

①⑨ RÉPUBLIQUE FRANÇAISE
—
**INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE**
—
COURBEVOIE
—

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

3 093 579

②① N° d'enregistrement national : **19 02266**

⑤① Int Cl⁸ : **G 06 F 9/48 (2019.01)**

①②

BREVET D'INVENTION

B1

⑤④ ORDONNANCEUR D'ÉVÉNEMENTS POUR MICROPROCESSEUR.

②② Date de dépôt : 07.03.19.

③③ Priorité :

④③ Date de mise à la disposition du public
de la demande : 11.09.20 Bulletin 20/37.

④⑤ Date de la mise à disposition du public du
brevet d'invention : 30.07.21 Bulletin 21/30.

⑤⑥ Liste des documents cités dans le rapport de
recherche :

Se reporter à la fin du présent fascicule

⑥⑥ Références à d'autres documents nationaux
apparentés :

○ Demande(s) d'extension :

⑦① Demandeur(s) : *THALES Société anonyme* — FR.

⑦② Inventeur(s) : ANDRADE ALFONSECA Damian,
TEIXEIRA Tony et GUGUEN Stéphane.

⑦③ Titulaire(s) : *THALES Société anonyme*.

⑦④ Mandataire(s) : *MARKS & CLERK FRANCE*.

FR 3 093 579 - B1



Description

Titre de l'invention : Ordonnanceur d'évènements pour micro- processeur.

- [0001] L'invention concerne un ordonnanceur d'évènements ou de tâches ou « scheduler » d'un système d'exploitation ou OS (Operating System).
- [0002] Les systèmes d'exploitation gèrent les priorités et les interruptions des tâches. Le temps d'occupation du microprocesseur est utilisé pour faire des sauvegardes de contexte lorsqu'une interruption arrive. De ce fait, le temps d'occupation du microprocesseur ne peut pas être utilisé à 100%, car des marges doivent être prises pour le système d'exploitation.
- [0003] Lorsqu'un bus de communication est partagé par plusieurs équipements, une marge pour le remplissage de la bande passante est prise afin de gérer d'éventuelles collisions entre les tâches. La capacité de traitement d'informations par un bus n'est donc pas exploitée dans sa totalité.
- [0004] Pour augmenter le temps de calcul utile d'un microprocesseur, i.e., temps disponible uniquement pour le calcul, une solution est d'utiliser des dispositifs pour fonctionner plus rapidement afin de pouvoir tenir le temps réel avec le système d'exploitation et les fonctions de l'utilisateur.
- [0005] Au niveau des communications entre plusieurs équipements, l'art antérieur a donc tendance à augmenter le nombre de connexions ou encore à augmenter la bande passante afin de garantir le traitement de toutes les demandes par le bus.
- [0006] Les solutions connues par le demandeur conduisent donc à l'utilisation de composants plus performants, plus coûteux et à une augmentation de connexions dans le cas d'une communication.
- [0007] Sur un microprocesseur, lorsque de nombreuses tâches doivent être exécutées simultanément, un système d'exploitation doit être implémenté pour gérer l'accès aux unités de calcul, car les opérations se font en série. Si les contraintes temps réel sont importantes, la vitesse de calcul doit être incrémentée pour arriver à exécuter toutes les opérations nécessaires et pour assurer la gestion des entrées et des sorties au niveau du microprocesseur. La gestion des entrées et des sorties, l'interruption de tâches avec leur sauvegarde de contexte et d'autres opérations font qu'il n'est pas possible d'utiliser de manière optimale les capacités de calcul du microprocesseur. Sur un composant programmable de type FPGA (Field Programmation Gate Array), il est possible d'implémenter de manière simultanée plusieurs opérations et d'optimiser ainsi les calculs. La gestion des entrées/sorties peut aussi se faire sans contraindre les calculs. Le débogage des algorithmes est difficile à réaliser et l'implémentation des al-

algorithmes nécessite des ressources importantes ce qui conduit à une augmentation de la taille et du coût d'un FPGA.

[0008] L'idée de la présente demande de brevet repose notamment sur une pré planification des évènements au niveau d'un microprocesseur et les communications de manière déterministe, sans avoir recours à une gestion des priorités des interruptions. Les évènements sont exécutés dans un ordre précis de façon à ne jamais avoir deux évènements qui s'exécutent à la fois et sans interrompre l'exécution d'un évènement.

[0009] Dans la suite de la description, le mot « évènement » désigne indifféremment des tâches à exécuter par un processeur ou un système d'exploitation, des messages, ou tout autre évènement ou toute autre action exécuté par un microprocesseur.

[0010] L'invention concerne un procédé pour répartir temporellement les évènements à transmettre dans une fenêtre temporelle initiale $T_i=[t_0, t_{max}]$, lesdits évènements étant exécutés par un premier microprocesseur, la répartition temporelle étant réalisée par un dispositif programmable caractérisé en ce qu'il comporte au moins les étapes suivantes :

- identifier au moins deux évènements E_1, E_j , à exécuter par le premier microprocesseur,
- identifier pour chaque évènement E_1, E_j , sa fréquence d'exécution F_1, F_j , selon des fréquences multiples de deux, la durée T_1, T_j , d'exécution d'un évènement,
- insérer un premier évènement E_1 et calculer la disponibilité temporelle T_r restante dans la fenêtre temporelle initiale T_i afin de déterminer la taille maximale en temps et la fréquence maximale admissible d'un futur évènement E_j ,
- à partir de la disponibilité temporelle, de la taille d'un évènement et de sa fréquence admissible, calculer pour chaque futur évènement E_j à exécuter et pour chacune des n occurrences dudit évènement dans la fenêtre temporelle initiale, une valeur de décalage temporel dt tels que $t_0+dt_1(E_j), t_0+dt_2(E_j), t_0+dt_3(E_j), \dots, t_0+dt_n(E_j)$ correspondant au positionnement temporel du futur évènement E_j par rapport à l'instant de démarrage t_0 dans la fenêtre temporelle initiale,
- exécuter l'évènement E_j et vérifier si le temps d'exécution de l'évènement E_j par le microprocesseur, T_{jmicro} est conforme au temps d'exécution T_{jFPGA} attendu dans le FPGA.

[0011] Le procédé peut tenir compte de la priorité associée aux évènements à exécuter dans une fenêtre temporelle pour déterminer les instants d'exécution de chacun de ces évènements.

[0012] L'invention concerne aussi un système pour répartir temporellement les évènements à transmettre dans une fenêtre temporelle initiale $T_i=[t_0, t_{max}]$ caractérisé en ce qu'il comprend au moins les éléments suivants :

- un dispositif comprenant une mémoire partagée et un ordonnanceur d'évènements,

- ledit ordonnanceur est adapté à :

déterminer les instants d'exécution d'au moins deux évènements dans la fenêtre temporelle initiale $T_i=[t_0, t_{max}]$, en prenant en compte la disponibilité temporelle T_r restante dans la fenêtre temporelle initiale T_i après l'insertion d'un premier évènement, à partir de la disponibilité temporelle, de la taille d'un futur évènement et de sa fréquence admissible, à calculer pour chaque futur évènement E_2 à exécuter une valeur de décalage temporel dt correspondant au positionnement temporel du futur évènement E_2 par rapport à l'instant de démarrage t_0 dans la fenêtre temporelle initiale et à vérifier si le temps d'exécution de l'évènement E_2 par le microprocesseur, T_{2micro} est conforme au temps d'exécution T_{2FPGA} ,

- exécuter l'évènement et vérifier si le temps d'exécution de l'évènement est conforme au temps d'exécution attendu,

- un premier microprocesseur configuré pour exécuter les évènements mémorisés dans une mémoire partagée, sur réception d'un ordre d'interruption INT émis par ledit dispositif et pour transmettre le temps d'exécution d'un évènement à l'ordonnanceur.

[0013] L'ordonnanceur est, par exemple, configuré pour calculer la disponibilité temporelle restante dans chacune des n fenêtres générées par le positionnement de l'évènement exécuté n fois dans la fenêtre temporelle initiale.

[0014] Les dessins annexés illustrent l'invention :

[0015] [fig.1] un schéma d'une architecture parallèle pour la mise en œuvre de l'invention,

[0016] [fig.2] un schéma illustrant les échanges entre les éléments de la figure 1,

[0017] [fig.3] illustre l'étape de configuration des tâches et des messages de communication,

[0018] [fig.4] illustre l'étape de planification suivant l'étape de configuration,

[0019] [fig.5] illustre un système mettant en œuvre le procédé selon l'invention.

[0020] Le principe mis en œuvre dans le procédé et le système selon l'invention reposent notamment sur une gestion des évènements traités par un microprocesseur et des communications de manière déterministe, sans avoir recours à une gestion des priorités des interruptions, en fonctionnement normal. Les tâches vont être exécutées dans un ordre précis, de façon à ne jamais avoir deux évènements qui s'exécutent à la fois et sans interrompre l'exécution d'un évènement. La communication et la gestion des évènements ou tâches sont organisées temporellement de manière à faire interagir tous les équipements du réseau de manière synchronisée et sans aucune interruption.

[0021] La figure 1 est un schéma représentant une architecture de bus parallèle et les échanges entre un premier microprocesseur 1 et le FPGA 2.

[0022] Le microprocesseur 1 comprend un programme 4 et une mémoire partagée 3 avec une mémoire du FPGA. Ces éléments sont connus de l'homme du métier et ne seront pas détaillés, le partage des mémoires se faisant selon un principe connu de l'homme du métier. Le microprocesseur comprend un timer qui est activé lorsqu'une in-

terruption d'exécution d'un évènement ou d'une tâche s'active. Lorsque l'exécution de la tâche est terminée, le timer s'arrête et le microprocesseur transmet le temps d'exécution de la tâche au FPGA qui va alors arrêter son timer, avant d'effectuer la vérification de temps d'exécution d'un évènement. L'enchaînement de ces étapes est illustré à la figure 2.

[0023] Le dispositif programmable FPGA 2 comprend une mémoire partagée 5, un gestionnaire de données et de communication 6 et un ordonnanceur 7 objet de l'invention. Le gestionnaire de données et de communication échange des informations (des données) avec la mémoire partagée du microprocesseur. L'ordonnanceur 7 gère les interruptions au niveau du microprocesseur et va permettre l'exécution des évènements de façon déterministe et un par un comme il a été indiqué précédemment. C'est l'ordonnanceur qui détermine l'instant de positionnement d'un évènement à exécuter et qui va calculer comment positionner temporellement les différents évènements à exécuter dans la fenêtre temporelle. Le gestionnaire de données et de communication est par exemple un bus de communication partagé par plusieurs équipements.

[0024] La figure 2 illustre des échanges entre le microprocesseur et le FPGA.

[0025] Le gestionnaire de données et de communication du FPGA transmet un ordre d'écriture de données du FPGA vers la mémoire partagée 3 du microprocesseur 1, 21. En même temps, l'ordonnanceur 7 transmet un signal d'interruption INT vers le programme 4 implémenté sur le microprocesseur 1, 22. Le microprocesseur sur réception du signal d'interruption lit les données qui se trouvent dans sa mémoire partagée 3, 23. Les données lues sont transmises au programme 4 qui va sélectionner, 24, une application à exécuter selon le contenu des données et exécuter, 25, l'application sélectionnée. L'exécution de l'application génère des données qui vont être écrites dans la mémoire partagée 3 et qui vont ensuite être transmises dans la mémoire partagée 5 du FPGA. L'ordonnanceur 7 connaissant la durée d'exécution d'un évènement et son positionnement temporel dans une fenêtre temporelle va vérifier le temps d'exécution, 27, puis lire les données, 28.

[0026] La figure 3 illustre l'étape de planification réalisée par l'ordonnanceur selon l'invention, des tâches et des messages de communication à transmettre, dans l'exemple, trois tâches et deux messages. Les tâches et les messages à envoyer sont identifiés. Par défaut, les tâches et les messages sont empilés les uns derrière les autres au niveau d'une pile, non représentée sur la figure pour des raisons de simplification, selon la disponibilité temporelle restante dans une fenêtre temporelle. Pour chaque message et pour chaque tâche, le microprocesseur identifie sa fréquence d'exécution, selon des fréquences multiples de deux, la durée d'exécution T_t de la tâche ou la durée T_m du message (espace temporel pris par la tâche ou le message), et éventuellement les messages et/ou les tâches qui précèdent. C'est l'évènement le plus lent qui détermine la

vitesse de répétition du cycle, correspondant à une fenêtre temporelle. A chaque interruption d'exécution d'un évènement, le timer T_{micro} du microprocesseur 1 est activé, en même temps le timer T_{FPGA} du FPGA est activé. Ces deux timers permettent notamment de vérifier le temps d'exécution d'un évènement, au niveau du microprocesseur et dans le FPGA. A la fin de l'exécution de l'évènement, le timer du microprocesseur s'arrête et transmet le temps mesuré au FPGA, qui sur réception de cette information arrête son timer.

[0027] Après insertion d'un nouvel évènement (tâche ou message), l'ordonnanceur recalcule la disponibilité temporelle d'utilisation des ressources du microprocesseur, dans la fenêtre temporelle. L'ordonnanceur va recalculer après chaque positionnement d'un évènement à exécuter, en tenant compte de la « taille temporelle d'exécution » de l'évènement, de l'intervalle de temps restant dans la fenêtre temporelle et disponible pour que le microprocesseur puisse exécuter d'autres évènements. Dans le cas où un évènement est exécuté n fois dans la fenêtre temporelle, l'ordonnanceur va effectuer le calcul de l'intervalle restant dans la fenêtre temporelle pour chaque occurrence de l'évènement. L'ordonnanceur va calculer l'intervalle de temps restant pour les n fenêtres temporelles ainsi générées dans la fenêtre temporelle (T_0 à T_{max}), ceci est par exemple illustré avec le quatrième évènement exécuté quatre fois dans la fenêtre temporelle initiale $[T_0, T_{\text{max}}]$ et donc conduira à quatre valeurs de décalage temporels à utiliser pour l'exécution des évènements, ce qui est illustré sur la figure par $t_0+dt_1(E_4)$, $t_0+dt_2(E_4)$, $t_0+dt_3(E_4)$, $t_0+dt_4(E_4)$. De manière générique pour un évènement j , avec un nombre d'occurrence n dans la fenêtre temporelle, si les conditions de disponibilité temporelle sont remplies, on aura n instants temporels de positionnement de l'évènement j , tels que $t_0+dt_1(E_j)$, $t_0+dt_2(E_j)$, $t_0+dt_3(E_j)$, ... $t_0+dt_n(E_j)$.

[0028] Sur l'exemple de la figure 3, les messages et les tâches sont programmés selon un ordre d'exécution défini au préalable par un utilisateur ou une application donnée, l'ordre d'exécution est garanti si le champ « priorité ou previous », i.e. tâche ou message précédent est renseigné.

[0029] A partir des informations données lors de l'étape de planification, l'ordonnanceur va définir les instants d'exécution des tâches et/ou des évènements. Pour cela, l'ordonnanceur prend en compte la valeur de fréquence minimum f_{min} de l'ensemble des tâches et messages considérés. Cette fréquence (f_{min}) détermine le cycle de répétition des évènements ainsi que la fenêtre temporelle maximale (T_0 à T_{max}), T_0 est l'instant où le processeur commence à exécuter un évènement, T_{max} la borne supérieure. Puis il détermine une valeur de décalage temporel (t_x) ou offset pour un évènement (tâche ou message) qui doit être exécuté ($T_0 + t_x$). Sur la figure 4, les décalages temporels ou offsets conduisent à un ordre d'exécution avec : le premier message transmis à $t_0=0$, le cinquième évènement exécuté à $t=t_0+100\text{ms}$, le deuxième

évènement exécuté à $t = t_0 + 125\text{ms}$, le troisième évènement exécuté à $t = t_0 + 175\text{ms}$, le quatrième message est transmis à $t = t_0 + 200\text{ms}$. Sur chaque nouveau cycle, les offsets temporels se répètent.

- [0030] Pour le calcul de la fenêtre temporelle restante, après avoir positionné le quatrième évènement par exemple, ce dernier étant exécuté n fois dans la fenêtre temporelle, l'ordonnanceur 7 calcule l'intervalle restant pour les fenêtres temporelles générées à chaque positionnement d'une occurrence du quatrième évènement, $[t = (t_0 + 200\text{ms} + \text{taille en ms de l'évènement}) + (t_0 + n * 200\text{ms})]$ où n correspond au nombre de fois où le quatrième évènement apparaît dans la fenêtre temporelle.
- [0031] Le procédé selon l'invention garantit ainsi que tous les évènements planifiés seront exécutés un par un, sans jamais se croiser temporellement et en tenant compte de leur fréquence d'exécution. L'ordonnanceur 7 génère les informations servant à générer des interruptions afin d'exécuter tous les évènements, tous les messages et les tâches planifiés séquentiellement.
- [0032] A la fin de l'exécution d'un évènement, l'ordonnanceur d'évènements ou « scheduler » contrôle si le temps d'exécution correspond à l'attendu. Le temps mesuré d'exécution d'un évènement par le premier microprocesseur 1 est comparé avec le temps mesuré par le FPGA, selon un principe connu de l'homme du métier. Ceci permet notamment de vérifier que le microprocesseur ne s'est pas arrêté pendant l'exécution de l'évènement avec une erreur ou est resté bloqué dans une boucle infinie.
- [0033] La figure 5 illustre un exemple d'implémentation de l'ordonnanceur d'évènements selon l'invention. Dans cette configuration, les tâches annexes sont laissées au FPGA. Un protocole de communication connu de l'homme du métier permet l'échange de données entre le premier microprocesseur 1 et le FPGA 2, ainsi que la vérification de la bonne exécution de la tâche de calcul souhaitée dans le créneau temporel prévu. La puissance de calcul du microprocesseur est ainsi optimisée.
- [0034] Paramétrage tâche :
- Le FPGA contient un ordonnanceur d'évènements selon l'invention où l'ordre d'exécution des évènements ou des tâches a été obtenu par la mise en œuvre du procédé selon l'invention décrit ci-dessus,
 - Le FPGA écrit (WR) sur une mémoire DPRAM partagée avec le microprocesseur l'identifiant (ID) de la tâche à exécuter ainsi que les paramètres d'entrée (DATA IN),
 - Le FPGA lance une interruption INT vers le premier microprocesseur 1 et démarre un temporisateur ou Timer FPGA,
- [0035] Exécution tâche ou évènement
- Le premier microprocesseur initialise un Timer microprocesseur,
 - Le premier microprocesseur 1 reçoit une interruption du FPGA et lit (RD) la mémoire d'échange DPRAM,

Le contenu de cette lecture (DATA IN) correspond à la tâche qui devra être exécutée ainsi que les paramètres d'entrée,

Le premier microprocesseur exécute la tâche correspondante,

Le premier microprocesseur lance une écriture de mémoire partagée DPRAM 3 et retourne le Timer microprocesseur, l'identifiant de la tâche qui vient d'être exécutée (ID) ainsi que les paramètres de sortie (DATA OUT),

[0036] Vérification tâche

- Le FPGA contrôle son Timer en comparant avec le Timer du premier microprocesseur, selon une méthode connue de l'homme du métier,
- Le FPGA peut utiliser les paramètres de sortie (DATA OUT),
- Un nouveau cycle peut être lancé.

[0037] L'ordonnanceur d'évènements selon l'invention présente notamment comme avantages :

- la gestion des tâches d'un microprocesseur et des communications de manière déterministe, sans avoir recours à une gestion des priorités des interruptions. Les tâches sont exécutées dans un ordre précis de façon à ne jamais avoir deux évènements qui s'exécutent à la fois et sans interrompre l'exécution d'un évènement ou d'une tâche,
- une utilisation à 100% de la bande passante est possible, en l'absence de marge à prévoir pour la gestion d'éventuelles collisions,
- le fonctionnement synchrone de plusieurs équipements permet l'utilisation de la totalité de la bande passante, car il n'y a pas de marge à prévoir pour la gestion des collisions, en fonctionnement normal du système,
- la communication et la gestion des tâches sont organisées temporellement de manière à faire interagir tous les équipements du réseau de manière synchronisée et sans aucune interruption.

Revendications

- [Revendication 1] Procédé pour répartir temporellement les évènements à transmettre dans une fenêtre temporelle initiale $T_i=[t_0, t_{max}]$, lesdits évènements étant exécutés par un premier microprocesseur (1), la répartition temporelle étant réalisée par un dispositif programmable (2) caractérisé en ce qu'il comporte au moins les étapes suivantes :
- identifier au moins deux évènements E_1, E_j , à exécuter par le premier microprocesseur,
 - identifier pour chaque évènement E_1, E_j , sa fréquence d'exécution F_1, F_j , selon des fréquences multiples de deux et la durée T_1, T_j , d'exécution d'un évènement,
 - insérer un premier évènement E_1 et calculer la disponibilité temporelle T_r restante dans la fenêtre temporelle initiale T_i afin de déterminer la taille maximale en temps et la fréquence maximale admissible d'un futur évènement E_j ,
 - à partir de la disponibilité temporelle, de la taille d'un évènement et de sa fréquence admissible, calculer pour chaque futur évènement E_j à exécuter et pour chacune des n occurrences dudit évènement dans la fenêtre temporelle initiale, une valeur de décalage temporel dt tels que $t_0 + dt_1(E_j), t_0 + dt_2(E_j), t_0 + dt_3(E_j), \dots, t_0 + dt_n(E_j)$ correspondant au positionnement temporel du futur évènement E_j par rapport à l'instant de démarrage t_0 dans la fenêtre temporelle initiale,
 - exécuter l'évènement E_j et à la fin de l'exécution de l'évènement transmettre le temps d'exécution de la tâche au dispositif programmable (2), puis vérifier si le temps d'exécution de l'évènement E_j par le microprocesseur, T_{jmicro} est conforme au temps d'exécution T_{jFPGA} attendu dans le dispositif programmable (2).
- [Revendication 2] Procédé selon la revendication 1 caractérisé en ce qu'il tient compte de la priorité associée aux évènements à exécuter dans une fenêtre temporelle pour déterminer les instants d'exécution de chacun de ces évènements.
- [Revendication 3] Système pour répartir temporellement les évènements à transmettre dans une fenêtre temporelle initiale $T_i=[t_0, t_{max}]$ caractérisé en ce qu'il comprend au moins les éléments suivants :
- un dispositif programmable (2) comprenant une mémoire partagée (5) et un ordonnanceur d'évènements (7),
 - ledit ordonnanceur (7) est adapté à

déterminer les instants d'exécution d'au moins deux évènements dans la fenêtre temporelle initiale $T_i=[t_0, t_{max}]$, en prenant en compte la disponibilité temporelle T_r restante dans la fenêtre temporelle initiale T_i après l'insertion d'un premier évènement, à partir de la disponibilité temporelle, de la taille d'un futur évènement et de sa fréquence admissible, à calculer pour chaque futur évènement E_2 à exécuter une valeur de décalage temporel dt correspondant au positionnement temporel du futur évènement E_2 par rapport à l'instant de démarrage t_0 dans la fenêtre temporelle initiale et à vérifier si le temps d'exécution de l'évènement E_2 par le microprocesseur, T_{2micro} est conforme au temps d'exécution T_{2FPGA} ,

- exécuter l'évènement et vérifier si le temps d'exécution de l'évènement est conforme au temps d'exécution attendu,
- un premier microprocesseur (1) configuré pour exécuter les évènements mémorisés dans une mémoire partagée (3), sur réception d'un ordre d'interruption INT émis par ledit dispositif (2) et pour transmettre le temps d'exécution d'un évènement à l'ordonnanceur (7).

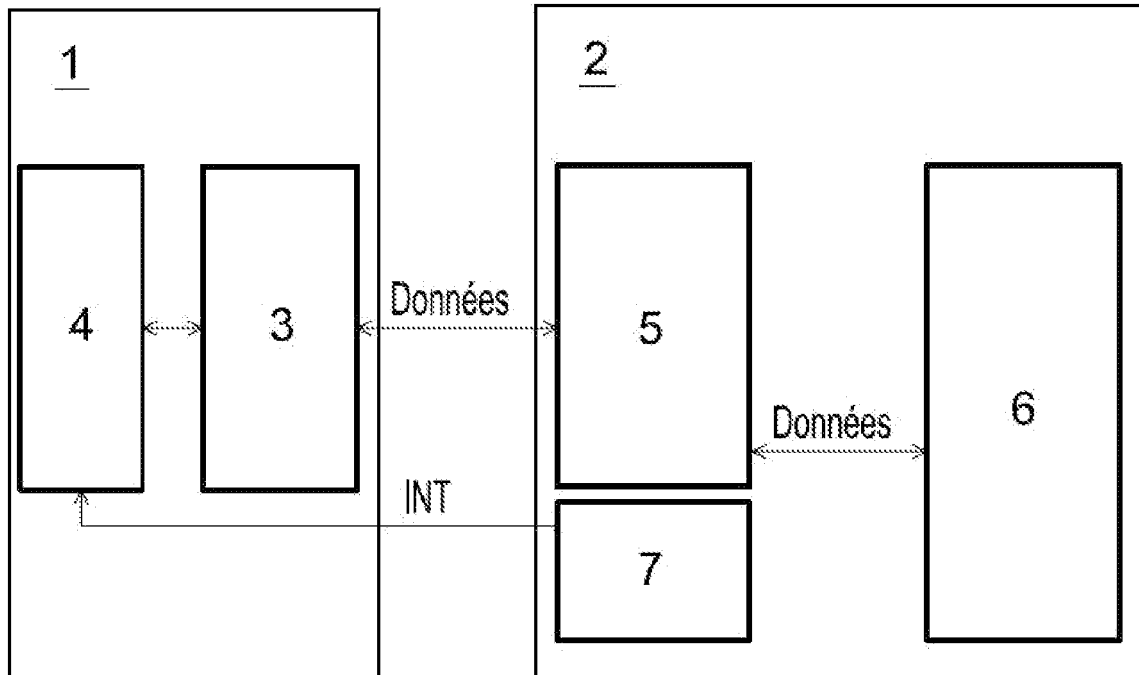
[Revendication 4]

Système selon la revendication 3 caractérisé en ce que l'ordonnanceur est configuré pour calculer la disponibilité temporelle restante dans chacune des n fenêtres générées par le positionnement de l'évènement exécuté n fois dans la fenêtre temporelle initiale.

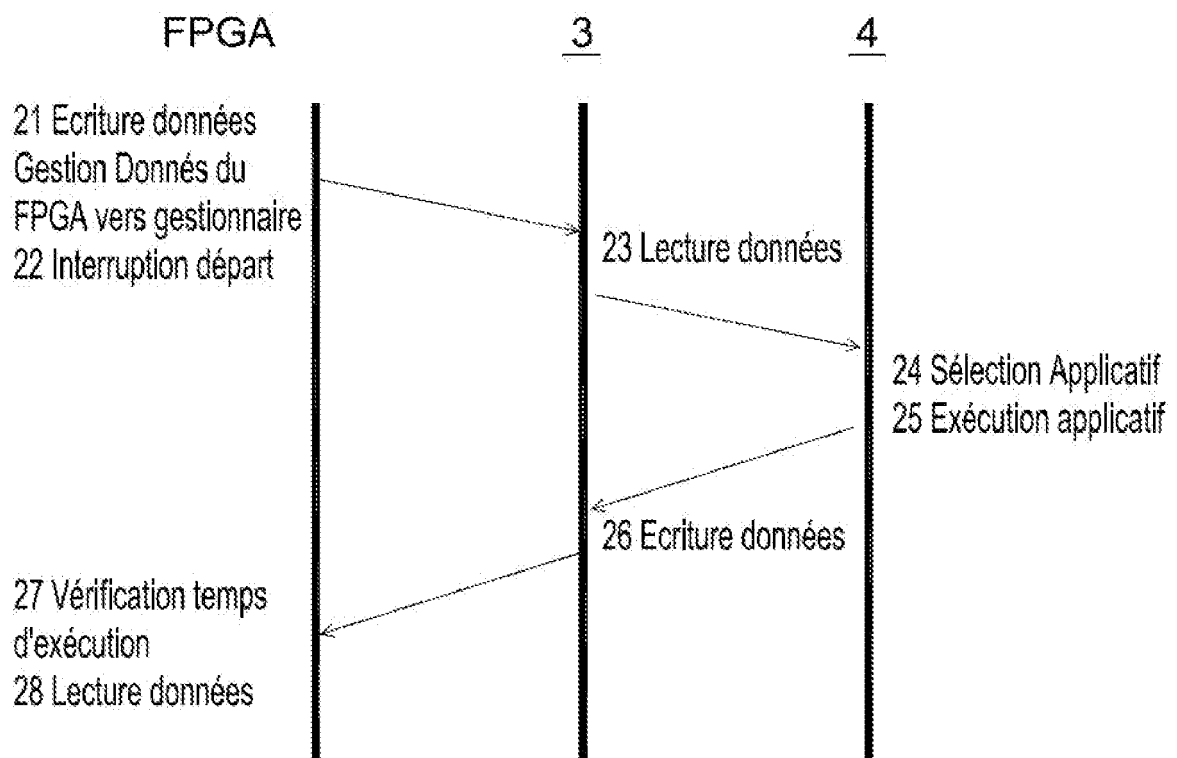
[Revendication 5]

Système selon l'une des revendications 3 ou 4 caractérisé en ce que le dispositif programmable est de type FPGA (Field Programmable Gate Array).

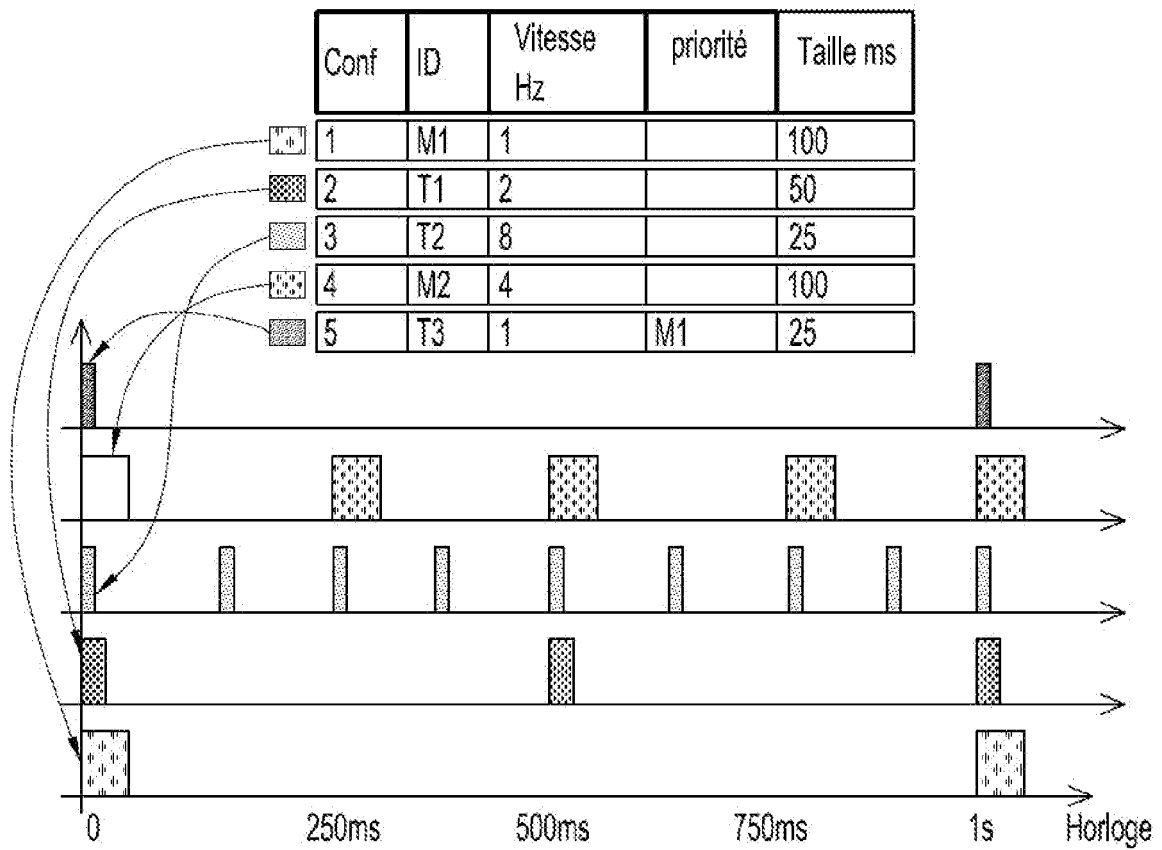
[Fig. 1]



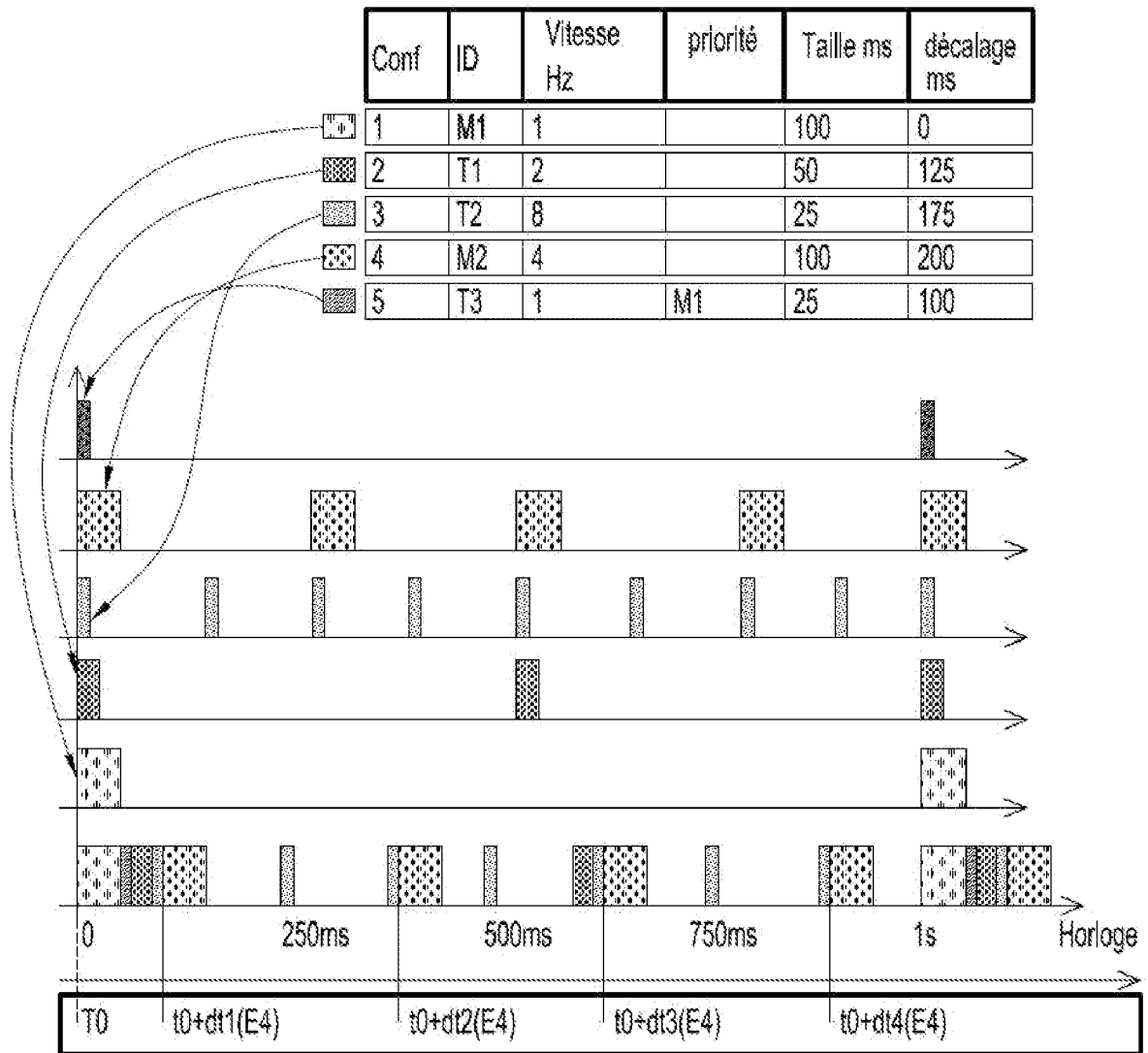
[Fig. 2]



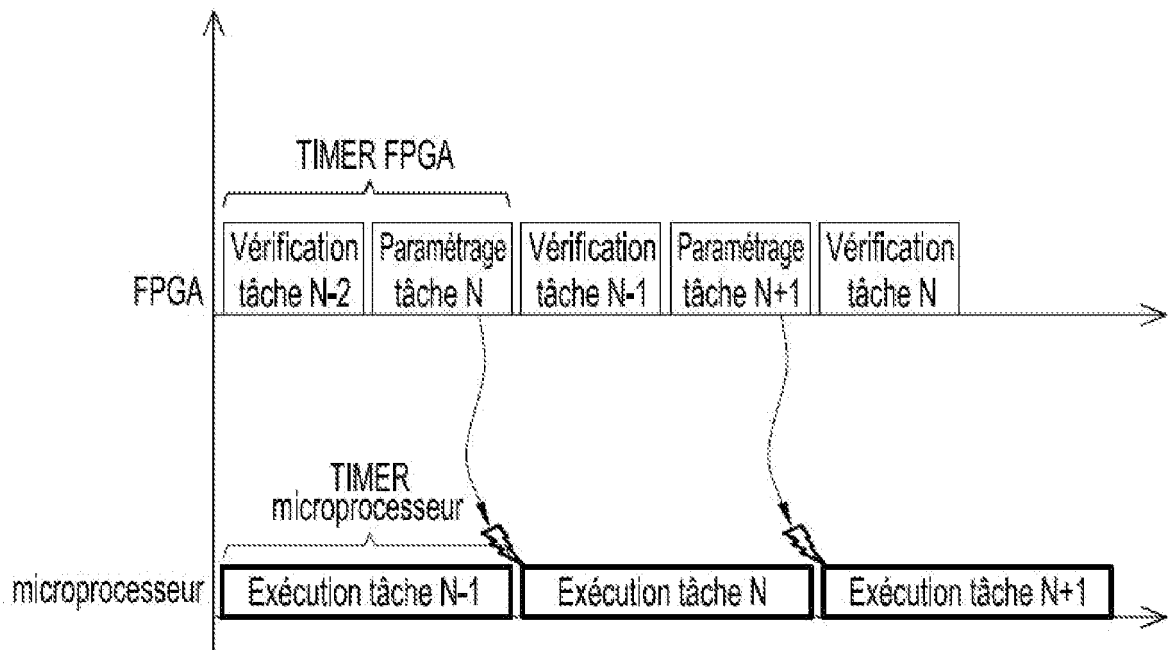
[Fig. 3]



[Fig. 4]



[Fig. 5]



RAPPORT DE RECHERCHE

articles L.612-14, L.612-53 à 69 du code de la propriété intellectuelle

OBJET DU RAPPORT DE RECHERCHE

L'I.N.P.I. annexe à chaque brevet un "RAPPORT DE RECHERCHE" citant les éléments de l'état de la technique qui peuvent être pris en considération pour apprécier la brevetabilité de l'invention, au sens des articles L. 611-11 (nouveau) et L. 611-14 (activité inventive) du code de la propriété intellectuelle. Ce rapport porte sur les revendications du brevet qui définissent l'objet de l'invention et délimitent l'étendue de la protection.

Après délivrance, l'I.N.P.I. peut, à la requête de toute personne intéressée, formuler un "AVIS DOCUMENTAIRE" sur la base des documents cités dans ce rapport de recherche et de tout autre document que le requérant souhaite voir prendre en considération.

CONDITIONS D'ETABLISSEMENT DU PRESENT RAPPORT DE RECHERCHE

Le demandeur a présenté des observations en réponse au rapport de recherche préliminaire.

Le demandeur a maintenu les revendications.

Le demandeur a modifié les revendications.

Le demandeur a modifié la description pour en éliminer les éléments qui n'étaient plus en concordance avec les nouvelles revendications.

Les tiers ont présenté des observations après publication du rapport de recherche préliminaire.

Un rapport de recherche préliminaire complémentaire a été établi.

DOCUMENTS CITES DANS LE PRESENT RAPPORT DE RECHERCHE

La répartition des documents entre les rubriques 1, 2 et 3 tient compte, le cas échéant, des revendications déposées en dernier lieu et/ou des observations présentées.

Les documents énumérés à la rubrique 1 ci-après sont susceptibles d'être pris en considération pour apprécier la brevetabilité de l'invention.

Les documents énumérés à la rubrique 2 ci-après illustrent l'arrière-plan technologique général.

Les documents énumérés à la rubrique 3 ci-après ont été cités en cours de procédure, mais leur pertinence dépend de la validité des priorités revendiquées.

Aucun document n'a été cité en cours de procédure.

**1. ELEMENTS DE L'ETAT DE LA TECHNIQUE SUSCEPTIBLES D'ETRE PRIS EN
CONSIDERATION POUR APPRECIER LA BREVETABILITE DE L'INVENTION**

US 2014/089889 A1 (THE MATHWORKS, INC.)
27 mars 2014 (2014-03-27)

**2. ELEMENTS DE L'ETAT DE LA TECHNIQUE ILLUSTRANT L'ARRIERE-PLAN
TECHNOLOGIQUE GENERAL**

EP 3 438 821 A1 (ALIBABA GROUP HOLDING
LIMITED GRAND KAYMAN)
6 février 2019 (2019-02-06)

WO 2010/092146 A1 (INCHRON GMBH)
19 août 2010 (2010-08-19)

WO 2014/170569 A1 (KRONO-SAFE)
23 octobre 2014 (2014-10-23)

**3. ELEMENTS DE L'ETAT DE LA TECHNIQUE DONT LA PERTINENCE DEPEND
DE LA VALIDITE DES PRIORITES**

NEANT