

⑭ **DEMANDE DE BREVET D'INVENTION**

A1

⑮ Date de dépôt : 08.01.93.

⑯ Priorité :

⑰ Date de la mise à disposition du public de la demande : 13.07.94 Bulletin 94/28.

⑱ Liste des documents cités dans le rapport de recherche préliminaire : *Se reporter à la fin du présent fascicule.*

⑲ Références à d'autres documents nationaux apparentés :

⑳ Demandeur(s) : *CEGELEC Société Anonyme — FR.*

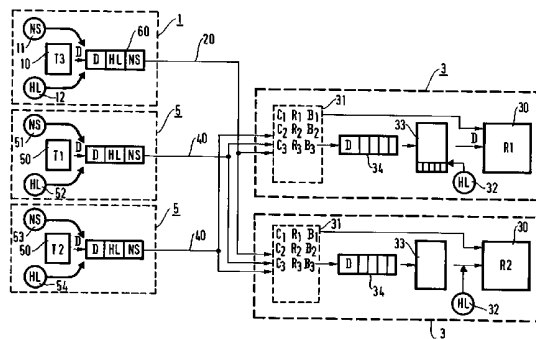
㉑ Inventeur(s) : Herz Dominique.

㉒ Titulaire(s) :

㉓ Mandataire : SOSPI Prugneau Philippe.

⑳ **Système de synchronisation de tâches redondantes.**

㉔ Un système de synchronisation de tâches redondantes pour une installation de contrôle/commande comprenant des interfaces déportées (5) dans lesquelles résident des tâches (50) émettrices de données, un poste de conduite (1) dans lequel réside une tâche (10) émettrice de données et des calculateurs de contrôle d'automatisme (3) dans lesquels résident des tâches redondantes (30) réceptrices desdites données. Le système de synchronisation est agencé pour former des messages (60) en sortie des tâches émettrices de données incluant une estampille temporelle (HL) et un code (NS) identifiant une tâche, et exploiter des constantes de temps préalablement enregistrées et récupérées à partir dudit code pour synchroniser les tâches redondantes de façon à ce qu'elles prennent en compte les données selon une chronologie identique.



FR 2 700 401 - A1



Système de synchronisation de tâches redondantes

L'invention se rapporte à une installation de contrôle/commande, comprenant des premiers processeurs, au moins une paire de seconds processeurs reliés aux premiers processeurs, des tâches émettrices de données et résidant chacune dans un premier processeur, une paire de tâches redondantes réceptrices des mêmes données et résidant chacune dans un second processeur, et un système de synchronisation des tâches redondantes pour assurer que les tâches redondantes prennent en compte lesdites données fournies par les tâches émettrices selon une chronologie identique, le système de synchronisation étant agencé pour:

- encapsuler les données fournies par les tâches émettrices dans des messages incluant chacun une estampille temporelle, les messages étant envoyés vers les seconds processeurs;

- déterminer pour chaque message une date de stabilité du message à partir de l'estampille temporelle du message;

- maintenir en attente, dans chaque second processeur, les messages reçus par ce processeur, détecter des messages stables parmi les messages maintenus en attente, et fournir les données encapsulées dans les messages stables à la tâche redondante dudit second processeur selon un ordre chronologique donné par les estampilles temporelles des messages.

L'invention s'applique plus particulièrement à la synchronisation de tâches redondantes résidant dans des calculateurs de contrôle d'automatisme (niveau 1 du modèle CIM) d'une installation de contrôle/commande industriel.

La redondance active sur les entrées de tâches dupliquées est une technique connue pour mettre en oeuvre des procédures de tolérance aux fautes. Les tâches redondantes prennent en entrée les mêmes données et exécutent le même programme de manière qu'une commutation de la sortie des tâches redondantes est possible en cas de

défaillance d'une des tâches redondantes par exemple. Ces tâches redondantes sont telles qu'elles ne réagissent qu'aux données d'entrée fournies par les tâches émettrices. Elles restent isolées de leur environnement au cours du traitement
5 de ces données. Les messages transmis par un premier processeur ont un ordre d'émission qui reste identique à leur ordre de réception par un second processeur. Pour avoir une redondance active, il est nécessaire que les tâches redondantes aient le même comportement. Compte tenu des
10 hypothèses ci-dessus sur les tâches redondantes, une condition suffisante pour obtenir une équivalence de comportement est que les tâches redondantes prennent les données d'entrée dans le même ordre, c'est-à-dire soient synchronisées.

15 On connaît du document ACM Computing Surveys -Vol. 22 - N°4 - Dec 1990: "Implementing Fault-Tolerant Services Using State Machine Approach: A tutorial" de F. Schneider, un système de synchronisation de tâches redondantes dans lequel l'estampille temporelle incluse dans chaque message
20 est constituée par une date d'envoi du message. Cette date d'envoi est donnée classiquement par une horloge du processeur dans lequel réside la tâche émettrice de données. Les granularités des horloges des processeurs sont suffisamment petites pour qu'en aucun cas, deux messages
25 produits en sortie du même processeur aient la même estampille temporelle. Les seconds processeurs ont eux aussi une horloge locale et les horloges des processeurs sont éventuellement resynchronisées en cas de besoin comme cela est connu pour qu'elles aient la même référence de temps.

30 Le fonctionnement du système de synchronisation de tâches redondantes connu du document précédent est le suivant.

Une constante de temps unique est déterminée au préalable, à partir de temps de transfert de données mesurés
35 entre chaque tâche émettrice et chaque tâche redondante. La

constante de temps est égale au maxima de ces temps de transfert mesurés.

Les messages reçus par un second processeur sont placés dans une file d'attente où ils sont ordonnés selon l'ordre croissant de leur estampille temporelle. Une date de stabilité, pour chaque message est calculée, cette date étant égale au cumul de l'estampille temporelle et de la constante de temps. Un message en entrée d'un processeur est dit stable lorsqu'aucun autre message ayant une estampille inférieure à celle de ce message ne peut plus arriver en entrée du processeur. Par conséquent un message stable est détecté lorsque l'horloge du second processeur donne une date supérieure à la date de stabilité du message. Celui-ci est alors prélevé de la file d'attente et la donnée qu'il contient est fournie à la tâche redondante résidant dans ce second processeur.

Ce processus est effectué de la même façon sur l'autre second processeur.

De cette manière, les tâches redondantes prennent en compte les données qu'elles reçoivent selon une chronologie identique donnée par les estampilles temporelles des messages encapsulant ces données.

Ce système de synchronisation connu présente l'inconvénient suivant.

Sur la figure 1, une installation de contrôle/commande est représentée schématiquement et comprend un poste 1 de conduite (PC) (niveau 2 du modèle CIM) relié, au travers d'un réseau 2 de transmission de messages supportant un protocole de communication par exemple apériodique, à des calculateurs de contrôle d'automatisme 3 (CA1,CA2). Les calculateurs de contrôle d'automatisme 3 sont reliés, au travers d'un réseau 4 de transmission de messages supportant un protocole de communication par exemple périodique, à des interfaces 5 déportées (E/S1,E/S2) qui reçoivent des données ou transmettent des données à un procédé physique 6 à contrôler. Généralement, le processus de traitement de

données reliant les interfaces déportées aux calculateurs de contrôle d'automatisme est temps réel. Les caractéristiques des réseaux 2 et 4 sont telles que les temps de transmission des messages entre les interfaces déportées 5 et les processeurs 3 sont très inférieurs aux temps de transmission des messages entre le poste 1 de conduite et les calculateurs 3.

Du fait que la constante de temps utilisée pour la détermination de la date de stabilité de chaque message est uniforme pour tous les messages, ceux ci sont retardés en moyenne d'une même valeur de temps. Il en résulte qu'un premier message faisant partie d'un processus de traitement temps réel et ayant un temps de transfert petit (message provenant de l'interface déportée) par rapport à celui d'un second message (message provenant du poste de conduite) peut être retardé d'un temps beaucoup plus important que le second message si le second message bien que reçu après le premier message par le calculateur de contrôle d'automatisme, a une estampille plus petite que celle du premier message. Par conséquent, ce système connu de synchronisation de tâches redondantes est susceptible d'entraîner des dépassements de temps de réponse pour le processus temps réel et n'est donc pas adapté à une telle installation de contrôle/commande.

Le but de l'invention est de remédier à cet inconvénient.

A cet effet, l'invention a pour objet une installation de contrôle/commande telle que définie précédemment, caractérisée en ce que le système de synchronisation est en outre agencé pour:

- associer aux messages des codes d'identification identifiant chacun une tâche émettrice de données;
- enregistrer des constantes de temps en correspondance avec lesdits codes d'identification;
- augmenter les estampilles temporelles des messages avec lesdites constantes de temps pour déterminer les dates

de stabilité des messages et définir l'ordre chronologique de prise en compte des données des messages stables par la tâche redondante résidant dans ce processeur de manière à privilégier la prise en compte des données encapsulées dans
5 les messages associés à un ou certains codes d'identification.

Selon un mode de réalisation particulièrement simple, le système de synchronisation est agencé pour inclure dans chaque message un code d'identification identifiant une
10 tâche émettrice.

Le système de synchronisation est agencé pour maintenir en attente les messages reçus par un second processeur dans une file d'attente de messages, ordonner les messages en attente selon l'ordre croissant des estampilles
15 temporelles des messages augmentées chacune d'une première constante de temps C, déterminer les messages stables parmi les messages maintenus en attente à partir des estampilles temporelles des messages augmentées chacune d'une seconde constante de temps R.

20 Avantageusement, l'invention s'étend à un procédé pour déterminer les premières et secondes constantes de temps pour le système de synchronisation ci-dessus dans lequel les tâches émettrices comprennent une ou des tâches prioritaires et une ou des tâches non prioritaires. Ce procédé assure que
25 les temps d'attente, dans la file d'attente, des messages pour les tâches prioritaires sont réduits au maximum. Si il n'y a qu'une seule tâche prioritaire, les messages pour cette tâche ont un temps d'attente nul. Par ailleurs les temps d'attente des messages pour les tâches non
30 prioritaires sont égaux entre eux et réduits au maximum.

Un exemple de réalisation de l'invention est décrit ci-dessous en détail en référence aux dessins.

La figure 1 représente une installation de contrôle/commande.

35 La figure 2 représente la structure d'un message.

La figure 3 représente un enregistrement dans une file d'attente de messages.

La figure 4 représente sous forme de blocs fonctionnels, le système de synchronisation de tâches selon l'invention pour l'installation de contrôle/commande de la figure 1.

La figure 5 est un organigramme illustrant une manière d'implémenter la détection de messages stables pour le système de synchronisation.

10 Sur la figure 4, deux tâches T1,T2 émettrices de données D portant la référence 50 résident dans des interfaces déportées 5. Chaque interface déportée 5 est constitué classiquement d'un processeur avec mémoire et équipement périphérique. Elles comportent chacune une
15 horloge locale 52,54.

Une tâche T1 émettrice de données D portant la référence 10 réside dans le poste de conduite 1. Le poste de conduite est constitué d'un processeur avec mémoire et équipement périphérique. Il comporte aussi une horloge
20 locale 12.

Deux tâches redondantes R1,R2 réceptrices des données D fournies par les tâches T1,T2,T3 portent la référence 30 et résident dans les calculateurs de contrôle d'automatisme 3. Chaque calculateur de contrôle d'automatisme 3 est
25 constitué aussi d'un processeur avec mémoire et périphérique. Ils comportent chacun une horloge locale 32.

Le système de synchronisation des tâches R1 et R2 est maintenant décrit.

Il comporte d'abord des tâches interfaces de sortie
30 (non représentées), réparties dans le poste de conduite 1 et les interfaces déportées 5, qui forment des messages 60 en sorties des tâches émettrices de données 10,50.

Ces tâches interfaces de sortie se couplent aux tâches émettrices par l'intermédiaire de files d'entrée/sortie (non
35 représentées) de données pour assurer le passage des données sur les réseaux 2,4 vers les calculateurs 3. De cette façon,

la structure et le fonctionnement des tâches émettrices sont indépendants de la synchronisation des tâches redondantes.

Comme représenté sur la figure 2, chaque message 60 comprend une zone 61 de donnée dans laquelle est inscrite une donnée D fournie par une tâche émettrice, une zone 62 dans laquelle est inscrite la date HL constituant une estampille temporelle donnée par une horloge locale 12,52,54 au moment de la formation du message et une zone 63 dans laquelle est inscrit un code d'identification de tâche ou de site NS. Chaque tâche interface de sortie code les messages qu'elle forme selon le protocole de communication adéquat pour les envoyer sur les réseaux 20 et 40.

Selon un exemple de réalisation simple, le code d'identification des messages est une valeur numérique. Par exemple, les messages formés en sortie de la tâche T1 ont un code d'identification égal à 1. Les messages formés en sortie de la tâche T2 ont un code d'identification égal à 2 et les messages formés en sortie de la tâche T3 ont un code d'identification égal à 3. De cette manière, les messages 60 peuvent être différenciés facilement en fonction de leurs origines respectives.

Le système de synchronisation des tâches redondantes R1,R2 comprend ensuite des tâches interfaces d'entrée réparties dans les calculateurs 3. Chaque tâche interface d'entrée a pour fonction de saisir les messages 60 reçus par un calculateur 3, retarder éventuellement les messages reçus dans une file d'attente 33 et fournir les données encapsulés dans les messages à la tâche redondante 30 quand les messages sont stables.

Par la suite, le mode de fonctionnement du système de synchronisation sera décrit en faisant référence uniquement à une tâche interface d'entrée pour une tâche redondante 30 résidant dans un calculateur 3.

Cette tâche interface d'entrée exploite des constantes de temps précalculées et enregistrées en mémoire ou calculées en ligne dans le calculateur 3 pour déterminer la

date de stabilité des messages et l'ordre de prise en compte des données fournies à la tâche redondante 30.

Pour déterminer ces constantes de temps, il est nécessaire de mesurer d'abord les temps de transfert minima et maxima d'une donnée D entre chaque tâche émettrice 10,50 et la tâche redondante 30. Dans le cas présent, m_1, m_2, m_3 sont les temps de transfert minima d'une donnée D entre les tâches T1, T2, T3 et la tâche R1. M_1, M_2, M_3 sont les temps de transfert maxima d'une données D entre les tâches T1, T2, T3 et la tâche R1. Un temps de transfert est égal à la différence entre la date locale de réception d'une données par une tâche 30 dans un processeur 3 et la date locale d'émission de cette donnée par une tâche 10,50 dans un processeur 1,5.

Dans l'exemple de réalisation de l'invention, il est souhaitable de privilégier les messages provenant des interfaces déportées 5 par rapport à ceux provenant du poste de conduite 1. Les tâches T1, T2 sont donc des tâches prioritaires.

On détermine ensuite des premières constantes de temps C_1, C_2, C_3 respectivement pour les tâches T1, T2, T3 à partir des relations suivantes, où M est égal à 2 et N est égal à 3.

$$C_j = \text{moy}_k, k=j - \min_{k \rightarrow M+1, N}(\text{moy}_k) - \max_{k \rightarrow 1, M}(\text{diff}_k) - \max_{k \rightarrow M+1, N}(\text{diff}_k) \text{ pour } j \rightarrow 1, M$$

$$25 \quad C_j = \text{moy}_k, k=j - \min_{k \rightarrow M+1, N}(\text{moy}_k) \text{ pour } j \rightarrow M+1, N$$

$$\text{si } \min_{k \rightarrow 1, M}(\text{moy}_k) - \min_{k \rightarrow M+1, N}(\text{moy}_k) - \max_{k \rightarrow 1, M}(\text{diff}_k) - \max_{k \rightarrow M+1, N}(\text{diff}_k) \geq 0$$

et

$$C_j = \text{moy}_k, k=j - \min_{k \rightarrow 1, M}(\text{moy}_k) \text{ pour } j \rightarrow 1, M$$

$$C_j = \text{moy}_k, k=j - \min_{k \rightarrow 1, M}(\text{moy}_k) + \max_{k \rightarrow 1, M}(\text{diff}_k) + \max_{k \rightarrow M+1, N}(\text{diff}_k) \text{ pour } j \rightarrow M+1, N$$

$$30 \quad \text{si } \min_{k \rightarrow 1, M}(\text{moy}_k) - \min_{k \rightarrow M+1, N}(\text{moy}_k) - \max_{k \rightarrow 1, M}(\text{diff}_k) - \max_{k \rightarrow M+1, N}(\text{diff}_k) < 0$$

où

$$\text{moy}_k = (m_k + M_k) / 2 \text{ pour } k \rightarrow 1, N$$

$$\text{diff}_k = (M_k - m_k) / 2 \text{ pour } k \rightarrow 1, N.$$

On détermine ensuite des secondes constantes de temps
 5 R_1, R_2, R_3 respectivement pour les tâches T_1, T_2, T_3 à partir de
 la relation suivante, dans laquelle les indices i et j vont
 de 1 à 3.

$$R_i = \max(M_j - C_j) + C_i, j \neq i$$

De préférence, les constantes de temps C_1, C_2, C_3 et
 10 R_1, R_2, R_3 sont préalablement enregistrées dans le calculateur
 3 pour être récupérées sélectivement à partir d'un code
 d'identification. En variante, les valeurs m_k et M_k des
 temps de transfert mesurés sont enregistrées dans le
 calculateur 3 pour être récupérées sélectivement à partir
 15 d'un code d'identification, le calcul des constantes de
 temps C et R se faisant en ligne.

En réponse aux messages 60 reçus en entrée du
 calculateur 3, la tâche interface d'entrée décode ces
 messages et forme des enregistrements 34 qui sont maintenus
 20 en attente dans une file d'attente 33. Chaque enregistrement
 34, comme visible en figure 3, comporte un champ 35 dans
 lequel est inscrit la donnée D d'un message, un champ 36
 dans lequel est inscrite l'estampille HL du message
 augmentée de la constante C_j dont l'indice j correspond ou
 25 est égal au code d'identification NS du message, un champ 37
 dans lequel est inscrit le code d'identification NS du
 message, un champ 38 dans lequel est inscrite la valeur
 d'estampille HL du message augmentée de la constante de
 temps R_j dont l'indice j correspond ou est égal au code
 30 d'identification NS du message, et un champ 39 dans lequel
 est inscrite une variable booléenne de contrôle F
 positionnée à "Faux".

La valeur du champ 38 d'un enregistrement correspond à
 la date de stabilité d'un message.

35 Par ailleurs des constantes booléennes B sont
 enregistrées dans le calculateur 3 en correspondance avec

les codes d'identification. Pour un code d'identification i , la valeur de la constante booléenne est "Vrai" si:

$$\max(M_j - C_j) - (m_i - C_i) > 0 \text{ pour } j \neq i$$

Dans ce cas, l'expression $\max(M_j - C_j) - (m_i - C_i)$ 5 donne le temps d'attente maximal d'une donnée d'un message en attente d'être stable. Dans le cas contraire, ce temps d'attente est nul.

Ces constantes booléennes seront exploités pour éviter de maintenir en attente les messages qui sont stables 10 en entrée du processeur 3.

Le fonctionnement de la tâche interface d'entrée est maintenant décrit en référence à la figure 5.

Les messages 60 transitant par les réseaux 2 et 4 sont reçus en séquence en entrée du calculateur de contrôle 15 d'automatisme 3 par la tâche interface d'entrée.

Un message 60 reçu est décodé en 100 pour récupérer la donnée D , l'estampille HL et le code d'identification NS du message.

A partir du code d'identification NS , les constantes 20 de temps C_j, R_j et la constante booléenne B_j dont l'indice j correspond au code d'identification NS sont récupérées en 110.

La constante booléenne B_j est comparée à la valeur "Vrai" en 120. Si elle n'est pas égale à "Vrai", la donnée D 25 du message reçu est fournie directement à la tâche redondante $R1$ en 130 puisque ce message est stable. Dans le cas contraire, un enregistrement 34 est formé à partir du message 60 en 140 et est placé dans la file d'attente 33 de façon à ce que les enregistrements 34 soient ordonnés selon 30 l'ordre croissant des valeurs de champ 36. Si deux enregistrements consécutifs dans la file d'attente 33 ont la même valeur de champ 36, celui ayant la plus faible valeur de champ 37 précède l'autre dans la file d'attente.

Après les étapes 130 et 140, les enregistrements 34 35 dans la file d'attente 33 sont balayés en 150. Pour chaque enregistrement 34 balayé, la variable de contrôle F dans le

champ 39 est comparée à la valeur "Vrai" en 160. Si cette variable de contrôle est égale à "Vrai", le traitement continue en 150 pour l'enregistrement 34 suivant jusqu'à ce que tous les enregistrements aient été balayés. Dans le cas contraire, la date de stabilité dans le champ 38 pour l'enregistrement balayé est comparée en 170 à la date actuelle fournie par l'horloge locale 32 du calculateur de contrôle d'automatisme 3. Si cette date de stabilité est inférieure à la date actuelle, la variable de contrôle F du champ 39 est positionnée à "Vrai" en 180 et le traitement continue en 150 et sinon en 150. Quand tous les enregistrements 34 ont été balayés dans la file d'attente 33, le traitement se poursuit en 190. Dans cette étape, si le premier enregistrement 34 de la file d'attente 33 (celui ayant un champ 36 de plus faible valeur) a une variable de contrôle F égale à "Vrai", une première donnée D dans le champ 35 de cet enregistrement est fournie à la tâche R1. L'enregistrement est simultanément enlevé de la file d'attente qui est réorganisée en conséquence. Si de nouveau la variable de contrôle F du premier enregistrement de la file d'attente est égale à "Vrai", une seconde donnée D dans le champ 35 de cet enregistrement est fournie à la tâche R1 qui la prendra en compte après la première donnée D. Cette procédure est répétée jusqu'à ce que la variable F soit égale à "Faux", auquel cas le traitement est terminé.

Les étapes 100 à 190 sont répétées pour chaque nouveau message 63 reçu en entrée pour ce calculateur de contrôle d'automatisme. La tâche interface d'entrée ci-dessus décrite fonctionne de la même façon pour l'autre tâche redondante résidant dans l'autre calculateur de contrôle d'automatisme, les constantes de temps devant seulement être adaptées aux temps de transfert de données pour ce calculateur. L'invention peut par conséquent s'appliquer facilement à un nombre de calculateurs de contrôle d'automatisme et de tâches redondantes supérieur à 2.

REVENDICATIONS

1.) Une installation de contrôle/commande, comprenant des premiers processeurs (1,5), au moins une paire de
5 seconds processeurs (3) reliés aux premiers processeurs, des tâches émettrices de données (10,50) et résidant chacune dans un premier processeur, une paire de tâches redondantes (30) réceptrices des mêmes données et résidant chacune dans un second processeur, et un système de synchronisation des
10 tâches redondantes pour assurer que les tâches redondantes prennent en compte lesdites données fournies par les tâches émettrices selon une chronologie identique, le système de synchronisation étant agencé pour:

- encapsuler les données fournies par les tâches
15 émettrices dans des messages (60) incluant chacun une estampille temporelle (62), les messages étant envoyés vers les seconds processeurs;

- déterminer pour chaque message une date de stabilité du message à partir de l'estampille du message;

20 - maintenir en attente, dans chaque second processeur, les messages reçus par ce processeur, détecter des messages stables parmi les messages maintenus en attente, et fournir les données encapsulées dans les messages stables à la tâche redondante dudit second processeur selon un ordre
25 chronologique donné par les estampilles temporelles des messages,

c a r a c t é r i s é en ce que le système de synchronisation étant en outre agencé pour:

- associer aux messages des codes d'identification
30 (63) identifiant chacun une tâche émettrice de données;

- enregistrer des constantes de temps en correspondance avec lesdits codes d'identification;

- augmenter les estampilles temporelles des messages avec lesdites constantes de temps pour déterminer les dates
35 de stabilité des messages et définir l'ordre chronologique de prise en compte des données des messages stables par la

tâche redondante résidant dans ce processeur de manière à privilégier la prise en compte des données encapsulées dans les messages associés à un ou certains codes d'identification.

5

2.) Le système selon la revendication 1, dans lequel le système de synchronisation est agencé pour inclure dans chaque message un code d'identification identifiant une tâche émettrice.

10

3.) Le système selon l'une des revendications 1 ou 2, dans lequel le système de synchronisation est agencé pour maintenir en attente les messages reçus par un second processeur dans une file d'attente de messages (33),
 15 ordonner les messages en attente selon l'ordre croissant des estampilles temporelles des messages augmentées chacune d'une première constante de temps C, déterminer les messages stables parmi les messages maintenus en attente à partir des estampilles temporelles des messages augmentées chacune
 20 d'une seconde constante de temps R.

4.) Un procédé pour déterminer les premières et secondes constantes de temps pour une installation de contrôle/commande selon la revendication 3, dans laquelle
 25 les tâches émettrices comprennent une ou plusieurs tâches émettrices prioritaires T_1, T_2, \dots, T_M et une ou plusieurs tâches non prioritaires T_{M+1}, \dots, T_N , le procédé consistant à:

- mesurer les temps de transfert minima m_k et maxima M_k d'une donnée entre chaque tâche émettrice T_k et une tâche
 30 redondante,

- calculer les premières constantes C_j , l'indice j allant de 1 à N , à partir des relations suivantes:

$$C_j = \text{moy}_{k, k=j} - \min_{k \rightarrow M+1, N}(\text{moy}_k) - \max_{k \rightarrow 1, M}(\text{diff}_k) - \max_{k \rightarrow M+1, N}(\text{diff}_k) \text{ pour } j \rightarrow 1, M$$

$$C_j = \text{moy}_{k, k=j} - \min_{k \rightarrow M+1, N}(\text{moy}_k) \text{ pour } j \rightarrow M+1, N$$

si $\min_{k \rightarrow 1, M}(\text{moy}_k) - \min_{k \rightarrow M+1, N}(\text{moy}_k) - \max_{k \rightarrow 1, M}(\text{diff}_k) - \max_{k \rightarrow M+1, N}(\text{diff}_k) \geq 0$

et

$C_j = \text{moy}_{k, k=j} - \min_{k \rightarrow 1, M}(\text{moy}_k)$ pour $j \rightarrow 1, M$

$C_j = \text{moy}_{k, k=j} - \min_{k \rightarrow 1, M}(\text{moy}_k) + \max_{k \rightarrow 1, M}(\text{diff}_k) + \max_{k \rightarrow M+1, N}(\text{diff}_k)$ pour $j \rightarrow M+1, N$

5 si $\min_{k \rightarrow 1, M}(\text{moy}_k) - \min_{k \rightarrow M+1, N}(\text{moy}_k) - \max_{k \rightarrow 1, M}(\text{diff}_k) - \max_{k \rightarrow M+1, N}(\text{diff}_k) < 0$

où

$\text{moy}_k = (m_k + M_k) / 2$ pour $k \rightarrow 1, N$

$\text{diff}_k = (M_k - m_k) / 2$ pour $k \rightarrow 1, N$

- et calculer les secondes constantes de temps R_j ,

10 l'indice j allant de 1 à N , à partir de la relation suivante:

$R_j = \max (M_k - C_k) + C_j$ pour $k \neq j$

1/3

FIG.1

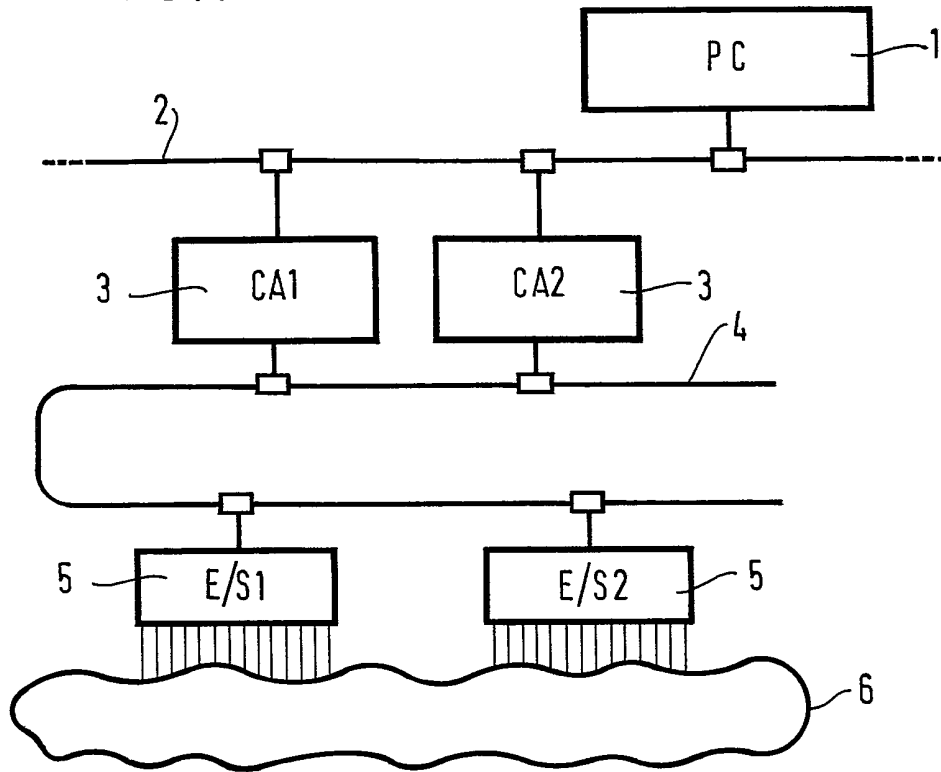


FIG.2

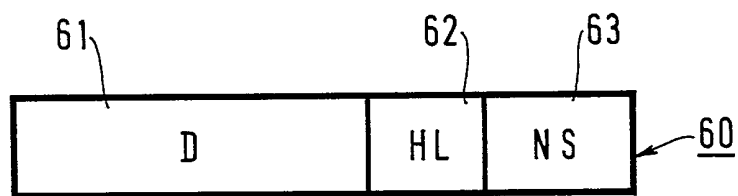


FIG.3

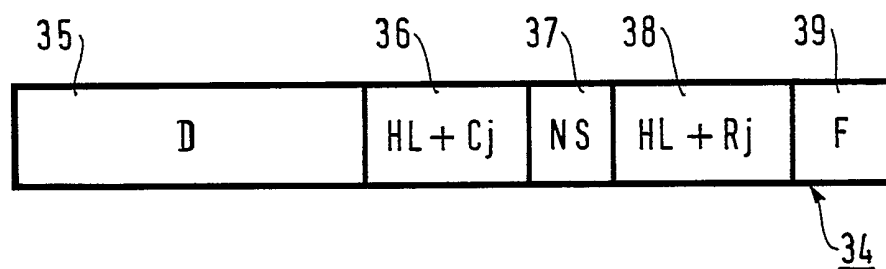


FIG. 4

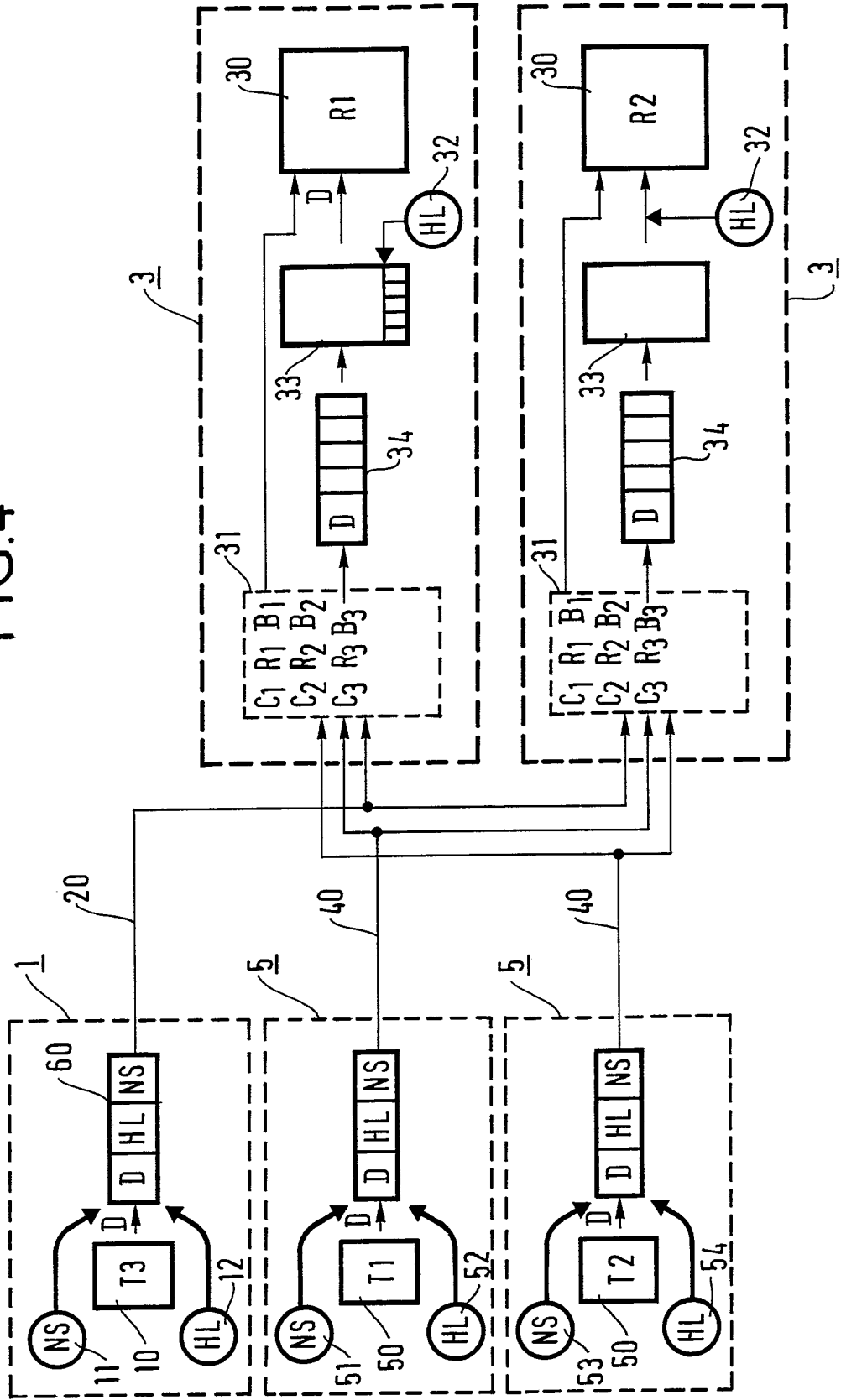
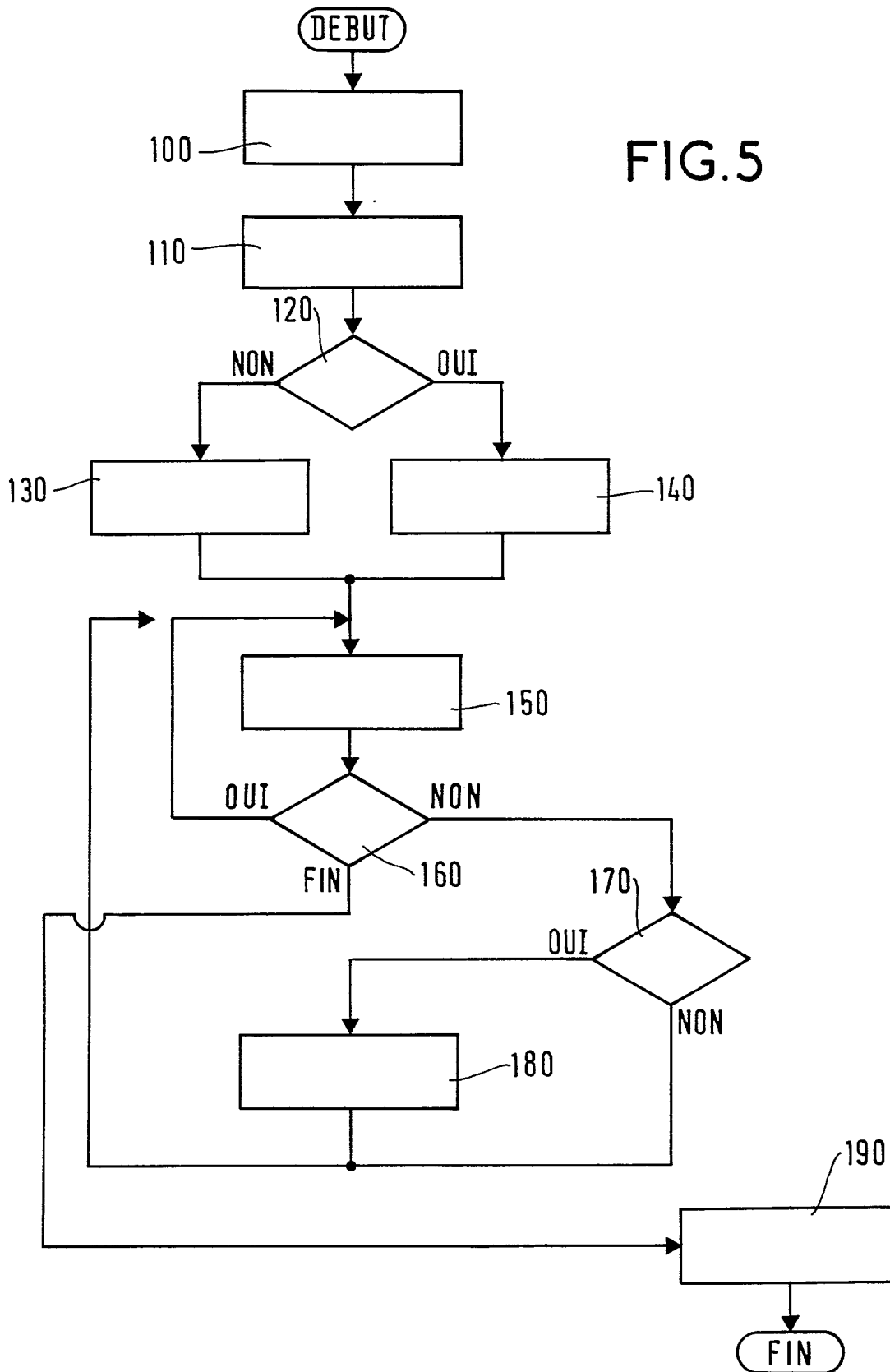


FIG.5



INSTITUT NATIONAL
de la
PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE PRELIMINAIRE
établi sur la base des dernières revendications
déposées avant le commencement de la recherche

FR 9300147
FA 480666

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
A	US-A-5 168 443 (MCLAUGHLIN ET AL.) * colonne 4, ligne 17 - ligne 32 * * colonne 8, ligne 23 - colonne 9, ligne 44 * * figures 2,6 *	1
A	COMPUTER vol. 23, no. 10, Octobre 1990, LOS ALAMITOS US pages 33 - 42 K. G. SHIN ET AL. 'Fault-tolerant clock synchronization in distributed systems' * page 36, colonne 2, ligne 23 - colonne 3, ligne 7 *	1
A	EP-A-0 445 954 (IBM) * colonne 3, ligne 43 - colonne 5, ligne 6 * * figures 1,2 *	1
		DOMAINES TECHNIQUES RECHERCHES (Int. Cl.5)
		G06F G05B
Date d'achèvement de la recherche 22 SEPTEMBRE 1993		Examineur MASCHE C.
<p>CATEGORIE DES DOCUMENTS CITES</p> <p>X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinent à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire</p> <p>T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons * : membre de la même famille, document correspondant</p>		

1

EPO FORM 1503 03.82 (FCM13)