

①9 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
COURBEVOIE

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

3 087 982

②1 N° d'enregistrement national : 18 60117

⑤1 Int Cl⁸ : H 04 L 5/22 (2019.01)

①2

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 31.10.18.

③0 Priorité :

④3 Date de mise à la disposition du public de la demande : 01.05.20 Bulletin 20/18.

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

⑥0 Références à d'autres documents nationaux apparentés :

Demande(s) d'extension :

⑦1 Demandeur(s) : COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES Etablissement public — FR.

⑦2 Inventeur(s) : HEBBACHE FAROUK, JAN MATHIEU, BRANDNER FLORIAN et PAUTET LAURENT.

⑦3 Titulaire(s) : COMMISSARIAT A L'ENERGIE ATOMIQUE ET AUX ENERGIES ALTERNATIVES Etablissement public.

⑦4 Mandataire(s) : BREVALEX Société à responsabilité limitée.

⑤4 **PROCEDE ET CIRCUIT DE MULTIPLEXAGE TEMPOREL D'ACCES CONCURRENTS A UNE RESSOURCE INFORMATIQUE.**

⑤7 L'invention porte sur un procédé mis en œuvre par ordinateur d'arbitrage entre des programmes informatiques cherchant à accéder concurremment à une ressource partagée en émettant chacun une requête d'accès. Le procédé réalise un accès multiple à répartition dans le temps selon lequel le temps est divisé en créneaux temporels chacun alloué à un programme critique pour un accès à la ressource partagée, chaque créneau temporel comprenant une pluralité d'unités de temps.

Le procédé exploite une marge de traitement associée à chaque programme critique pour retarder une échéance de traitement d'une requête d'accès émise par le programme critique.

Le procédé comprend, à chaque unité de temps, une étape de sélection d'une requête d'accès en attente et une étape de détermination d'une autorisation de traitement immédiat de la requête d'accès sélectionnée. Cette détermination comprend pour une unité de temps ne correspondant pas au début d'un créneau temporel, lorsque le programme critique auquel le prochain créneau temporel est alloué n'a pas émis la requête sélectionnée, l'autorisation du traitement immédiat de la requête sélectionnée si la marge de traitement du programme critique auquel le prochain cré-

neau temporel est alloué est supérieure à un seuil.

FR 3 087 982 - A1



PROCÉDÉ ET CIRCUIT DE MULTIPLEXAGE TEMPOREL D'ACCÈS CONCURRENTS À UNE RESSOURCE INFORMATIQUE

DESCRIPTION

DOMAINE TECHNIQUE

Le domaine de l'invention est celui des systèmes informatiques temps-réel pour
5 lesquels le temps d'exécution des tâches, et notamment le temps d'exécution dans le pire
des cas (dit WCET pour « Worst Case Execution Time »), doit être connu pour en assurer
la validation et en garantir la sûreté. L'invention vise plus particulièrement à améliorer le
taux d'utilisation d'une ressource partagée par des programmes informatiques
s'exécutant en parallèle tout en garantissant l'estimation du WCET de ces programmes au
10 moyen d'une politique d'arbitrage à multiplexage temporel pour l'accès à la ressource
partagée.

ÉTAT DE LA TECHNIQUE ANTÉRIEURE

Les systèmes temps-réel doivent réagir de manière fiable, ce qui implique à la fois
d'être certain du résultat produit par leurs programmes mais aussi de connaître le temps
15 qu'ils prennent pour s'exécuter. Les temps d'exécution pire-cas sont ainsi des données
fondamentales pour la validation et la sûreté de tels systèmes temps-réel, et encore plus
dans le contexte des systèmes temps-réels autonomes (robotique, voiture autonome,
GPS) pour lesquels la sûreté de fonctionnement est primordiale.

Cependant calculer un WCET, à la fois garanti (majorant strict) et pas trop
20 pessimiste afin de réduire les coûts et la complexité de tels systèmes temps-réels, est un
problème difficile à résoudre sur des architecture matérielles multi-cœurs en raison de la
concurrence entre les différents programmes s'exécutant en parallèle pour l'accès aux
ressources partagées, typiquement une mémoire partagée. Pour chaque requête d'accès
à une telle ressource partagée émise par un programme, il faut en effet
25 systématiquement considérer la situation la plus défavorable induite par la politique
d'arbitrage utilisée pour gérer l'accès à cette ressource partagée. Ceci induit un

pessimisme important dans la valeur du WCET obtenue et donc un faible taux d'utilisation de cette ressource partagée lors de l'exécution des programmes. Ce problème de sous-utilisation de la ressource partagée est amplifié avec la multiplication de programmes non-soumis à des contraintes temporelles (dits programmes non-critiques) qui s'exécutent en parallèle des programmes temps-réel (dits programmes critiques). Les requêtes d'accès générées par ces programmes non-critiques impactent la situation la plus défavorable qui doit être considérée pour les requêtes d'accès émises par les programmes critiques et accentuent donc le pessimisme des WCET calculés et la sous-utilisation de la ressource partagée.

Il est possible d'éliminer par construction toute concurrence entre les requêtes d'accès émises par les différentes programmes en ayant recours à une politique d'arbitrage à multiplexage temporel (en anglais « Time-Division Multiplexing », TDM). Selon cette politique, le temps est divisé en créneaux temporels chacun alloué à un programme prédéfini pour un accès exclusif à la ressource partagée. Le temps d'accès à la ressource partagée par un programme peut alors être facilement borné (il correspond au délai maximal jusqu'à parvenir au prochain créneau temporel alloué au programme), la bande passante offerte à un programme est indépendante des autres programmes et les WCET des programmes peuvent ainsi être déterminés.

Toutefois, le temps d'accès à la ressource partagée par un programme dépend de l'ordonnancement des créneaux temporels qui lui sont affectés. Or cet ordonnancement est généralement statique car réalisé lors de la conception du système, avant l'exécution des programmes. Il comprend généralement l'affectation d'une séquence de créneaux temporels à différents programmes, cette séquence se répétant périodiquement (on parle de période TDM). Mais les créneaux temporels d'une période TDM ne sont utilisés par les programmes que s'ils ont une requête d'accès à la ressource partagée à émettre. Lorsque cela n'est pas le cas, ces créneaux temporels sont donc inutilisés. Une politique d'arbitrage TDM basique est donc dite oisive car ces créneaux temporels ne sont pas récupérés par les autres programmes afin de diminuer leur temps d'accès à la ressource partagée.

Par ailleurs, il est admis que pour les programmes temps-réel, il peut exister en moyenne un facteur 10 entre le temps d'exécution d'une instance du programme et son WCET. L'oisiveté de la politique d'arbitrage TDM associée à cette caractéristique des programmes temps réel génère un faible taux d'utilisation de la ressource partagée. Ce problème est amplifié lorsque le nombre de programmes augmente car dans ce cas la longueur d'une période TDM est également augmentée. Un autre facteur d'amplification de ce phénomène est la présence au sein des systèmes considérés d'un nombre croissant de programmes non-critiques pour lesquels l'affectation de créneaux augmente la latence des requêtes des programmes critiques ainsi que la latence des requêtes de ces programmes non-critiques, en contradiction avec leur objectif d'avoir les meilleurs performances d'exécution en moyenne.

La publication de Farouk Hebbache, Mathieu Jan, Florian Brandner et Laurent Pautet, intitulée « Dynamic Arbitration of Memory Requests with TDM-like Guarantees », Proceedings of the CRTS 2017 workshop, présente une amélioration à une telle politique d'arbitrage TDM qui permet d'en réduire l'oisiveté. Selon cette amélioration, dite TMDds (« dynamic TDM with slack counters »), une échéance de traitement est associée à chaque requête d'accès d'un programme critique qui correspond à l'instant auquel le traitement de cette requête doit être terminé au plus tard, ce qui permet de rendre possible la détermination des WCET. Cette échéance correspond tout simplement à la fin du prochain créneau temporel alloué au programme critique qui suit l'émission de la requête d'accès.

L'arbitrage peut alors être rendu dynamique, tant que les échéances de traitement sont respectées. Pour ce faire, un compteur de marge de traitement (« slack counter » en anglais) est associé à chaque programme critique. Ce compteur indique le nombre de créneaux temporels séparant la terminaison effective du traitement d'une requête d'accès de son échéance de traitement. Ce compteur est utilisé pour retarder le cas échéant l'échéance de traitement de la prochaine requête d'accès du programme critique. L'arbitrage réalisé à chaque début de créneau temporel consiste alors à sélectionner la requête d'accès pendante ayant l'échéance de traitement la plus proche, indépendamment du programme critique à qui ce créneau temporel est alloué.

Si cette politique d'arbitrage TDMds permet d'améliorer l'utilisation de la ressource partagée, elle reste néanmoins perfectible. En effet, pour les systèmes considérés, la longueur des créneaux temporels doit être supérieure ou égale au temps maximum mis par une requête, émise de manière isolée, pour être traitée par la ressource partagée. Or si par exemple la ressource partagée est une mémoire, le temps d'accès d'une requête en écriture est inférieur au temps d'accès d'une requête en lecture, puisque dans ce second cas la mémoire doit émettre les données demandées à destination du programme requérant. Par ailleurs, le temps d'accès à une mémoire peut dépendre de l'historique des requêtes émises par les autres programmes. Une borne supérieure de traitement des requêtes doit donc être identifiée pour déterminer la longueur des créneaux temporels. Toutefois, le traitement de la plupart des requêtes va par définition se terminer avant cette borne, de sorte que la politique d'arbitrage TDMds demeure oisive.

EXPOSÉ DE L'INVENTION

L'invention a pour objectif de proposer une politique d'arbitrage améliorée qui permette d'optimiser l'utilisation de la ressource partagée tout en garantissant la prédictibilité temporelle du traitement des requêtes d'accès à la ressource partagée issues de programmes critiques.

A cet effet, l'invention propose un procédé mis en œuvre par ordinateur d'arbitrage entre des programmes informatiques cherchant à accéder concurremment à une ressource partagée en émettant chacun une requête d'accès. Le procédé réalise un accès multiple à répartition dans le temps selon lequel le temps est divisé en créneaux temporels chacun alloué à un programme critique pour un accès à la ressource partagée, chaque créneau temporel comprenant une pluralité d'unités de temps. Une marge de traitement est associée à chaque programme critique pour retarder une échéance de traitement d'une requête d'accès émise par le programme critique. Le procédé comprend, à chaque unité de temps, une étape de sélection d'une requête d'accès parmi une ou plusieurs requêtes d'accès en attente et une étape de détermination d'une autorisation de traitement immédiat de la requête d'accès sélectionnée. L'étape de détermination comprend pour une unité de temps ne correspondant pas au début d'un

créneau temporel, lorsque le programme critique auquel le prochain créneau temporel est alloué n'a pas émis la requête sélectionnée, l'autorisation du traitement immédiat de la requête sélectionnée si la marge de traitement du programme critique auquel le prochain créneau temporel est alloué est supérieure à un seuil.

5 Certains aspects préférés mais non limitatifs de ce procédé sont les suivants :

- l'étape de détermination comprend, pour une unité de temps ne correspondant pas au début d'un créneau temporel, lorsque le programme critique auquel le prochain créneau temporel est alloué n'a pas émis la requête sélectionnée et lorsque la marge de traitement du programme critique auquel le prochain créneau temporel est alloué est inférieure au seuil, la mise en attente de la requête sélectionnée ;

10

- le seuil correspond à l'écart temporel entre l'unité de temps à laquelle l'étape de détermination est réalisée et l'unité de temps correspondant au début du prochain créneau temporel ;

15

- l'étape de détermination comprend pour une unité de temps ne correspondant pas au début d'un créneau temporel, lorsque le programme critique auquel le prochain créneau temporel est alloué a émis la requête sélectionnée, l'autorisation du traitement immédiat de la requête sélectionnée ;

20

- l'étape de détermination comprend, pour une unité de temps correspondant au début d'un créneau temporel, l'autorisation du traitement immédiat de la requête sélectionnée ;

25

- l'étape de sélection comprend, en présence d'une requête d'accès en attente d'un programme critique et d'une requête d'accès en attente d'un programme non-critique, la sélection de la requête d'accès du programme non-critique si l'échéance de traitement de la requête d'accès du programme critique est postérieure à la fin du prochain créneau temporel ;

- la marge de traitement d'un programme critique est mise à jour à chaque terminaison d'un traitement d'une requête d'accès du programme critique pour correspondre au nombre d'unités de temps séparant la terminaison du traitement de la requête d'accès et l'échéance de traitement de la requête d'accès ;

- l'échéance de traitement d'une requête d'accès d'un programme critique correspond à la fin d'un créneau temporel alloué au programme critique qui est le premier à débiter après une date correspondant à une date d'émission de la requête d'accès retardée par ajout de la marge de traitement du programme critique ;

- 5 - il comprend en outre la préemption au profit d'un programme critique d'un premier programme ayant émis une requête d'accès à la ressource partagée en attente de traitement, ladite préemption comprenant le calcul d'une échéance de traitement de ladite requête correspondant à la fin d'un créneau temporel alloué au programme critique.

10 BRÈVE DESCRIPTION DES DESSINS

D'autres aspects, buts, avantages et caractéristiques de l'invention apparaîtront mieux à la lecture de la description détaillée suivante de formes de réalisation préférées de celle-ci, donnée à titre d'exemple non limitatif, et faite en référence aux dessins annexés sur lesquels :

- 15 - les figures 1a et 1b sont des schémas illustrant le principe de l'arbitrage réalisé dans l'invention pour autoriser ou non un accès immédiat à la ressource partagée ;
- la figure 2 est une machine d'états représentant un enchaînement d'étapes du procédé selon l'invention ;
- 20 - la figure 3 est un chronogramme représentant un exemple d'arbitrage opéré par le procédé selon l'invention ;
- la figure 4 est un schéma d'un circuit configuré pour mettre en œuvre le procédé selon l'invention.

EXPOSÉ DÉTAILLÉ DE MODES DE RÉALISATION PARTICULIERS

- 25 L'invention porte sur un procédé mis en œuvre par ordinateur d'arbitrage entre des programmes informatiques cherchant à accéder concurremment à une ressource partagée en émettant chacun une requête d'accès. En référence à la figure 4 qui sera décrite plus en détail ultérieurement, ce procédé est typiquement mis en œuvre sur une

architecture matérielle multi-cœurs où chaque cœur C_0, C_1, C_m est configuré pour exécuter un programme susceptible de requérir des accès à la ressource partagée lors de son exécution. Chaque cœur peut disposer d'une mémoire cache privée et la ressource partagée est par exemple une mémoire partagée MM. Une implémentation matérielle configurée pour mettre en œuvre ce procédé prend la forme d'un circuit de multiplexage temporel CMT interposé entre les cœurs et la ressource partagée.

Comme pour l'arbitrage TDMds décrit dans la publication précitée, le procédé selon l'invention réalise un accès multiple à répartition dans le temps selon lequel le temps est divisé en créneaux temporels (*slots* en anglais) chacun alloué à un programme critique pour un accès à la ressource partagée, chaque créneau temporel comprenant une pluralité d'unités de temps (une unité de temps est par exemple un cycle d'horloge). On a représenté sur les figures 1a, 1b et 3 un exemple d'une telle division temporelle en créneaux de tailles identiques alternativement alloué à un premier programme critique A (les créneaux portant un numéro impair) et à un deuxième programme critique B (les créneaux portant un numéros pair).

Un programme critique est par exemple un programme soumis à des contraintes temporelles et pour lequel on souhaite pouvoir borner le temps d'exécution dans le pire des cas (WCET). Ainsi, comme pour l'arbitrage TDMds, le procédé selon l'invention associe une échéance de traitement à chaque requête d'accès émise par un programme critique. Cette échéance correspond à l'instant auquel le traitement de la requête du programme critique doit être terminé au plus tard et doit être respectée. Le procédé porte également sur des requêtes d'accès émises par des programmes non soumis à des contraintes temporelles, i.e. des programmes non-critiques, et pour lesquelles aucune notion d'échéance de traitement n'est nécessaire.

Et le procédé selon l'invention exploite lui aussi une marge de traitement associée à chaque programme critique, cette marge de traitement étant utilisée pour éventuellement retarder une échéance de traitement d'une requête d'accès émise par le programme critique, notamment lorsque le programme critique a accumulé une marge de traitement suffisante. De telle manière, une requête d'accès à la ressource émise par

un programme peut être traité sur un créneau qui n'est pas alloué par construction à ce programme.

Selon TDMs, les décisions d'arbitrage sont prises au début de chaque créneau temporel et ne sont basées que sur les requêtes en attente de traitement. Il est alors possible de retarder une requête d'un programme critique, en fonction de l'échéance de traitement de cette requête. La marge de traitement n'est pas exploitée lors des décisions d'arbitrage et n'a qu'un effet sur le calcul des échéances de traitement des requêtes.

L'invention est basée sur le fait de reconnaître qu'une marge de traitement est également valide lorsqu'un programme critique n'a pas (encore) émis une requête d'accès à la ressource. En particulier, il est possible de déterminer une borne inférieure de l'échéance de traitement associée à n'importe quelle requête émise par le programme critique auquel le prochain créneau temporel est alloué (même lorsque le programme n'a pas encore émis une requête). La ressource partagée peut alors démarrer le traitement de n'importe laquelle des requêtes en attente si cette borne inférieure est postérieure à la fin du prochain créneau temporel. Cela garantit que la mémoire peut traiter partiellement la requête dans le créneau temporel actuel, tout en complétant ce traitement dans le créneau suivant sans violer le comportement dans le pire des cas.

L'invention propose ainsi d'exploiter les marges de traitement, non pas à chaque début de créneau temporel, mais à chaque unité de temps. La marge de traitement associée au programme critique auquel le prochain créneau temporel est alloué est consultée pour déterminer si une requête peut immédiatement être traitée par la ressource partagée. Cette consultation a pour objectif de vérifier si un débordement du temps de traitement de la requête sur le prochain créneau temporel ne remet pas en cause le respect de l'échéance de traitement associée à une éventuelle requête qui serait émise par le programme critique auquel ce prochain créneau temporel est alloué.

Ainsi dans le cadre de l'invention, la marge de traitement d'un programme critique est mise à jour à chaque terminaison d'un traitement d'une requête d'accès du programme critique pour correspondre au nombre d'unités de temps séparant la terminaison du traitement de la requête d'accès et l'échéance de traitement de la

requête d'accès. Et l'échéance de traitement d'une requête d'accès d'un programme critique correspond à la fin du créneau temporel alloué au programme critique qui est le premier à débiter après une date correspondant à une date d'émission de la requête d'accès retardée par ajout de la marge de traitement du programme critique.

5 On a représenté sur les figures 1a et 1b un exemple dans lequel à une unité de temps correspondant au temps t , l'accès à la ressource est libéré et il est vérifié si une requête en attente peut être traitée. Cette unité de temps t ne correspond pas au début du créneau temporel alloué au programme critique B qui débute à t_0 . Le créneau suivant, alloué au programme critique A, débute à t_1 .

10 Sur la figure 1a, la marge de traitement MA1 du programme critique A auquel le prochain créneau temporel est alloué (créneau n°3) est suffisamment importante pour permettre de retarder l'échéance de traitement d'une requête qui serait émise par le programme A entre t et t_1 . Au lieu que cette échéance corresponde à la fin I1 du prochain créneau temporel alloué au programme A, elle est retardée pour correspondre à la fin E1
15 du créneau temporel n°5 alloué au programme A qui suit une date correspondant à une date d'émission de la requête d'accès retardée par ajout de la marge de traitement du programme critique MA1. Cela revient à déterminer que la marge de traitement MA1 est supérieure au nombre d'unités de temps séparant t et t_1 .

20 Sur la figure 1b, la marge de traitement MA2 du programme critique A auquel le prochain créneau temporel est alloué n'est pas suffisamment importante pour permettre de retarder l'échéance de traitement d'une requête qui serait émise par le programme A entre t et t_1 . Son échéance de traitement correspond donc à la fin E2 du prochain créneau temporel (n°3) associé au programme A. Cela revient à déterminer que la marge de traitement MA2 est inférieure au nombre d'unités de temps séparant t et t_1 . Dans un tel
25 cas de figure, la requête en attente ne peut pas être traitée car son traitement pourrait déborder sur le prochain créneau temporel n°3 en empêchant le traitement d'une requête que formulerait le programme critique A et donc compromettre le respect de l'échéance de traitement du programme critique A.

On a représenté sur la figure 2, une machine d'états représentative d'un mode de réalisation du procédé selon l'invention. Sur cette figure, la référence « Rac » désigne l'émission d'une requête d'accès à la ressource partagée.

Si cette émission est réalisée par un programme critique « Cr », il est procédé lors
5 d'une opération « DC » au calcul de l'échéance de traitement de la requête émise par le programme critique. Ce calcul exploite la marge de traitement accumulé par le programme critique lors du traitement de sa précédente requête d'accès à la ressource pour venir éventuellement retarder l'échéance de traitement au-delà de la fin du prochain créneau temporel alloué à ce programme critique (i.e. à la fin d'un créneau
10 temporel alloué au programme critique postérieur au prochain créneau temporel alloué au programme critique).

Si cette émission est réalisée par un programme non-critique « NCr » et qu'il existe une ou plusieurs autres requêtes non-critiques pendantes, il est procédé à un arbitrage
15 « NC-AR » des requêtes non-critiques selon une politique d'arbitrage quelconque (par exemple priorité fixe ou premier arrivé-premier servi) pour venir en sélectionner une.

A chaque unité de temps, il est vérifié si l'accès mémoire est possible. Dans la négative (*idle*), les requêtes pendantes sont mises en attente. Dans l'affirmative (*idle*), il est procédé à la sélection d'une requête (Select) et à la détermination d'une autorisation de traitement immédiat de la requête d'accès sélectionnée.

La sélection comprend le choix, parmi les requêtes de programmes critiques, de la
20 requête présentant l'échéance de traitement la plus proche. Le résultat de la sélection est donc soit la requête critique présentant l'échéance de traitement la plus proche, soit la requête non critique résultant de l'arbitrage « NC-AR ». Par exemple, en présence d'une requête d'accès en attente d'un programme critique et d'une requête d'accès en attente
25 d'un programme non-critique, il est procédé à la sélection de la requête d'accès du programme non-critique si l'échéance de traitement du programme critique est postérieure à la date de fin du prochain créneau temporel.

La détermination d'une autorisation de traitement immédiat de la requête d'accès sélectionnée comprend, pour une unité de temps ne correspondant pas au début d'un
30 créneau temporel ($t \neq t_0$), lorsque le programme critique auquel le prochain créneau

temporel est alloué n'a pas émis la requête sélectionnée (« Next=Rs » et « N »), l'autorisation du traitement immédiat de la requête sélectionnée (« RS ») si la marge de traitement du programme critique auquel le prochain créneau temporel est alloué est supérieure à un seuil (« $t+MT_{next}>t_1$ » et « O »). Si la marge de traitement est inférieure

5 au seuil, la requête sélectionnée est mise en attente (« Att »). Le seuil peut correspondre comme on l'a vu précédemment à l'écart temporel entre l'unité de temps t à laquelle l'étape de détermination est réalisée et l'unité de temps t_1 correspondant au début du prochain créneau temporel. Alternativement, le seuil peut correspondre à la taille d'un créneau temporel.

10 La détermination d'une autorisation de traitement immédiat de la requête d'accès sélectionnée comprend par ailleurs, pour une unité de temps ne correspondant pas au début d'un créneau temporel ($t \neq t_0$), lorsque le programme critique auquel le prochain créneau temporel est alloué a émis la requête sélectionnée (« Next=Rs » et « O »), l'autorisation du traitement immédiat de la requête sélectionnée (« Rs »).

15 Et la détermination d'une autorisation de traitement immédiat de la requête d'accès sélectionnée comprend par ailleurs, pour une unité de temps correspondant au début d'un créneau temporel ($t=t_0$), l'autorisation du traitement immédiat de la requête sélectionnée (« Rs »).

20 Une fois la requête sélectionnée traitée, il est procédé, lorsque cette requête est issue d'un programme critique (« $Rs=Cr$ »), à la mise à jour « MAJ MTs » de la marge de traitement de ce programme critique. De telle manière, la marge de traitement d'un programme critique est mise à jour à chaque terminaison d'un traitement d'une requête d'accès du programme critique pour correspondre au nombre d'unités de temps séparant la terminaison du traitement de la requête d'accès et l'échéance de traitement de la

25 requête d'accès.

On a représenté sur la figure 3 un exemple de mise en œuvre de ce procédé avec deux programmes critiques A et B, un programme non-critique c et des créneaux ayant une taille de 8 unités de temps. Sur cette figure, la valeur courante de la marge de traitement est indiquée en exposant de chaque requête (par exemple la requête A_1 du

programme critique A bénéficie d'une marge de traitement de 10Δ , soit 10 unités de temps).

5 Une première requête d'accès A_0 du programme critique A est tout d'abord émise au cours du créneau n°1 alloué au programme A. La marge de traitement de A est nulle et son échéance de traitement est fixée à la fin du prochain créneau alloué à A (créneau n°3). A ce stade, la marge de traitement de B (propriétaire du prochain créneau n°2) est nulle et il n'est pas possible de réaliser un traitement immédiat de A_0 avant le début du prochain créneau n°2. Le traitement de A_0 dure six unités de temps et A a donc accumulé une marge de traitement de 10 unités de temps.

10 Une première requête d'accès B_0 du programme critique B est émise au cours du créneau n°2. Le prochain créneau n°3 est alloué au programme A qui a accumulé suffisamment de marge de traitement pour que soit autorisé le traitement immédiat de la requête B_0 sans risque que le débordement du traitement de B_0 sur le créneau n°3 ne vienne remettre en cause le respect de l'échéance de traitement associée à une éventuelle requête émise par le programme critique A auquel le créneau n°3 est alloué.
15 Le programme B voit sa marge de traitement passer à 10 unités de temps.

Au cours du créneau n°4, une première requête d'accès c_0 du programme non-critique c et une deuxième requête d'accès B_1 du programme critique B sont simultanément émises. La requête c_0 du programme non-critique est sélectionnée car le programme critique B n'est pas propriétaire du prochain créneau n°5, et est immédiatement traitée car le programme critique A propriétaire du prochain créneau n°5 a accumulé suffisamment de marge de traitement (en l'occurrence 10 unités de temps).
20 Puis dès la terminaison de la requête d'accès c_0 au cours du créneau n°5, il est procédé au traitement immédiat de la requête B_1 car le programme B est propriétaire du prochain créneau n°6.
25

Au cours du traitement de la requête B_1 , une deuxième requête d'accès A_1 du programme critique A et une deuxième requête d'accès c_1 du programme non-critique c sont émises. A la terminaison du traitement de la requête B_1 au cours du créneau n°6, celui-ci voit sa marge de traitement passer à 7 unités de temps. La requête A_1 est
30 sélectionnée et immédiatement traitée car le programme A est propriétaire du prochain

créneau n°7. Le traitement de la requête A_1 se termine au cours du créneau n°6, et la marge de traitement du programme A passe à 11 unités de temps.

A la terminaison du traitement de la requête A_1 deux requêtes sont en compétition : la deuxième requête d'accès c_1 du programme non-critique c et une troisième requête d'accès B_2 du programme critique B. La deuxième requête d'accès c_1 est sélectionnée car l'échéance de traitement de la requête B_2 est postérieure à la fin du créneau n°7, puis immédiatement traitée car la marge de traitement du programme A propriétaire du prochain créneau n°7 est suffisante. A la terminaison du traitement de la requête c_1 , la requête B_2 est immédiatement traitée car le programme B est propriétaire du prochain créneau n°8. A la terminaison du traitement de la requête B_2 au cours du créneau n°8, celui-ci voit sa marge de traitement passer à 5 unités de temps. Une troisième requête d'accès A_2 du programme critique A est ensuite émise qui au vu de la marge de traitement dont bénéficie alors le programme A dispose d'une échéance de traitement à la fin du créneau n°11. Néanmoins, le programme A est propriétaire du prochain créneau n°9 et sa requête A_2 peut être immédiatement traitée.

On a représenté sur la figure 4 une implémentation possible d'un circuit de multiplexage temporel CMT configuré pour mettre en œuvre le procédé selon l'invention. Le circuit CMT est interposé entre des cœurs C_0 , C_1 , C_m exécutant chacun un programme et une ressource partagée. Une marge de traitement est associée à chaque cœur exécutant un programme critique. Un compteur peut pour cela être implémenté sous la forme d'un registre stockant un nombre d'unités de temps non-utilisées par le cœur. Ce compteur fait partie du contexte d'exécution du programme critique, et sa valeur est donc sauvegardée/restaurée à chaque préemption/reprise de l'exécution du programme.

Le circuit CMT comprend une unité d'arbitrage AR et un multiplexeur MUX commandé par l'unité d'arbitrage pour réaliser l'accès sélectif de l'un des cœurs à la ressource partagée. Le circuit CMT comprend par ailleurs une unité de calcul d'échéance de traitement DC_0 , DC_1 , DC_m associée à chacun des cœurs. Cette unité est configurée pour réceptionner une requête d'accès R_0 , R_1 , R_m émise par un programme exécuté sur le cœur auquel elle est associée et pour calculer l'échéance de traitement de la requête lorsque celle-ci est émise par un programme critique. L'unité DC_0 , DC_1 , DC_m peut

comprendre des registres dédiés pour stocker l'échéance calculée ainsi que la marge de traitement courante.

Si la requête est émise par un programme non-critique, l'unité DC0, DC1, DCm vient simplement passer la requête à une unité NC.AR en charge de réaliser un arbitrage entre
 5 requêtes non-critiques selon une politique d'arbitrage quelconque. Elle peut venir stocker une valeur prédéfinie dans le registre dédié à l'échéance, par exemple la valeur maximale pouvant être prise par une échéance.

Le circuit d'arbitrage CA comprend par ailleurs une unité Min.extractor configurée pour recevoir des unités de calcul d'échéance de traitement DC0, DC1, DCm la valeur
 10 stockée dans leur registre dédié à l'échéance, déterminer quelle est la plus petite valeur et fournir à l'unité d'arbitrage AR un identifiant de la requête ayant l'échéance la plus proche par rapport à l'instant courant.

Le circuit d'arbitrage CA comprend également une unité SL configurée pour recevoir des unités de calcul d'échéance de traitement DC0, DC1, DCm la valeur stockée dans leur
 15 registre dédié à la marge et fournir à l'unité d'arbitrage AR la marge de traitement correspondant au programme critique auquel le prochain créneau temporel est alloué.

A chaque unité de temps, si une requête en attente peut être traitée, l'unité d'arbitrage AR sélectionne une requête (venant de l'unité Min.extractor ou de l'unité NC.AR) et vérifie si elle peut être traitée immédiatement. Pour ce faire, si l'unité de temps
 20 correspond au début d'un créneau temporel, la requête est traitée. Sinon, l'unité d'arbitrage détermine si le traitement de la requête peut déborder sur le créneau temporel suivant. Comme on l'a vu précédemment, un tel débordement sur le créneau temporel suivant est autorisé si l'une ou l'autre des deux conditions suivantes est remplie :

- 25 - Le créneau suivant est alloué au programme émetteur de la requête sélectionnée ;
- Le programme auquel le créneau suivant est alloué a accumulé suffisamment de marge pour permettre un débordement de la requête sélectionnée sur son créneau. À titre d'exemple, une marge suffisante correspond à la taille d'un
 30 créneau.

A la terminaison du traitement d'une requête émise par un programme critique, l'unité d'arbitrage AR en notifie les unités de calcul d'échéance de traitement DC0, DC1, DCm via un lien Ack, ce qui déclenche la mise à jour du calcul de la marge temporelle associée au programme critique dont la requête vient d'être traitée.

5 Dans un mode de réalisation possible de l'invention, les cœurs peuvent réaliser du multitâche préemptif, c'est-à-dire distribuer du temps-processeur entre différents programmes. Un programme s'exécutant sur l'un des cœurs peut ainsi être préempté au profit d'un autre programme. Le programme en cours d'exécution est alors interrompu pour permettre à l'autre programme de s'exécuter.

10 Or un créneau temporel est en réalité associé à chaque cœur. Ce créneau lui est alloué lorsque le cœur exécute un programme critique. En revanche, lorsque le cœur exécute un programme non-critique, le créneau peut être exploité pour le traitement d'autres requêtes en provenance d'autres cœurs. Dans le cas où un programme critique souhaite préempter un programme non-critique (selon l'ordonnement du multitâche
15 préemptif) et que ce programme non-critique est en attente de traitement d'une requête d'accès à la ressource partagée, cette requête doit être traitée pour que le programme non-critique puisse effectivement être préempté.

 Dans le cadre de ce mode de réalisation, afin de borner l'attente de traitement de la requête non-critique, celle-ci hérite de la criticité du programme critique. Le cœur devient
20 alors critique et le créneau temporel qui lui est associé lui est à nouveau dédié pour le traitement de la requête. Le calcul de l'échéance se base donc sur la prochaine occurrence du créneau temporel qui lui est à nouveau dédié. Dans une variante possible, la valeur actuelle de la marge de traitement du programme critique préemptant le programme non-critique peut être ajoutée pour déterminer cette échéance.

25 Il est possible également qu'un programme critique vienne préempter un autre programme critique ayant une requête en attente de traitement du fait d'une marge de traitement permettant le traitement d'autres requêtes. Un calcul similaire peut alors être réalisé afin de réduire le délai d'attente avant la réalisation de la préemption.

 Le procédé selon l'invention peut ainsi comprendre une étape de préemption au
30 profit d'un programme critique d'un premier programme ayant émis une requête d'accès

à la ressource partagée en attente de traitement. Cette étape de préemption comprend le calcul, éventuellement en tenant compte de la marge de traitement du programme critique, d'une échéance de traitement de ladite requête correspondant à la fin d'un créneau temporel alloué au programme critique.

5 Le circuit de multiplexage temporel CMT peut réaliser une telle gestion des préemptions de la manière suivante. Le cœur réalisant la préemption en notifie l'unité de calcul d'échéance qui lui est associé. Si un programme non-critique A est préempté au profit d'un programme critique B, la présence d'une requête issue du programme non-critique A dans l'unité NC.AR est vérifiée. Si une telle requête existe, alors celle-ci est
10 retirée de l'unité NC.AR et est considérée comme critique dans la suite des étapes du procédé précédemment décrit en considérant potentiellement la marge de traitement du programme critique B pour le calcul de l'échéance de traitement désormais associée au programme A. Si un programme critique A est préempté au profit d'un programme critique B, de manière similaire, la marge de traitement du programme critique B peut
15 être utilisée pour calculer la nouvelle échéance de traitement du programme A.

L'invention n'est pas limitée au procédé précédemment décrit mais s'étend également à un circuit de multiplexage temporel d'accès concurrents à une ressource partagée requis par des programmes informatiques, configuré pour mettre en œuvre le procédé, et notamment au circuit CMT précédemment décrit.

20

25

REVENDEICATIONS

1. Procédé mis en œuvre par ordinateur d'arbitrage entre des programmes informatiques cherchant à accéder concurremment à une ressource partagée (MM) en émettant chacun une requête d'accès, le procédé réalisant un accès multiple à répartition dans le temps selon lequel le temps est divisé en créneaux temporels chacun alloué à un programme critique pour un accès à la ressource partagée, chaque créneau temporel comprenant une pluralité d'unités de temps,

procédé dans lequel une marge de traitement (MA_1 , MA_2) est associée à chaque programme critique pour retarder une échéance de traitement (I_1) d'une requête d'accès émise par le programme critique,

procédé caractérisé en ce qu'il comprend, à chaque unité de temps, une étape de sélection d'une requête d'accès (B_0 , c_0 , c_1) parmi une ou plusieurs requêtes d'accès en attente et une étape de détermination d'une autorisation de traitement immédiat de la requête d'accès sélectionnée, ladite étape de détermination comprenant pour une unité de temps ne correspondant pas au début d'un créneau temporel, lorsque le programme critique (A) auquel le prochain créneau temporel est alloué n'a pas émis la requête sélectionnée, l'autorisation du traitement immédiat de la requête sélectionnée (B_0 , c_0 , c_1) si la marge de traitement du programme critique auquel le prochain créneau temporel est alloué est supérieure à un seuil.

2. Procédé selon la revendication 1, dans lequel l'étape de détermination comprend, pour une unité de temps ne correspondant pas au début d'un créneau temporel, lorsque le programme critique auquel le prochain créneau temporel est alloué (B) n'a pas émis la requête sélectionnée (A_0) et lorsque la marge de traitement du programme critique auquel le prochain créneau temporel est alloué est inférieure au seuil, la mise en attente de la requête sélectionnée.

3. Procédé selon l'une des revendications 1 à 2, dans lequel le seuil correspond à l'écart temporel entre l'unité de temps à laquelle l'étape de détermination est réalisée et l'unité de temps correspondant au début du prochain créneau temporel.

5 4. Procédé selon l'une des revendications 1 à 3, dans lequel ladite étape de détermination comprend pour une unité de temps ne correspondant pas au début d'un créneau temporel, lorsque le programme critique (A) auquel le prochain créneau temporel est alloué a émis la requête sélectionnée, l'autorisation du traitement immédiat de la requête sélectionnée (A_1).

10 5. Procédé selon l'une des revendications 1 à 4, dans lequel ladite étape de détermination comprend, pour une unité de temps correspondant au début d'un créneau temporel, l'autorisation du traitement immédiat de la requête sélectionnée (A_0).

15 6. Procédé selon l'une des revendications 1 à 5, dans lequel l'étape de sélection comprend, en présence d'une requête d'accès (B_1) en attente d'un programme critique et d'une requête d'accès (c_0) en attente d'un programme non-critique, la sélection de la requête d'accès du programme non-critique (c_0) si l'échéance de traitement de la requête d'accès (B_1) du programme critique est postérieure à la fin du prochain créneau temporel.

20 7. Procédé selon l'une des revendications 1 à 6, dans lequel la marge de traitement d'un programme critique est mise à jour à chaque terminaison d'un traitement d'une requête d'accès du programme critique pour correspondre au nombre d'unités de temps séparant la terminaison du traitement de la requête d'accès et
25 l'échéance de traitement de la requête d'accès.

8. Procédé selon la revendication 7, dans lequel l'échéance de traitement d'une requête d'accès d'un programme critique correspond à la fin d'un créneau temporel alloué au programme critique qui est le premier à débiter après une date correspondant
30 à une date d'émission de la requête d'accès retardée par ajout de la marge de traitement du programme critique.

9. Procédé selon l'une des revendications 1 à 8, comprenant en outre la préemption au profit d'un programme critique d'un premier programme ayant émis une requête d'accès à la ressource partagée en attente de traitement, ladite préemption
5 comprenant le calcul d'une échéance de traitement de ladite requête correspondant à la fin d'un créneau temporel alloué au programme critique.

10. Circuit de multiplexage temporel (CMT) d'accès concurrents à une ressource partagée requis par des programmes informatiques, caractérisé en ce qu'il est configuré
10 pour mettre en œuvre le procédé selon l'une des étapes 1 à 9.

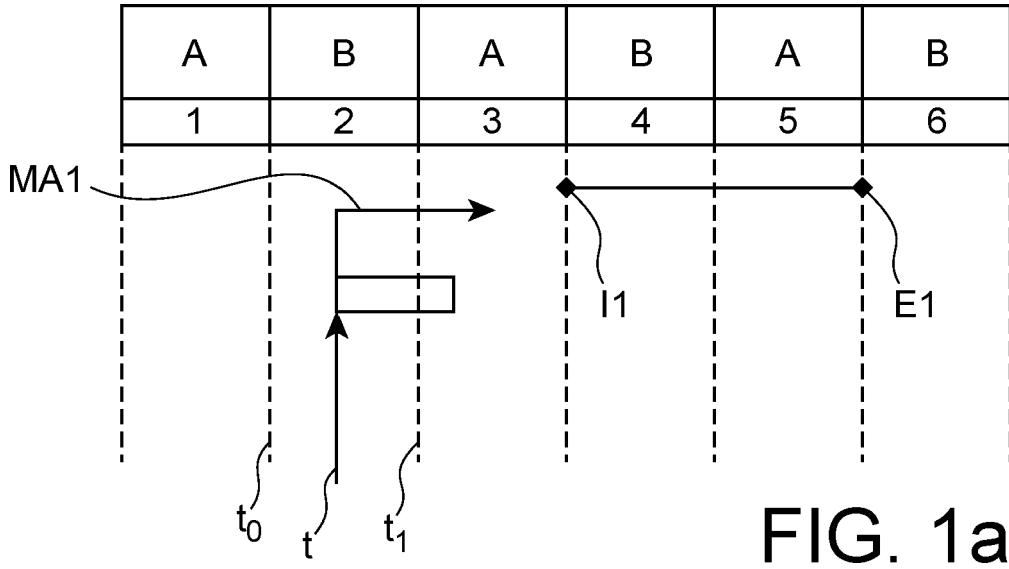


FIG. 1a

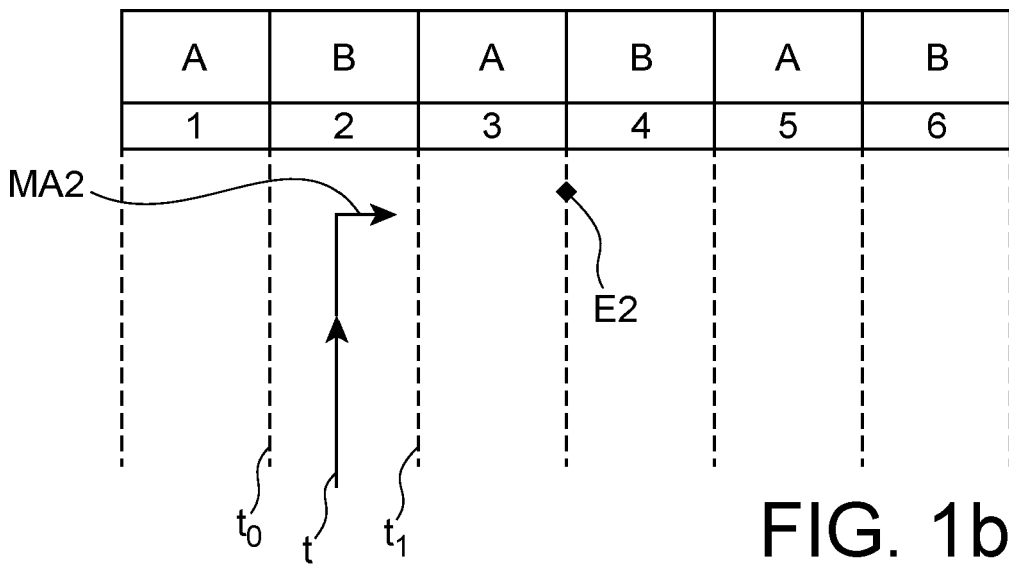


FIG. 1b

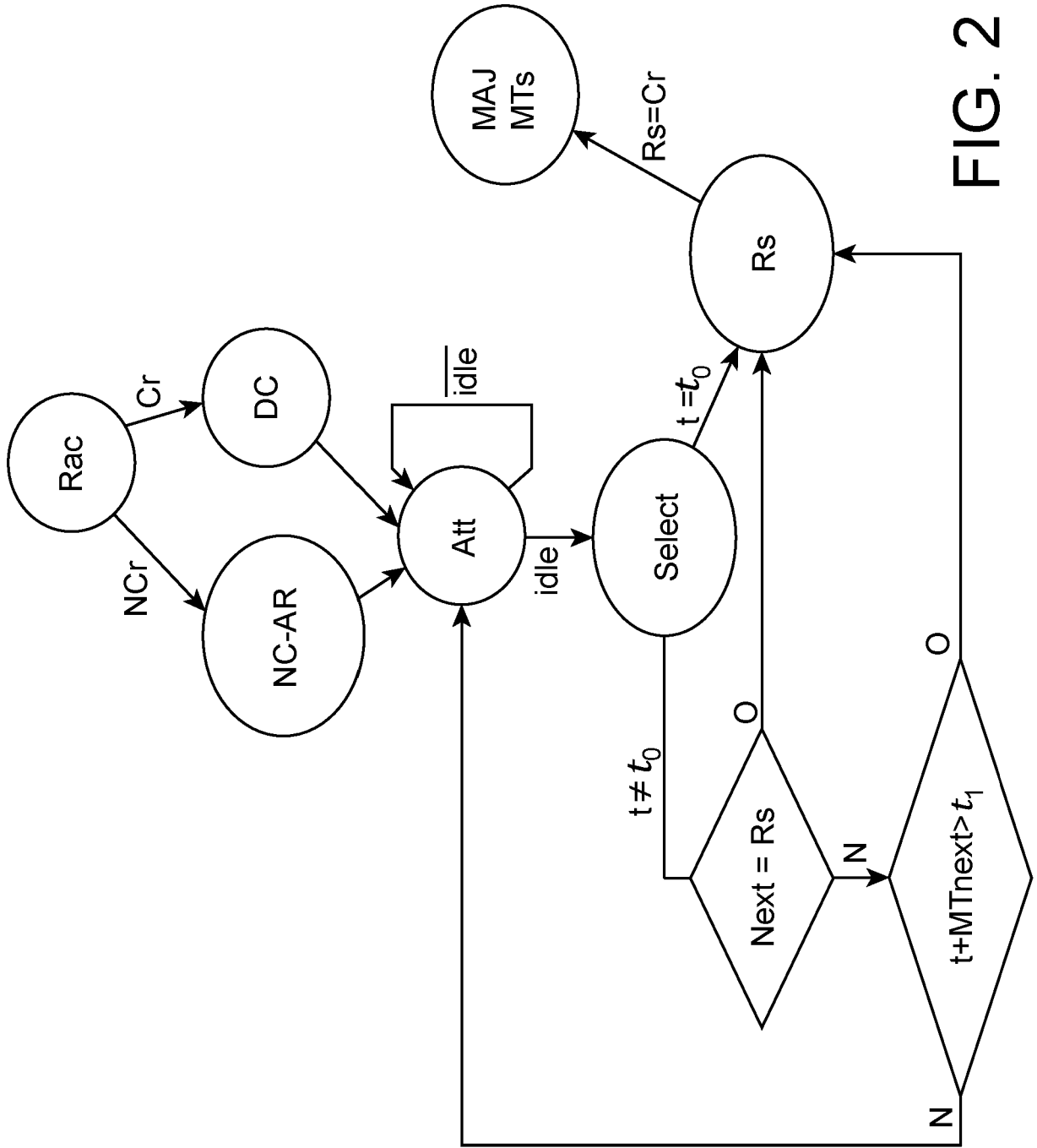


FIG. 2

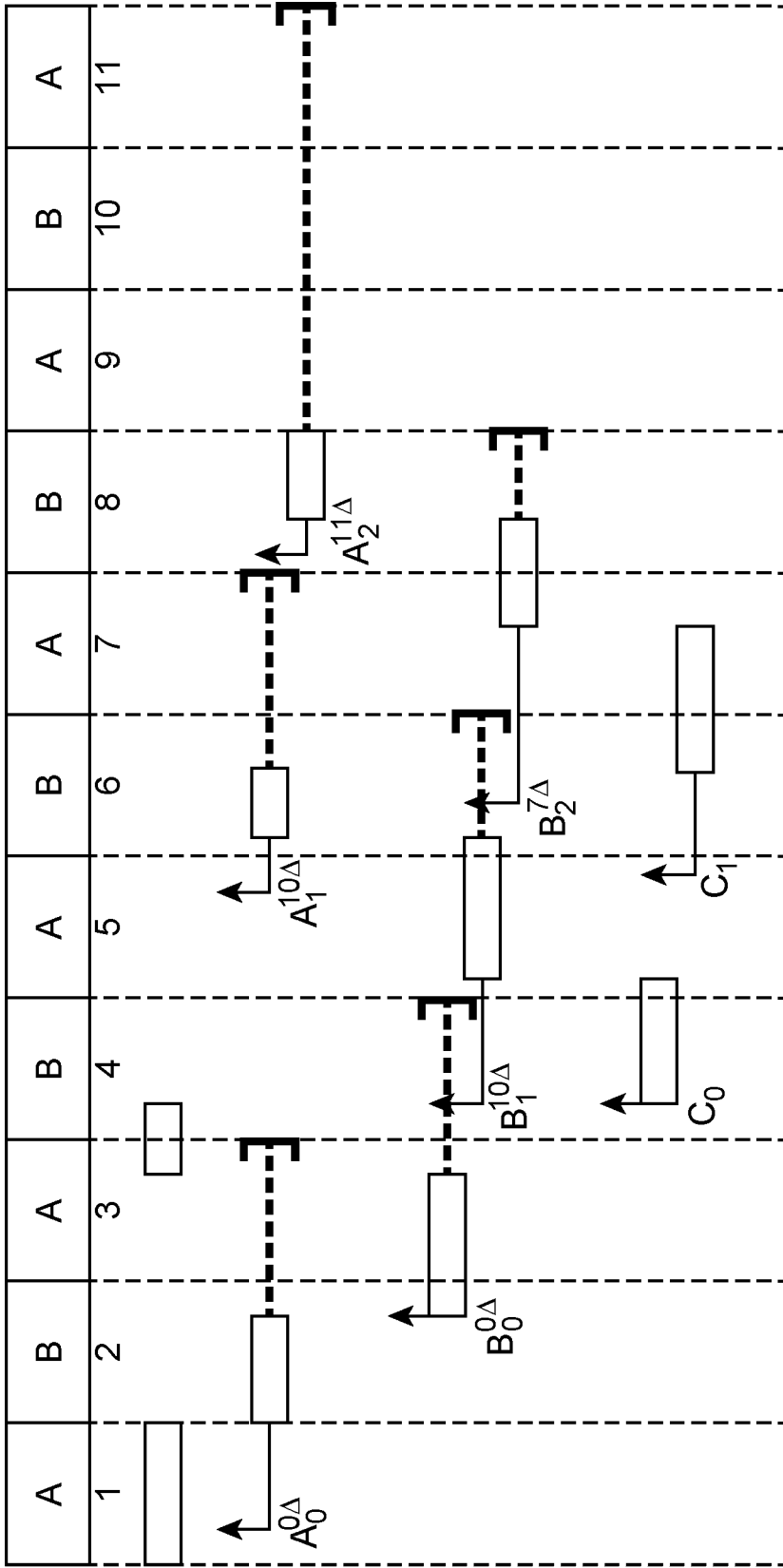


FIG. 3

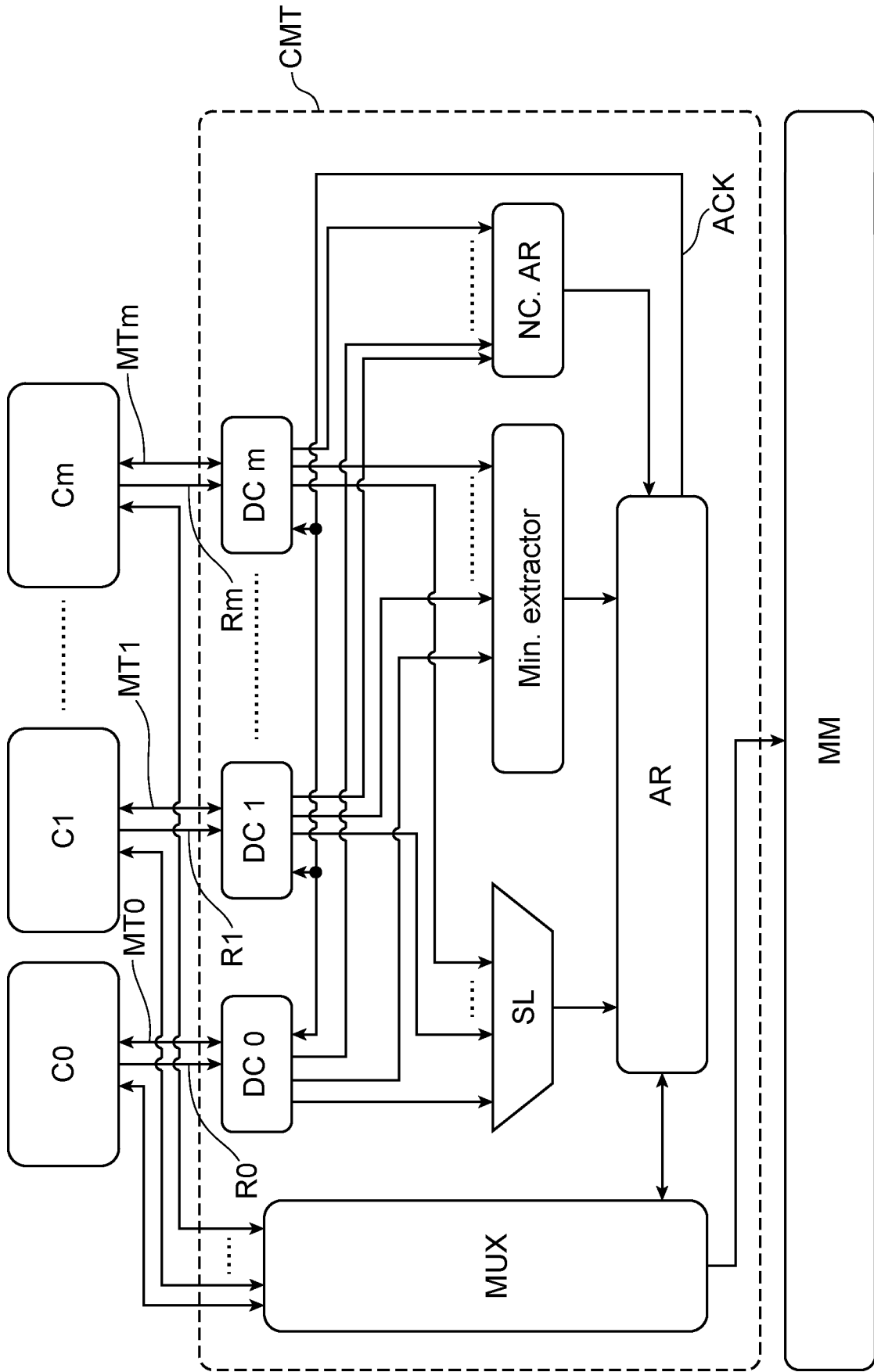


FIG. 4

**RAPPORT DE RECHERCHE
 PRÉLIMINAIRE**

 établi sur la base des dernières revendications
 déposées avant le commencement de la recherche

 N° d'enregistrement
 national

 FA 860758
 FR 1860117

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	EP 0 886 218 A2 (ADVANCED MICRO DEVICES INC.) 23 décembre 1998 (1998-12-23) * abrégé * * colonne 1, ligne 37 - colonne 2, ligne 2 * * colonne 4, lignes 29-44 * * colonne 4, ligne 55 - colonne 5, ligne 14 * * colonne 5, ligne 32 - colonne 7, ligne 38 * * colonne 7, ligne 55 - colonne 8, ligne 9 * * figure 3 *	1-10	H04L5/22
A	EP 1 550 953 A1 (SIEMENS AG) 6 juillet 2005 (2005-07-06) * le document en entier *	1-10	DOMAINES TECHNIQUES RECHERCHÉS (IPC)
A	US 2014/223114 A1 (LSI CORPORATION) 7 août 2014 (2014-08-07) * le document en entier *	1-10	
A	US 2009/217280 A1 (HONEYWELL INTERNATIONAL INC.) 27 août 2009 (2009-08-27) * le document en entier *	1-10	
Date d'achèvement de la recherche		Examineur	
27 juin 2019		Tomàs Blanch, F	
CATÉGORIE DES DOCUMENTS CITÉS		T : théorie ou principe à la base de l'invention	
X : particulièrement pertinent à lui seul		E : document de brevet bénéficiant d'une date antérieure	
Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie		à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.	
A : arrière-plan technologique		D : cité dans la demande	
O : divulgation non-écrite		L : cité pour d'autres raisons	
P : document intercalaire		
		& : membre de la même famille, document correspondant	

**ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE
RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 1860117 FA 860758**

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche préliminaire visé ci-dessus.

Les dits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du **27-06-2019**

Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets, ni de l'Administration française

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
EP 0886218	A2	23-12-1998	DE 69827879 T2	03-11-2005
			EP 0886218 A2	23-12-1998
			JP 4625549 B2	02-02-2011
			JP H1125036 A	29-01-1999
			US 6061361 A	09-05-2000

EP 1550953	A1	06-07-2005	AUCUN	

US 2014223114	A1	07-08-2014	AUCUN	

US 2009217280	A1	27-08-2009	AUCUN	
