

12 DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 31.01.22.

30 Priorité :

43 Date de mise à la disposition du public de la
demande : 04.08.23 Bulletin 23/31.

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) : STMICROELECTRONICS (ALPS)
SAS Société par actions simplifiée — FR.

72 Inventeur(s) : BENHAMMADI Jawad.

73 Titulaire(s) : STMICROELECTRONICS (ALPS) SAS
Société par actions simplifiée.

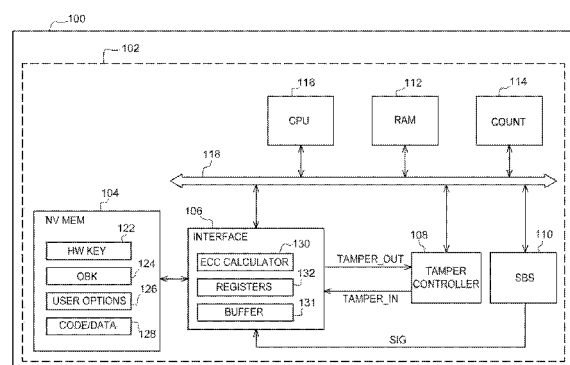
74 Mandataire(s) : CABINET BEAUMONT.

54 Mémoire non volatile sécurisée.

57 Mémoire non volatile sécurisée

La présente description concerne un procédé comprenant :- le chargement, à partir d'une mémoire non volatile (104) d'un circuit (102) vers un circuit de calcul (130), d'un premier paramètre de sécurité du circuit et d'un premier code correcteur d'erreur stocké en association avec le premier paramètre de sécurité ;- la vérification, par le circuit de calcul (130), du premier paramètre de sécurité et du premier code correcteur d'erreur afin de déterminer si un ou plusieurs des bits du paramètre de sécurité sont erronés ; et- s'il est déterminé que deux bits du paramètre de sécurité sont erronés, le chargement d'une valeur par défaut du premier paramètre dans un registre (132).

Figure pour l'abrégé: Fig. 1



Description

Titre de l'invention : *Mémoire non volatile sécurisée*

Domaine technique

- [0001] La présente description concerne le domaine des procédés et circuits pour la sécurité des circuits électroniques, et en particulier un procédé et un circuit permettant de protéger un circuit électronique comprenant une mémoire non volatile.

Technique antérieure

- [0002] Les mémoires non volatiles, telles que les mémoires de type FLASH, peuvent stocker des données sensibles en termes de confidentialité et sécurité telles que par exemple des clefs de chiffrement ou des paramètres de sécurité. La séquence de démarrage d'un circuit électronique est une étape sensible en termes de sécurité, car elle implique en général le traitement de ces données sensibles.
- [0003] Bien que des solutions existent pour rendre les données sensibles immuables, il serait souhaitable de protéger d'avantage ces données lorsqu'elles sont chargées de la mémoire non volatile vers d'autres mémoires.

Résumé de l'invention

- [0004] Il existe un besoin d'améliorer les méthodes de protections de la confidentialité et/ou la sécurité du contenu d'une mémoire non volatile d'un dispositif électronique.
- [0005] Un mode de réalisation pallie tout ou partie des inconvénients des méthodes de protection connus.
- [0006] Un mode de réalisation prévoit un procédé comprenant :
- le chargement, à partir d'une mémoire non volatile d'un circuit vers un circuit de calcul, d'un premier paramètre de sécurité du circuit et d'un premier code correcteur d'erreur stocké en association avec le premier paramètre de sécurité ;
 - la vérification, par le circuit de calcul, du premier paramètre de sécurité et du premier code correcteur d'erreur afin de déterminer si un ou plusieurs des bits du paramètre de sécurité sont erronés ; et
 - s'il est déterminé que deux bits du paramètre de sécurité sont erronés, le chargement d'une valeur par défaut du premier paramètre dans un registre.
- [0007] Selon un mode de réalisation, la valeur par défaut correspond à une valeur imposant au circuit un état de fonctionnement sécurisé.
- [0008] Selon un mode de réalisation, le procédé ci-dessus comprend, s'il est déterminé que deux bits du paramètre de sécurité sont erronés, la transmission d'un signal de sortie d'une interface de la mémoire non volatile vers un circuit de contrôle du circuit, et la réalisation d'une ou plusieurs contre-mesures par le circuit de contrôle.
- [0009] Selon un mode de réalisation, le circuit de contrôle est configuré pour incrémenter un

compteur suite à la réception du signal de sortie, et pour réinitialiser le circuit lorsqu'une valeur seuil est atteinte par le compteur.

- [0010] Selon un mode de réalisation, le procédé ci-dessus comprend, s'il est déterminé que deux bits du paramètre de sécurité sont erronés, la réalisation par l'interface de la mémoire d'une ou plusieurs contre-mesures.
- [0011] Selon un mode de réalisation, le procédé ci-dessus comprend :
 - la réception, par l'interface de la mémoire non volatile, d'un premier signal de commande, le premier signal de commande étant composé de m bits, pouvant prendre 2^m valeurs différentes, seulement q valeurs parmi ces 2^m valeurs étant autorisées, chacune de ces q valeurs étant différente des autres par au moins deux valeurs de bits, où m est supérieure ou égale à 2 et q est inférieure ou égale à $2^m / 2$;
 - la comparaison de la valeur du premier signal de commande, par l'interface de la mémoire non volatile, avec chacune des q valeurs autorisées ; et
 - si la valeur du premier signal de commande ne correspond à aucune des q valeurs autorisées, la réalisation d'une ou plusieurs contre-mesures, comprenant par exemple la transmission du signal de sortie par l'interface de la mémoire au circuit de contrôle du circuit.
- [0012] Selon un mode de réalisation, le procédé ci-dessus comprend en outre la réception par l'interface de la mémoire non volatile, à partir du circuit de contrôle du circuit, d'un signal d'entrée d'avertissement indiquant la détection d'un événement suspect, et la réalisation par l'interface d'une ou plusieurs contre-mesures.
- [0013] Selon un mode de réalisation, l'une ou plusieurs contre-mesures comprennent l'interdiction d'accès à une première zone de la mémoire non volatile suite à la réception du signal d'entrée d'avertissement.
- [0014] Selon un mode de réalisation, la première zone de la mémoire non volatile contient au moins une clef de chiffrement.
- [0015] Selon un mode de réalisation, l'une ou plusieurs contre-mesures comprennent le remplacement la valeur d'une clef de chiffrement, stockée dans la mémoire non volatile, par une valeur factice.
- [0016] Selon un mode de réalisation, le chargement dans le registre du premier paramètre de sécurité du circuit est effectué avant la fin d'une séquence de démarrage du circuit.
- [0017] Selon un mode de réalisation, la mémoire non volatile est une mémoire de type FLASH.
- [0018] Un mode de réalisation prévoit un circuit comprenant:
 - une mémoire non volatile stockant un premier paramètre de sécurité, et un premier code correcteur d'erreur, stocké en association du premier paramètre dans la mémoire non volatile ; et
 - un circuit de calcul configuré pour :

vérifier le premier paramètre de sécurité et le premier code correcteur d'erreur afin de déterminer si un ou plusieurs des bits du paramètre de sécurité sont erronés ; et s'il est déterminé que deux bits du paramètre de sécurité sont erronés, charger une valeur par défaut du premier paramètre de sécurité dans un registre.

[0019] Selon un mode de réalisation, le circuit ci-dessus comprend une interface de la mémoire non volatile configurée pour, lorsqu'il est déterminé que deux bits du paramètre de sécurité sont erronés, transmettre un signal de sortie vers un circuit de contrôle du circuit configuré pour réaliser une ou plusieurs contre-mesures.

Brève description des dessins

[0020] Ces caractéristiques et avantages, ainsi que d'autres, seront exposés en détail dans la description suivante de modes de réalisation particuliers faite à titre non limitatif en relation avec les figures jointes parmi lesquelles :

[0021] la [Fig.1] représente de façon très schématique et sous forme de blocs, un mode de réalisation d'un dispositif électronique selon un mode de réalisation de la présente description ;

[0022] la [Fig.2] est un organigramme représentant des opérations d'un procédé de protection d'un circuit selon un exemple de réalisation de la présente description ;

[0023] la [Fig.3] est un organigramme représentant des opérations d'un procédé de protection d'un circuit selon un autre exemple de réalisation de la présente description ; et

[0024] la [Fig.4] est un organigramme représentant des opérations d'un procédé de protection d'un circuit selon un autre exemple de réalisation de la présente description.

Description des modes de réalisation

[0025] De mêmes éléments ont été désignés par de mêmes références dans les différentes figures. En particulier, les éléments structurels et/ou fonctionnels communs aux différents modes de réalisation peuvent présenter les mêmes références et peuvent disposer de propriétés structurelles, dimensionnelles et matérielles identiques.

[0026] Par souci de clarté, seuls les étapes et éléments utiles à la compréhension des modes de réalisation décrits ont été représentés et sont détaillés. En particulier, la conception de dispositifs de traitement est bien connue de la personne du métier et certains éléments n'ont pas été détaillés dans la description qui suit.

[0027] Sauf précision contraire, lorsque l'on fait référence à deux éléments connectés entre eux, cela signifie directement connectés sans éléments intermédiaires autres que des conducteurs, et lorsque l'on fait référence à deux éléments reliés (en anglais "coupled") entre eux, cela signifie que ces deux éléments peuvent être connectés ou être reliés par l'intermédiaire d'un ou plusieurs autres éléments.

[0028] Dans la description qui suit, lorsque l'on fait référence à des qualificatifs de position

absolue, tels que les termes "avant", "arrière", "haut", "bas", "gauche", "droite", etc., ou relative, tels que les termes "dessus", "dessous", "supérieur", "inférieur", etc., ou à des qualificatifs d'orientation, tels que les termes "horizontal", "vertical", etc., il est fait référence sauf précision contraire à l'orientation des figures.

- [0029] Sauf précision contraire, les expressions "environ", "approximativement", "sensiblement", et "de l'ordre de" signifient à 10 % près, de préférence à 5 % près.
- [0030] Dans la présente description, le terme « événement suspect » est utilisé pour désigner une altération de la valeur d'un ou plusieurs signaux transmis dans un circuit intégré, ou d'une ou plusieurs données stockées dans une mémoire volatile ou non volatile d'un circuit intégré, d'origine inconnue. Cette altération est par exemple provoquée par une attaque à l'encontre du circuit intégré, par exemple par une attaque par injection de faute, ou est causée par une autre source d'erreur, telle que l'environnement du circuit, par exemple par l'effet de radiation.
- [0031] La [Fig.1] représente, de façon très schématique et sous forme de blocs, un dispositif électronique 100 comprenant un circuit intégré 102 selon un mode de réalisation de la présente description.
- [0032] Le dispositif électronique 100 est par exemple une carte électronique telle qu'une carte à microcircuits, un matériel à usage informatique, un circuit à microprocesseur, etc.
- [0033] Le circuit intégré 102 comprend par exemple une mémoire non volatile 104 (NV MEM), par exemple une mémoire FLASH. La mémoire 104 est reliée à une interface mémoire 106 (INTERFACE) elle-même reliée à un circuit de contrôle 108 (TAMPER CONTROLLER) du circuit intégré 102, ainsi qu'à un circuit 110 (SBS) du circuit intégré 102. A titre d'exemple, le circuit 110 est un circuit de configuration du démarrage et de la sécurité du circuit intégré 102 (en anglais « System configuration, Boot and Security »). Le circuit 110 est par exemple configuré pour transmettre un ou plusieurs signaux SIG à l'interface 106. Par exemple les signaux sont des signaux de commande transmis par le circuit 110, par exemple lors du démarrage du dispositif 100, afin d'informer la mémoire non volatile 104 des fonctionnalités et/ou ressources du circuit 102 étant accessibles lors de l'étape en cours du démarrage.
- [0034] Le circuit 102 comprend, par exemple, en outre une mémoire volatile 112 (RAM), un compteur 114 (COUNT) tel que par exemple un compteur monotone, et un processeur générique 116 (CPU). Le processeur 116, la mémoire volatile 112 et le compteur 114 sont par exemple reliés à l'interface mémoire 106, au circuit de contrôle 108 et au circuit 110 par un bus 118, comprenant par exemple un bus de données.
- [0035] La mémoire non volatile 104 comprend en outre une zone 122 stockant une clef de chiffrement (HW KEY), une zone 124 stockant d'autres clefs de chiffrement telles que des Option Bytes Keys (OBK), et une zone 126 stockant des données de configuration

(USER OPTIONS) telles que par exemple des paramètres de sécurité du circuit 102 choisis par l'utilisateur pour être utilisés lors d'une séquence de démarrage du dispositif 100. La zone 126 comprend en outre, pour au moins certaines des valeurs de données qu'elle contient, un code correcteur d'erreur (ECC - en anglais « Error-Correcting Code »). Chaque code correcteur d'erreur est stocké dans la zone 126 en association avec l'une des valeurs de données correspondante. La combinaison de chaque valeur de donnée avec le code correcteur d'erreur associé forme un codage de données comprenant de la redondance au codage et permet alors de détecter, et dans certains cas de corriger, une ou plusieurs erreurs dans la valeur de donnée ayant lieu pendant ou après son enregistrement. Par exemple, le code correcteur d'erreur permet de détecter et/ou de corriger une ou plusieurs erreurs intervenant lors d'une transmission de la donnée, entre la mémoire non volatile 104 et le circuit de contrôle 108.

[0036] Une zone 128 de la mémoire non volatile 104 stocke par exemple des codes et/ou des données 128 (CODE/DATA). Par exemple la zone 128 stocke des codes de démarrage et/ou des codes applicatifs.

[0037] L'interface mémoire 106 comprend en outre un circuit de calcul 130 (ECC CALCULATOR) configuré pour calculer, à partir d'une valeur de donnée stockée dans la zone 126, un code de vérification et pour comparer ce code de vérification avec le code correcteur d'erreur associé à la valeur de donnée. A titre d'exemple, l'interface mémoire 106 comprend en outre au moins une mémoire tampon 131 (BUFFER) dans lequel sont chargés, à partir de la zone 126, la valeur de donnée et son code correcteur d'erreur associé. Le circuit de calcul 130 est alors par exemple configuré pour effectuer le calcul du code de vérification de la valeur de donnée à partir de la mémoire tampon 131 et pour comparer le code de vérification avec le code correcteur d'erreur contenu dans la mémoire tampon 131.

[0038] Lorsque le code de vérification calculé correspond au code correcteur d'erreur, la valeur de donnée est traitée comme étant intègre, et est par exemple chargée dans un parmi plusieurs registres 132 (REGISTERS), compris par exemple dans l'interface mémoire 106. Cette valeur de donnée représente un ou plusieurs paramètres de sécurité qui modifient par exemple le comportement du processeur générique 116. Par exemple, ces paramètres de sécurité définissent des règles d'accès aux données stockées dans la mémoire non volatile 104, telles que l'autorisation ou l'interdiction de lire et/ou de modifier ces données. Dans un autre exemple, ces paramètres permettent de définir l'une ou plusieurs zones de la mémoire non volatile 104 accessibles juste en lecture, et/ou l'une ou plusieurs zones de la mémoire non volatile 104 accessibles dans un mode sécurisé. Encore dans un autre exemple, ces paramètres permettent d'activer ou de désactiver un ou plusieurs circuits cryptographiques (non illustrés) du circuit

intégré 102.

- [0039] Lorsque le code de vérification calculé diffère du code correcteur d'erreur, cela indique qu'une altération d'au moins un bit de la valeur de donnée a eu lieu. Selon le codage utilisé pour générer le code correcteur d'erreur et la taille de ce code par rapport à la taille de la valeur de donnée, il est également possible d'identifier le nombre de bits erronés jusqu'à une certaine limite, et de corriger un ou plusieurs bits erronés de la valeur de donnée. Dans un exemple de réalisation, il est possible de détecter et corriger un bit erroné dans la valeur de donnée, ou de détecter la présence d'au moins deux bits erronés dans la valeur de donnée sans pouvoir les corriger.
- [0040] Dans les applications visées par la présente description, lorsqu'au moins deux bits de la valeur de donnée sont déterminés par le circuit de calcul 130 comme étant erronés, il est considéré qu'un événement suspect, tel qu'une attaque sur la mémoire non volatile 104, a lieu. Dans le cas de détection d'un événement suspect et dans les modes de réalisation décrits, on prévoit le déclenchement de contre-mesures visant à sécuriser le circuit intégré 102. A titre d'exemple, afin de sécuriser le circuit intégré 102, une valeur par défaut est alors stockée dans au moins un des registres 132 en remplacement de la valeur de donnée erronée. La valeur par défaut correspond par exemple à une valeur d'un paramètre de sécurité imposant un mode sécurisé au circuit 102, tel qu'un démarrage sécurisé.
- [0041] Dans les modes de réalisation décrits, l'interface 106 de la mémoire non volatile 104 est configurée pour transmettre un signal (TAMPER_OUT) au circuit de contrôle 108 en cas de détection d'un événement suspect. Le circuit de contrôle 108 est configuré pour, par exemple, suite à la réception du signal TAMPER_OUT, déclencher d'autres contre-mesures. A titre d'exemple, à chaque réception du signal TAMPER_OUT, le circuit de contrôle 108 commande l'incrémentation du compteur 114 et, dès que le compteur 114 atteint une valeur seuil, le circuit de contrôle 108 est par exemple configuré pour réinitialiser le circuit 102. A titre d'exemple, la valeur seuil vaut entre 3 et 10.
- [0042] Le circuit de contrôle 108 est en outre configuré pour transmettre un signal d'avertissement (TAMPER_IN) à l'interface 106 de la mémoire non volatile 104 lorsqu'un événement suspect est détecté hors de la mémoire non volatile 104. Suite à la réception du signal TAMPER_IN, l'interface 106 est par exemple configurée pour déclencher des contre-mesures visant à protéger la confidentialité du contenu de la mémoire 104 et en particulier du contenu des zones 122 à 128.
- [0043] La [Fig.2] est un organigramme représentant des opérations d'un procédé de protection du circuit intégré 102 de la [Fig.1] selon un exemple de réalisation de la présente description. Les étapes du procédé de la [Fig.2] sont implémentées, par exemple, par la mémoire non volatile 104 et l'interface mémoire 106, sous contrôle du

processeur générique 116.

- [0044] Dans une étape 200 (DUMP USER OPTION), une valeur de donnée, par exemple contenue dans la zone 126 de la mémoire non volatile 104 de la [Fig.1], ainsi que le code correcteur d'erreur associé, sont par exemple chargés dans la mémoire tampon 131 du circuit de calcul 130. Par exemple, dans le cas où il y a plusieurs valeurs de données à traiter dans la zone 126, ces valeurs sont par exemple associées à des valeurs d'index ou adresses mémoire correspondantes. La valeur de donnée et le code correcteur d'erreur chargés dans l'étape 200 correspondent par exemple à ceux associés avec une valeur courante de l'index ou de l'adresse mémoire.
- [0045] Dans une étape 201 (COMPUTE ECC), le circuit de calcul 130 calcule un code de vérification à partir de la valeur de donnée. Le code de vérification est par exemple calculé selon le même algorithme que celui utilisé pour le calcul du code correcteur d'erreur associé à la valeur de donnée et stocké en association de la valeur de donnée. En particulier, avant l'enregistrement de la valeur de donnée dans la zone 126, le code correcteur d'erreur a par exemple été calculé, par le circuit 130 ou par un autre circuit, de sorte que la valeur de donnée et le code correcteur d'erreur associé soient enregistrés au même moment dans la zone 126. Le circuit de calcul 130 est en outre configuré pour comparer le code de vérification avec le code correcteur d'erreur associé à la valeur de donnée.
- [0046] Dans une étape 202 (DBL_ECC ERROR ?), le circuit de calcul 130 détermine, sur la base de la comparaison entre le code de vérification et le code correcteur d'erreur, si au moins deux bits de la valeur de donnée, chargée dans l'étape 200, sont erronés.
- [0047] Dans le cas où le circuit de calcul 130 détermine que moins de deux bits de la valeur de donnée sont erronés (branche N), le procédé continue dans une étape 203 (LOAD USER OPTION IN REGISTER) dans laquelle la valeur de donnée est par exemple chargée dans au moins un des registres 132. Dans le cas où un bit de la valeur de donnée est erroné, le circuit de calcul 130 est par exemple configuré pour corriger ce bit avant de charger la valeur de donnée dans l'au moins un des registres 132. Il serait également possible que, au lieu de charger la valeur de donnée dans au moins un des registres 132, la valeur de donnée soit chargée dans au moins un élément de mémorisation du circuit 102, tel que par exemple la mémoire volatile 112 ou un registre distinct des registres 132 (non illustrés en [Fig.1]).
- [0048] Dans le cas où le circuit de calcul 130 détermine, lors de l'étape 202, qu'une double erreur a eu lieu (branche Y), c'est à dire qu'au moins deux bits de la valeur de donnée sont erronés, le procédé continue dans une étape 204 (SET DBL_ECC_ERR=1). Lors de l'étape 204, une valeur, par exemple la valeur 1, est affectée à une variable DBL_ECC_ERR, qui est par exemple initialement à 0. La variable DBL_ECC_ERR est par exemple stockée dans un des registres 132 et son état indique si une valeur de

donnée chargée lors de l'étape 200 a été détectée comme présentant au moins deux bits erronés. A titre d'exemple, lorsque la variable DBL_ECC_ERR a pour état 0, cela indique qu'aucune double erreur n'a été détectée sur les valeurs de donnée préalablement traitées. Lorsque la variable DBL_ECC_ERR a pour état 1, cela indique qu'au moins une des valeurs de donnée, chargées dans une réalisation de l'étape 200, présente une double erreur et, dans ce cas, qu'un événement suspect a eu lieu. Les valeurs 0 et 1 sont présentées à titre d'exemple et ne sont bien sûr pas limitatives.

[0049] Suite à la réalisation de l'étape 203 ou cas échéant de l'étape 204, le procédé continue dans une étape 205 (END ?) dans laquelle, il est déterminé, par exemple par le processeur générique 116, si une autre valeur de donnée stockée dans la zone 126 est à charger, en association d'un code correcteur d'erreur, dans la mémoire tampon 131. S'il est déterminé qu'il reste au moins une valeur de donnée à charger (branche N), le procédé continue dans une étape 206 (INCREMENT USER OPTION INDEX). A titre d'exemple, lors de la réalisation de l'étape 206, l'adresse mémoire de la nouvelle valeur de donnée à charger est déterminée, par exemple par le processeur générique 116, et est transmise à la mémoire non volatile 104. Le procédé reprend ensuite à l'étape 200. La nouvelle valeur de donnée à charger est par exemple déterminée en incrémentant la valeur d'index courante.

[0050] Dans le cas où, dans l'étape 205, il est déterminé qu'il n'y a plus de valeur de donnée à charger (branche Y), le procédé continue dans une étape 207 (DBL_ECC_ERR=1 ?) dans laquelle l'état de la variable DBL_ECC_ERR est vérifié, par exemple par le processeur générique 116. Lorsque l'état de la variable DBL_ECC_ERR indique qu'aucune double erreur n'a été détectée (branche N), par exemple lorsque la variable vaut 0, le procédé se termine sur une étape 208 (END). A titre d'exemple, suite à l'étape 208, les valeurs de données sont utilisées par le circuit intégré 102 dans une séquence de démarrage.

[0051] Dans le cas où, lors de la réalisation de l'étape 207, il est déterminé qu'une double erreur a été détectée (branche Y), et par conséquent qu'un événement suspect a eu lieu, le procédé continue dans une étape 209 (COUNTERMEASURES). L'étape 209 consiste en la réalisation d'une ou plusieurs contre-mesures assurant la protection du circuit 102.

[0052] Selon un mode de réalisation, une contre-mesure réalisée en cas de détection d'un événement suspect consiste à charger des valeurs par défaut dans au moins un des registres 132 en remplacement des valeurs de données. Les valeurs par défaut correspondent par exemple à des valeurs assurant un démarrage et/ou un fonctionnement sécurisé du circuit 102. A titre d'exemple, les valeurs par défaut permettent de restreindre l'accès à la mémoire non volatile 104.

[0053] Selon un mode de réalisation, une autre contre-mesure réalisée en cas de détection

d'un événement suspect consiste à remplacer la valeur de la clef de chiffrement HW KEY stockée dans la zone 122, par une valeur factice. La clef de chiffrement HW KEY sera alors inutilisable car sa valeur ne permettra pas d'effectuer le déchiffrement des données concernées.

- [0054] Selon un mode de réalisation, une autre contre-mesure réalisée en cas de détection d'un événement suspect consiste à bloquer tout accès à la zone 124 stockant les clefs de chiffrement OBK.
- [0055] Selon un mode de réalisation, une autre contre-mesure réalisée en cas de détection d'un événement suspect consiste à transmettre, par l'intermédiaire de l'interface 106 de la mémoire non volatile 104 et au circuit de contrôle 108 du circuit intégré 102, le signal TAMPER_OUT. A titre d'exemple, le circuit de contrôle 108 est configuré pour commander l'incrémementation du compteur 114 suite à la réception du signal TAMPER_OUT. Le processeur générique 116 est par exemple configuré pour déclencher une réinitialisation du circuit intégré 102 lorsque le compteur atteint une certaine valeur seuil.
- [0056] Bien que, dans l'exemple des figures 1 et 2, le code de vérification soit calculé dans l'interface 106 de la mémoire volatile 104, dans d'autres mode de réalisation, la valeur de donnée et le code correcteur d'erreur associé sont chargés dans un registre hors de l'interface mémoire 106 et de la mémoire non volatile 104, par exemple dans un registre (non illustré en [Fig.1]) compris dans le processeur 116. Le calcul du code de vérification, ainsi que la comparaison avec le code correcteur d'erreur, sont alors effectués en dehors de la mémoire non volatile 104.
- [0057] Bien que, dans l'exemple des figures 1 et 2, le code correcteur d'erreur soit utilisé pour détecter des événements suspects, de façon supplémentaire ou de façon alternative, il serait également possible de détecter des événements suspects par la mémoire volatile 104 sur la base d'autres moyens. Un exemple, basé sur des signaux de commande SIG, est maintenant décrit en relation avec la [Fig.3].
- [0058] La [Fig.3] est un organigramme représentant des opérations d'un procédé de protection du circuit 102 de la [Fig.1] selon un autre exemple de réalisation de la présente description. Les étapes du procédé de la [Fig.3] sont implémentées, par exemple, par la mémoire non volatile 104 et l'interface mémoire 106, sous contrôle du processeur générique 116.
- [0059] Dans une étape 300 (RECEIVE SIGNAL), le circuit 110 envoie un parmi les signaux de commande SIG à l'interface 106 de la mémoire non volatile 104.
- [0060] A titre d'exemple, le signal de commande transmis permet d'indiquer à la mémoire non volatile 104 si certaines fonctionnalités du circuit intégré 102 sont activées et/ou si certaines ressources du circuit intégré 102 sont accessibles. Dans un autre exemple, le signal de commande SIG transmis indique à la mémoire non volatile 104 quelles zones

mémoire la composant, par exemple quelles zones parmi des zones partitionnant la zone 128, doivent être accessibles ou inaccessibles à l'écriture et/ou à la lecture. A titre d'exemple, un signal de commande SIG de ce type est transmis à la mémoire non volatile 104 à chaque étape d'une séquence de démarrage permettant de verrouiller et/ou déverrouiller progressivement la mémoire non volatile 104.

- [0061] Selon un mode de réalisation de la présente description, les signaux de commande sont encodés sur un nombre m de bits, où m est un nombre entier par exemple compris entre 4 et 16. En particulier, le nombre m permet d'ajouter de la redondance dans l'encodage d'un signal de commande en le codant sur plus de bits que nécessaire. Plus particulièrement, sur les 2^m valeurs de signal de commande possibles, seulement un nombre q , inférieur ou égal à $2^m / 2$, de valeurs correspondent à une valeur de signal de commande valide et autorisée. Autrement dit, chaque valeur de signal de commande valide et autorisée diffère de chaque autre valeur de signal de commande valide et autorisée par au moins deux bits. Lorsqu'une valeur de signal de commande valide et autorisée est modifiée par un évènement suspect, les valeurs d'au moins deux bits parmi les m bits codant ce signal doivent donc être modifiées afin que la valeur du signal modifiée corresponde à une autre valeur valide et autorisée du signal de commande. A titre d'exemple, les valeurs de signaux de commande valides et autorisées sont mémorisées dans une liste, ou dans une table, stockée par exemple dans la mémoire non volatile 104 ou par une autre mémoire du circuit 102.
- [0062] Dans une étape 301 (AUTHORIZED SIGNAL VALUE ?), l'interface 106 vérifie si la valeur du signal de commande est valide et autorisée, par exemple en la comparant avec les valeurs mémorisées dans la liste ou dans la table des valeurs de signal valides et autorisées. Dans le cas où la valeur du signal de commande reçue est valide et autorisée (branche Y), le procédé reprend par exemple à l'étape 300 dans laquelle l'interface 106 est en attente de réception d'un nouveau signal de commande. Dans le cas où la valeur du signal de commande reçue par l'interface 106 ne correspond pas à une valeur de signal valide et autorisée (branche N), un évènement suspect, altérant la valeur du signal, a eu lieu. Le procédé continue dans une étape 302 (COUNTERMEASURES) consistant en la réalisation d'une ou plusieurs contre-mesures assurant la protection du circuit intégré 102. Les contre-mesures déployées sont par exemple les contre-mesures décrites en relation avec l'étape 209 de la [Fig.2].
- [0063] La [Fig.4] est un organigramme représentant des opérations d'un procédé de protection du circuit 102 selon un autre exemple de réalisation de la présente description. Les étapes du procédé de la [Fig.4] sont implémentées, par exemple, par le circuit de contrôle 108 et par la mémoire non volatile 104 et l'interface mémoire 106, sous contrôle du processeur générique 116.
- [0064] Dans une étape 400 (SUSPICIOUS EVENT DETECTION), un évènement suspect

est détecté par un mécanisme de détection à l'extérieur de la mémoire non volatile 104. A titre d'exemple, le circuit de contrôle 108 du circuit intégré 102 est configuré pour détecter la survenue d'un événement suspect, ayant par exemple lieu sur le contenu de la mémoire volatile 102, sur des données en cours de transmission sur le bus 118, etc. Des mécanismes de détection d'événements suspects sont connus de la personne du métier, et ne seront pas décrits ici en détail.

[0065] Suite à la détection d'un événement suspect hors de la mémoire non volatile 104 et de l'interface mémoire 106 lors de l'étape 400, le circuit de contrôle 108 est configuré pour, dans une étape 401 (ASSERT TAMPER_IN SIGNAL), transmettre le signal d'avertissement TAMPER_IN à l'interface 106 de la mémoire non volatile 104. Lors de l'étape 401, le circuit de contrôle 108 est par exemple en outre configuré pour déclencher l'incrémentement du compteur 114.

[0066] L'interface 106 déclenche alors, dans une étape 402 (COUNTERMEASURES IN NV MEM) des contre-mesures afin de protéger la mémoire non volatile 104. Les contre-mesures déployées sont par exemple les contre-mesures décrites en relation avec l'étape 209 de la [Fig.2] telles que :

- le chargement des valeurs par défaut dans au moins un des registres 132 en remplacement des valeurs de données venant de la zone mémoire 126 ;
- le remplacement de la valeur de la clef de chiffrement HW KEY stockée dans la zone 122, par une valeur factice ; et/ou
- l'interdiction d'accès à la zone 124 stockant les clefs de chiffrement OBK.

[0067] Divers modes de réalisation et variantes ont été décrits. La personne du métier comprendra que certaines caractéristiques de ces divers modes de réalisation et variantes pourraient être combinées, et d'autres variantes apparaîtront à la personne du métier. Par exemple, d'autres contre-mesures que celles décrites peuvent être implémentées.

[0068] Enfin, la mise en œuvre pratique des modes de réalisation et variantes décrits est à la portée de la personne du métier à partir des indications fonctionnelles données ci-dessus.

Revendications

- [Revendication 1] Procédé comprenant :
- le chargement, à partir d'une mémoire non volatile (104) d'un circuit (102) vers un circuit de calcul (130), d'un premier paramètre de sécurité du circuit et d'un premier code correcteur d'erreur stocké en association avec le premier paramètre de sécurité ;
 - la vérification, par le circuit de calcul (130), du premier paramètre de sécurité et du premier code correcteur d'erreur afin de déterminer si un ou plusieurs des bits du paramètre de sécurité sont erronés ; et
 - s'il est déterminé que deux bits du paramètre de sécurité sont erronés, le chargement d'une valeur par défaut du premier paramètre dans un registre (132).
- [Revendication 2] Procédé selon la revendication 1, dans lequel la valeur par défaut correspond à une valeur imposant au circuit (102) un état de fonctionnement sécurisé.
- [Revendication 3] Procédé selon la revendication 1 ou 2, comprenant en outre, s'il est déterminé que deux bits du paramètre de sécurité sont erronés, la transmission d'un signal de sortie (TAMPER_OUT) d'une interface (106) de la mémoire non volatile (104) vers un circuit de contrôle (108) du circuit (102), et la réalisation d'une ou plusieurs contre-mesures par le circuit de contrôle.
- [Revendication 4] Procédé selon la revendication 3, dans lequel le circuit de contrôle (108) est configuré pour incrémenter un compteur (114) suite à la réception du signal de sortie (TAMPER_OUT), et pour réinitialiser le circuit (102) lorsqu'une valeur seuil est atteinte par le compteur.
- [Revendication 5] Procédé selon la revendication 3 ou 4, comprenant en outre, s'il est déterminé que deux bits du paramètre de sécurité sont erronés, la réalisation par l'interface (106) de la mémoire (104) d'une ou plusieurs contre-mesures.
- [Revendication 6] Procédé selon la revendication 3 ou 4, comprenant en outre :
- la réception, par l'interface (106) de la mémoire non volatile (104), d'un premier signal de commande (SIG), le premier signal de commande étant composé de m bits, pouvant prendre 2^m valeurs différentes, seulement q valeurs parmi ces 2^m valeurs étant autorisées, chacune de ces q valeurs étant différente des autres par au moins deux valeurs de bits, où m est supérieure ou égale à 2 et q est inférieure ou égale à $2^m / 2$;

- la comparaison de la valeur du premier signal de commande, par l'interface de la mémoire non volatile, avec chacune des q valeurs autorisées ; et

- si la valeur du premier signal de commande ne correspond à aucune des q valeurs autorisées, la réalisation d'une ou plusieurs contre-mesures, comprenant par exemple la transmission du signal de sortie (TAMPER_OUT) par l'interface de la mémoire au circuit de contrôle (108) du circuit (102).

[Revendication 7] Procédé selon la revendication 3 ou 4, comprenant en outre la réception par l'interface (106) de la mémoire non volatile (104), à partir du circuit de contrôle (108) du circuit (102), d'un signal d'entrée d'avertissement (TAMPER_IN) indiquant la détection d'un événement suspect, et la réalisation par l'interface d'une ou plusieurs contre-mesures.

[Revendication 8] Procédé selon l'une quelconque des revendications 5 à 7, dans lequel l'une ou plusieurs contre-mesures comprennent l'interdiction d'accès à une première zone (124) de la mémoire non volatile (104) suite à la réception du signal d'entrée d'avertissement (TAMPER_IN).

[Revendication 9] Procédé selon la revendication 8, dans lequel la première zone (124) de la mémoire non volatile contient au moins une clef de chiffrement (OBK).

[Revendication 10] Procédé selon l'une quelconque des revendications 5 à 9, dans lequel l'une ou plusieurs contre-mesures comprennent le remplacement la valeur d'une clef de chiffrement (HW KEY), stockée dans la mémoire non volatile (104), par une valeur factice.

[Revendication 11] Procédé selon l'une quelconque des revendications 1 à 10, dans lequel le chargement dans le registre (132) du premier paramètre de sécurité du circuit (102) est effectué avant la fin d'une séquence de démarrage du circuit.

[Revendication 12] Procédé selon l'une quelconque des revendications 1 à 11, dans lequel la mémoire non volatile (104) est une mémoire de type FLASH.

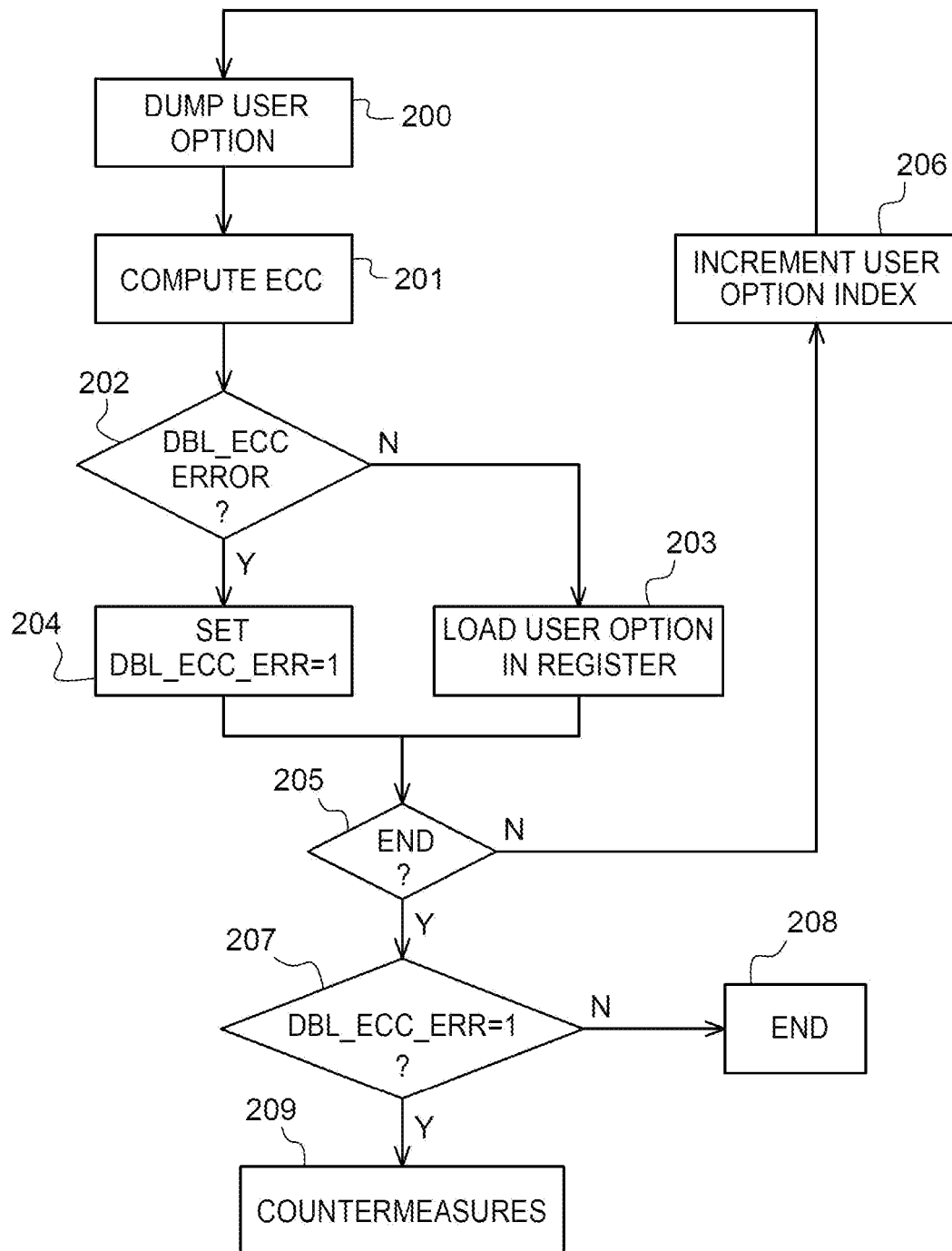
[Revendication 13] Circuit (102) comprenant:

- une mémoire non volatile (104) stockant un premier paramètre de sécurité, et un premier code correcteur d'erreur, stocké en association du premier paramètre dans la mémoire non volatile ; et
- un circuit de calcul (130) configuré pour :
vérifier le premier paramètre de sécurité et le premier code correcteur d'erreur afin de déterminer si un ou plusieurs des bits du paramètre de sécurité sont erronés ; et

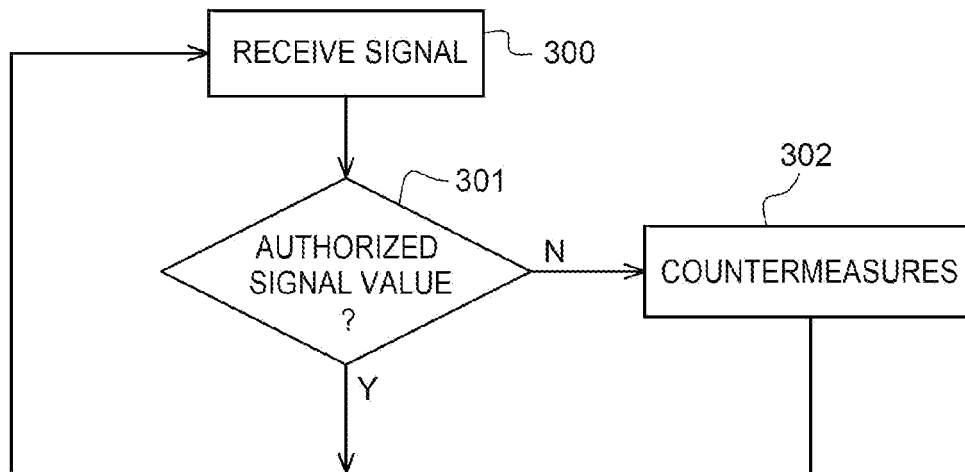
s'il est déterminé que deux bits du paramètre de sécurité sont erronés, charger une valeur par défaut du premier paramètre de sécurité dans un registre (132).

[Revendication 14] Circuit (102) selon la revendication 13, comprenant en outre une interface (106) de la mémoire non volatile (104) configurée pour, lorsqu'il est déterminé que deux bits du paramètre de sécurité sont erronés, transmettre un signal de sortie (TAMPER_OUT) vers un circuit de contrôle (108) du circuit configuré pour réaliser une ou plusieurs contre-mesures.

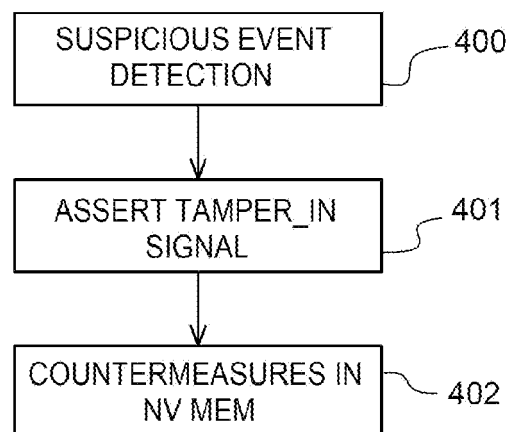
[Fig. 2]

**Fig. 2**

[Fig. 3]

**Fig. 3**

[Fig. 4]

**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 904419
FR 2200857

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	US 2019/028282 A1 (SHARIFI KAMRAN [CA] ET AL) 24 janvier 2019 (2019-01-24) * alinéas [0051], [0056], [0071], [0090], [0107], [0110] - [0124]; revendications 1-3,7; figures 9B, 10A, 10B, 11, 12 * -----	1-14	G06F11/08 G11C7/00 G11C29/42
X	US 10 733 291 B1 (MCLEOD CAMERON [US]) 4 août 2020 (2020-08-04) * le document en entier * -----	1-14	DOMAINES TECHNIQUES RECHERCHÉS (IPC) G06F
Date d'achèvement de la recherche		Examineur	
3 septembre 2022		Bauer, Regine	
CATÉGORIE DES DOCUMENTS CITÉS X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant			

ANNEXE AU RAPPORT DE RECHERCHE PRÉLIMINAIRE **RELATIF A LA DEMANDE DE BREVET FRANÇAIS NO. FR 2200857 FA 904419**

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 **03-09-2022**
 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
US 2019028282 A1	24-01-2019	US 2019028282 A1	24-01-2019
		US 2019028283 A1	24-01-2019

US 10733291 B1	04-08-2020	AUCUN	
