



DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITE DE COOPERATION EN MATIÈRE DE BREVETS (PCT)

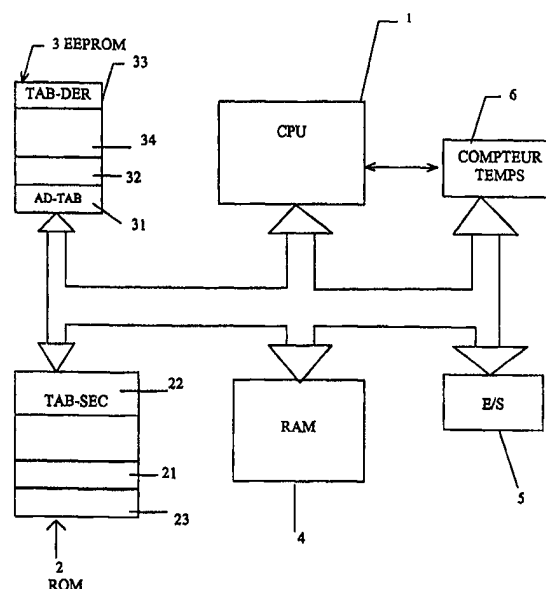
(51) Classification internationale des brevets ⁶ : G06F 9/32, 11/20, 9/26	A1	(11) Numéro de publication internationale: WO 98/57255 (43) Date de publication internationale: 17 décembre 1998 (17.12.98)
(21) Numéro de la demande internationale: PCT/FR98/01228 (22) Date de dépôt international: 12 juin 1998 (12.06.98) (30) Données relatives à la priorité: 97/07321 13 juin 1997 (13.06.97) FR (71) Déposant (pour tous les Etats désignés sauf US): BULL CP8 [FR/FR]; 68, route de Versailles, Boîte postale 45, F-78430 Louveciennes (FR). (72) Inventeur; et (75) Inventeur/Déposant (US seulement): NASSOR, Azad [FR/FR]; 12, chemin de Rambouillet, F-78450 Villepreux (FR). (74) Mandataire: CORLU, Bernard; Bull S.A., PC59C18, 68, route de Versailles, F-78434 Louveciennes Cedex (FR).		(81) Etats désignés: JP, US, brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Publiée <i>Avec rapport de recherche internationale.</i>

(54) Title: METHOD FOR MODIFYING CODE SEQUENCES AND RELATED DEVICE**(54) Titre:** PROCEDE DE MODIFICATION DE SEQUENCES DE CODE ET DISPOSITIF ASSOCIE**(57) Abstract**

The invention concerns a device for modifying code sequences written in a medium memory (2) comprising a central unit (1), said memory containing a main programme executable by the central unit (1) which also comprises a second non-volatile programmable memory (3), and a third working memory (4). The invention is characterised in that a rerouting table TAB_DER contained in the second programmable memory contains at least one field containing a new code sequence reference data, rerouting means for delayed rerouting of the executed code sequence towards the new code sequence written in one of the three memories and means in the new code sequence for returning to a point of the sequence before rerouting.

(57) Abrégé

La présente invention concerne un procédé de modification de séquences de code et le dispositif associé. Le dispositif de modification de séquences de code inscrites dans une mémoire (2) d'un support comprenant une unité centrale (1), ladite mémoire contenant un programme principal exécutable par l'unité centrale (1) qui comporte également une seconde mémoire (3) programmable non volatile, et une troisième mémoire de travail (4), est caractérisé en ce qu'une table de déroutement TAB_DER contenue dans la seconde mémoire programmable contient au moins un champ contenant une donnée de référence d'une nouvelle séquence de code, des moyens de déroutement permettant le déroutement différé de la séquence de code exécutée vers la nouvelle séquence de code inscrite dans une des trois mémoires et des moyens dans la nouvelle séquence de code permettant le retour en un point de la séquence de code exécutée avant le déroutement.



3...ELECTRICALLY-ERASABLE PROGRAMMABLE READ-ONLY MEMORY
 33...DEROUTING TABLE
 31...SYSTEM ZONE
 22...APPLICATION PROGRAMME
 2...READ-ONLY MEMORY
 1...CENTRAL PROCESSING UNIT
 4...RANDOM ACCESS MEMORY
 6...COUNTER
 5...INPUT/OUTPUT

UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	ES	Espagne	LS	Lesotho	SI	Slovénie
AM	Arménie	FI	Finlande	LT	Lituanie	SK	Slovaquie
AT	Autriche	FR	France	LU	Luxembourg	SN	Sénégal
AU	Australie	GA	Gabon	LV	Lettonie	SZ	Swaziland
AZ	Azerbaïdjan	GB	Royaume-Uni	MC	Monaco	TD	Tchad
BA	Bosnie-Herzégovine	GE	Géorgie	MD	République de Moldova	TG	Togo
BB	Barbade	GH	Ghana	MG	Madagascar	TJ	Tadjikistan
BE	Belgique	GN	Guinée	MK	Ex-République yougoslave de Macédoine	TM	Turkménistan
BF	Burkina Faso	GR	Grèce	ML	Mali	TR	Turquie
BG	Bulgarie	HU	Hongrie	MN	Mongolie	TT	Trinité-et-Tobago
BJ	Bénin	IE	Irlande	MR	Mauritanie	UA	Ukraine
BR	Brésil	IL	Israël	MW	Malawi	UG	Ouganda
BY	Bélarus	IS	Islande	MX	Mexique	US	Etats-Unis d'Amérique
CA	Canada	IT	Italie	NE	Niger	UZ	Ouzbékistan
CF	République centrafricaine	JP	Japon	NL	Pays-Bas	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norvège	YU	Yougoslavie
CH	Suisse	KG	Kirghizistan	NZ	Nouvelle-Zélande	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	République populaire démocratique de Corée	PL	Pologne		
CM	Cameroun	KR	République de Corée	PT	Portugal		
CN	Chine	KZ	Kazakstan	RO	Roumanie		
CU	Cuba	LC	Sainte-Lucie	RU	Fédération de Russie		
CZ	République tchèque	LI	Liechtenstein	SD	Soudan		
DE	Allemagne	LK	Sri Lanka	SE	Suède		
DK	Danemark	LR	Libéria	SG	Singapour		
EE	Estonie						

PROCEDE DE MODIFICATION DE SEQUENCES DE CODE ET
DISPOSITIF ASSOCIE

La présente invention concerne un procédé de modification de
5 séquences de code et le dispositif associé.

La présente invention concerne les programmes d'ordinateur et
notamment ceux destinés à être enregistrés dans un support de façon non
modifiable, du moins facilement. Ces supports sont intégrés dans un
système informatique comprenant entre autres, une unité centrale, une
10 mémoire de travail, une mémoire non volatile et des moyens d'entrée/sortie.
Plus spécifiquement, ce système informatique peut être incorporé dans une
carte à puce. Dans ce cas, la carte contient un circuit comportant au moins
un microprocesseur, une mémoire morte contenant un programme et
éventuellement des données, une mémoire de travail et une mémoire non
15 volatile programmable. Avantageusement, le circuit est conçu sous une
forme monolithique. La mémoire non volatile peut stocker des données et/ou
du code ; ainsi, le microprocesseur peut exécuter ce code de la même façon
que celui enregistré en mémoire morte. On trouve donc dans une carte deux
types de mémoire ; le contenu de la première est inscrit dès la fabrication du
20 circuit et non modifiable. Le contenu de la seconde est vierge au départ, les
valeurs seront inscrites au cours de l'utilisation normale de l'objet.

De nos jours, les cartes à puce peuvent répondre techniquement à de
nombreux besoins. Le programme incorporé dans la carte, appelé aussi
"système d'exploitation", permet d'adapter les fonctions de la carte à son
25 usage final. Actuellement, le système d'exploitation est stocké dans une
mémoire ROM qui est gravée lors de l'élaboration du circuit intégré. La
modification du programme en vue de répondre à de nouveaux besoins est
une opération longue qui pose un gros problème lorsque le client est pressé.

De plus, cette opération est très coûteuse ; c'est pourquoi beaucoup de "petits" clients qui désireraient acheter quelques milliers de cartes sont découragés et souvent se contentent d'une carte qui ne répond que partiellement à leur attente. Une solution consisterait à utiliser un masque
5 existant et à rajouter les fonctions que demande le client dans la mémoire programmable, ou à modifier celles existantes en ROM.

La possibilité d'introduire et d'exécuter un code supplémentaire en mémoire programmable offre l'avantage de pouvoir facilement rajouter de nouvelles fonctions à un programme ancien ou d'adapter d'anciennes
10 fonctions à des besoins spécifiques.

La demande de brevet FR 96/05454 de la demanderesse décrit un mécanisme particulier pour dérouter un programme lors de l'exécution de certaines instructions. La précédente invention consiste à établir à certains emplacements de la mémoire ROM, par des instructions respectives, des
15 points d'interrogation et des points d'orientation. Un point d'interrogation est référencé par un numéro et permet d'accéder à une routine dans la mémoire programmable s'il existe une séquence de code correspondant à l'adresse indiquée par ce numéro. Dans l'affirmative, un drapeau est positionné et une adresse de déroutement est mémorisée en mémoire RAM. Un point
20 d'orientation est actif si un point d'interrogation a préalablement été exécuté. Dans le cas favorable, le déroutement est déclenché en faisant effectuer au programme normal un saut à l'adresse programmée. La séquence de code à exécuter peut être en mémoire programmable ou en mémoire morte.

Ce mode de réalisation comporte cependant plusieurs problèmes, si
25 de nombreuses modifications dans le déroulement du programme doivent pouvoir être prises en compte. Dans ce dernier cas, il faut implémenter un grand nombre de points d'orientation dans la mémoire morte. A l'extrême limite, si l'on veut une grande adaptabilité, le programme contient plus de code permettant d'effectuer des déroutements que de code constituant le

programme principal. La multiplicité de ces points est un inconvénient majeur si la taille de la mémoire morte est limitée. D'autre part, le temps d'exécution s'en trouve augmenté proportionnellement au nombre de points. Si on limite le nombre de points de déroutement pour tenir compte des contraintes, le mode de réalisation perd en souplesse, car il ne permet pas
5 de dérouter un programme lors de l'exécution de n'importe quelle instruction.

La présente invention a pour objet un dispositif permettant de corriger certaines anomalies de fonctionnement d'un programme figé et permettre ainsi un déroulement correct, ou de rajouter facilement des fonctionnalités à
10 un programme existant, tout en optimisant la séquence de code à écrire.

Ce but est atteint par le fait que le dispositif de modification de séquences de code inscrites dans une mémoire d'un support comprenant une unité centrale capable d'exécuter ces séquences de code, ladite mémoire contenant un programme principal exécutable par l'unité centrale
15 qui comporte également une seconde mémoire programmable non volatile contenant éventuellement des nouvelles séquences de code exécutable, et une troisième mémoire de travail, est caractérisé en ce qu'une table de déroutement TAB_DER contenue dans la seconde mémoire programmable contient au moins un champ contenant une donnée de référence d'une
20 nouvelle séquence de code, des moyens de déroutement permettant le déroutement différé de la séquence de code exécutée vers la nouvelle séquence de code inscrite dans une des trois mémoires et des moyens dans la nouvelle séquence de code permettant le retour en un point de la séquence de code exécutée avant le déroutement.

25 Selon un autre but, la présente invention peut interrompre le déroulement normal d'un programme avant l'exécution de n'importe quelle instruction, même avec un nombre de points d'orientation limité.

Ce but est atteint par le fait que les moyens de déroutement consistent en des instructions d'orientation (IORi) activables et implantées

au préalable dans la mémoire contenant le code du programme principal, chaque instruction d'orientation étant associée à une référence i de la table de déroutement TAB_DER inscrite en mémoire programmable.

Selon une autre particularité, chaque instruction d'orientation (IORi)
5 activée déclenche l'exécution d'une séquence nouvelle de code comportant :

- des moyens de lire dans la table TAB_DER de la mémoire programmable un délai de temporisation ΔT_i correspondant à la référence de l'instruction d'orientation, ce délai de temporisation permettant de différer le déclenchement d'une interruption qui effectue un branchement à une
10 séquence nouvelle de code dont l'adresse (Adri) est indiquée dans la table, en association avec le délai de temporisation,

- des moyens de mémoriser l'adresse (Adri) dans une mémoire du dispositif et

- des moyens de lancer un compteur de temps du dispositif pour
15 décompter le temps nécessaire au délai de temporisation du branchement.

Selon un autre but, la présente invention permet de masquer certaines opérations dites sensibles effectuées par l'unité centrale.

Ce but est atteint par le fait que le dispositif de modification de séquences de code comporte une seconde table TAB_SEC mémorisée dans
20 la mémoire du dispositif et associant à chaque point (i) de déroutement un intervalle de temps $[\Delta T_{\text{mini}} ; \Delta T_{\text{maxi}}]$ associé au délai de temporisation ΔT_i préalable à l'exécution d'une nouvelle séquence de code, et des moyens de vérification que le délai de temporisation est autorisé par l'intervalle de temps associé fourni par cette table.

25 Selon une autre particularité, le dispositif de modification de séquences de code comporte des moyens permettant le décalage du délai ΔT_i de temporisation de la valeur de l'intervalle de temps $[\Delta T_{\text{mini}} ; \Delta T_{\text{maxi}}]$.

Selon une autre particularité, le dispositif de séquences de code comporte des moyens de déclencher un message d'erreur lorsque le délai de temporisation ΔT_i est dans l'intervalle de temps.

5 Selon une autre particularité, le dispositif de séquences de code comporte, consécutivement à la fin du délai de temporisation (ΔT_i), lorsque le compteur atteint la valeur nulle, des moyens de déclencher une interruption, des moyens de mémorisation de la valeur actuelle d'un registre compteur programme PC dans une pile, puis des moyens de dérouter le programme à l'adresse définie dans la partie de la mémoire ROM contenant
10 des vecteurs d'interruption, qui fournissent l'adresse du début de la séquence de code de l'interruption, des moyens de vérification que la valeur du registre compteur programme PC Val_PC mémorisée dans la pile n'est pas une valeur d'adresse de séquence sensible contenue dans une table TAB_SEC, et des moyens de modifier le déroulement des opérations.

15 Selon une autre particularité, les moyens de vérification testent si la valeur du registre compteur programme est contenue par TAB_SEC dans l'intervalle [Adrdeb_i , Adrfin_1], correspondant à une interruption du programme pendant une séquence sensible et les moyens de modifier le déroulement des opérations de la carte renvoient un message indiquant que
20 sa sécurité est atteinte et se bloque, soit les moyens de vérification testent si la valeur du registre compteur programme Val_PC est contenue dans l'intervalle]Adrfin_i , Adrdeb_i+1[correspondant à une interruption du programme pendant une séquence non sensible et les moyens de modifier le déroulement des opérations autorisent le programme à exécuter alors la
25 nouvelle séquence de code dont l'adresse de début a été mémorisée lors du traitement de l'instruction d'orientation (IORi).

Selon une autre particularité, le dispositif de modification de séquences de code comporte une source de fréquence pour le compteur
30 différente d'une source de fréquence permettant à l'unité centrale d'exécuter le programme, la valeur du délai de temporisation (ΔT_i) programmée dans la

table de déroutement TAB_DER étant calculée pour permettre d'interrompre le programme à une adresse déterminée, la table TAB_DER comporte pour chaque valeur du délai de temporisation, un élément supplémentaire contenant cette adresse déterminée et des moyens de comparer l'adresse
5 de l'instruction interrompue par l'interruption, à celle indiquée dans la table, et de provoquer une alarme.

Selon une autre particularité, le dispositif de séquences de code comporte des moyens de déclencher une alarme pour bloquer le support et indiquer une tentative de fraude par une écriture dans la mémoire.

10 Selon une autre particularité, chaque séquence nouvelle de code se termine par une instruction d'orientation pour recharger le compteur avec une nouvelle valeur de délai de temporisation (ΔT_i).

Un dernier but est un procédé de modification de séquences de code figées inscrites dans un support comportant une unité centrale et une
15 mémoire.

Ce but est atteint par le fait que le procédé de modification de séquences de code figées inscrites dans un support comportant une unité centrale et une mémoire consiste à prévoir dans au moins une séquence de code figée, au moins une instruction d'orientation (IOR_i), permettant de
20 dérouter l'exécution par une interruption différée d'un délai de temporisation du programme contenu dans la mémoire de vers une adresse déterminée, par une table de déroutement TAB_DER, en fonction d'une référence *i* associée à l'instruction d'orientation et dans un délai de temporisation, déterminé par le contenu d'une ligne de table correspondant à la référence *i*
25 de l'instruction d'orientation, une séquence nouvelle de code exécutable lors de l'interruption générée à la fin du délai de temporisation étant implantée à l'adresse contenue dans la table (TAB_DER).

Selon une autre particularité, une étape déclenchant l'interruption est précédée d'une étape de de vérification et le délai de temporisation n'est

pas inclus dans un intervalle défini par une seconde table TAB_SEC dite sécuritaire inscrite dans la mémoire non volatile du support.

D'autres caractéristiques et avantages de la présente invention apparaîtront plus clairement à la lecture de la description ci-après faite en
5 référence aux dessins annexés dans lesquels :

la figure 1 représente une vue schématique des circuits électroniques nécessaire à la mise en oeuvre de la présente invention ;

la figure 2A représente le schéma temporel des séquences de code nécessaire à la mise en oeuvre de la présente invention ;

10 la figure 2B représente le logigramme de la séquence de code correspondant à l'activation d'un point de déroutement ;

la figure 2C représente la séquence de code correspondant au traitement de l'interruption générée par le compteur ;

la figure 2D représente la table de déroutement TAB_DER ;

15 la figure 3 représente un diagramme temporel de l'imbrication des séquences de code du programme d'application de traitement de l'instruction d'orientation et de traitement de l'interruption, dans un exemple d'application de l'invention ;

la figure 4A représente la table de sécurité TAB_SEC ;

20 la figure 4B représente la modification du logigramme de la séquence de déroutement dans le cas d'une utilisation d'une table de sécurité conjointement avec la table de déroutement de l'invention.

La figure 5 représente le logigramme du programme d'écriture dans la table de déroutement.

La présente invention va maintenant être explicitée à l'aide d'un exemple qui se situe dans le domaine des "cartes à puce" et plus particulièrement des cartes à microprocesseurs. Ces cartes possèdent généralement un circuit intégré dont le schéma général est représenté dans la figure 1. Ce circuit dispose d'une Unité Centrale (1) reliée par un bus d'adresse et de donnée à une mémoire non volatile par exemple de type ROM (2), contenant le programme principal, une mémoire programmable non volatile de type EEPROM (3), une mémoire de travail RAM (4), une interface d'entrée/sortie I/O (5) et un compteur (6). Le compteur peut générer une interruption et interrompre ainsi le déroulement normal du programme exécuté par l'Unité Centrale. L'interruption est associée à un vecteur. Ce dernier est l'adresse de début de la routine d'interruption. C'est l'adresse de la première instruction de la séquence qui gère l'interruption. De façon intégrée ou externe au compteur, un moyen d'inhibition de l'interruption permet de retarder et même d'annuler l'interruption. Celle-ci sera alors prise en compte plus tard ou pas du tout.

La mémoire programmable (3) est divisée en plusieurs parties. Une première partie, appelée zone système, contient des informations systèmes non lisibles de l'extérieur, cette partie contient notamment les valeurs des pointeurs permettant de délimiter chacune des autres parties de la mémoire. Une seconde partie (32) est notée zone de donnée, elle est accessible de l'extérieur et contient principalement les données utilisateurs. Une troisième partie (33) contient une table d'orientation TAB_DER, cette table contient des éléments aux formats identiques composés d'au moins trois champs (N°Réf, ΔT_i et Adri). Une quatrième partie (34), notée zone de séquences, contient les séquences de code pouvant être appelées par le programme principal à partir des numéros de référence (N°Réf) du point d'orientation et de l'adresse (Adri) lue dans la table. Il est à noter que dans une variante de l'invention, la table d'orientation, ou une ou plusieurs séquences de code,

peuvent être chargées en mémoire de travail RAM plutôt qu'en mémoire programmable non volatile.

L'adresse de début, appelée AD_TAB, de la table d'orientation TAB_DER est mémorisée dans la zone système (31). Un emplacement
5 précis est prévu pour contenir cette valeur. L'écriture effective de cet emplacement constitue en soi l'indication que la table est bien présente et que les points d'orientation peuvent être opérationnels.

La mémoire ROM(2) qui contient, entre autres, le programme principal, est divisée en trois parties. La première (21) réalise l'initialisation
10 du programme lors d'une mise sous tension. La seconde partie (22) contient le programme d'application. La troisième partie (23) contient du code "dormant" dont le rôle est expliqué par la suite.

Lorsque le programme arrive à un point de déroutement ou d'orientation (220), il exécute une séquence de déroutement représentée à
15 la figure 2B, laquelle consiste en une première étape dans laquelle le programme examine si une interruption est déjà en cours. Dans le cas affirmatif, le programme est avorté. Dans le cas négatif, le programme se poursuit par l'étape 2202, à laquelle il examine si un indicateur correspondant au point de déroutement i est actif.

20 Dans la négative, le programme fait un traitement d'erreur du type décrit dans la demande précédente. Dans l'affirmative, le programme se poursuit par l'étape de recherche dans la table TAB_DER inscrite dans la troisième partie (33) de la mémoire EEPROM, la valeur (ΔTi) de programmation du compteur correspondant au déroutement i et l'adresse de
25 saut (Adrsi).

L'étape suivante 2205 permet le chargement du compteur de temporisation (6) avec la valeur (ΔTi) de temporisation donnée par la ligne i de la table TAB_DER. Puis l'étape suivante du programme 2206 permet le

lancement du compteur de temporisation. Sur certains microprocesseurs l'écriture d'une nouvelle valeur dans le compteur entraîne son activation, et dans ce cas les deux étapes 2205 et 2206 sont confondues. Après cette étape, la séquence de déroutement se rebranche par l'étape 2207 à l'instruction suivante du programme qu'elle était en train d'exécuter
5 lorsqu'elle a rencontré une instruction (IOR) de déroutement ou d'orientation, matérialisant un point de déroutement.

Lorsque le compteur, lancé à l'étape 2206, a écoulé le temps (ΔT_i) fourni par la ligne i de la table TAB_DER, le compteur (6) déclenche sur le microprocesseur (1) une interruption qui donne lieu au traitement d'une
10 séquence d'interruption (221 ; figure 2A). Cette séquence d'interruption, dont le détail est donné à la figure 2C, commence par une étape d'empilement éventuel du contexte de saut (2210) dans la mémoire RAM(4). Puis le traitement de l'interruption se poursuit par une étape (2211) d'exécution de la nouvelle séquence de code contenue, soit dans la quatrième partie (34)
15 de l'EEPROM(3), soit dans la troisième partie (23) de la ROM, à l'adresse mémorisée précédemment à l'étape 2204. Puis l'interruption se termine par l'étape (2212) d'utilisation éventuelle du contexte empilé pour revenir au programme principal, soit à l'instruction suivante (2212A) de celle exécutée avant l'interruption en utilisant, par exemple, une instruction RTI de retour
20 d'interruption, soit en faisant un saut (par exemple, par une instruction JUMP), comme représenté par la référence 2212B de la figure 2A.

On comprend ainsi que par l'enregistrement d'une table dans l'EEPROM, d'une séquence d'interruption dans la ROM et d'une séquence
25 de déroutement dans une mémoire volatile ou non, on peut intervenir en tout point du programme d'application mémorisé dans la ROM, et modifier ou ajouter des fonctionnalités nouvelles par la mise en oeuvre des parties de code dormant, ou de nouvelles parties de code inscrites en EEPROM par exemple.

Le point d'orientation existait avant la partie de programme que l'on veut maintenant modifier. Ce point d'orientation est doté d'un numéro de référence qui lui est propre, dans le cas présent "1".

La table TAB_DER a été introduite en mémoire programmable et
5 l'entrée correspondant au numéro de référence « 1 » renseigné de la façon suivante par exemple :

N° du point d'orientation	valeur du compteur	adresse de saut
1	$\Delta T1$	Adr_1

$\Delta T1$ a été calculée pour que cela corresponde à la durée nécessaire pour interrompre le déroulement du programme principal à l'adresse désirée Adr_2 (figure 3). Pour faire ce calcul, on prend en compte le nombre
10 d'instructions pour « atteindre » l'adresse Adr_2 et la fréquence de décompte du compteur.

Dans un premier temps, la présence de la table d'orientation est testée. On a dit précédemment que cela peut revenir à tester l'écriture ou non de l'emplacement contenant l'adresse de début de la table TAB_DER.
15 Ensuite, la présence d'une interruption en cours du compteur est testée. Si le compteur est en cours, il faut interdire l'exécution du point de déroutement, seule la séquence de code correspondant à un précédent point d'orientation sera exécutée. Sinon, le programme recherche les valeurs écrites dans les deuxième et troisième éléments correspondant au
20 point d'orientation numéro 1. Les valeurs $\Delta T1$ et Adr_1 sont extraites respectivement du premier et second champ de l'élément. Le registre de donnée du compteur est alors chargé avec la valeur $\Delta T1$ et, le vecteur d'interruption associé à l'interruption provoquée par le compteur est chargé avec la valeur Adr_1. Enfin, juste avant de sortir de la séquence du point
25 d'orientation, le compteur est lancé et la valeur stockée dans son registre de donnée décroît en fonction du temps.

La valeur ΔT_1 doit être déterminée avec une grande précision, elle dépend directement du nombre de cycles qui sépare le moment des exécutions de l'instruction de lancement du compteur et de la première instruction que l'on ne veut pas exécuter. Si le compteur est synchronisé
5 avec l'horloge de l'Unité Centrale, il suffit d'additionner le nombre de cycles de chacune des instructions séparant les deux instructions précédemment nommées. Si le compteur n'est pas synchronisé avec la même fréquence que l'U.C. le calcul étant approximatif est délicat.

Si la source de fréquence du compteur est différente de celle
10 permettant de dérouler le programme, alors le procédé de l'invention permet d'effectuer par exemple une tâche contrôle sécuritaire. En effet, les valeurs programmées dans la table TAB_DER permettent d'interrompre à une adresse déterminée, laquelle adresse a été indiquée comme élément supplémentaire dans la table TAB_DER, et une fois l'interruption active, on
15 peut comparer l'adresse de l'instruction interrompue à celle indiquée dans la table. Si elles ne sont pas égales, alors on peut conclure qu'il y a un dérèglement des sources de fréquence, dû peut être à une tentative de fraude, et le programme peut agir de la façon qui convient (en se bloquant par exemple).

20 Une façon préférée de réaliser l'invention consiste à élaborer une table (figure 2D) en mémoire programmable, le premier élément de cette table est le numéro de référence du point d'orientation, le second élément est une valeur de chargement du compteur et, le troisième élément est l'adresse de la séquence de code à exécuter.

25 Cette table (figure 2D) possède une taille maximale de 30 octets répartis en 6 lignes de 5 octets. Le premier élément de la table comprend un octet, ce qui permet d'installer dans le code ROM, 255 points d'orientation, numérotés de 1 à 255. La valeur zéro indique qu'il n'y a plus de valeur et la fin de table est alors atteinte. La seconde valeur s'exprime sur deux octets,

- ce qui autorise 65536 valeurs différentes de chargement du compteur. Enfin, l'adresse de saut est notée sur deux octets, valeur traditionnelle pour des microprocesseurs de type carte à puce. Cette adresse doit permettre de faire exécuter des séquences de codes écrites en mémoire programmable
- 5 EEPROM aussi bien que du code en ROM (code dormant).

- On peut remarquer que la sixième ligne de cette table TAB_DER (figure 2D) est à "00", il n'y a donc, dans le cas présent que 5 points d'orientation opérationnels : Les n°1,3 4 (2 entrées) et 6. Les points d'orientation 3 et 6 ont la même adresse de saut Adr_3, ce qui veut dire que
- 10 la modification de programme est la même pour ces deux points. Trois séquences de déroutement sont donc prévues. Ces séquences de déroutement peuvent être mémorisées dans la mémoire programmable et dans une partie du code ROM, ce code "dormant" pouvant être ainsi exécuté à l'aide des points d'orientation.

- 15 Une autre façon de gérer la table d'orientation, qui évite au programmeur de prévoir en mémoire une information correspondant à la taille de la table, consiste à prévoir un champ supplémentaire, qui contient la valeur d'adresse de l'élément suivant de la table d'orientation. Pour plus de détails, on pourra se référer à la demande de brevet référencée plus haut.

- 20 Avantageusement, une séquence nouvelle de code peut se terminer par un point d'orientation ce qui permet de recharger le compteur et donc de rajouter une nouvelle temporisation, comme le montre le tableau précédent pour le numéro de référence « 4 ». Cela est particulièrement utile pour réaliser des temporisations très longues qui dépassent la capacité du
- 25 compteur. On implémente alors dans la séquence de code, la gestion d'un compteur d'interruption qui est initialisé à une certaine valeur de départ, cette valeur représente le nombre de temporisations maximales à effectuer avant de programmer dans le compteur du temporisateur la valeur inscrite dans TAB_DER. A la suite de cette dernière temporisation, la partie de la

séquence de code relative à la modification du programme principal est exécutée.

Un exemple d'application de l'invention est représenté à la figure 3, dans laquelle un programme d'application écrit en mémoire ROM peut
5 contenir un certain nombre d'instructions d'interrogation (Ili), suivi de séquences de code les séparant de l'instruction d'orientation (IOi) correspondante, selon le principe de la demande de brevet français n° 96/05454 déposée par la demanderesse.

Ces instructions d'interrogation (Ili) permettent de déterminer si
10 l'instruction d'orientation (IORi) correspondante est active ou non, et dans le cas où elle est active, lorsque le programme arrive sur l'instruction d'orientation, de dérouter le programme vers un point d'orientation (i) ou déroutement, dont le numéro de référence pour l'instruction d'orientation (IOi) est i.

15 L'exécution de l'instruction d'orientation (IORi) déclenche plusieurs opérations. Elle déclenche d'abord la lecture de la table de dérivation ou de déroutement TAB_DER inscrite dans la zone 33 de la mémoire EEPROM pour déterminer si, à l'adresse i de cette table, les valeurs ΔT_1 de compteur et d'adresse (Adri) de saut ont été remplies et dans le cas où ces valeurs
20 sont présentes, les informations correspondantes ΔT_i valeur de compteur et Adri adresse de saut, sont mémorisées temporairement, par exemple, dans la mémoire RAM(4). Puis, l'instruction d'orientation se termine par le lancement du compteur de temps 6.

Une fois l'instruction d'orientation exécutée, le programme
25 d'application continue à dérouler sa séquence dans l'ordre, arrive à l'exécution de l'instruction repérée par l'adresse Adr_1 correspondant à l'exemple de code exécutable figurant en annexe 1, exécute la série d'instruction de cette séquence, puis arrive à l'instruction repérée par l'adresse Adr_2, pour laquelle on a choisi de modifier la valeur

correspondante utilisée dans l'opération ultérieure de l'instruction de la séquence du programme normalement enregistré en mémoire ROM par une autre valeur. Pour cela on détermine ΔT_1 de façon que l'interruption se produise avant que le programme exécute l'instruction de l'adresse (Adr_2) et l'interruption déroute le programme vers une séquence de code à l'adresse Adri. Ainsi, dans l'exemple donné du programme de l'annexe 1, on veut remplacer la valeur 10 par la valeur 20 dans la multiplication qui est effectuée ensuite à l'adresse Adr_3.

Ceci est obtenu par la séquence de code figurant à l'annexe 2. On conçoit ainsi que l'accumulateur B a été chargé avec la valeur 20 à la place de la valeur 10, ce qui va changer le résultat de la multiplication sans avoir à modifier le programme inscrit en ROM.

Ainsi par des inscriptions de séquences nouvelles de code en EEPROM dans la zone 34, comme celle représentée à la figure 3, dans le carré 341, par l'inscription d'une table de déroutement TAB_DER dans la zone 33 de l'EEPROM (3) et, par la mise en place d'instructions d'orientation dans le programme d'exploitation (Operating System) mémorisé dans la ROM de la carte à puce, on va pouvoir intervenir et modifier à volonté toutes les instructions par le choix de la temporisation ΔT_i et de l'adresse d'intervention Adri, à laquelle va se trouver la nouvelle séquence de code inscrite dans une des mémoires de la carte.

Le début de cette partie de programme de l'annexe 1 comporte l'appel à un point d'orientation, numéroté "01". Des instructions existent ensuite. Puis, une séquence particulière est décrite. Tout d'abord, la valeur de l'octet pointé par X est multipliée par la valeur 10, le résultat sur deux octets est stocké dans deux registres, puis le sous-programme d'écriture en mémoire programmable est exécuté. Le programme étant figé en ROM il n'est plus modifiable, or il s'avère que la valeur "10" doit être changée en "20". Nous

allons maintenant décrire comment la présente invention peut résoudre ce problème.

L'Unité Centrale continue alors l'exécution du programme principal (PP ; figure 3). Au moment où le microprocesseur va lancer l'exécution de l'instruction de l'adresse notée adr_2, le registre de donnée du compteur (6)
5 initialisé au point de déroutement avec la valeur ΔT_i , atteint la valeur nulle, ce qui déclenche une interruption (IT ; figure 3) qui permet le branchement de l'adresse Adri. L'instruction "LDB #010d" n'est donc pas exécutée. L'Unité Centrale est déroutée vers l'adresse Adri où l'on trouve la séquence de code
10 qui figure en annexe 2 :

Cette petite séquence de code décrite dans l'annexe 2 permet de charger la valeur "20" dans le registre B qui sert à la multiplication, puis de revenir dans le programme principal, c'est à dire l'adresse adr_3, immédiatement après l'instruction que l'on veut modifier ou ne pas exécuter.
15 L'instruction de multiplication à cette adresse prend en compte non plus la valeur "10" contenue à l'instruction précédente dans le code ROM, mais la valeur "20" contenue dans la nouvelle séquence de code.

La variante de l'invention ci-après prend en compte les besoins sécuritaires. Pour des raisons sécuritaires, il peut être important de ne pouvoir arrêter l'exécution d'une partie de programme par une interruption.
20 C'est le cas par exemple d'une authentification avec calcul cryptographique et comparaison entre des valeurs reçues et calculées. Ainsi, tant que la clé secrète est en mémoire de travail et donc visible par une séquence de déroutement, il ne faut pas autoriser d'interruptions. Ceci est possible en
25 inhibant au cours de l'exécution d'un programme "sensible" toutes les interruptions ou au moins celles du compteur.

L'inhibition des interruptions et leur réactivation s'effectuent chacune par une instruction interprétable par la quasi totalité des microprocesseurs. Ces instructions modifient généralement un bit du registre d'état du

microprocesseur qui, tant qu'il est actif, empêche le déclenchement de l'interruption, et tant qu'il est inactif, autorise les interruptions. La présence d'une instruction d'inhibition d'interruption en mémoire ROM avant une séquence sensible a pour conséquence que la séquence sensible est
5 définitivement protégée si l'instruction de réactivation est placée après la séquence sensible.

Une autre méthode pour interdire le déclenchement des interruptions lors de l'exécution d'une séquence sensible est de tester, lors de l'exploitation de la table TAB_DER, et ceci pour chaque point d'orientation,
10 la valeur de programmation du timer. Une table TAB_SEC est donc prévue en mémoire ROM ou en mémoire E²PROM verrouillée qui comprend pour chaque point d'orientation du programme principal, un couple de valeur [ΔT_{mini} , ΔT_{maxi}] définissant un intervalle dans lequel toute valeur de programmation du compteur est interdite. Avantageusement, la table
15 TAB_SEC peut comprendre plusieurs couples [ΔT_{mini} , ΔT_{maxi}] pour le même point d'orientation, comme représenté à la figure 4A.

La présence de la table TAB_SEC en ROM fige les séquences sensibles, comme précédemment, alors que la présence en E²PROM permet de modifier les zones de séquence sensible jusqu'au verrouillage par une
20 clé.

Pour les points d'orientation 3 et 5, deux intervalles de temps sont programmés.

Lors de l'exploitation des informations dans la table TAB_DER, le système d'exploitation teste si la valeur ΔT_i à écrire correspondant au point
25 d'orientation i n'appartient pas à des intervalles [ΔT_{mini} , ΔT_{maxi}]. Si c'est le cas, une séquence sensible serait interrompue, cela étant interdit, le système d'exploitation refuse l'écriture et renvoie un message d'erreur. Si par contre, correspondant à ce point d'orientation, la valeur ΔT_i n'est pas

incluse dans l'intervalle de temps non autorisé, l'écriture est menée à bien et le point d'orientation devient opérationnel.

Dans la variante où on utilise une table sécuritaire TAB_SEC, la séquence de déroutement correspond à la figure 4B. Cette séquence
5 comporte, outre les étapes de la figure 2B, les étapes suivantes, ajoutées entre l'étape 2204 et l'étape 2205 de la figure 2B :

- une première étape 22041, qui consiste à lire la table de sécurité TAB_SEC à la ligne i pour déterminer l'intervalle $[\Delta T_{\min i}, \Delta T_{\max i}]$, et vérifier, par exemple, que la valeur ΔT_i d'initialisation du compteur de
10 temporisation n'est pas incluse dans cet intervalle $[\Delta T_{\max i}, \Delta T_{\min i}]$. La logique opposée est également possible. Si la valeur ΔT_i n'est pas incluse, on poursuit par l'étape 2205. Si la valeur (ΔT_i) est incluse, on poursuit, soit par l'étape 22043A qui permet l'affichage d'un message d'erreur, soit par l'étape 22043B, qui exécute une instruction permettant de recharger le timer
15 avec la valeur (ΔT_i) augmentée de l'intervalle. Ceci diffère le déclenchement de l'interruption de la longueur de l'intervalle.

Cette dernière variante permet malgré tout de déclencher les interruptions, mais de ne pas pouvoir intervenir sur des séquences du programme d'application qui doivent être protégées car faisant partie des
20 portions sécuritaires.

Pour le reste, la séquence de déroutement se poursuit de la même façon que pour la figure 2B.

Enfin, le programme d'application ou une partie du code dormant peut inclure le programme d'écriture qui peut être appelé, par exemple, par une
25 orientation à une adresse fixe initiale, de façon à permettre par l'orientation à ce programme d'écriture, l'écriture de la table de déroutement TAB_DER selon la séquence décrite à la figure 5.

- Ce programme commence par une étape de réception d'ordre d'inscription d'un ou plusieurs éléments dans la table TAB_DER, les éléments ΔT_i , et Adri devant être écrits à la ligne i de l'élément dans la table. Cette étape est poursuivie par une étape de test sur le drapeau d'écriture de la table ECA pour vérifier s'il est actif ou inactif. Dans le cas où ECA est
- 5 actif, le programme se poursuit par un message d'erreur à l'étape 61. Dans le cas où ECA est inactif, le programme se poursuit par une étape 53 d'analyse de chaque élément ΔT_i par rapport à la valeur correspondante dans la table de sécurité TAB_SEC. Cette étape comporte un test
- 10 déterminer si la valeur ΔT_i est dans l'intervalle $[\Delta T_{\max i}, \Delta T_{\min i}]$. Dans l'affirmative, le programme se poursuit à l'étape 22043B par une modification du ΔT_i représentant la valeur de l'intervalle, de façon à différer l'interruption pour ne pas tomber dans la séquence interdite. Une variante représentée en parallèle consiste à envoyer un message d'erreur 22043A.
- 15 Ceci est représenté à l'étape 59 par l'opération consistant à remplacer (ΔT_i) par la valeur ($\Delta T_i + x$), x étant l'intervalle ($\Delta T_{\max i} - \Delta T_{\min i}$). Dans le cas où le ΔT n'est pas dans l'intervalle ou après modification de la valeur ΔT_i , le programme se poursuit par l'étape 55, où on positionne à l'état actif, le drapeau ECA d'écriture de la table.
- 20 Cette étape effectuée, le programme se poursuit par l'étape 56 de mise à jour de la table de déroutement et de vérification d'écriture. Cette vérification s'effectue par un test représenté à l'étape 57, dans le cas où le test confirme l'écriture correcte, le programme se poursuit par l'examen de savoir s'il y a un autre élément à écrire, dans la négative, le programme se
- 25 termine à l'étape 62, dans l'affirmative, le programme se reboucle à l'étape 53.

Si le test de vérification est négatif, le programme se poursuit par l'étape 60 de positionnement du drapeau d'écriture à l'état inactif.

Dans une variante de l'invention par la table TAB_DER il est même possible de protéger certaines parties de codes sans gêner les possibilités de modification des autres parties.

Une troisième variante consiste à stocker dans la table TAB_SEC en
5 ROM les adresses début et fin associées à chacune des séquences sensibles. Les valeurs d'adresses se déterminent aisément et ne sont pas le résultat d'un calcul pour déterminer une durée comme précédemment.

Par exemple

TAB_SEC : où chaque ligne représente les adresses de début et de fin des
10 séquences sensibles

Adresse de début	Adresse de fin
Adrdeb_1	Adrfin_1
Adrdeb_2	Adrfin_2

Avantageusement, les fenêtres d'adresses [Adrdeb_i , Adrfin_1] sont mémorisées par adresses croissantes, ceci facilitant la lecture de la table. Par conséquent, les valeurs situées dans les intervalles]Adrfin_i , Adrdeb_i+1[sont des valeurs d'adresses du programme non sensible.

15 La vérification de l'interruption d'une séquence sensible s'effectue consécutivement à la fin de la durée de temporisation. Lorsque le compteur atteint la valeur nulle, une interruption est déclenchée et la valeur actuelle du PC est mise dans la pile, puis le programme se dérouté à l'adresse définie dans la partie de la mémoire ROM plus communément appelée :
20 "vecteur d'interruption". Le concepteur du programme a pris soin d'initialiser la valeur du vecteur correspondant à l'interruption générée par la fin du comptage avec l'adresse du début de la séquence de code de l'interruption.

Ladite routine est en ROM, et donc non modifiable, ceci pour des raisons de sécurité, elle sera donc toujours exécutée.

Au début de ladite séquence de code de l'interruption, on vérifie que la valeur du PC : Val_PC mémorisée dans la pile (c'est-à-dire l'adresse de l'instruction que le microprocesseur devait exécuter s'il n'y avait pas eu d'interruption) n'est pas une valeur d'adresse de séquence sensible. Si la table TAB_SEC n'existe pas ou est vide, le test n'est pas effectué et le programme exécute directement la nouvelle séquence de code. Sinon, le programme de la routine extrait la valeur Val_PC de la pile et recherche dans la table TAB_SEC entre quelles valeurs d'adresse elle se situe.

Si les valeurs sont du type [Adrdeb_i , Adrfi_n_i], le programme a été interrompu pendant une séquence sensible, la carte peut renvoyer un message indiquant que sa sécurité est atteinte et peut se bloquer. Si les valeurs sont du type]Adrfi_n_i , Adrdeb_i+1[le programme a été interrompu pendant une séquence non sensible, le programme exécute alors la nouvelle séquence de code dont l'adresse de début a été mémorisée lors du traitement du point d'orientation.

D'autres modifications font également partie de l'esprit de l'invention.

Annexe 1

Voici ci-dessous une partie de code exécutable d'un programme principal, ce code est écrit en assembleur MOTOROLA 6805 :

5	adr_0	LDX	#01	Chargement du pointeur avec numéro du point d'orientation
		JSR	INS_ORT	Saut au point d'orientation
		NOP		
				
10	adr_1	LDX	#080H	Chargement de la valeur héra- décimale 80 dans le pointeur X.
		LDA	,X	Chargement de l'accumulateur A avec le contenu de l'adresse 80.
15		BEQ	V_ZERO	Branchement si égal zéro à V_ZERO, sinon instruction suivante
	adr_2	LDB	#010d	Chargement de l'accumulateur B avec la valeur décimale 10
20	adr_3	MUL		multiplication de A par B
		STA	Reg_H	Mémorisation du contenu de H dans l'accumulateur A.
		STB	Reg_L	Mémorisation du contenu de L dans l'accumulateur B.
25		JSR	Ecrit_Mot	
	V_ZERO	JMP	Suite	

Annexe 2

30		Adri	LDB	#020d
			JMP	adr_3

REVENDICATIONS

1. Dispositif de modification de séquences de code dans un support comprenant une unité centrale (1) capable d'exécuter ces séquences de code, une première mémoire (2) contenant un programme comprenant au moins une séquence de code exécutable par l'unité centrale (1), et une seconde mémoire programmable (3), caractérisé en ce qu'une table de déroutement TAB_DER contenue dans la seconde mémoire programmable contient au moins un champ contenant une donnée de référence d'une nouvelle séquence de code stockée dans une desdites mémoires, des moyens de déroutement permettant un déroutement différé depuis une séquence de code dudit programme en cours d'exécution vers ladite nouvelle séquence de code, et des moyens prévus dans la nouvelle séquence de code pour permettre, une fois la nouvelle séquence de code exécutée, un retour en un point de ladite séquence de code du programme.

2. Dispositif de modification de séquences de code selon la revendication 1 caractérisé en ce que les moyens de déroutement consistent en des instructions d'orientation (IORi) activables et implantées au préalable dans la mémoire contenant le code du programme, chaque instruction d'orientation étant associée à une référence i de la table de déroutement TAB_DER inscrite en mémoire programmable (3).

3. Dispositif de modification de séquences de code selon la revendication 2, caractérisé en ce que chaque instruction d'orientation (IORi) activée déclenche l'exécution d'une séquence nouvelle de code comportant :

- des moyens de lire dans la table TAB_DER de la mémoire programmable (3) un délai de temporisation ΔT_i correspondant à la référence de l'instruction d'orientation, ce délai de temporisation permettant

de différer le déclenchement d'une interruption qui effectue un branchement à une séquence nouvelle de code dont l'adresse (Adri) est indiquée dans la table, en association avec le délai de temporisation,

5 - des moyens de mémoriser l'adresse (Adri) dans une mémoire du dispositif et

 - des moyens de lancer un compteur (6) de temps du dispositif pour décompter le temps nécessaire au délai de temporisation du branchement.

4. Dispositif de modification de séquences de code selon la revendication 3, caractérisé en ce qu'il comporte une seconde table
10 TAB_SEC mémorisée dans la mémoire du dispositif et associant à chaque point (i) de déroutement un intervalle de temps [ΔT_{mini} ; ΔT_{maxi}] associé au délai de temporisation ΔT_i préalable à l'exécution d'une nouvelle séquence de code, et des moyens de vérification que le délai de temporisation est autorisé par l'intervalle de temps associé fourni par cette table.

15 5. Dispositif de modification de séquences de code selon la revendication 4, caractérisé en ce qu'il comporte des moyens permettant le décalage du délai ΔT_i de temporisation de la valeur de l'intervalle de temps [ΔT_{mini} ; ΔT_{maxi}].

20 6. Dispositif de modification de séquences de code selon la revendication 4, caractérisé en ce qu'il comporte des moyens de déclencher un message d'erreur lorsque le délai de temporisation ΔT_i est dans l'intervalle de temps.

7. Dispositif de modification de séquences de code selon la revendication 3, caractérisé en ce qu'il comporte consécutivement à la fin du
25 délai de temporisation (ΔT_i), lorsque le compteur (6) atteint la valeur nulle, des moyens de déclencher une interruption, des moyens de mémorisation de la valeur actuelle d'un registre compteur programme PC dans une pile,

puis des moyens de dérouter le programme à l'adresse définie dans la partie de la mémoire ROM contenant des vecteurs d'interruption, qui fournissent l'adresse du début de la séquence de code de l'interruption, des moyens de vérifier que la valeur du registre compteur programme PC Val_PC
5 mémorisée dans la pile n'est pas une valeur d'adresse de séquence sensible contenue dans une table TAB_SEC, et des moyens de modifier le déroulement des opérations.

8. Dispositif de modification de séquences de code selon la revendication 7, caractérisé en ce que, soit les moyens de vérification
10 testent si la valeur du registre compteur programme est contenue par TAB_SEC dans l'intervalle [Adrdeb_i , Adrfin_1], correspondant à une interruption du programme pendant une séquence sensible et les moyens de modifier le déroulement des opérations de la carte renvoient un message indiquant que sa sécurité est atteinte et se bloque, soit les moyens de
15 vérification testent si la valeur du registre compteur programme Val_PC est contenue dans l'intervalle]Adrfin_i , Adrdeb_i+1[correspondant à une interruption du programme pendant une séquence non sensible et les moyens de modifier le déroulement des opérations autorisent le programme à exécuter alors la nouvelle séquence de code dont l'adresse de début a été
20 mémorisée lors du traitement de l'instruction d'orientation (IORi).

9. Dispositif de modification de séquences de code selon la revendication 1, caractérisé en ce qu'il comporte une source de fréquence pour le compteur (6), différente d'une source de fréquence permettant à l'unité centrale (1) d'exécuter le programme, la valeur du délai de
25 temporisation (ΔT_i) programmée dans la table de déroutement TAB_DER étant calculée pour permettre d'interrompre le programme à une adresse déterminée, la table TAB_DER comporte pour chaque valeur du délai de temporisation, un élément supplémentaire contenant cette adresse déterminée et des moyens de comparer l'adresse de l'instruction

interrompue par l'interruption, à celle indiquée dans la table, et de provoquer une alarme.

10. Dispositif de modification de séquences de code selon la revendication précédente, caractérisé en ce qu'il comporte des moyens de
5 déclencher une alarme pour bloquer le support et indiquer une tentative de fraude par une écriture dans la mémoire.

11. Dispositif de modification de séquences de code selon la revendication 1 ou 3, caractérisé en ce que chaque séquence nouvelle de code se termine par une instruction d'orientation pour recharger le compteur
10 (6) avec une nouvelle valeur de délai de temporisation (ΔT_i).

12. Procédé de modification de séquences de code figées inscrites dans un support comportant une unité centrale et une mémoire (2), caractérisé en ce qu'il consiste à prévoir dans au moins une séquence de code figée, au moins une instruction d'orientation (IOR_i), permettant de
15 dérouter, par une interruption différée d'un délai de temporisation, l'exécution d'un programme contenu dans la mémoire vers une adresse déterminée, par une table de déroutement TAB_DER, en fonction d'une référence i associée à l'instruction d'orientation et dans un délai de temporisation, déterminé par le contenu d'une ligne de table correspondant
20 à la référence i de l'instruction d'orientation, une séquence nouvelle de code exécutable lors de l'interruption générée à la fin du délai de temporisation étant implantée à l'adresse contenue dans la table (TAB_DER).

13. Procédé selon la revendication 12, caractérisé en ce qu'une étape déclenchant l'interruption est précédée d'une étape de vérification,
25 consistant à vérifier que le délai de temporisation n'est pas inclus dans un intervalle défini par une seconde table TAB_SEC dite sécuritaire inscrite dans la mémoire non volatile du support.

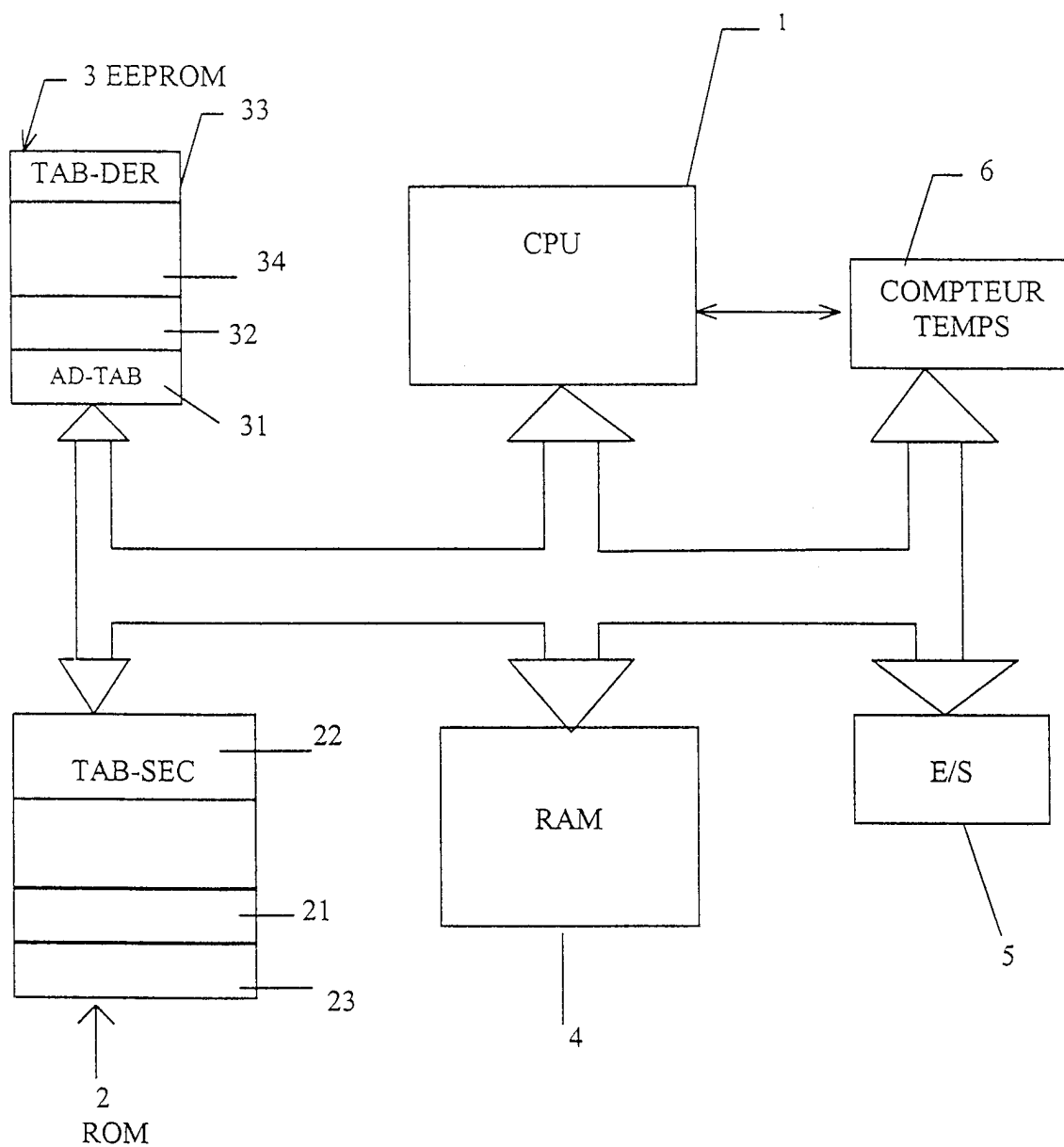
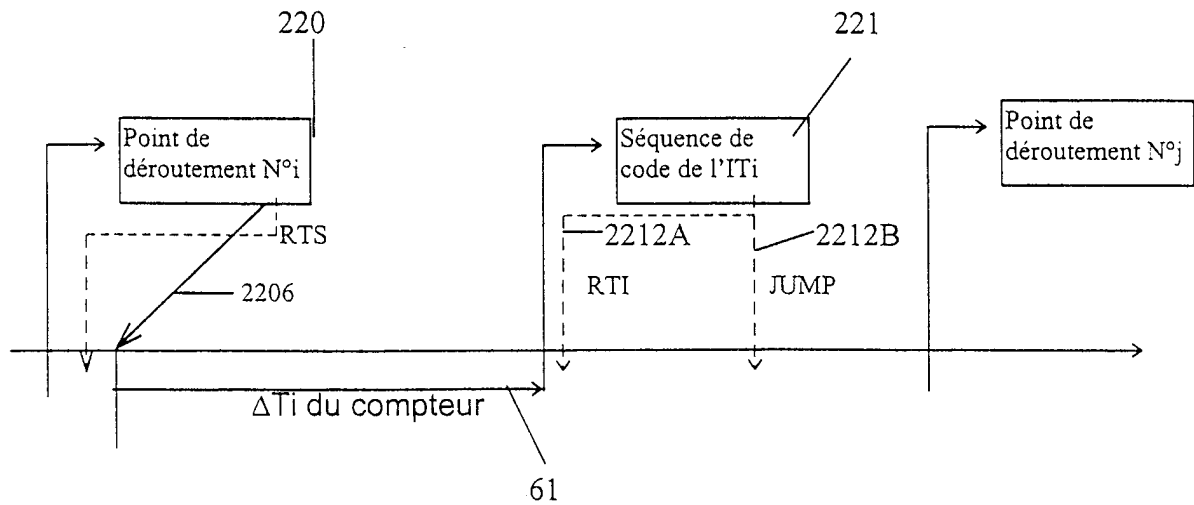
FIG. 1

FIG. 2A**FIG. 2C**

Saut à l'adresse préalablement chargée dans le vecteur d'IT

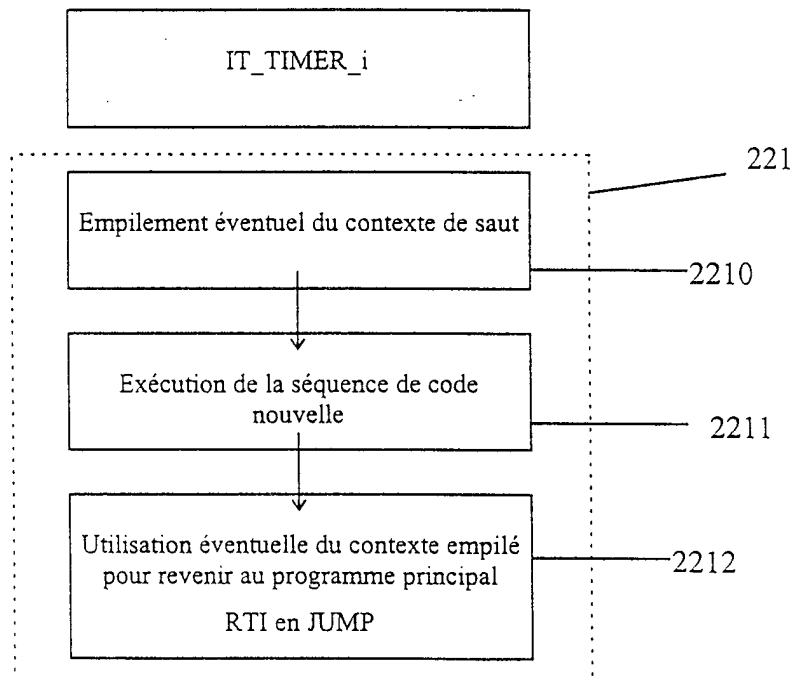


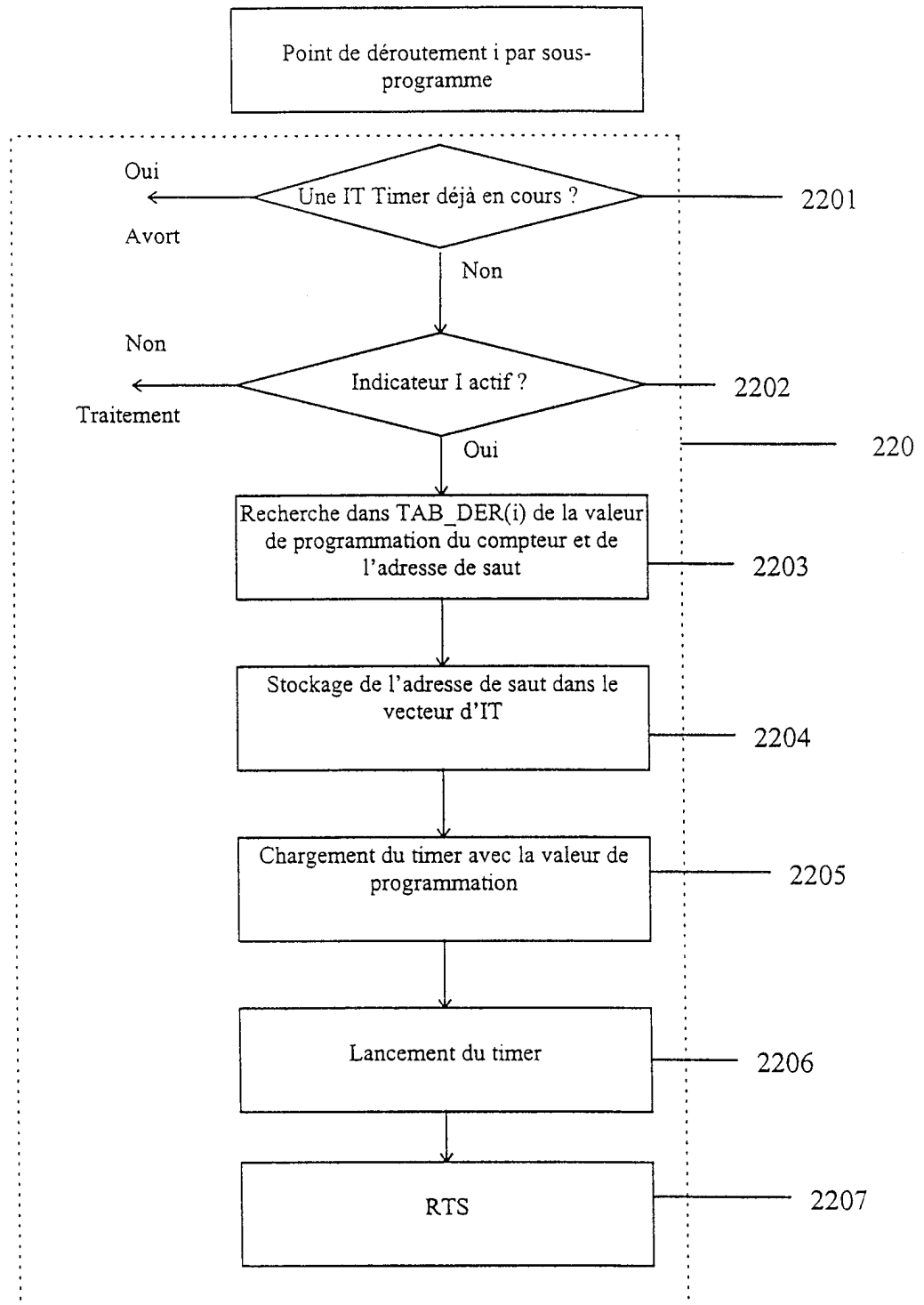
FIG. 2 B

FIG. 2D

TAB-DER

N° du point d'orientation	valeur du compteur	adresse de saut
1	$\Delta T1$	Adr_1
3	$\Delta T3$	Adr_3
4	$\Delta T4_1$	Adr_4_1
4	$\Delta T4_2$	Adr_4_2
6	$\Delta T6$	Adr_3
0	0000	0000

FIG. 4A

TAB-SEC

N° du point d'orientation	valeur minimale	valeur maximale
1	ΔT_{min1}	ΔT_{max1}
2	ΔT_{min2}	ΔT_{max2}
3	ΔT_{min3-1}	ΔT_{max3-1}
3	ΔT_{min3-2}	ΔT_{max3-2}
4	ΔT_{min4}	ΔT_{max4}
5	ΔT_{min5-1}	ΔT_{max5-1}
5	ΔT_{min5-2}	ΔT_{max5-2}
6	ΔT_{min6}	ΔT_{max6}

5/7

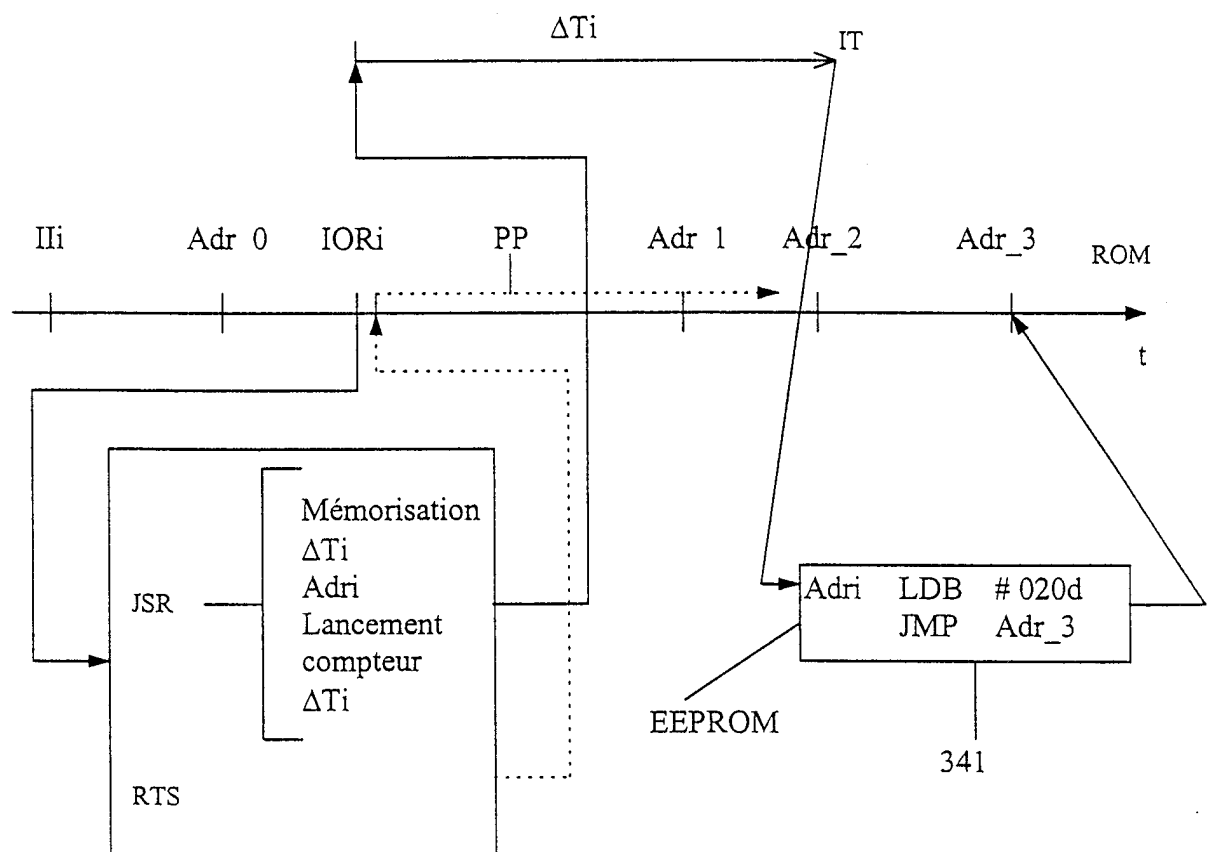
FIG 3

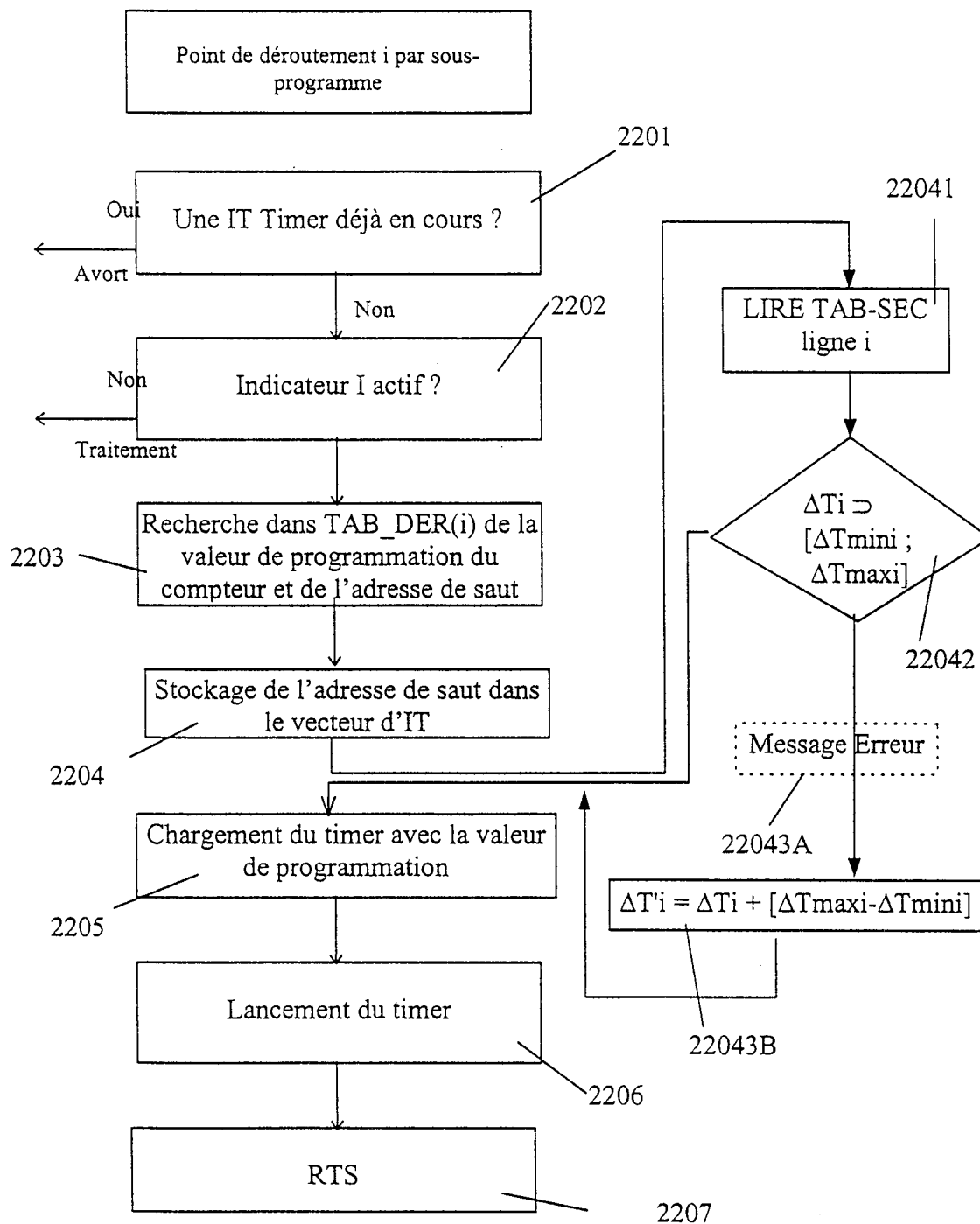
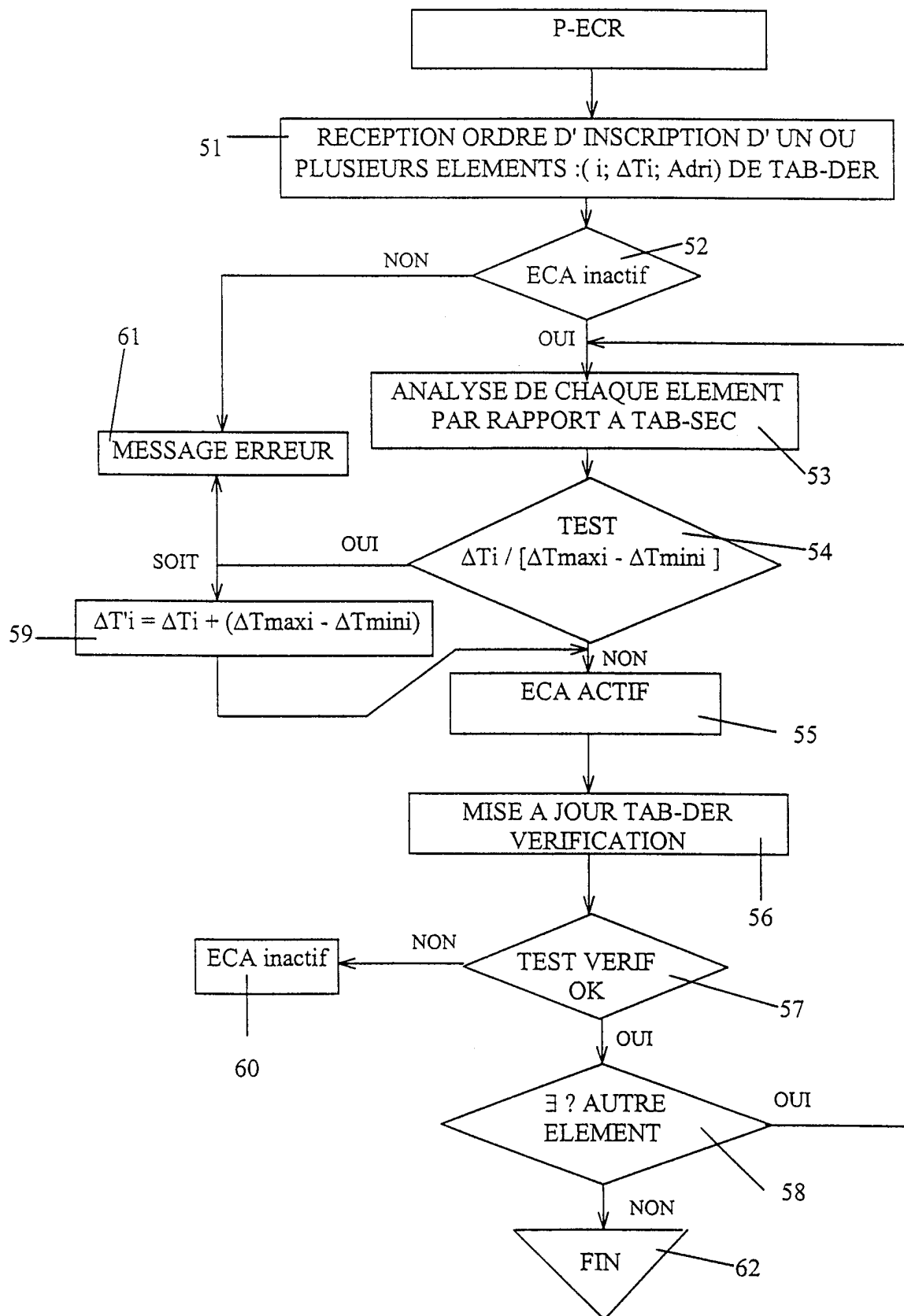
FIG 4B

FIG. 5



INTERNATIONAL SEARCH REPORT

International Application No

PCT/FR 98/01228

A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 G06F9/32 G06F11/20 G06F9/26

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	"MAINTAINABLE ROS CODE THROUGH THE COMBINATION OF ROM AND EEPROM" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 32, no. 9A, 1 February 1990, pages 273-276, XP000083070 see the whole document ---	1,2
Y	US 4 831 517 A (CROUSE RICHARD S ET AL) 16 May 1989 see the whole document ---	1,2
A	US 5 485 629 A (DULONG CAROLE) 16 January 1996 see column 18, line 61 - column 19, line 30 --- -/--	3

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

19 August 1998

Date of mailing of the international search report

27/08/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Daskalakis, T

INTERNATIONAL SEARCH REPORT

International Application No

PCT/FR 98/01228

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 623 665 A (SHIMADA KEIICHIRO ET AL) 22 April 1997 see the whole document ---	1,2
A	US 4 542 453 A (PATRICK MICHAEL J ET AL) 17 September 1985 see the whole document -----	1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/FR 98/01228

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 4831517 A	16-05-1989	EP 0263447 A	13-04-1988
		JP 63108434 A	13-05-1988

US 5485629 A	16-01-1996	NONE	

US 5623665 A	22-04-1997	JP 5189268 A	30-07-1993
		CA 2086883 A	14-07-1993
		EP 0551870 A	21-07-1993

US 4542453 A	17-09-1985	EP 0092646 A	02-11-1983
		JP 58151648 A	08-09-1983

RAPPORT DE RECHERCHE INTERNATIONALE

L. ande internationale No

PCT/FR 98/01228

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 6 G06F9/32 G06F11/20 G06F9/26

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)

CIB 6 G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si cela est réalisable, termes de recherche utilisés)

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
Y	"MAINTAINABLE ROS CODE THROUGH THE COMBINATION OF ROM AND EEPROM" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 32, no. 9A, 1 février 1990, pages 273-276, XP000083070 voir le document en entier ---	1, 2
Y	US 4 831 517 A (CROUSE RICHARD S ET AL) 16 mai 1989 voir le document en entier ---	1, 2
A	US 5 485 629 A (DULONG CAROLE) 16 janvier 1996 voir colonne 18, ligne 61 - colonne 19, ligne 30 ----- -/-	3

☒ Voir la suite du cadre C pour la fin de la liste des documents

☒ Les documents de familles de brevets sont indiqués en annexe

° Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée

19 août 1998

Date d'expédition du présent rapport de recherche internationale

27/08/1998

Nom et adresse postale de l'administration chargée de la recherche internationale

Office Européen des Brevets, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Fonctionnaire autorisé

Daskalakis, T

RAPPORT DE RECHERCHE INTERNATIONALE

Recherche Internationale No

PCT/FR 98/01228

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	US 5 623 665 A (SHIMADA KEIICHIRO ET AL) 22 avril 1997 voir le document en entier ---	1,2
A	US 4 542 453 A (PATRICK MICHAEL J ET AL) 17 septembre 1985 voir le document en entier -----	1

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Requête internationale No

PCT/FR 98/01228

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 4831517 A	16-05-1989	EP 0263447 A JP 63108434 A	13-04-1988 13-05-1988
US 5485629 A	16-01-1996	AUCUN	
US 5623665 A	22-04-1997	JP 5189268 A CA 2086883 A EP 0551870 A	30-07-1993 14-07-1993 21-07-1993
US 4542453 A	17-09-1985	EP 0092646 A JP 58151648 A	02-11-1983 08-09-1983