 15 . effectue deux lectures successives du registre de synchronisation (205) de l'autre microprocesseur (200) dit deuxième microprocesseur, ces deux lectures étant espacées d'un nombre entier N_1 de microcycles comptés au moyen des cycles machine du premier microprocesseur (100),
- 20 . arme son registre de synchronisation (105) au bout d'un nombre entier N_2 de microcycles comptés au moyen de ses cycles machine à partir de sa première lecture du registre de synchronisation (205) du deuxième microprocesseur (200),
- 25 . désarme son registre de synchronisation (105) au bout d'un nombre entier $N_3 + 1$ de microcycles comptés au moyen de ses cycles machine depuis l'armement effectif de son registre de synchronisation (105),
- 30 . répète l'ensemble des trois opérations précédentes au bout d'un nombre entier N_4 de microcycles comptés au moyen de ses cycles machine depuis l'armement effectif de son registre de synchronisation (105), tant que le registre de synchronisation (205) du deuxième microprocesseur (200) ne reflète pas un état de synchronisation en lui apparaissant constamment armé au cours des deux lectures successives qu'il en effectue
- 35 . et, une fois que le registre de synchronisation (205) du deuxième microprocesseur (200) lui apparaît constamment armé, effectue son programme au bout d'un nombre entier N_5 de microcycles comptés au moyen de ses cycles machine depuis le désarmement de son

registre de synchronisation (105),
et d'autre part pour que l'autre microprocesseur dit deuxième
microprocesseur (200)

- arme son registre de synchronisation (205),

5 • désarme son registre de synchronisation (205) au bout du
nombre entier $N_1 + 1$ de microcycles comptés au moyen de ses cycles
machine depuis l'armement de son registre de synchronisation (205),

- effectue deux lectures successives du registre de synchroni-
sation (105) du premier microprocesseur (100) au bout du nombre
10 entier N_2 de microcycles comptés au moyen de ses cycles machine
depuis l'armement effectif de son registre de synchronisation (205),
ces deux lectures étant espacées du nombre entier N_3 de microcycles
comptés au moyen de ses cycles machine

- répète l'ensemble de ces trois opérations au bout d'un
15 nombre N_6 de microcycles différent du nombre N_4 , les microcycles
étant comptés au moyen de ses cycles machine à partir de sa première
lecture, tant que le registre de synchronisation (105) du premier
microprocesseur (100) ne reflète pas un état de synchronisation en
lui apparaissant constamment armé au cours des deux lectures suc-
20 sives qu'il effectue

- et une fois que le registre de synchronisation (105) du
premier microprocesseur (100) lui est apparu constamment armé,
effectue son programme au bout d'un nombre entier $N_5 + 1$ de micro-
cycles comptés au moyen de ses cycles machine depuis sa deuxième
25 lecture.

2/ Procédé selon la revendication 1, caractérisé en ce que le nom-
bre (N_1) de microcycles au bout duquel le deuxième microproces-
seur (200) répète l'armement et le désarmement de son registre de
synchronisation (205) ainsi que les deux lectures successives du
30 registre de synchronisation (105) du premier microprocesseur (100)
diffère d'un microcycle du nombre N_4 de microcycles au bout duquel
le premier microprocesseur (100) répète des opérations analogues.

3/ Procédé selon la revendication 1, caractérisé en ce que le
nombre (N_1) de microcycles au bout duquel le deuxième microproces-
seur (200) répète l'armement et le désarmement de son registre de
35

synchronisation (205) ainsi que les deux lectures successives du registre de synchronisation (205) du premier microprocesseur (100) est égal :

- au nombre N_4 diminué d'une unité si le deuxième microprocesseur (200) a vu le registre de synchronisation (105) du premier microprocesseur (100) armé au cours de sa première lecture et désarmé au cours de sa deuxième lecture,

5 - au nombre N_4 augmenté d'une unité si le deuxième microprocesseur (200) a vu le registre de synchronisation (105) du premier microprocesseur (100) désarmé au cours de sa première lecture et armé au cours de sa deuxième lecture

10 - et à la somme des nombres N_1 et N_4 augmenté d'une unité si le deuxième microprocesseur (200) a vu le registre de synchronisation (105) du premier microprocesseur (100) constamment désarmé au cours de ses deux lectures.

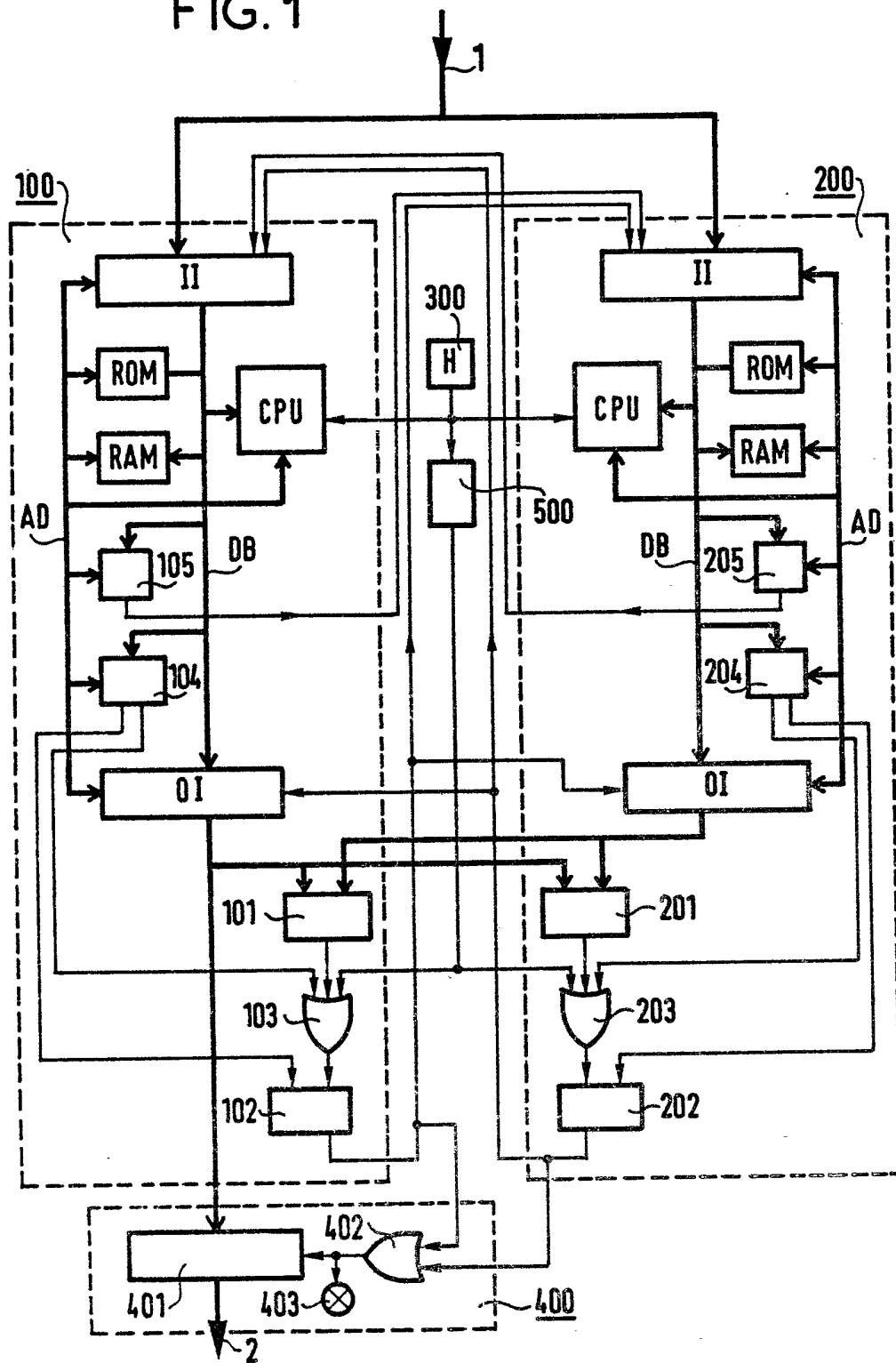
15 4/ Procédé selon la revendication 1, caractérisé en ce qu'il comporte en outre un test de synchronisation consistant à commander à chaque microprocesseur (100, 200), après qu'il ait vu le registre de synchronisation (205, 105) de l'autre (200, 100) constamment armé au cours des deux lectures successives qu'il en effectue et avant qu'il exécute son programme, une troisième lecture du registre de synchronisation (205, 105) de l'autre microprocesseur (200, 100) au cours de laquelle ce dernier registre (205, 105) doit apparaître désarmé, un résultat inverse conduisant, pour le microprocesseur concerné (100, 200) à une reprise du procédé de synchronisation depuis l'origine.

20 5/ Procédé selon la revendication 1 appliqué à des microprocesseur (100, 200) identiques devant effectuer en parallèle une même tâche à partir de données identiques disponibles sur des entrées communes et à l'aide de programmes distincts mais identiques, chacun des microprocesseurs (100, 200) étant muni d'un interface de sortie (OI) dans lequel il inscrit ses informations de sortie et d'un comparateur (101, 201) connecté en entrées aux interfaces de sortie (OI), ce procédé étant caractérisé en ce qu'il consiste en outre, pour maintenir le synchronisme, à commander parallèlement à chacun des micro-

processeurs (100, 200) au cours de l'exécution de leurs programmes, de transférer toutes les données en provenance de l'extérieur dans les interfaces de sortie (OI) avant leur utilisation et à recommencer cette opération tant que les comparateurs (101, 201) 5 détectent une différence entre les données délivrées par les interfaces de sortie (OI)

1/3

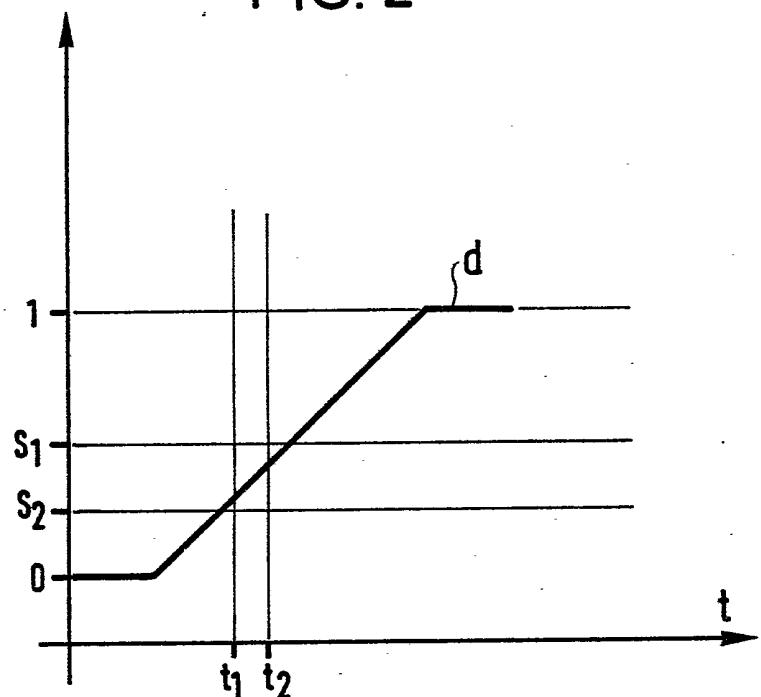
FIG. 1



2513409

2/3

FIG. 2



3/3

