

①9 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

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

2 846 764

②1 N° d'enregistrement national : **02 13753**

⑤1 Int Cl⁷ : G 06 F 13/16

⑫

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 04.11.02.

③0 Priorité :

④3 Date de mise à la disposition du public de la demande : 07.05.04 Bulletin 04/19.

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 :

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

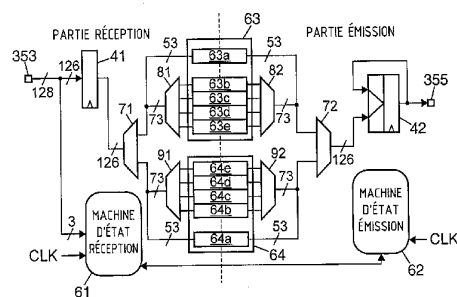
⑦2 Inventeur(s) : RAMANADIN BERNARD.

⑦3 Titulaire(s) :

⑦4 Mandataire(s) : CABINET PLASSERAUD.

⑤4 **PERFECTIONNEMENT AUX SYSTEMES ELECTRONIQUES COMPRENANT UN BUS SYSTEME.**

⑤7 Un module de resynchronisation destiné à être utilisé dans un système électronique comprenant un bus système comprend des moyens de chevauchement pour faire chevaucher les transactions destinées à et/ou provenant dudit module fonctionnel associé. Les moyens de chevauchement comprennent un premier circuit tampon (63) et au moins un second circuit tampon (64) connectés en parallèle, adaptés chacun pour stocker des données de transaction d'une transaction déterminée.



FR 2 846 764 - A1



PERFECTIONNEMENT AUX SYSTEMES ELECTRONIQUES COMPRENANT UN BUS SYSTEME

La présente invention se rapporte aux systèmes électroniques comprenant un bus système, en particulier les microprocesseurs.

De tels systèmes comprennent des modules fonctionnels connectés au bus système via des ports de communication appropriés. On entend par
5 module fonctionnel un module qui permet d'assurer une fonctionnalité déterminée du système. Dans un microprocesseur, de tels modules fonctionnels sont par exemple un cœur de CPU (« Central Processing Unit »), des interfaces permettant l'accès des ressources externes, etc.

Les modules fonctionnels échangent des transactions à travers le bus
10 système. Le terme transaction désigne des opérations initiées par un module fonctionnel maître et effectuées par un module fonctionnel esclave, essentiellement des écritures et des lectures de données. Les transactions suivent des chemins logiques relativement long, qui affectent le temps de propagation des transactions. Or, chaque module fonctionnel, et chaque port
15 de communication des modules fonctionnels, doivent respecter des contraintes de relation de temps (« timing constraints »), qui sont définies comme un pourcentage d'un cycle d'horloge.

De plus, certains modules fonctionnels connectés au bus sont lents en émission ("slow sender") et/ou en réception ("slow receiver"). Il s'ensuit qu'un
20 module fonctionnel esclave peut insérer des cycles d'attente pendant une transaction. Ce type de transaction augmente le temps d'occupation du bus.

C'est pourquoi il est souhaitable de prévoir des moyens pour relâcher les contraintes de relation de temps sur le bus, lorsque les temps de propagation sont critiques compte tenu de la fréquence de fonctionnement du
25 système.

A cet effet, on connaît des moyens pour faire chevaucher ("pipeline" en anglais) les transactions. Un tel chevauchement ("pipelining" en anglais), permet à un module fonctionnel d'émettre ou de recevoir une transaction alors

qu'une autre transaction est en train d'être transmise sur le bus ou exécutée, respectivement.

Dans l'art antérieur, il est connu d'insérer une structure mémoire de type FIFO ("First-in First-out", qui signifie premier entré premier sorti) entre un
5 module fonctionnel et le bus système, pour l'un seulement ou les deux sens de communication (émission et réception). Les transactions passent à travers cette structure mémoire.

La taille de la file doit être suffisante pour stocker toutes les unités de données de protocole ("Protocol Data Unit" en anglais, ou PDU) appartenant
10 aux transactions qui se chevauchent. Par conséquent, si on veut faire chevaucher un nombre déterminé P de transactions s'étalant chacune sur un nombre déterminé Q d'unités de données de protocole, il faut prévoir une file de PxQ registres en série. Il s'ensuit que les transactions doivent systématiquement traverser PxQ registres, ce qui nécessite une durée
15 correspondant à PxQ coups d'horloge. Cette solution introduit donc de la latence sans réduire le temps d'occupation du bus.

Cette latence est encore plus pénalisante lorsque les transactions sont de taille variable, c'est-à-dire qu'elles s'étalent sur un nombre variable d'unités de données de protocole. Par exemple, pour une transaction qui s'étend sur au
20 maximum quatre unités de données de protocole, il faut une file de huit registres de long pour stocker entièrement deux transactions de taille maximum. Or, même les transactions plus courtes, par exemple ne contenant qu'une unique unité de données de protocole, doivent traverser la file de huit registres.

25 La solution proposée par l'invention consiste à insérer un module de resynchronisation (aussi appelé " retiming buffer module", en anglais) entre un module fonctionnel et le bus. Ce module de resynchronisation comprend des circuits tampons ("buffer circuits" en anglais) qui peuvent stocker des données de transaction d'une transaction déterminée. Du point de vue du protocole du
30 bus système, un tel module fonctionnel est transparent. Il a uniquement pour fonction de recevoir, de stocker, puis de transmettre des données d'une

transaction provenant de ou destinée à un module fonctionnel auquel il est associé.

Un premier aspect de l'invention concerne ainsi un système électronique comprenant un bus système et des modules fonctionnels qui peuvent échanger des transactions, éventuellement de taille variable, à travers ledit bus système. Le système comprend au moins un module de resynchronisation disposé entre un module fonctionnel associé et ledit bus système. Le module de resynchronisation comprend des moyens de chevauchement pour faire chevaucher les transactions destinées à et/ou provenant dudit module fonctionnel associé. Les moyens de chevauchement comprennent un premier circuit tampon et au moins un second circuit tampon connectés en parallèle, adaptés chacun pour stocker des données de transaction d'une transaction déterminée.

Dans un mode de réalisation, les moyens de chevauchement comprennent en outre :

- un port d'entrée pour recevoir une transaction d'entrée, ladite transaction d'entrée contenant des données de transaction déterminées ;
- des premiers moyens de gestion pour stocker lesdites données de transaction dans ledit premier circuit tampon ou dans ledit second circuit tampon si ledit premier circuit tampon n'est pas vide, ou pour refuser ladite transaction d'entrée si ledit premier circuit tampon et ledit second circuit tampon ne sont pas vides ;
- des seconds moyens de gestion pour requérir une transaction de sortie dès que lesdites données de transaction sont stockées, pour générer ladite transaction de sortie à partir desdites données de transaction stockées, et pour vider le circuit tampon contenant lesdites données de transaction dès que ladite seconde transaction est transmise ; et,
- un port de sortie pour transmettre ladite transaction de sortie sous la commande desdits seconds moyens de gestion.

Les premiers moyens de gestion peuvent comprendre une première machine d'état et un démultiplexeur, commandé par ladite première machine

d'état. De même, les seconds moyens de gestion peuvent comprendre une seconde machine d'état et un multiplexeur commandé par ladite seconde machine d'état.

De préférence, lorsque les transactions sont de taille variable, le
5 premier circuit tampon et/ou le second circuit tampon comprennent chacun des registres tampons connectés en parallèle en nombre au moins égal au nombre maximum d'unités de données de protocole que peut contenir une transaction. Chacun de ces registres est adapté pour stocker les données de transaction d'une unité de données de protocole respective de la transaction. Ainsi, le
10 temps de transmission des transactions ayant moins de cellules que le nombre maximum de cellules que peut contenir une transaction n'est pas contraint par ce nombre maximum.

Le premier circuit tampon et/ou le second circuit tampon peuvent aussi comprendre un registre tampon qui est adapté pour stocker des données de
15 transaction communes à toutes les unités de données de protocole de la transaction. De cette manière, la taille des registres qui stockent les unités de données de protocole peut être réduite.

Dans un exemple, le système comprend en outre un registre tampon de sortie disposé en amont du port de sortie. Ce registre comprend :

20 - un premier registre à bascules comprenant un nombre déterminé N de bascules ayant chacune une entrée de données, une sortie de données et une entrée d'activation ;

- un deuxième registre à bascules comprenant N bascules, ayant
25 chacune une entrée de données, une sortie de données et une entrée d'activation ;

- un troisième registre à bascule comprenant une bascule ayant une entrée de données, une sortie de données et une entrée d'activation ;

- un multiplexeur de sortie ayant N premières entrées, N secondes entrées, N sorties et une entrée de sélection.

30 Les entrées de données des N bascules du premier registre reçoivent respectivement N signaux d'entrée. Les sorties de données des N bascules du

premier registre sont respectivement reliées aux N premières entrées du multiplexeur de sortie. Les N sorties du multiplexeur de sortie délivrent N signaux de sortie respectifs. Les entrées de données des N bascules du deuxième registre sont respectivement reliées aux N sorties du multiplexeur de sortie. Les N sorties de données des N bascules du deuxième registre sont respectivement reliées aux N secondes entrées du multiplexeur de sortie. La sortie de données du troisième registre est reliée à l'entrée de sélection du multiplexeur de sortie. L'entrée de données du troisième registre reçoit un signal d'activation. Enfin, les entrées d'activation des bascules des premier, deuxième et troisième registres reçoivent un même signal d'horloge.

Un second aspect de l'invention se rapporte à un module de resynchronisation destiné à être utilisé dans un système selon le premier aspect. Ce module comprend des moyens de chevauchement pour faire chevaucher les transactions destinées à et/ou provenant dudit module fonctionnel associé. Les moyens de chevauchement comprennent un premier circuit tampon et au moins un second circuit tampon connectés en parallèle, dont chacun est adapté pour stocker des données de transaction.

Un troisième aspect de l'invention se rapporte à un procédé de gestion d'un système électronique comprenant un bus système et des modules fonctionnels qui peuvent échanger des transactions, de taille éventuellement variable, à travers ledit bus système. Le procédé comprend les étapes consistant à :

- prévoir au moins un module de resynchronisation disposé entre un module fonctionnel associé et ledit bus système ;
- prévoir dans ledit module de resynchronisation des moyens de chevauchement pour faire chevaucher les transactions destinées à et/ou provenant dudit module fonctionnel associé ; et,
- prévoir dans lesdits moyens de chevauchement un premier tampon et au moins un second circuit tampon connectés en parallèle, dont chacun est adapté pour stocker des données de transaction d'une transaction terminée.

Un mode de mise en œuvre du procédé comprend les étapes consistant à :

- recevoir une transaction d'entrée, ladite transaction d'entrée contenant des données de transaction déterminées ;
- 5 - stocker lesdites données de transaction dans ledit premier circuit tampon ou dans ledit second circuit tampon si ledit premier circuit tampon n'est pas vide, ou pour refuser ladite transaction d'entrée si ledit premier circuit tampon et ledit second circuit tampon ne sont pas vides,
- requérir une transaction de sortie dès que lesdites données de
10 transaction sont stockées ;
- générer ladite transaction de sortie à partir desdites données de transaction stockées ;
- vider le circuit tampon contenant lesdites données de transaction dès que ladite seconde transaction est transmise ;
- 15 - transmettre ladite transaction de sortie via un port de sortie sous la commande desdits seconds moyens de gestion.

Lorsque les transactions sont de taille variable, l'étape consistant à stocker les données de transaction dans le premier circuit tampon ou dans le second circuit tampon comprend de préférence les étapes consistant à :

- 20 - prévoir dans chacun des premier et second circuits tampon des registres tampons connectés en parallèle, en nombre au moins égal au nombre maximum d'unités de données de protocole que peut contenir une transaction ;
et,
- stocker les données de transaction de chaque unité de données de
25 protocole de la transaction dans l'un respectif des registres tampon du circuit tampon.

De plus, l'étape consistant à stocker lesdites données de transaction dans le premier circuit tampon ou dans le second circuit tampon peut aussi comprendre les étapes consistant à :

- prévoir en outre dans chacun des premier et second circuits tampon un registre tampon supplémentaire connecté en parallèle ; et,

- stocker dans ledit registre tampon des données de transaction communes à toutes les unités de données de protocole de la transaction.

5 D'autres caractéristiques et avantages de l'invention apparaîtront encore à la lecture de la description qui va suivre. Celle-ci est purement illustrative et doit être lue en regard des dessins annexés sur lesquels :

- la figure 1 est un schéma simplifié d'un microprocesseur comprenant un bus système ;

10 - la figure 2 est un diagramme illustrant l'organisation en couche du protocole Super Hyway ;

- la figure 3 est un schéma illustrant la connection d'un module initiateur et d'un module cible au bus Super Hyway ;

15 - la figure 4 montre des chronogrammes de signaux du bus Super Hyway pendant une requête à quatre cellules ;

- la figure 5 et la figure 6 sont des schémas illustrant l'agencement dans un système des modules de resynchronisation selon la présente l'invention ;

20 - la figure 7 est un schéma détaillé d'un mode de réalisation d'une partie d'un module de resynchronisation selon la présente l'invention ;

- la figure 8 est un schéma illustrant le principe d'une règle de conception avantageuse d'un module de resynchronisation selon l'invention ;

25 - la figure 9 illustre un premier mode de réalisation d'un circuit tampon de sortie d'un exemple de module de resynchronisation selon la présente l'invention ; et,

- la figure 10 illustre un second mode de réalisation d'un circuit tampon de sortie d'un exemple de module de resynchronisation selon la présente l'invention.

Dans la présente description de l'invention, on considère l'exemple d'un système électronique ayant un bus appelé "Super Hyway". C'est un bus système pour les microprocesseurs ayant une architecture à chevauchement. Il a été conçu pour fonctionner à 200 MHz. Le bus Super Hyway est la moelle épinière des SOC ("System on Chip", qui signifie système sur puce) développés par STMicroelectronics et Hitachi. Il est bien entendu que le principe de l'invention ne se limite pas, toutefois, à cet exemple.

Le bus Super Hyway supporte un protocole de transactions basé sur la transmission de paquets. Ce protocole constitue la couche la plus haute de l'infrastructure de communication entre deux modules fonctionnels (maître et esclave) connectés au bus. Dans la terminologie du bus Super Hyway, les modules maîtres sont appelés initiateurs ("initiators" en anglais) et les modules esclaves sont appelés cibles ("target" en anglais).

Une transaction définit la communication entre un port initiateur du module initiateur et un port cible du module cible. Il s'agit d'un cycle complet de traitement de données qui est exécuté en réponse à une demande et qui constitue un tout en soi, avec un début et une fin. Les transactions du protocole Super Hyway sont de deux types seulement : des requêtes et des réponses. Cependant, le protocole est le même dans les deux sens, interrogation et réponse. Seuls le nom et le nombre des signaux du bus qui sont utilisés sont différents d'un type de transaction à l'autre. Les données d'une transaction du protocole Super Hyway sont comprises dans une ou plusieurs unités de données de protocole appelées cellules. Une transaction peut contenir une, deux ou quatre cellules. C'est en ce sens qu'on dit qu'elle est de taille variable. Dans un exemple, chaque cellule contient 8 octets, c'est-à-dire 64 bits.

La figure 1 donne un schéma des principaux éléments compris dans un exemple de système électronique dans lequel l'invention peut être mise en œuvre. Cet exemple de système est le microprocesseur ST50 développé par STMicroelectronics.

Le microprocesseur ST50 est un SOC d'évaluation destiné à tester et démontrer les capacités du cœur ("Core" en anglais) SH5 dans un système complet. Le SH5 est un cœur de CPU ("Central Processing Unit"), le dernier

membre de la famille SuperH, développée conjointement par Hitachi et STMicroelectronics. Le microprocesseur ST50 a été conçu pour être utilisé dans des systèmes de développement et des plate-formes d'application pour les développeurs et utilisateurs du cœur SH5.

5 Dans l'exemple, le système comprend un cœur 11, un bus système 12 et des modules fonctionnels 13 à 16.

Le cœur 11 est le SH5. C'est un processeur RISC à 64 bits, haute performance, faible consommation et de taille réduite. Il est dédié aux applications embarquées, spécialement dans le domaine de la télévision numérique, de la télécommunication et l'automobile. Ce cœur possède des fonctionnalités adaptées aux applications multimédias, accessibles au travers de son jeu d'instruction SIMD ("Single Instruction Multiple Data"). Une unité virgule flottante optionnelle, conforme à la norme IEEE 754, permet d'effectuer des calculs graphiques 3D. L'architecture du jeu d'instructions possède un mode compatible avec le ST40, qui est le microprocesseur de la génération précédente, et un mode 64 bits natif pour de hautes performances ou pour les modules multimédias. Des fonctions de débogage non-intrusives rendent possible le développement de systèmes temps réel. Ces fonctionnalités comprennent un mécanisme de traçabilité temps réel. Ce cœur est fourni sous la forme d'un ASIC pour être implémenté dans des SOC.

Le bus système 12 est un bus Super Hyway. C'est un bus cadencé à 200 MHz, autorisant des taux de transferts allant jusqu'à 3,2 Gigaoctets/s. Il a une largeur de 64 bits de données et est "full duplex". Il supporte des modes de transfert en rafale. Cela permet des transferts de 32 octets par transaction. Ce bus possède un arbitre 12a ("arbiter" en anglais) qui attribue l'accès au bus selon la technique du tourniquet.

Un premier module fonctionnel 13 est constitué d'une interface PCI ("Peripheral Component Interconnect"). L'interface PCI est d'une largeur de 32 bits. Elle autorise les fréquences de fonctionnement de 33 et 66 MHz. Elle permet d'adresser un espace mémoire continu de 512 Mo et accepte jusqu'à 4 maîtres sur le bus. L'intégration d'une interface PCI permet la connexion au

microprocesseur ST50 de nombreux périphériques standards, avant même que des blocs spécifiques soient développés pour être intégrés au SOC.

Un deuxième module fonctionnel 14 comprend une interface P-Bus ("Peripheral Bus"), qui assure le rôle de passerelle vers un autre bus permettant l'extension du système à des composants périphériques.

Un troisième module fonctionnel 15 comprend une interface dite EMI ("External Memory Interface", qui signifie interface de mémoire externe). L'EMI permet de connecter des mémoires de type RAM, DRAM ou SDRAM avec des bus de données de 16, 32 ou 64 bits, et des fréquences d'horloge comprises entre 66 et 133 MHz. On peut utiliser des DRAMs de taille 16 Mo, 64 Mo, 128 Mo, 256 Mo, 512 Mo. Grâce à deux signaux CS (Chip Select), l'EMI peut adresser jusqu'à 1 Go.

Enfin, un quatrième module fonctionnel 16 comprend une interface dite FEMI ("Flash EMI"). La FEMI permet de connecter des mémoires de type Flash/ROM avec des bus de données de 8, 16, 32 bits et un bus d'adresses de 26 bits. Elle permet des transferts en mode rafale synchrone et rafale asynchrone. Elle supporte le protocole MPX et peut fonctionner aux fréquences 50, 66 et 100 MHz.

Le système comprend d'autres éléments, notamment une horloge système qui assure le cadencement des modules fonctionnels et du bus, qui ne sont pas représentés à la figure 1.

La diagramme de la figure 2 illustre la structure en couches du protocole de communication du bus Super Hyway. Comme de nombreux protocoles de réseaux, le protocole Super Hyway est en effet décomposé en couches. Il possède quatre couches qui sont les couches transaction, paquet, cellule et physique.

Une transaction définit la communication entre deux modules fonctionnels du système. Cette communication peut être un simple octet ou une structure de données complète. Pour l'implémentation sur le bus Super Hyway, les transactions sont découpées en une série d'opérations mémoire. La couche paquet sépare les opérations en une paire requête / réponse. A ce

niveau, les paquets contiennent une quantité fixe de données qui doit être rendue compatible avec l'implémentation du bus. La couche cellule casse donc les paquets en une série de cellules, qui ont toutes la bonne taille. En effet, la taille du bus peut varier suivant ses implémentations. Dans le cas du ST50, la

5 partie données du bus fait 64 bits de large. Les paquets sont donc décomposés en cellules de 64 bits de données. La couche physique, comme son nom l'indique réalise l'encodage physique de ces cellules en ajoutant des informations de structure et de contrôle de flux.

De plus amples détails sur chacune des couches du protocole sont

10 donnés ci-dessous.

Couche transaction

Un module communique avec le reste du système en utilisant des transactions standards. Ces transactions définissent les opérations qui peuvent être générées lorsqu'un module communique avec le système. Il s'agit

15 typiquement d'opérations dans une mémoire comme des lectures ou des écritures de différentes tailles. La transaction comporte toutes les informations nécessaires au système pour définir de manière non ambiguë l'opération demandée ainsi que ses paramètres.

Les opérations possibles sont :

20 - l'opération LOAD (sur M octets) permet de lire un mot aligné de M octets du module cible vers le module initiateur. Les valeurs possibles pour M sont 1, 2, 4, 8, 16, 32 ;

- l'opération STORE (sur M octets) permet d'écrire un mot aligné de M octets du module initiateur dans le module cible. Cette opération écrase la

25 donnée précédente à l'adresse spécifiée. Les valeurs autorisées pour M sont 1, 2, 4, 8, 16, 32 ;

- l'opération SWAP (sur M octets) échange la valeur d'un mot aligné de M octets du module initiateur avec la donnée stockée à l'emplacement spécifié dans le module cible. La donnée qui était à l'origine dans le module cible est

30 retournée au module initiateur. Les valeurs autorisées pour M sont 4 et 8 ;

- l'opération FLUSH retourne une réponse lorsqu'une copie de la donnée, associée à une adresse physique déterminée, qui devrait être stockée dans un module cible est effectivement présente à l'adresse physique spécifiée. Le module cible retient alors une copie de la donnée ;

- 5 - l'opération PURGE retourne une réponse lorsqu'une copie de la donnée, associée à une adresse physique déterminée, qui devrait être stockée dans un module cible est effectivement présente à l'adresse physique spécifiée. La donnée est alors supprimée du module cible.

Couche paquet

- 10 La couche paquet sépare les transactions en requêtes et en réponses. Les systèmes développés sur la base du bus Super Hyway sont composés de modules de deux types seulement : initiateur et cible.

Les modules initiateurs initient les opérations systèmes. Ils les commandent en envoyant une opération et une adresse. Les opérations envoyées par les modules initiateurs à un module cible sont les requêtes.

Les modules cibles reçoivent les requêtes et effectuent les opérations correspondantes. Le résultat de l'opération est renvoyé au module initiateur au moyen d'une autre transaction, qui est une réponse.

Couche cellule

- 20 La couche cellule scinde les paquets en sous-ensembles compatibles avec la taille du bus Super Hyway. En effet, les spécifications du bus Super Hyway ne déterminent pas la taille du bus de données. Suivant l'implémentation, celui-ci peut mesurer 16, 32 ou 64 bits de large. Dans le cas du ST50, le bus de données du bus Super Hyway mesure 64 bits de large.

- 25 Les cellules d'un paquet de requête sont notées 1, 2, 3 et 4. Les cellules d'un paquet de réponse sont notées a, b, c, et d.

Comme nous l'avons vu en décrivant les différentes transactions possibles sur le bus Super Hyway, le nombre d'octets à transmettre peut être de 1, 2, 4, 8, 16 ou 32. Pour des opérations transmettant 1, 2, 4 ou 8 octets, 30 une seule cellule suffit car elle peut à elle seule transmettre les 64 bits de

données. Pour une opération transférant 16 octets, deux cellules sont nécessaires. Et pour une opération transférant 32 octets, quatre cellules sont nécessaires. Dit autrement, la taille du paquet (en nombre de cellules) dépend de l'opération.

5 Couche physique

L'interface du bus Super Hyway est constituée d'une série de signaux, notamment des signaux de données et des signaux d'adresses. La couche physique permet de donner à ces signaux les valeurs permettant de transmettre les cellules. La couche physique encode donc les cellules et ajoute
10 des informations de structure et de contrôle de flux.

Le schéma de la figure 3 donne une illustration des différents signaux du bus Super Hyway 12 et leur relation avec le module initiateur 41 et le module cible 42 qui sont parties à la transaction. Les signaux associés à la requête sont:

- 15 - req (request) : est actif lorsque le module initiateur est prêt à envoyer la donnée ;
- gnt (grant) : est actif lorsque le système est prêt à recevoir des données du module initiateur ;
- eop (end of packet) : indique la dernière cellule du paquet ;
- 20 - add[31:3] (address) : encodent l'adresse de la cible de la transaction ;
- opc[7:0] (opcode) : encodent l'opération associée à la transaction ;
- src[7:0] (source identity) : définissent l'identificateur de la source ;
- tid[7:0] (transaction identity) : définissent l'identificateur de la transaction ;
- 25 - msk[7:0] (mask) ou be[7:0] (byte enable) : indiquent les octets valides dans la cellule ; et
- data[63:0] (data) : encodent la donnée portée par la cellule.

Et voici les signaux associés à la réponse:

- r_req (response request) : est actif lorsque le module cible est prêt à envoyer la donnée ;
- r_gnt (response grant) : est actif lorsque le système est prêt à recevoir des données du module cible ;
- 5 - r_eop (response end of packet): indique la dernière cellule du paquet réponse ;
- r_opc[7:0] (response opcode): encodent la réponse du module cible à l'opération ;
- r_data[63:0] (response data) : encodent la donnée associée à la
- 10 cellule de réponse ;
- r_tid[7:0] (response indentifier) : encodent une copie de l'identificateur de transaction ; et
- r_src[7:0] (response source) : encodent une copie de l'identificateur de source.
- 15 Afin de donner une illustration du fonctionnement de la couche physique du protocole pendant un exemple de transaction, les chronogrammes de la figure 4 illustrent l'état des signaux du bus durant une requête à quatre cellules (a priori la plus complexe).
- Lorsqu'un module initiateur génère une requête, il place les
- 20 informations valides sur le bus (signaux data, msk, opc, add, tid et eop) et place la ligne req à 1. Le système signale qu'il est prêt à recevoir la transaction en plaçant la ligne gnt à 1. Le système (plus particulièrement l'arbitre du bus Super Hyway) utilise alors le champ adresse pour sélectionner le module cible auquel la transaction est destinée.
- 25 Le champ opcode indique quelle est l'opération associée à la transaction. Ce champ se rapporte donc au module cible qui devra effectuer l'opération mais aussi au bus Super Hyway. En effet, le champ opcode indique le nombre de cellules contenues dans le paquet. Le champ src porte l'identificateur du module initiateur qui génère la requête. Le champ tid permet
- 30 d'identifier la transaction de façon non ambiguë.

Les champs opc, add, src, tid restent constants tout au long d'une transaction. C'est pourquoi ces champs définissent ce qui est appelé un en-tête ("Header" en anglais). La dernière cellule d'un paquet est signalée par le signal eop. Celui-ci est à 1 lorsque la cellule transmise est la dernière du paquet. Il est
5 toujours à 0 sinon.

Une requête commence lorsque le module initiateur place la ligne req à 1 et se termine lorsque les trois signaux req, gnt et eop sont simultanément à 1. Une fois que le module initiateur a placé la ligne req à 1, il est obligé de terminer la transaction commencée. C'est-à-dire qu'il ne pourra pas replacer la
10 ligne req à 0 avant que la condition $eop=gnt=req=1$ n'ait été rencontrée. De même, lorsque la ligne gnt a été placée à 1, le système doit pouvoir recevoir la totalité de la transaction. C'est à dire qu'il ne pourra pas replacer la ligne gnt à 0 avant que la condition $eop=gnt=req=1$ n'ait été rencontrée.

Les chronogrammes illustrant une requête à une cellule et une requête
15 à deux cellules se déduisent facilement de ce qui précède. De même, les transactions de réponses répondent aux mêmes règles. Il faut simplement remplacer req par r_req, gnt par r_gnt, eop par r_eop, opc par r_opc, tid par r_tid et data par r_data. On notera en outre que les signaux msk (ou be) et add n'ont pas d'équivalent ici car ils ne servent pas pour la réponse.

20 Un module fonctionnel lent en émission peut avoir besoin de temps entre l'émission de deux cellules d'un même paquet. Dans ce cas, il insère un cycle d'attente en amenant temporairement le signal req (pour un SIF initiateur) ou le signal r_req (pour un SIF cible) à 0. Pendant la durée de ce cycle, les autres signaux sont maintenus à leurs niveaux respectifs.

25 Le schéma de la figure 5 illustre le détail de la connexion de modules fonctionnels au bus système selon un exemple de mise en œuvre de l'invention. Un module fonctionnel 31 d'un premier type comprend au moins un port initiateur INI et au moins un port cible TGT. Dit autrement, le module fonctionnel 31 peut être initiateur ou cible, selon la transaction qui est
30 concernée. Un autre module fonctionnel 32 d'un deuxième type ne comprend qu'un ou plusieurs ports cibles TGT. D'autres modules fonctionnels peuvent aussi être connectés au bus 12. On notera qu'un troisième type de module

fonctionnel (non représenté) peut n'avoir qu'un ou plusieurs ports initiateurs INI. Chaque port INI ou TGT comprend en fait deux ports : l'un Req pour émettre des requêtes, et l'autre Resp pour recevoir des réponses.

On notera que, dans l'exemple de système qui a été décrit plus haut en regard de la figure 2, tous les modules fonctionnels 11 à 16 sont du premier type précité.

Afin d'alléger les contraintes de relations temporelles, il est prévu de disposer des modules de resynchronisation entre certains au moins des modules fonctionnels et le bus système. Ces modules de resynchronisation sont appelés SIF ("SuperHyway InterFace"). Le rôle de ces SIFs est de casser le long chemin combinatoire qui va du port initiateur au port cible des modules fonctionnels qui sont partie à une transaction donnée. Pour cela, ils comprennent des moyens pour faire chevaucher les transactions. Dit autrement, ils assurent une fonction de tampon pour les transactions.

Comme les ports initiateurs et les ports cibles sont différents, il est préférable de prévoir deux types de module de resynchronisation, un pour chaque type de port. Dans ce cas, il y a un SIF initiateur et un SIF cible. Les SIFs sont destinés à être insérés autour du bus Super Hyway, chacun entre un module fonctionnel associé et le bus.

Dans l'exemple représenté à la figure 5, un SIF initiateur 33 est ainsi disposé entre le port INI du module initiateur 31 et le bus 12. De même, deux SIF cibles 34 et 35 sont disposés entre les ports TGT des modules cibles respectivement 31 et 32 d'une part, et le bus 12 d'autre part. Le module 31 possède en effet à la fois un port initiateur et un port cible.

Toutefois, en variante, on peut créer un unique type de SIF de structure plus complexe, qui est adapté pour être associé indistinctement à un module initiateur ou à un module cible.

Grâce aux SIFs, le flux de données d'une transaction entre un module initiateur et un module cible est décomposé en plusieurs flux de données intermédiaires : un premier flux intermédiaire entre le module initiateur et le SIF initiateur auquel il est associé ; un deuxième flux intermédiaire entre le SIF

initiateur et le SIF cible, via le bus 12 ; et enfin une troisième flux intermédiaire entre le SIF cible et le module cible auquel il est associé.

On notera que, lorsque seul le module initiateur est relié au bus via un SIF initiateur, le module cible étant directement relié au bus, le deuxième flux intermédiaire n'existe pas. Ledit troisième flux intermédiaire (qui en fait est le second) a alors lieu entre le SIF initiateur et le port cible, via le bus 12.

Inversement, lorsque seul le module cible est relié au bus via un SIF cible, le module initiateur étant directement relié au bus, le deuxième flux intermédiaire n'existe pas. Le premier flux intermédiaire a alors lieu entre le module initiateur et le SIF cible, via le bus 12. Et le troisième flux intermédiaire (qui en fait est le second) a lieu entre le SIF cible et le module cible.

On notera que les flux de données intermédiaires entre un SIF et le module fonctionnel auquel il est associé obéissent aux règles de la couche la plus basse (i.e. la couche physique) du protocole du bus système. De la sorte, la présence du SIF est complètement transparente pour le module fonctionnel auquel il est associé, pour les autres modules fonctionnels, et pour le bus. En particulier, le module fonctionnel associé à un SIF reste accessible par le reste du système via son adresse propre, le SIF n'ayant pas d'adresse mais étant physiquement connecté entre le module fonctionnel et le bus.

Dans un exemple de mise en œuvre, chaque module SIF comprend deux blocs principaux, un bloc requête pour transmettre les requêtes qui viennent du module initiateur et un bloc réponse pour transmettre les réponses qui viennent du module cible. Dit autrement, les SIFs traitent les données au niveau de la couche paquet du protocole.

A titre d'exemple, la figure 6 montre le détail de l'agencement du SIF cible 35 entre le bus Super Hyway 12 et le module cible 32 de la figure 3. Un bloc requête 351 du SIF cible 35 accepte ou refuse une requête venant d'un module initiateur (non représenté) à travers le bus 12. S'il accepte, il reçoit sur un port d'entrée 353 la requête venant du bus 12, stocke les données de requête dans une structure tampon, puis les émet sur un port de sortie 355 vers le port Req du module cible 32 lorsqu'une requête demandée par le SIF

cible 35 et destinée au module cible 32 est acceptée par le système. A l'inverse, un bloc réponse 352 accepte ou refuse une réponse venant du module cible 32 auquel il est associé. S'il accepte, il reçoit sur un port d'entrée 354 la réponse venant du port Resp du module cible 32, stocke les données de
5 réponse dans une structure tampon, puis les émet sur un port de sortie 356 vers le bus 12 lorsqu'une réponse demandée par le module SIF cible 35 et destinée au module initiateur (non représenté) est acceptée par le système.

On va maintenant décrire la micro-architecture des SIFs. Au préalable, tournons nous vers la figure 7 qui illustre une règle de conception avantageuse
10 pour la réalisation de la micro-structure des SIFs.

Suivant cette règle, on fait en sorte qu'un maximum des signaux d'entrée soit reçu sur les entrées de données respectives d'un registre d'entrée 41. De plus, on fait en sorte qu'un minimum des signaux d'entrée soit reçu sur des entrées de commande d'éléments synchrones formant une
15 logique combinatoire 43 ("Glue" en anglais et dans le jargon de l'Homme du métier). Enfin, on fait en sorte que la totalité des signaux de sortie soit délivré sur les sorties de données respectives d'un registre de sortie 42. Dans un exemple, les registres d'entrée et de sortie sont des registres à bascules de type D ("flip-flop", en anglais). La règle impose alors de concevoir la micro-
20 architecture du SIF de telle manière qu'un maximum de signaux d'entrées arrive sur des entrées D et que tous les signaux de sorties soient délivrés par des sorties Q de telles bascules. Les SIFs initiateur et cible étant très similaires, on considère ici le cas d'un SIF cible seulement. L'exposé qui suit est néanmoins également valable pour un SIF initiateur, moyennant les
25 adaptations nécessaires, qui sont à la portée de l'Homme du métier. Il suffit de se rappeler qu'un SIF initiateur reçoit des requêtes et délivre des réponses, respectivement du et au module initiateur auquel il est associé, alors que pour un SIF cible c'est l'inverse : il reçoit des réponses et délivre des requêtes respectivement du et au module cible auquel il est associé. De plus, le
30 protocole est identique pour les requêtes et les réponses, seul le nom et le nombre de signaux du bus qui interviennent sont différents. La micro-

architecture du bloc réponse d'un SIF est donc très similaire à celle du bloc requête de ce SIF.

C'est pourquoi, à titre d'exemple, le schéma de la figure 8 montre la micro-architecture du bloc requête d'un SIF cible.

5 La micro-architecture de chaque bloc se décompose en une partie réception et une partie émission. La partie réception du bloc requête gère la réception et le stockage des requêtes venant du bus Super Hyway. La partie émission du bloc requête gère la réémission des données de requête stockées
10 machine d'état, respectivement 61 et 62. A la figure 8, la zone contrôlée par une machine d'état est délimitée par une ligne verticale en pointillés, la partie réception et la partie émission étant représentées respectivement à gauche et à droite de cette ligne.

15 Par exemple, les machines d'état sont des machines d'état avec des sorties synchronisées. Ceci permet de donner des marges pour les contraintes de relation de temps. Il peut s'agir de machines d'état de type Moore ou de type Mealy. Les signaux de contrôle délivrés par ces machines d'état ne sont pas représentés pour ne pas surcharger le schéma.

20 Le port d'entrée 353 reçoit les signaux du bus 12 qui ont été présentés plus haut en regard du schéma de la figure 3. Par souci de clarté, ces signaux sont représentés à la figure 8 par un trait unique correspondant à 128 lignes de bus.

25 En accord avec la règle de conception présentée plus haut en regard du schéma de la figure 7, la partie réception comprend un registre d'entrée 41, comprenant des bascules pour recevoir 126 bits de données qui se répartissent comme suit : un bit de données qui est porté par le signal eop, 28 bits de données portés par le signal add[31:3], 8 bits de données portés par le signal opc[7:0], 8 bits de données portés par le signal src[7:0], 8 bits de données portés par le signal et tid[7:0], 8 bits de données portés par le signal
30 msk[7:0] (ou be[7:0]), et 64 bits de données d'une respective des 4 cellules (au plus) de la transaction qui sont portés par le signal data[63:0].

De même, la partie émission comprend un registre de sortie 42 comprenant des bascules pour délivrer les 128 bits de données de la requête. Deux modes de réalisation du registre de sortie 42 seront décrits plus loin en regard des schémas des figures 9 et 10, respectivement.

5 La machine d'état 61 reçoit les signaux req, gnt et eop, totalisant 3 lignes. De plus, la machine d'état 61 et la machine d'état 62 reçoivent le signal d'horloge CLK qui est le signal de l'horloge système (dont la fréquence maximum est égale à 200 Mhz). Ainsi, chaque machine d'état est synchrone de l'horloge système.

10 Le module SIF contient deux circuits tampon 63 et 64. Chacun des circuits tampons 63 et 64 est adapté pour stocker les données de transaction d'une transaction déterminée. En particulier ils comprennent chacun un registre d'en-tête respectivement 63a et 64a, comprenant des bascules pour stocker les 53 bits de données portés par les signaux d'en tête add[31:3], opc[7:0],
15 src[7:0], tid[7:0] et eop. De plus, ils comprennent chacun quatre registres de cellule respectivement 63b à 63e et 64b à 64e. Chaque registre de cellule comprend des bascules pour stocker 73 bits de données qui se décomposent comme suit : 64 bits de données d'une respectivement des 4 cellules (au plus) de la transaction qui sont portés par le signal data[63:0], 8 bits de données qui sont
20 portés par le signal msk[7:0] (ou be[7:0]), et un bit de données qui est porté par le signal eop.

En résumé, chaque circuit tampon peut stocker 4 cellules plus l'en-tête de la transaction qui reste constant tout au long de la transaction. Bien entendu, le nombre de registres des circuits tampon, et leur taille, dépendent
25 de l'implémentation du bus. Ici il y a quatre registres de cellule car une transaction peut contenir au plus quatre cellules. Mais ceci n'est qu'un exemple.

Bien entendu, il est possible de prévoir plus de deux circuits tampon tels que les circuits 63 et 64, pour faire chevaucher plus de deux transactions.
30 Dans l'application considérée ici, il a été estimé que la plus grande complexité de la micro-structure qui en résulte ne se justifiait pas compte tenu des effets observés sur le temps de propagation des transactions.

La partie réception comprend un démultiplexeur 71 qui reçoit en entrée les 126 bits de sortie du registre d'entrée 41 et les oriente soit vers le circuit tampon 63 soit vers le circuit tampon 64, sous la commande de la machine d'état 61. Dans le premier cas, un autre démultiplexeur 81 à quatre sorties permet d'orienter les (au plus) quatre cellules d'un paquet qui sont reçues successivement, vers l'un des registres 63b à 63^e du circuit tampon 63. De même, dans le second cas, un autre démultiplexeur 91 à quatre sorties permet d'orienter les (au plus) quatre cellules d'un autre paquet qui sont reçues successivement, vers l'un des registres 64b à 64e du circuit tampon 64.

10 Le registre d'entrée 41, les démultiplexeurs 71, 81 et 91, et les circuits tampons 63 et 64 sont pilotés de façon synchrone par la machine d'état 61 de la partie réception.

La partie émission comprend un multiplexeur 72 pour orienter vers le registre de sortie 42, soit les données de transaction stockées dans le circuit tampon 63 soit celles stockées dans le circuit tampon 64, sous la commande de la machine d'état 62. Dans le premier cas, les (au plus) quatre cellules d'un paquet stockées dans les registres 63b à 63e du circuit tampon 63 sont successivement orientées vers le multiplexeur 72 à l'aide d'un multiplexeur 82 à quatre entrées. De même, dans le second cas, les (au plus) quatre cellules d'un paquet stockées dans les registres 64b à 64e du circuit tampon 64 sont successivement orientées vers le multiplexeur 72 à l'aide d'un multiplexeur 92 à quatre entrées.

25 Les circuits tampon 63 et 64, les multiplexeurs 72, 82 et 92 ainsi que le registre de sortie 42 sont pilotés de façon synchrone par le machine d'état 62 de la partie réception. Le fait d'avoir deux circuits tampons (ici dans le bloc requête) permet au SIF de faire chevaucher les transactions (ici les requêtes).

30 Le fonctionnement du bloc requête du SIF cible est le suivant. Lorsque le SIF accepte une première transaction (requête), les données de la transaction sont stockées dans un premier des circuits tampons qui est vide. Le SIF demande alors au plus tôt une transaction (requête) au module fonctionnel (cible) auquel il est associé. Lorsque cette transaction est acceptée, les données de transaction sont lues dans ledit premier circuit tampon pour être

émises vers le module cible associé au SIF. Le premier circuit tampon est alors vidé. Pendant ce temps, une seconde transaction peut être acceptée par le SIF et les données de transaction correspondantes sont stockées dans le deuxième circuit tampon qui est vide. Dès que possible (notamment dès que la transaction qu'il a demandé est terminée, si elle est en cours), le SIF demande une seconde transaction au module fonctionnel (cible) associé afin d'émettre les données de transaction stockées dans le second circuit tampon et de pouvoir vider celui-ci.

Les deux machines d'état communiquent entre elles pour indiquer quel circuit tampon est prêt à être lu ou lequel vient d'être vidé et peut à nouveau stocker des données de transaction. On notera que si les deux circuits tampons 63 et 64 sont non vides, le SIF refuse la transaction s'il en reçoit une. On rappelle ainsi qu'il est possible d'augmenter le nombre de circuits tampon au-delà de deux, pour pouvoir stocker des données de transaction de plus de deux transactions à la fois.

On peut constater que cette micro-architecture respecte une des contraintes définies au début du projet. En effet, un maximum des entrées est sur des entrées D de bascules (celles du registre d'entrée 41). Seuls les signaux d'entrée qui arrivent directement sur la machine d'état 61 (à savoir les signaux req, gnt et eop dans l'exemple) traversent la logique combinatoire. Il s'agit des signaux de contrôle de flux. Il n'est en effet pas possible de les décaler d'un coup d'horloge et donc de leur faire traverser un registre.

On va maintenant décrire deux exemples de réalisation du registre de sortie 42 qui est disposé en amont du port de sortie 355. Par souci de généralité, on considère que ce registre transmet un nombre déterminé N de signaux. Dans l'exemple, N est égal à 126.

Dans un mode de réalisation classique, représenté schématiquement à la figure 9, le registre de sortie 42 comprend un unique registre à bascules 436, et un multiplexeur d'entrée 435. Les bascules sont par exemple des bascules de type D ("flip-flop", en anglais).

Le registre tampon de sortie comprend un unique registre à bascules 436 ou registre d'entrée, et un multiplexeur d'entrée 435.

Le registre 436 comprend N bascules, ayant chacune une entrée de données, une sortie de données et une entrée d'activation. Le multiplexeur 435
5 comprend N premières entrées 435a, N secondes entrées 435b, N sorties 435c et une entrée de sélection 435d.

Les N entrées 435a du multiplexeur reçoivent respectivement les N signaux d'entrées data_in, qui sont les données délivrées par la sortie du multiplexeur 72. Les N sorties 435c du multiplexeur sont respectivement reliées
10 aux N entrées de données du registre 436. Les N sorties de données 436b des bascules respectives du registre 436 délivrent N signaux de sortie data_out, respectivement. Ces N signaux de sortie sont en outre délivrés sur les N entrées 435b du multiplexeur 435.

Les N entrées d'activation 436c des bascules respectives du registre
15 436 reçoivent un signal d'horloge CLK. En outre, un signal shw_sifini_gnt est reçu sur l'entrée de sélection 435d du multiplexeur 435. Ce signal est un signal d'activation qui est par exemple délivré par une machine d'état.

Le registre 436 en combinaison avec le multiplexeur 435 permet de verrouiller les valeurs des signaux de sortie lorsque le signal shw_sifini_gnt est
20 à 0.

Cet art antérieur respecte la règle de conception précitée, puisque les signaux de sortie sont délivrés par des sorties de données de bascules.

Par contre, le signal shw_sifini_gnt attaque N entrées d'activation, à savoir les entrées 435d des N bascules du multiplexeur 435. Ceci nécessite
25 souvent d'introduire ce qu'on appelle un "arbre de bufferisation" ("buffer tree" en anglais), qui alourdit la micro-structure et introduit de la latence en entrée du registre tampon.

La figure 10 montre un mode de réalisation préféré du registre de sortie 42, comprenant trois registres à bascules et un multiplexeur de sortie. Les
30 bascules sont par exemple des bascules de type D.

Le registre 42 comprend ainsi un premier registre à bascules 431 ou registre d'entrée de données. Le registre 431 comprend N bascules ayant chacune une entrée de données, une sortie de données et une entrée d'activation. Les N entrées de données 431a desdites N bascules reçoivent
5 respectivement les N signaux d'entrée data_in. Les N sorties de données 431b desdites N bascules sont reliées à N premières entrées respectives 434a d'un multiplexeur de sortie 434, ayant N premières entrées 434a, N secondes entrées 434b, N sorties 434c et une entrée de sélection 434d. Les N sorties 434c du multiplexeur de sortie 434 délivrent les N signaux de sortie data_out.

10 Le registre 42 comprend en outre un deuxième registre à bascules 432 ou registre de verrouillage. Le registre 432 a également N bascules, chacune ayant une entrée de données, une sortie de données et une entrée d'activation. Les N entrées de données 432a des N bascules du registre 432 sont respectivement reliées aux N sorties 434b du multiplexeur de sortie 434, pour
15 respectivement recevoir les N signaux de sortie. De plus, les N sorties de données des bascules du registre 432 sont respectivement reliées aux N secondes entrées du multiplexeur de sortie 434.

Le registre 42 comprend en outre un troisième registre à bascule 433 ou registre d'activation. Le registre 433 comprend une bascule qui a une entrée
20 de données 433a, une sortie de données 433b et une entrée d'activation 433c. L'entrée de données 433a reçoit le signal d'activation shw_sifini_gnt délivré par une machine d'état faisant partie de la logique combinatoire 43. La sortie de données 433b est reliée à l'entrée de sélection du multiplexeur de sortie 434.

Les entrées d'activation 431c, 432c et 433c des bascules respectives
25 des registres 431, 432 et 433, respectivement reçoivent le signal CLK, c'est-à-dire le signal d'horloge système.

Le fonctionnement du registre tampon de sortie 42 de la figure 3 est le suivant.

Supposons qu'à un instant déterminé, le signal shw_sifini_gnt passe de
30 0 à 1. Au coup d'horloge suivant, la valeur 1 est stockée dans le registre d'activation 433. En outre, les valeurs courantes des N signaux d'entrée sont

stockées dans le registre d'entrée 432. Les N premières entrées 434a du multiplexeur de sortie 434 sont sélectionnées par le signal délivré par la sortie de données 433b du registre 433. Par conséquent les valeurs courantes des N signaux d'entrée sont délivrées par les N sorties 434c du multiplexeur 434 en tant que les N signaux de sortie, et ce quelles que soient les valeurs antérieures des signaux de sortie qui étaient stockées dans le registre 432. Au coup d'horloge suivant, les nouvelles valeurs des N signaux de sortie sont stockées dans le registre de verrouillage 432.

Supposons maintenant que le signal `shw_sifini_gnt` repasse à 0. Au coup d'horloge suivant, la valeur 0 est stockée dans le registre d'activation 433. Les secondes entrées 434b du multiplexeur 434 sont donc sélectionnées. Par conséquent, les valeurs des N signaux de sortie, qui ont précédemment été stockées dans le registre de verrouillage 432, sont délivrées par les N sorties du multiplexeur 434 en tant que les N signaux de sortie, et ce quelles que soient les (éventuellement nouvelles) valeurs des N signaux d'entrée. Dit autrement les valeurs des N signaux de sortie sont verrouillées par le registre de verrouillage 432.

L'inconvénient précité du mode de réalisation de la figure 9 n'existe pas avec le mode de réalisation de la figure 10. En effet, le signal `shw_sifini_gnt` entre sur l'entrée de données de la bascule 433 uniquement. Par conséquent, ce mode de réalisation est préférable pour les valeurs élevées de N (typiquement pour N supérieur à 10).

Certes le mode de réalisation de la figure 10 nécessite $2N+1$ bascules alors que celui de la figure 9 n'en comprend que N. Lorsque N est grand, ceci revient à doubler le nombre de bascules, ce qui a des conséquences sur la surface de silicium occupée par le SIF.

De plus, le mode de réalisation de la figure 10 ne respecte pas la règle de conception présentée plus haut en regard de la figure 6, puisque les signaux de sortie sont délivrés par un multiplexeur de sortie. Ceci introduit de la latence en sortie. Néanmoins, cette latence est acceptable eu égard au gain de temps en entrée du registre 42.

D'une manière générale, l'insertion de plusieurs couches de registres dans les SIFs ajoute de la latence dans les transactions. Toutefois, le temps d'occupation du bus Super Hyway par les SIFs qui y sont connectés, ce temps étant géré par l'arbitre 12a du bus, est réduit à son minimum. Il s'ensuit que, globalement, le fonctionnement du système est accéléré.

En pratique, la mise en œuvre de l'invention a permis une synthèse du bus Super Hyway à 266 MHz. C'est une fréquence à laquelle, sans la mise en œuvre de l'invention, le fonctionnement du système aurait été impossible.

REVENDICATIONS

1. Système électronique comprenant un bus système et des modules fonctionnels qui peuvent échanger des transactions à travers ledit bus système, caractérisé en ce qu'il comprend au moins un module de resynchronisation disposé entre un module fonctionnel associé et ledit bus système, ledit module de resynchronisation comprenant des moyens de chevauchement (351,352) pour faire chevaucher les transactions destinées à et/ou provenant dudit module fonctionnel associé, et en ce que lesdits moyens de chevauchement comprennent un premier circuit tampon (63) et au moins un second circuit tampon (64) connectés en parallèle, adaptés chacun pour stocker des données de transaction d'une transaction déterminée.

2. Système selon la revendication 1, caractérisé en ce que les moyens de chevauchement comprennent en outre :

- un port d'entrée (353,354) pour recevoir une transaction d'entrée, ladite transaction d'entrée contenant des données de transaction déterminées ;

- des premiers moyens de gestion (61,71) pour stocker lesdites données de transaction dans ledit premier circuit tampon ou dans ledit second circuit tampon si ledit premier circuit tampon n'est pas vide, ou pour refuser ladite transaction d'entrée si ledit premier circuit tampon et ledit second circuit tampon ne sont pas vides ;

- des seconds moyens de gestion (62,72) pour requérir une transaction de sortie dès que lesdites données de transaction sont stockées, pour générer ladite transaction de sortie à partir desdites données de transaction stockées, et pour vider le circuit tampon contenant lesdites données de transaction dès que ladite seconde transaction est transmise ; et,

- un port de sortie (355,356) pour transmettre ladite transaction de sortie sous la commande desdits seconds moyens de gestion.

3. Système selon la revendication 2, caractérisé en ce que les transactions étant de taille variable, le premier circuit tampon et/ou le second circuit tampon comprennent chacun des registres tampons (63a-63e;64a-63e) connectés en parallèle en nombre au moins égal au nombre maximum d'unités de données de protocole que peut contenir une transaction, chacun de ces registres étant adapté pour stocker les données de transaction d'une unité de données de protocole respective de la transaction.

4. Système selon la revendication 3, caractérisé en ce que le premier circuit tampon et/ou le second circuit tampon comprennent en outre un registre tampon qui est adapté pour stocker des données de transaction communes à toutes les unités de données de protocole de la transaction.

5. Système selon l'une quelconque des revendications 2 à 4, caractérisé en ce que les premiers moyens de gestion comprennent une première machine d'état (61) et un démultiplexeur (71), commandé par ladite première machine d'état.

6. Système selon l'une quelconque des revendications 2 à 5, caractérisé en ce que les seconds moyens de gestion comprennent une seconde machine d'état (62) et un multiplexeur (72) commandé par ladite seconde machine d'état.

7. Système selon l'une quelconque des revendications 1 à 6, caractérisé en ce qu'il comprend en outre un registre tampon de sortie (43) disposé en amont du port de sortie, et qui comprend :

- un premier registre à bascules (431) comprenant un nombre déterminé N de bascules ayant chacune une entrée de données, une sortie de données et une entrée d'activation ;

- un deuxième registre à bascules (432) comprenant N bascules, ayant chacune une entrée de données, une sortie de données et une entrée d'activation ;
- un troisième registre à bascule (433) comprenant une bascule ayant
5 une entrée de données, une sortie de données et une entrée d'activation ;
- un multiplexeur de sortie (434) ayant N premières entrées (434a), N secondes entrées (434b), N sorties (434c) et une entrée de sélection (434d) ;
et dans lequel :
 - les entrées de données (431a) des N bascules du premier registre
10 reçoivent respectivement N signaux d'entrée ;
 - les sorties de données (431b) des N bascules du premier registre sont respectivement reliées aux N premières entrées du multiplexeur de sortie ;
 - les N sorties du multiplexeur de sortie délivrent N signaux de sortie
15 respectifs ;
 - les entrées de données (432a) des N bascules du deuxième registre sont respectivement reliées aux N sorties du multiplexeur de sortie ;
 - les N sorties de données (432b) des N bascules du deuxième registre sont respectivement reliées aux N secondes entrées du multiplexeur de sortie ;
 - la sortie de données (433b) du troisième registre est reliée à l'entrée
20 de sélection du multiplexeur de sortie (434) ;
 - l'entrée de données (433a) du troisième registre reçoit un signal d'activation (shw_sifini_gnt) ; et,
 - les entrées d'activation (431c, 432c, 433c) des bascules des premier,
25 deuxième et troisième registres reçoivent un même signal d'horloge (CLK).

8. Module de resynchronisation destiné à être utilisé dans un système selon l'une quelconque des revendications 1 à 7, caractérisé en ce qu'il comprend des moyens de chevauchement (351,352) pour faire chevaucher les

transactions destinées à et/ou provenant dudit module fonctionnel associé, et en ce que lesdits moyens de chevauchement comprennent un premier circuit tampon (63) et au moins un second circuit tampon (64) connectés en parallèle, et adaptés chacun pour stocker des données de transaction d'une transaction
5 déterminée

9. Module de resynchronisation selon la revendication 8, caractérisé en ce que les moyens de chevauchement comprennent :

- un port d'entrée (353,354) pour recevoir une transaction d'entrée,
10 ladite transaction d'entrée contenant des données de transaction déterminées ;

- des premiers moyens de gestion (61,71) pour stocker lesdites données de transaction dans ledit premier circuit tampon ou dans ledit second circuit tampon si ledit premier circuit tampon n'est pas vide, ou pour refuser ladite transaction d'entrée si ledit premier circuit tampon et ledit second circuit
15 tampon ne sont pas vides,

- des seconds moyens de gestion (62,72) pour requérir une transaction de sortie dès que lesdites données de transaction sont stockées, pour générer ladite transaction de sortie à partir desdites données de transaction stockées, et pour vider le circuit tampon contenant lesdites données de transaction dès
20 que ladite seconde transaction est transmise ;

- un port de sortie (355,356) pour transmettre ladite transaction de sortie sous la commande desdits seconds moyens de gestion.

10. Module de resynchronisation selon la revendication 9, caractérisé
25 en ce que, les transactions étant de taille variable, le premier circuit tampon et/ou le second circuit tampon comprennent chacun des registres tampons (63a-63e;64a-63e) connectés en parallèle en nombre au moins égal au nombre maximum d'unités de données de protocole que peut contenir une transaction, chacun de ces registres tampon étant adapté pour stocker les données de
30 transaction d'une unité de données de protocole respective de la transaction.

11. Module de resynchronisation selon la revendication 10, caractérisé en ce que le premier circuit tampon et/ou le second circuit tampon comprennent en outre un registre tampon supplémentaire qui est adapté pour stocker des données de transaction communes à toutes les unités de données de protocole de la transaction.

12. Module de resynchronisation selon l'une quelconque des revendications 9 à 11, caractérisé en ce que les premiers moyens de gestion comprennent une première machine d'état (61) et un démultiplexeur (71), commandé par ladite première machine d'état.

13. Module de resynchronisation selon l'une quelconque des revendications 9 à 12, caractérisé en ce que les seconds moyens de gestion comprennent une seconde machine d'état (62) et un multiplexeur (72) commandé par ladite seconde machine d'état.

14. Module de resynchronisation selon l'une quelconque des revendications 9 à 13, caractérisé en ce qu'il comprend en outre un registre tampon de sortie (43) disposé en amont du port de sortie, et qui comprend :

- un premier registre à bascules (431) comprenant un nombre déterminé N de bascules ayant chacune une entrée de données, une sortie de données et une entrée d'activation ;

- un deuxième registre à bascules (432) comprenant N bascules, ayant chacune une entrée de données, une sortie de données et une entrée d'activation ;

- un troisième registre à bascule (433) comprenant une bascule ayant une entrée de données, une sortie de données et une entrée d'activation ;

- un multiplexeur de sortie (434) ayant N premières entrées (434a), N secondes entrées (434b), N sorties (434c) et une entrée de sélection (434d) ;

et dans lequel :

- les entrées de données (431a) des N bascules du premier registre reçoivent respectivement N signaux d'entrée ;
- les sorties de données (431b) des N bascules du premier registre sont respectivement reliées aux N premières entrées du multiplexeur de
5 sortie ;
- les N sorties du multiplexeur de sortie délivrent N signaux de sortie respectifs ;
- les entrées de données (432a) des N bascules du deuxième registre sont respectivement reliées aux N sorties du multiplexeur de sortie ;
- 10 - les N sorties de données (432b) des N bascules du deuxième registre sont respectivement reliées aux N secondes entrées du multiplexeur de sortie ;
- la sortie de données (433b) du troisième registre est reliée à l'entrée de sélection du multiplexeur de sortie (434) ;
- l'entrée de données (433a) du troisième registre reçoit un signal
15 d'activation (shw_sifini_gnt) ; et,
- les entrées d'activation (431c, 432c, 433c) des bascules des premier, deuxième et troisième registres reçoivent un même signal d'horloge (CLK).

15. Procédé de gestion d'un système électronique comprenant un bus
20 système et des modules fonctionnels qui peuvent échanger des transactions de taille variable à travers ledit bus système, caractérisé en ce qu'il comprend les étapes consistant à :

- prévoir au moins un module de resynchronisation disposé entre un module fonctionnel associé et ledit bus système ;
- 25 - prévoir dans ledit module de resynchronisation des moyens de chevauchement (351,352) pour faire chevaucher les transactions destinées à et/ou provenant dudit module fonctionnel associé ; et,
- prévoir dans lesdits moyens de chevauchement un premier circuit tampon (63) et au moins un second circuit tampon (64) connectés en parallèle,

adaptés chacun pour stocker des données de transaction d'une transaction déterminée.

16. Procédé selon la revendication 15, caractérisé en ce qu'il
5 comprend les étapes consistant à :

- recevoir une transaction d'entrée, ladite transaction d'entrée contenant des données de transaction déterminées ;

- stocker lesdites données de transaction dans ledit premier circuit tampon (63) ou dans ledit second circuit tampon (64) si ledit premier circuit
10 tampon n'est pas vide, ou pour refuser ladite transaction d'entrée si ledit premier circuit tampon et ledit second circuit tampon ne sont pas vides,

- requérir une transaction de sortie dès que lesdites données de transaction sont stockées ;

- générer ladite transaction de sortie à partir desdites données de
15 transaction stockées ;

- vider le circuit tampon contenant lesdites données de transaction dès que ladite seconde transaction est transmise ;

- transmettre ladite transaction de sortie via un port de sortie (355,356) sous la commande desdits seconds moyens de gestion.

20

17. Procédé selon la revendication 16, caractérisé en ce que l'étape consistant à stocker les données de transaction dans ledit premier circuit tampon ou dans ledit second circuit tampon comprend les étapes consistant à :

- prévoir dans chacun des premier et second circuits tampon des registres tampons (63a-63e;64a-63e) connectés en parallèle, en nombre au
25 moins égal au nombre maximum d'unités de données de protocole que peut contenir une transaction ;

- stocker les données de transaction de chaque unité de données de protocole de la transaction dans l'un respectif des registres tampon du circuit
30 tampon.

18. Procédé selon la revendication 17, caractérisé en ce que l'étape consistant à stocker lesdites données de transaction dans ledit premier circuit tampon ou dans ledit second circuit tampon comprend les étapes consistant à :

5 - prévoir en outre dans chacun des premier et second circuits tampon un registre tampon supplémentaire (63a,64a) connecté en parallèle ;

- stocker dans ledit registre tampon des données de transaction communes à toutes les unités de données de protocole de la transaction.

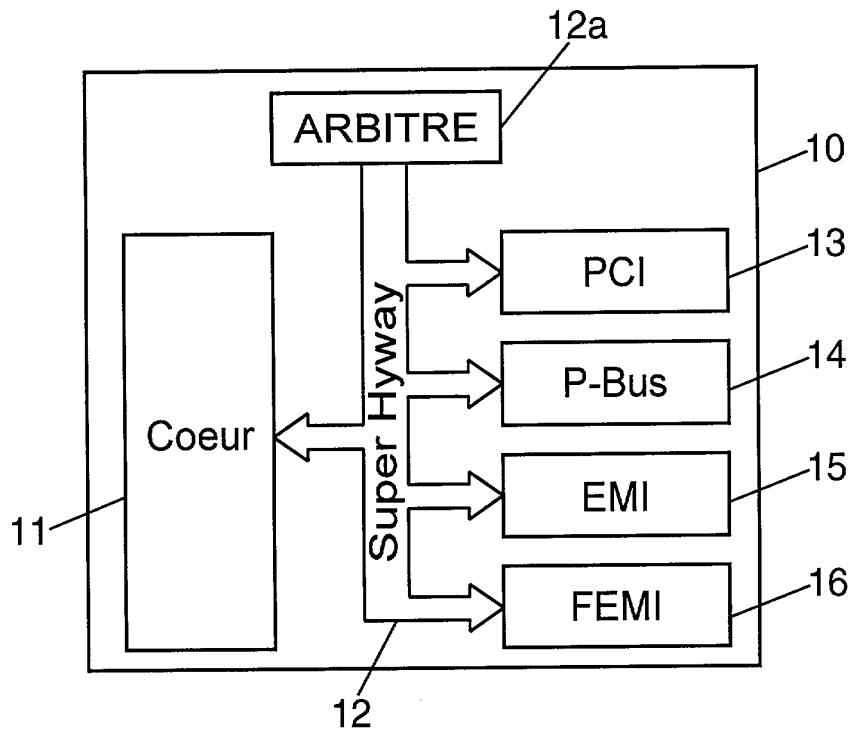


FIG. 1

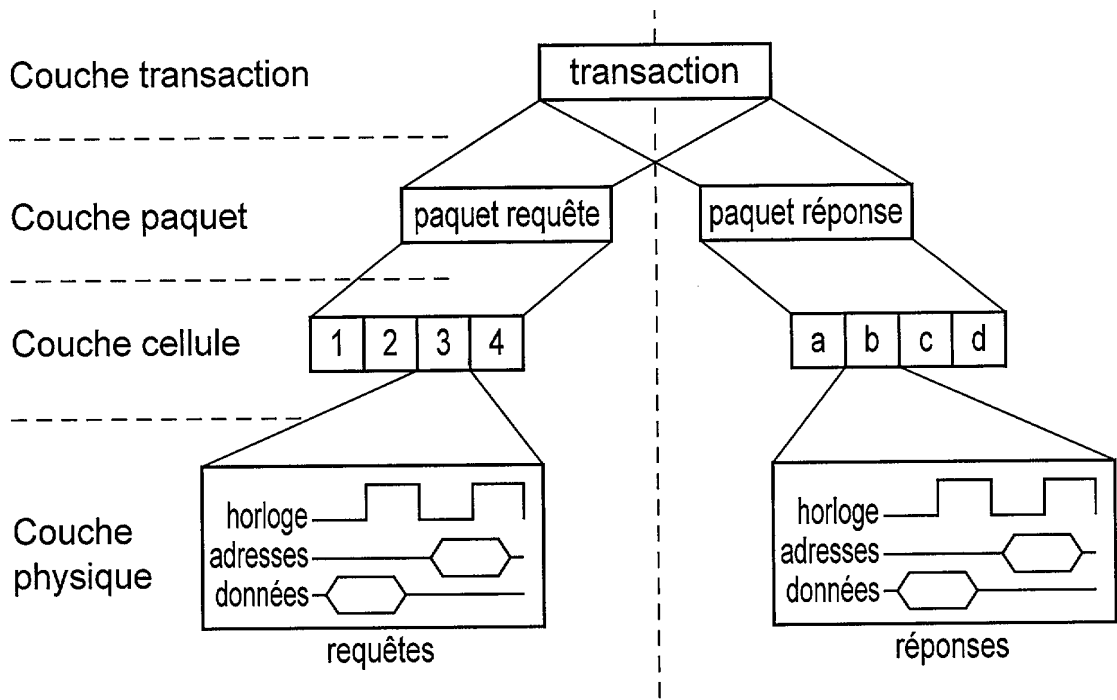


FIG. 2

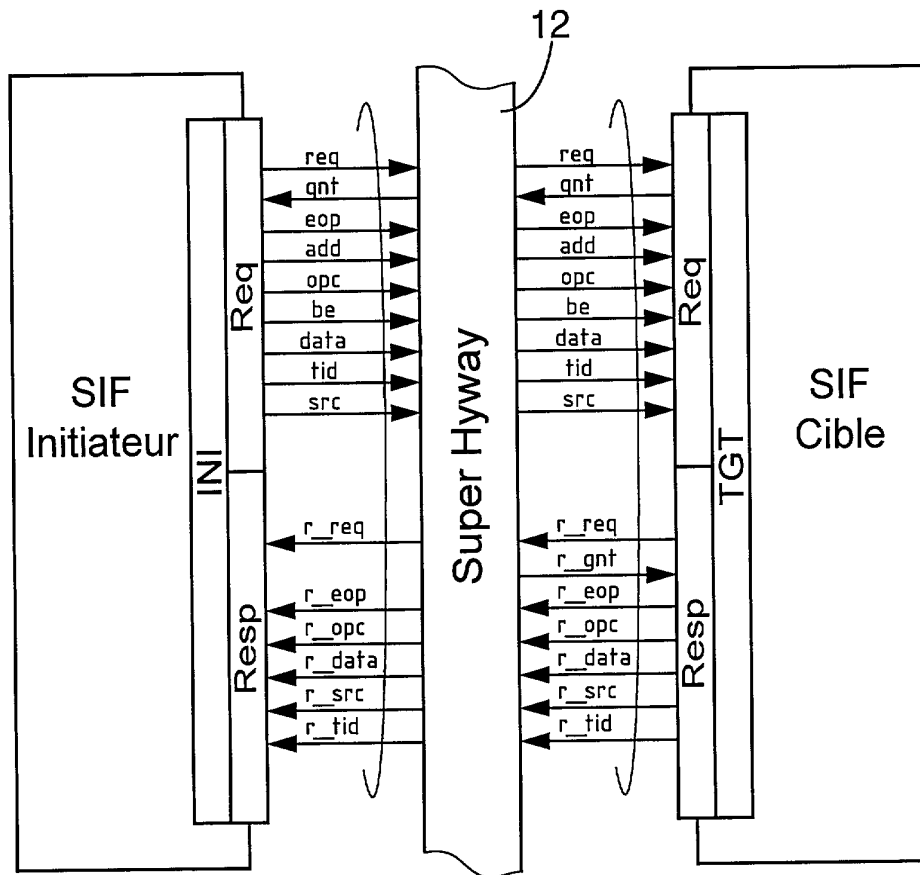


FIG. 3

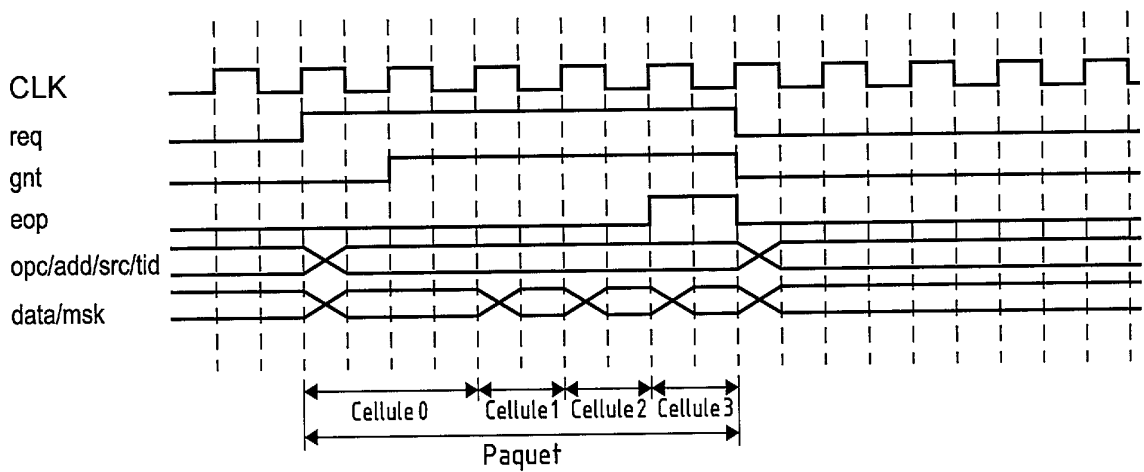


FIG. 4

3/4

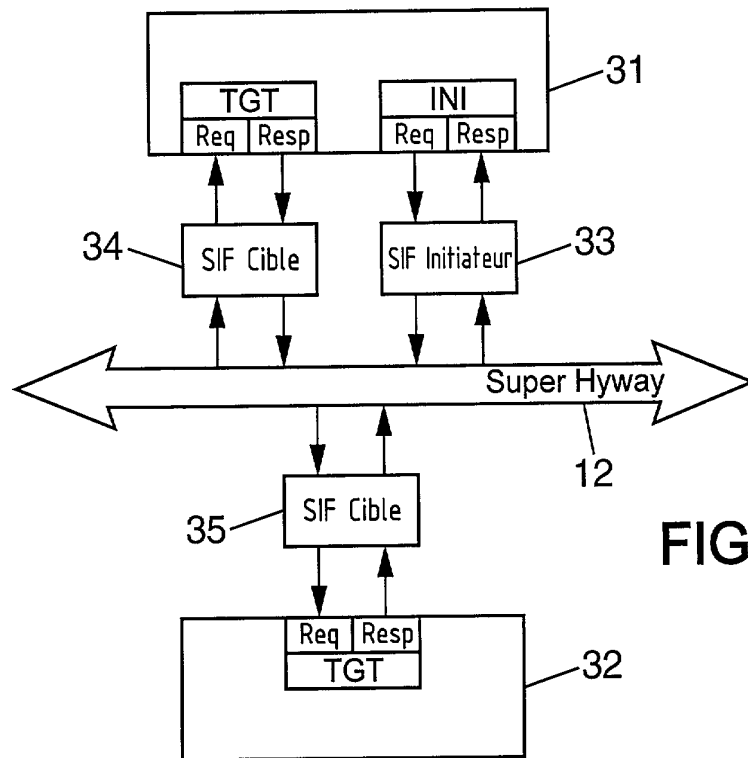


FIG. 5

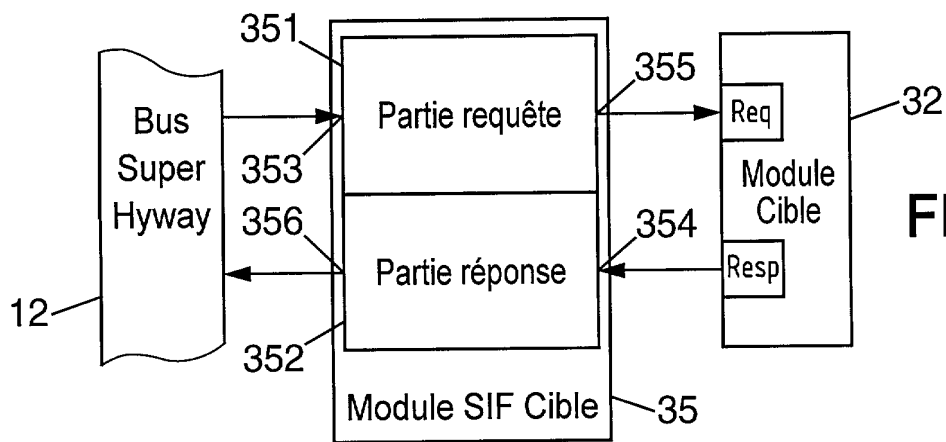


FIG. 6

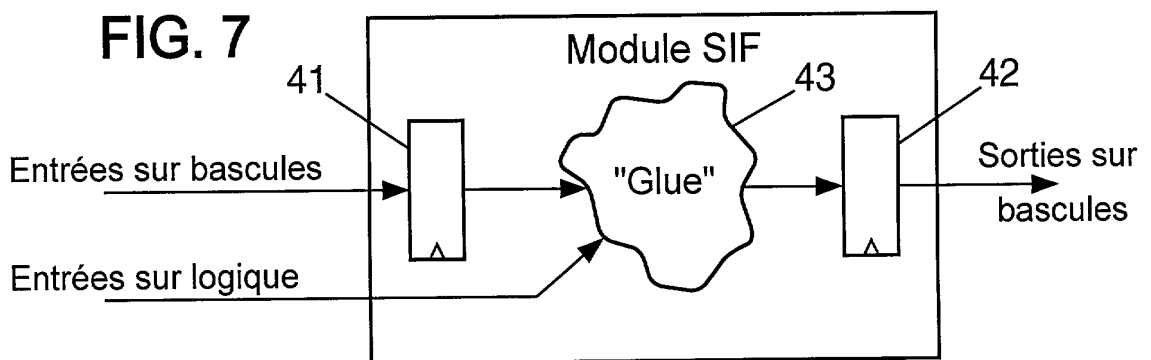


FIG. 7

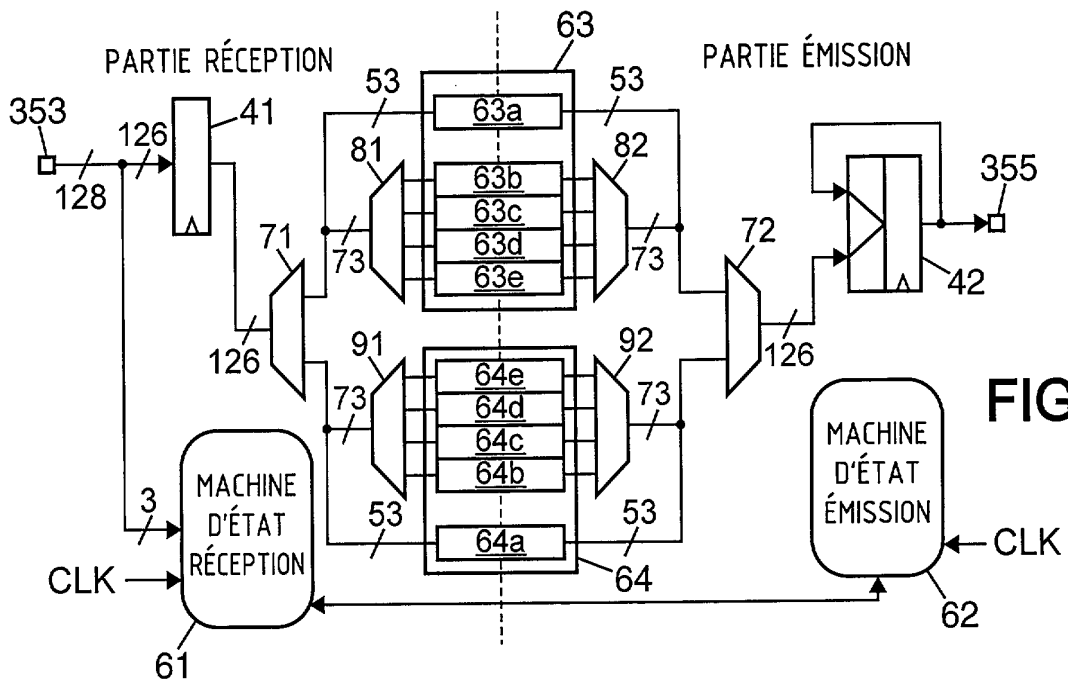


FIG. 8

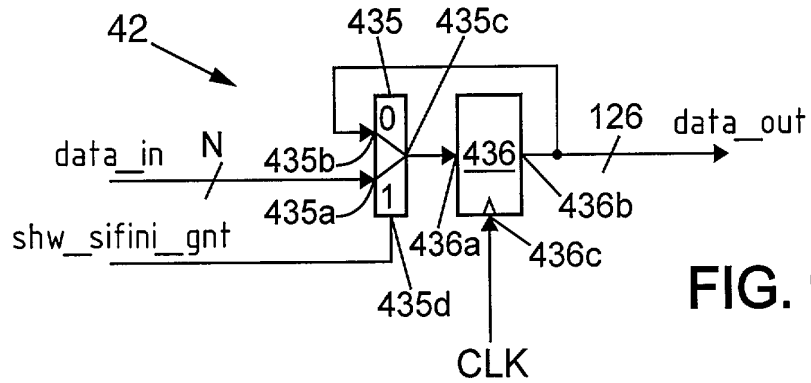


FIG. 9

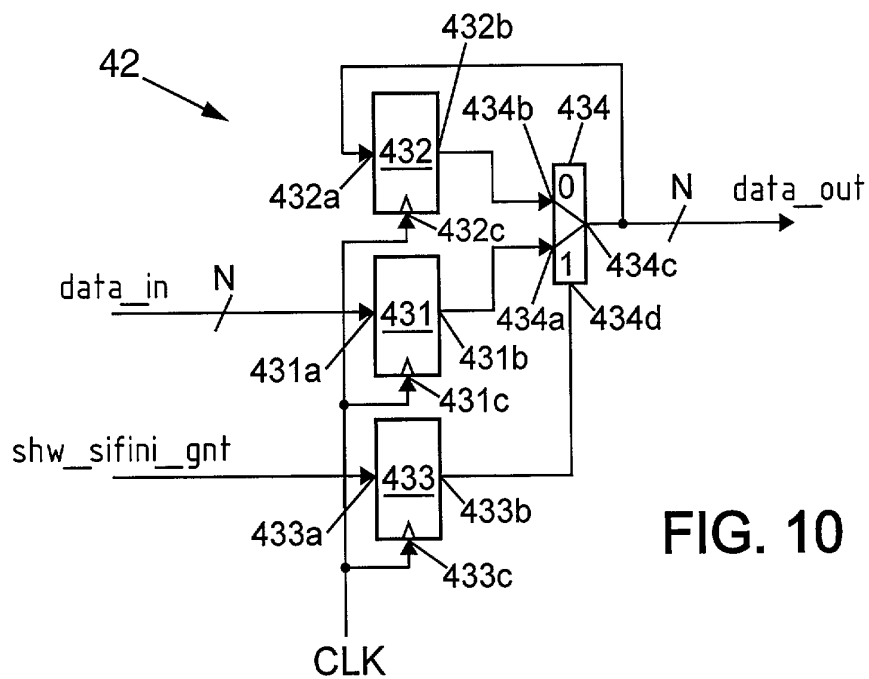


FIG. 10



**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**

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

N° d'enregistrement
national

FA 626959
FR 0213753

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 A	US 5 960 215 A (TANAKA KOICHI ET AL) 28 septembre 1999 (1999-09-28) * colonne 1, ligne 66 - colonne 2, ligne 33 * * colonne 4, ligne 29 - colonne 5, ligne 28 *	1-3, 15-18 4-14	G06F13/16
A	--- US 6 252 600 B1 (KOOB CHRISTOPHER EDWARD ET AL) 26 juin 2001 (2001-06-26) * colonne 3, ligne 34 - colonne 4, ligne 30 * * colonne 5, ligne 14 - colonne 6, ligne 18 * * abrégé *	1-18	
A	--- US 6 260 095 B1 (GOODRUM ALAN L) 10 juillet 2001 (2001-07-10) * colonne 3, ligne 32 - colonne 4, ligne 14 * * colonne 5, ligne 64 - colonne 8, ligne 36 * * abrégé; figures 2A,2B *	1-18	
			DOMAINES TECHNIQUES RECHERCHÉS (Int.CL.7)
			G06F
Date d'achèvement de la recherche		Examineur	
28 août 2003		Nguyen Xuan Hiep, C	
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 0213753 FA 626959**

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 **28-08-2003**
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 5960215 A	28-09-1999	AUCUN	
US 6252600 B1	26-06-2001	AUCUN	
US 6260095 B1	10-07-2001	US 5815677 A DE 69721209 D1 EP 0851362 A1 JP 10247172 A	29-09-1998 28-05-2003 01-07-1998 14-09-1998