

- **ENSEIRB** -



# LE BUS INDUSTRIEL PCI

**Patrice KADIONIK**  
[kadionik@enseirb.fr](mailto:kadionik@enseirb.fr)  
<http://www.enseirb.fr/~kadionik>

# TABLE DES MATIERES

<b>1. Le bus PCI</b> .....	<b>3</b>
<b>1.1. Historique</b> .....	<b>3</b>
<b>1.2. Présentation du PCI</b> .....	<b>3</b>
<b>1.3. Fonctionnement</b> .....	<b>5</b>
❖ Transfert de données.....	5
❖ Description des signaux.....	5
❖ Description des transferts.....	5
❖ Gestion de l'accès bus PCI.....	8
❖ Signaux de contrôle.....	8
❖ Gestion des interruptions.....	8
❖ Gestion des configurations.....	8
<b>2. Comparaison des bus PCI et VME</b> .....	<b>9</b>
<b>2.1. Origines des bus industriels</b> .....	<b>9</b>
❖ Les besoins.....	9
❖ Evolution des besoins de l'électronique.....	9
<b>2.2. Comparaison des bus</b> .....	<b>10</b>
❖ Interconnexions.....	10
❖ Technologie.....	10
❖ Fonctionnement.....	11
❖ Performances.....	12
❖ Fonctions intelligentes.....	13
<b>2.3. Les Normes</b> .....	<b>14</b>
<b>2.4. Evolutivité</b> .....	<b>15</b>
❖ Evolution du bus.....	15
❖ Nouveaux standards.....	15
<b>3. Conclusion</b> .....	<b>16</b>
<b>4. Bibliographie</b> .....	<b>17</b>

## **1. LE BUS PCI**

### **1.1. Historique**

Le bus PCI (Peripheral Component Interconnect) est une norme développée initialement par INTEL, avec une première révision (1.0) distribuée en 1992. Cette toute première définition présentait le PCI comme un bus local, proche des ressources, gravitant autour du processeur central. Après seulement un an, et donc en Avril 1993, la spécification du bus a été étendue de manière à inclure un environnement plus large, avec des ressources disposées sur des cartes enfichables dans une connectique normée. A ce niveau, le PCI a atteint un niveau de définition qui lui permet d'être exploité sur une carte mère avec processeur, capable de recevoir ces interfaces par cartes d'extension.

Enfin, en 1995, la révision 2.1 de cette norme, a incorporé des clarifications, ainsi que les spécifications nécessaires à la migration du bus vers la technologie de cadencement à 66 MHz.

Une révision 2.2 est actuellement finalisée, elle devrait inclure des définitions sur la gestion de l'énergie des alimentations.

### **1.2. Présentation du PCI**

Le bus PCI est un bus synchrone supportant un multiplexage des signaux d'adressages et de données. Il est avant tout prévu pour travailler avec des systèmes 32 bits, dans un fonctionnement en échange entre un maître et un esclave sous le contrôle d'un arbitre. L'interface PCI prévoit l'utilisation d'un minimum de signaux avec un protocole décorélé des protocoles de transferts des microprocesseurs existants. D'après la spécification du bus, un minimum de 49 signaux est nécessaire pour un module capable d'assurer des fonctions de maître, tandis que les autres signaux normalisés peuvent n'être que des options. La Figure 1 : Liste des signaux du bus PCI suivant la norme montre d'ailleurs les interconnexions entre le bus et un module.

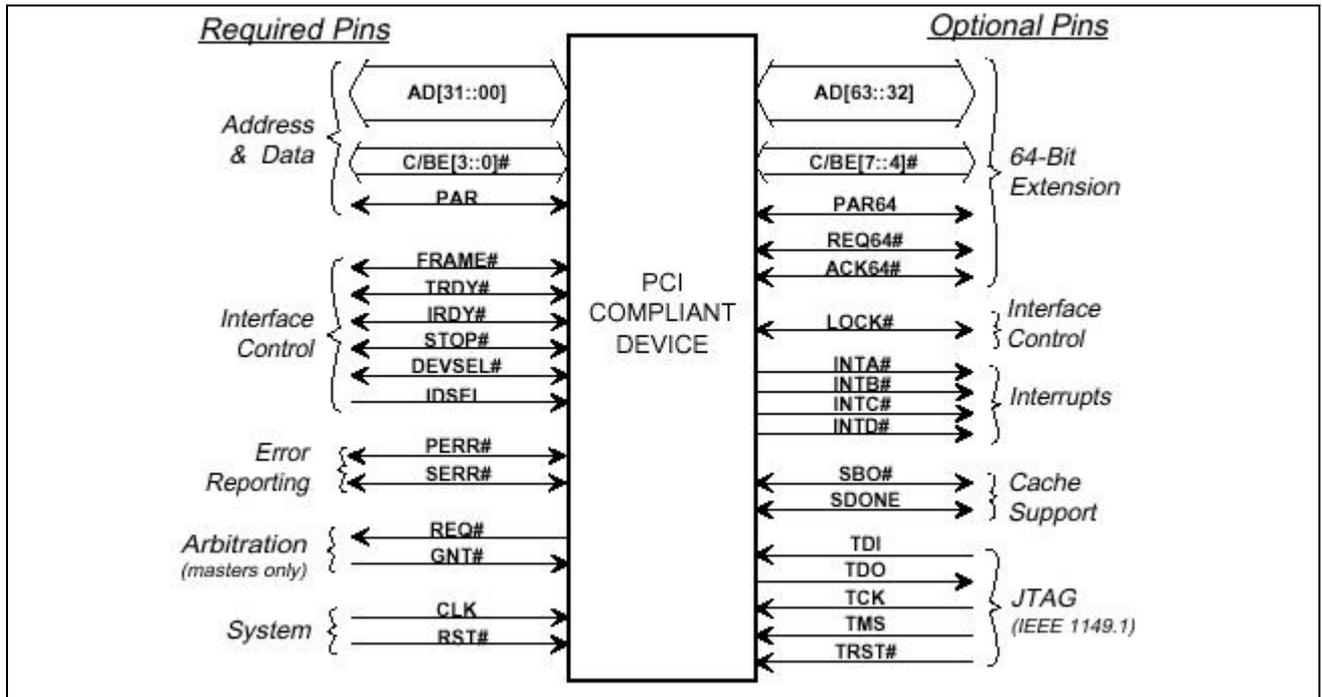


Figure 1 : Liste des signaux du bus PCI suivant la norme

De plus, la spécification du bus autorise l'interconnexion, sur le bus PCI, jusqu'à quatre cartes entre elles. Egalement, la norme prévoit l'emploi de passerelles (bridges) d'un bus PCI vers un autre bus PCI, ou bien un bus d'un autre type. Dans le cas de l'utilisation de bridges PCI/PCI, jusqu'à 256 passerelles sont possibles. Enfin, le bus comprend obligatoirement une unité de gestion d'accès au bus, tandis qu'il peut intégrer dans ce même arbitre un gestionnaire d'interruptions. La spécification 2.1 suggère une architecture classique d'utilisation du bus PCI, comme en Figure 2 : Schéma bloc d'une architecture PCI.

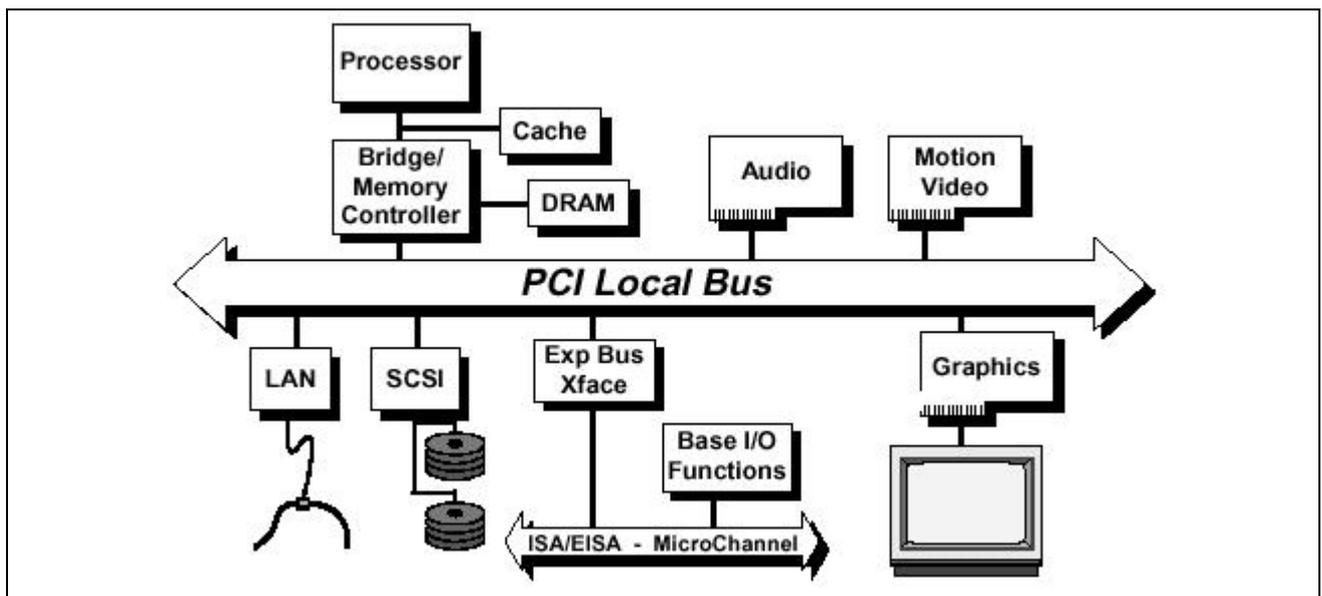


Figure 2 : Schéma bloc d'une architecture PCI

### 1.3. Fonctionnement

#### ❖ Transfert de données

Le principe de transfert de données sur le bus PCI exploite les échanges par paquets (burst), qui mettent en jeu la quasi-totalité des signaux de la configuration minimum.

#### ❖ Description des signaux

Pour le transfert de données, le PCI se compose donc :

- Du bus d'adresses et de données multiplexé, noté AD[31..00], qui permet donc d'exploiter 32 bits en adressage, comme en donnée. On obtient donc un espace adressable de 4 Go, sachant que la norme réclame un décodage complet sur l'ensemble des bits nécessaires par les esclaves. Ce bus prend la fonction de porteur d'adresses dans la phase dite d'adressage sur le bus, tandis qu'à l'issue de cette phase, il devient bus de données.
- Du bus de commandes et de validation d'octets, noté C/BE[3..0]\*, qui permet, durant la phase d'adressage, de transmettre la commande de bus qui va être réalisée, tandis qu'à cette issue, il devient bus de validation des octets circulant en groupes de 32 bits sur le bus de données.
- Du bus de contrôle qui regroupe les signaux gérés par le maître :  
FRAME\* : Départ et validation de transaction.  
IRDY\* : Etat du maître de l'échange.

En réponse l'esclave fournit :

- TRDY\* : Etat de l'esclave dans échange.
- DEVSEL\* : Indique la sélection d'un esclave s'étant reconnu après l'adressage.
- STOP\* : Demande d'arrêt de l'échange.

#### ❖ Description des transferts

Pour exécuter un transfert sur le bus PCI, le maître concerné doit d'abord s'assurer de la disponibilité du bus pour son propre usage. Cette phase étant réalisée avec succès, le maître engage l'accès bus en positionnant l'adresse et la commande, en même temps que le FRAME\*. Après un front d'horloge, le maître positionne alors l'IRDY\*, qui signale qu'il est paré par le transfert, mais surtout termine la phase d'adressage. Les esclaves potentiellement concernés sont alors amenés à répondre au maître consécutivement aux décodage d'adresses qu'ils ont chacun réalisés. Le signal DEVSEL\* positionné indique alors qu'un esclave s'est reconnu pour l'échange, et le positionnement de TRDY\* indique qu'il est prêt pour le transfert. Le maître positionne alors les signaux de validation des octets C/BE[3..0]\*, et dans le cas de l'écriture, émet la donnée. L'esclave positionne, ou lit, la donnée et informe qu'il est paré pour la suite en conservant TRDY\*. Le maître termine enfin l'échange, ou l'esclave réclame la fin du transfert par STOP\*, comme l'illustre la Figure 3 : Opération de base en écriture fournie par la spécification.

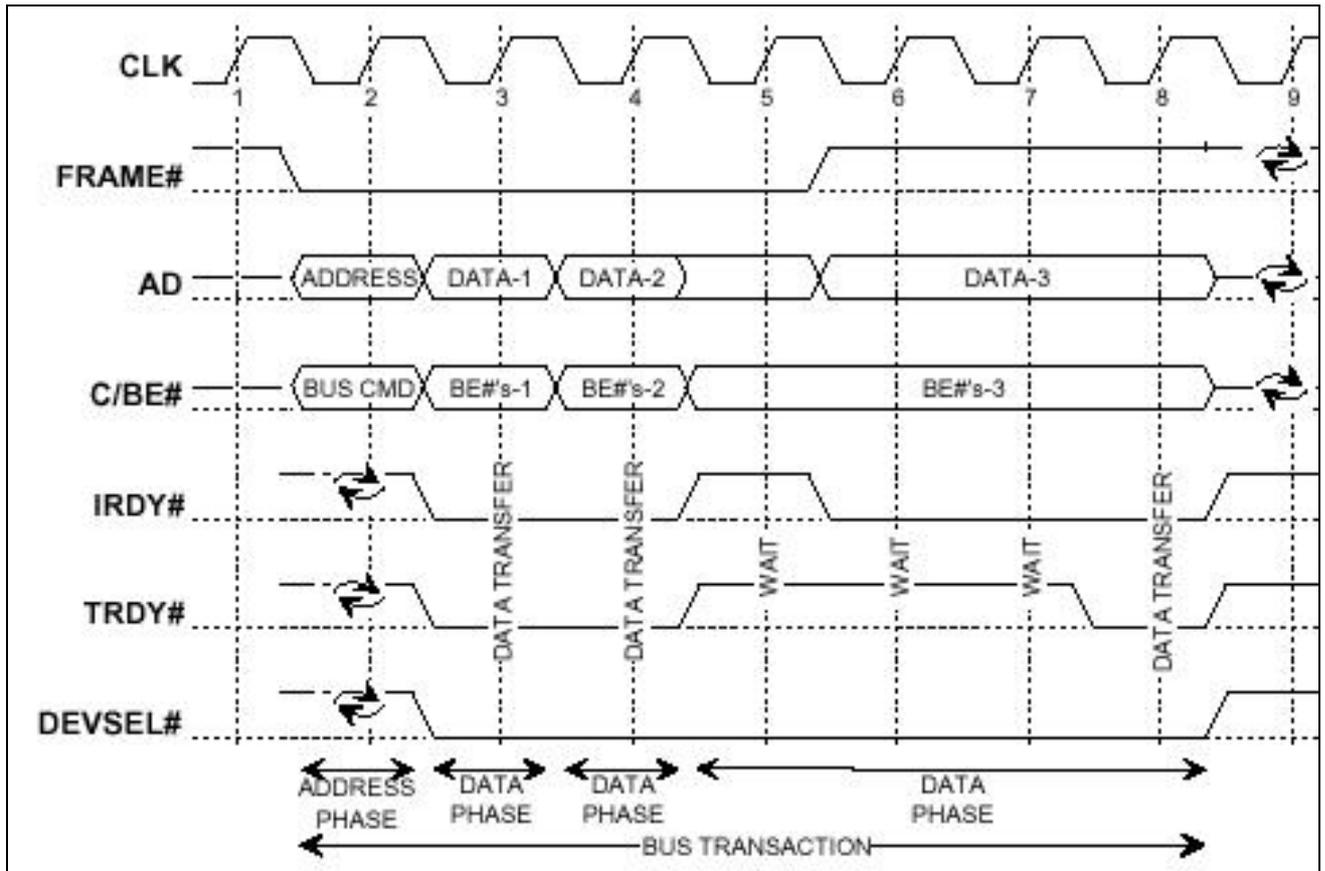


Figure 3 : Opération de base en écriture

Le bus PCI spécifie de nombreuses façons de mettre fin aux transferts de données. Ainsi, la transaction peut naturellement se terminer à l'initiative du maître, mais aussi bien suivant celle de l'esclave, avec dans ces cas, échange, ou non, de données.

Une autre fonctionnalité du bus PCI est de permettre des opérations différées. Ainsi, une transaction peut être initialisée par un maître, ce qui, du point de vue esclave, engage un traitement local. Seulement, l'esclave mémorise la commande reçue et demande la déconnexion. Il lui appartient ensuite de solder la tâche qui lui a été assignée. Enfin, le maître vient recontacter l'esclave et reçoit alors l'information d'acquittement de la tâche.

L'avantage de cette fonctionnalité est de permettre la libération du bus pour en permettre l'utilisation par d'autres maîtres. Ces transactions différées sont de cinq types :

- PWR est une écriture en mémoire postée. Une transaction qui se termine sur le bus d'origine avant d'aboutir sur le bus de destination.
- DRR est une requête de lecture différée. Une transaction qui doit se terminer sur le bus de destination avant d'être reçue sur le bus d'origine.
- DWR est une requête d'écriture différée. Une transaction qui doit se terminer sur le bus de destination avant d'être reçue sur le bus d'origine.
- DRC est un acquittement de lecture différée. Une transaction qui s'est terminée sur le bus de destination et qui remonte maintenant vers le bus d'origine.
- DWC est un acquittement d'écriture différée. Une transaction qui s'est terminée sur le bus de destination et dont l'acquittement remonte maintenant vers le bus d'origine.

Par ailleurs, le bus PCI autorise la réservation de ressource par l'intermédiaire d'un signal LOCK\*. Ainsi, un maître peut se réserver l'exploitation d'un esclave pour se garantir les résultats d'une tâche, vis à vis de l'intervention d'un autre maître. Après avoir réussi sa réservation, le maître est alors libre d'accéder à sa ressource et de prolonger, ou non, son exclusivité. La Figure 4 : Opération d'accès exclusif sur PCI, illustre un maître ayant réalisé une réservation et accédant à sa ressource. En fin de cycle, il lui appartient de libérer, ou non, cet état.

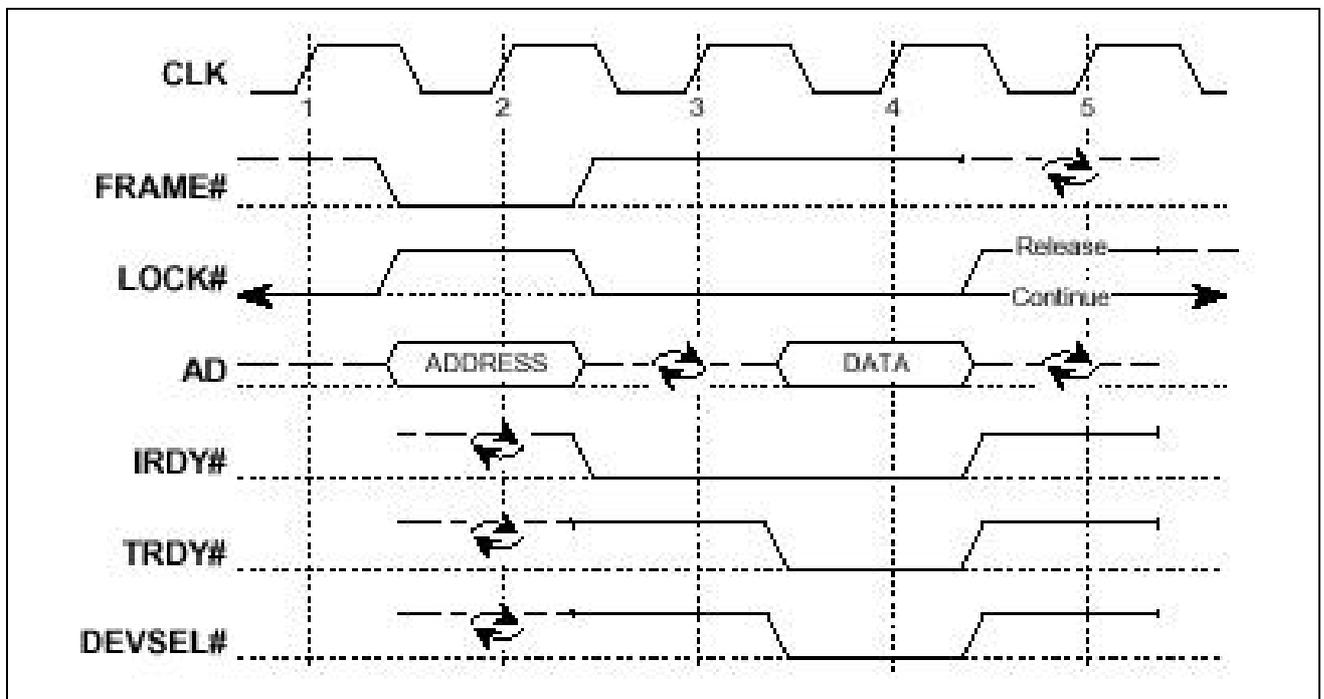


Figure 4 : Opération d'accès exclusif sur PCI

### ❖ Gestion de l'accès bus PCI

Un arbitre de bus est notamment dédié à la fonction de gestion des demandes d'accès par les différents maîtres potentiels. Le protocole utilisé est basé sur une requête émise par le maître qui doit alors recevoir une autorisation renvoyée par l'arbitre. Chaque maître dispose de sa propre liaison de requête-acquittement vers l'arbitre. La Figure 5 : Gestion de l'accès bus sur PCI illustre cette connectique. L'arbitrage est à la discrétion du concepteur en matière de niveau de priorité dans les requêtes.

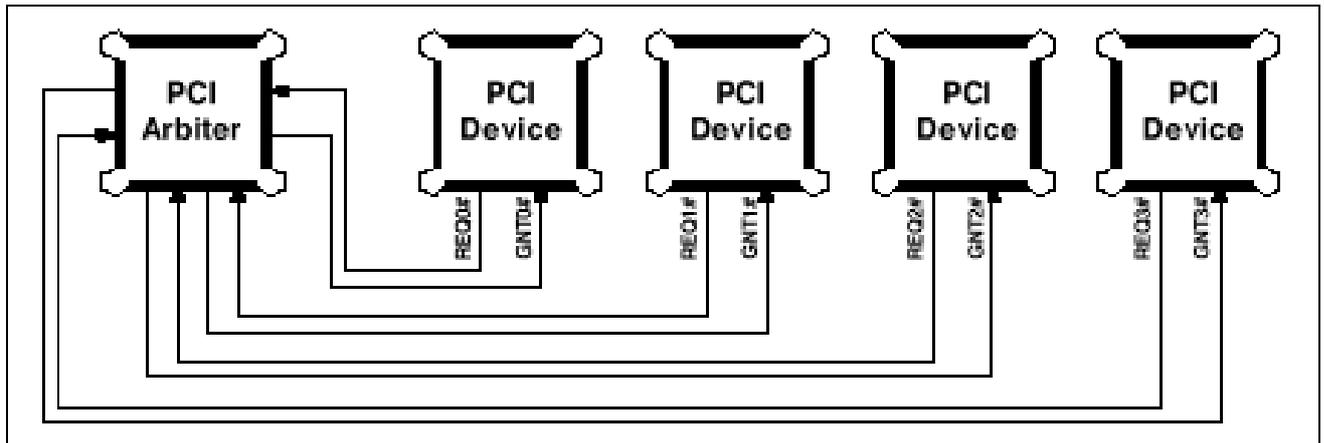


Figure 5 : Gestion de l'accès bus sur PCI

### ❖ Signaux de contrôle

Le bus PCI étant synchrone, il intègre un signal d'horloge global CLK. Une commande d'initialisation par le signal RST\* est également distribuée à tous les agents du bus. Par ailleurs, les agents du PCI gèrent la parité par un signal PAR calculé sur les bus AD[31..0] et C/BE[3..0]\*. Une détection d'erreur pouvant entraîner le positionnement d'un signal d'erreur PERR\*. Enfin, une erreur système, considérée comme grave, est signalée par SERR\* vers, en général, l'arbitre du bus.

### ❖ Gestion des interruptions

Les interruptions sur le PCI sont optionnelles par rapport à la configuration minimale nécessaire. Lorsqu'elles sont utilisées, elles sont au nombre de quatre et fonctionnent de manière asynchrone. Cependant, seule l'interruption INTA\* est distribuée à tous les agents et les autres lignes d'interruptions ne peuvent être utilisées que si la ressource connectée est multifonction. Ceci signifie que les interruptions sont au moins partagées par un «OU» câblé. Le contrôleur d'interruption du bus est laissé à la discrétion du concepteur en ce qui concerne son câblage aux lignes d'interruptions.

Lorsqu'une interruption est prise en compte par le gestionnaire, l'acquittement et le transfert du vecteur correspondant se font classiquement par le bus, mais en utilisant le bus de commande C/BE[3..0]\* positionné spécifiquement.

### ❖ Gestion des configurations

La définition du PCI fournit une description complète d'un espace de configuration que doit assurer tout composant interfaçable sur le bus. Ainsi, il est possible par logiciel de gérer

intégralement les configurations de fonctionnement des divers agents sur le bus. A cet effet, les composants offrent un espace de 256 octets inclus dans des registres. Les contenus de ces différents registres conditionnent en grande partie les propriétés de fonctionnement des composants concernés. Ils incluent également un codage descriptif des fonctionnalités du module associé.

Cet ensemble de registres est implanté de manière à pouvoir être modifié au niveau local du module concerné par un processeur local s'il est disponible, ou encore par une mémoire externe. Cependant, c'est surtout au niveau de l'accessibilité par le bus PCI que cet espace est intéressant. Ainsi, un maître sur le bus peut venir accéder, en utilisant des cycles spéciaux, aux registres de configuration d'un esclave. Ce principe permet donc une gestion dynamique des différents agents sur le bus.

## 2. COMPARAISON DES BUS PCI ET VME

### 2.1. Origines des bus industriels

#### ❖ Les besoins

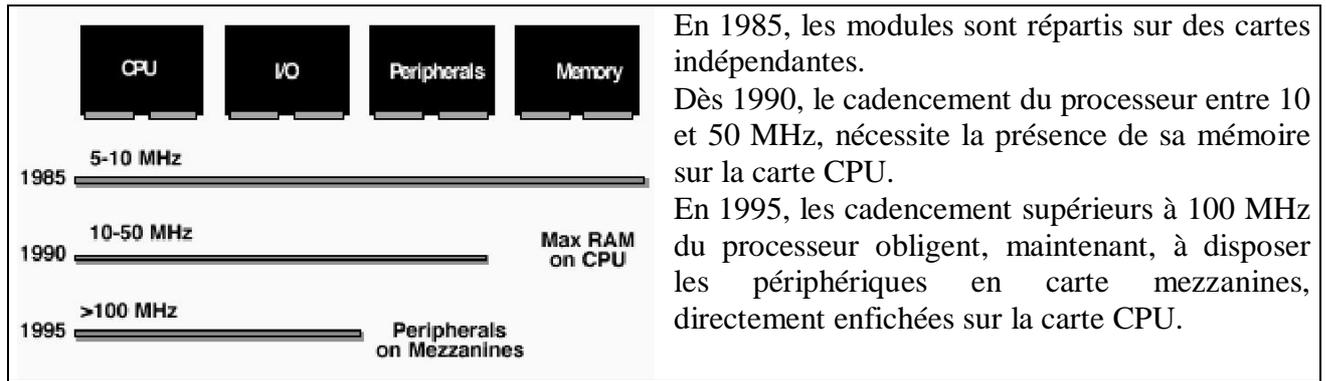
Le domaine industriel réclame une architecture robuste qui lui permette de lier entre eux différents modules électroniques de manière à constituer un système informatique complet (unités de traitements, de stockage de données ou de contrôle de périphériques). Par ailleurs, dans les années 80, les développements de cartes de traitements utilisaient particulièrement les processeurs de la famille MOTOROLA 680xx. Ainsi, le bus VME s'impose comme une normalisation des protocoles de transferts entre différentes unités qui exploitent des cœurs de calcul avec des processeurs presque directement interfaçables avec le bus.

Le bus VME offre également une définition claire et solide des caractéristiques exploitées dans un environnement précis. Ainsi, les châssis VME supportent une connectique rigoureuse et éprouvée, en adéquation avec une exploitation dans le milieu industriel.

D'un autre côté, le PCI est né comme un bus local, soit donc, un bus de bas niveau pour un processeur. Son intérêt est surtout d'apporter un potentiel de débit d'informations beaucoup plus élevé que ceux disponibles sur les bus existants. Par ailleurs, le PCI est décorélé du type de processeur principal intervenant sur le bus, sachant que des composants dédiés d'interfaçage permettent à ce type de ressource de se connecter au bus. En conséquence, ce bus est apparu comme parfaitement adapté à l'utilisation dans le domaine du PC (Personal Computer).

#### ❖ Evolution des besoins de l'électronique

Alors que les capacités d'intégration vont croissantes, les surfaces occupées par les différents composants, ou les modules, s'interfaçant dans un système se réduisent. De plus, les performances, en termes de cadencement s'élèvent, aussi, la diminution des distances entre les composants devient une nécessité. La Figure 6 : Evolution des cadences des systèmes, illustre ceci.



En 1985, les modules sont répartis sur des cartes indépendantes.  
 Dès 1990, le cadencement du processeur entre 10 et 50 MHz, nécessite la présence de sa mémoire sur la carte CPU.  
 En 1995, les cadencement supérieurs à 100 MHz du processeur obligent, maintenant, à disposer les périphériques en carte mezzanines, directement enfichées sur la carte CPU.

**Figure 6 : Evolution des cadences des systèmes**

Cette évolution amène naturellement à l'utilisation de bus de types locaux, comme le PCI, du moins au niveau de l'environnement processeur.

## 2.2. Comparaison des bus

### ❖ Interconnexions

Le bus VME est spécifié pour être distribué sur un châssis par une carte fond de panier qui supporte alors deux configurations avec 9 et 20 emplacements de cartes filles. Le format de ces cartes peut être du type simple ou double Europe. Les cartes s'enfichent sur le fond de panier par deux connecteurs de 96 points, bien adaptés à l'enfichage et au déenfichage des modules. Par ailleurs, le bus VME est adapté en impédance aux deux terminaisons du routage de fond de panier. Il peut s'agir d'une adaptation passive par pont résistif et diode Schottky pour les signaux rapides, ou bien, active avec adjonction d'un montage d'amplificateur opérationnel (chez Schroff).

Pour le bus PCI, la spécification donne une orientation très claire vers une conception pour PC. Le bus est donc distribué par une carte mère qui fait office de fond de panier, bien qu'elle ne soit pas exclusivement dédiée au support du bus. Il est alors possible de connecter 4 cartes filles en plus du contrôleur de bus. Par contre, la connectique en elle-même, bien que destinée à garantir certaines qualités électriques, n'est absolument pas fiable pour un environnement industriel.

Par ailleurs, le bus suit des recommandations quant au routage et à la disposition des composants d'interfaces, de même que des signaux doivent être référencés par des résistances de tirages. Mais, il n'y a pas d'adaptation des signaux sur les terminaisons de bus.

### ❖ Technologie

Le bus VME est d'une technologie TTL avec une alimentation principale de 5V distribuée pour 3A maximum par carte. Le bus inclut également une alimentation de secours de 5V permettant de sauvegarder des contextes en mémoire vive lors de coupures d'alimentation. Enfin un couple +12V et -12V est fourni pour des tensions auxiliaires.

Les signaux transitant sur le bus sont de cinq types qui sont du 3 états standard ou courant élevé, totem pôle standard ou courant élevé, et circuit collecteur ouvert.

Le bus PCI en de technologie CMOS 5V avec une distribution auxiliaire de +12V et -12V. Le bus est cependant prévu et orienté pour un fonctionnement en technologie 3V3, qui devient une alimentation quasi-standard dans les conceptions de carte de traitement à base de processeurs.

Les signaux transitant sur le bus sont de trois types qui sont du 3 états standard, totem pôle standard, et circuit collecteur ouvert.

## ❖ Fonctionnement

### Gestion du bus

La gestion du bus VME est réalisée par un arbitre qui dispose d'une situation géographique sur le châssis, il est en position la plus à gauche pour une vue de face. De plus, il existe une hiérarchie, également géographique, dans la disposition des différents modules sur le bus, et donc le niveau de priorité potentiel d'une carte va décroissant à mesure que l'on s'éloigne de l'arbitre. Par ailleurs, les lignes de requêtes vis à vis de l'arbitre sont, elles aussi, hiérarchisées, ou bien, à priorité tournante. Avec ce principe de gestion, le bus VME est donc capable de gérer véritablement un ensemble multiprocesseur ayant chacun une priorité hiérarchique à plat ou bien en cascade.

Pour un système PCI, l'arbitrage utilisé, bien que reposant sur le même principe que le VME, ne fait pas de distinction dans les différentes requêtes qui peuvent lui arriver. Bien qu'il soit possible de créer, pour le concepteur, un arbitrage plus élaboré, sachant que dans ce cas, il s'agit d'une étude à réaliser à part entière. Dans le cadre de la norme, donc, l'arbitrage entre les différents maîtres potentiels du bus se fait en autorisant indifféremment les demandeurs à occuper le bus. Dans une telle configuration, un montage multiprocesseur n'est guère envisageable, et il vaut mieux s'attacher à un ensemble maître principal vers différents esclaves.

On peut également noter que lorsque le bus PCI est occupé, il ne peut être employé pour une tâche secondaire. Pour le bus VME, ses sous-ensembles VMX et VMS, qui disposent de leurs propres arbitrage, sont capables de fonctionner alors que le bus DTB est en transfert.

### Gestion des interruptions

Le VME intègre un gestionnaire centralisé, ou plusieurs répartis, pour gérer les interruptions sur le bus. Il s'agit également d'une fonctionnalité hiérarchisée, qui exploite le chaînage des signaux pour établir une hiérarchie secondaire géographique des cartes, vis à vis de l'apparition et de l'acquiescement des IT (InTerruption). Les différents modules présents dans un châssis peuvent donc bénéficier d'une voie d'interruption que l'arbitre peut facilement discerner parmi les autres, de manière à accorder plus ou moins de priorité dans le traitement à engager pour cette IT. Cette architecture permet plus particulièrement la gestion des tâches en temps réel.

Les interruptions sur le PCI, sont d'abord optionnelles, d'après la norme, et au nombre de quatre. Cependant, seule une est partagée par tous les agents du bus, tandis que les autres ne sont utilisables que si l'agent connecté supporte plusieurs fonctions. Par conséquent, lorsqu'une IT apparaît sur le bus, il n'y a pas de moyen, à priori, de discerner quel en est l'agent émetteur. Naturellement, des adaptations sont possibles au niveau du gestionnaire d'IT, mais encore une fois, elles sont à l'initiative du concepteur, et restent limitées. Par conséquent, on retrouve bien un principe de gestion du bus qui tend à n'inclure qu'un seul

processeur central pour le bus complet. L'utilisation de cette gestion pour des traitements temps réels n'est pas adaptée.

### Transferts de données

Les transferts sur le VME se font entre un maître et un esclave en utilisant classiquement un cycle de lecture-modification-écriture qui est tout à fait adapté aux traitements des I/O. Par ailleurs, pour des besoins importants en quantité de données, il est possible d'exploiter un cycle DMA, qui permet de transférer un flot de données en continu. Enfin, le système est capable de gérer les besoins en réservation de ressources que peuvent exprimer les différents processeurs de l'ensemble. Le VME offre donc des possibilités assez larges d'exploitation pour le transfert de données. Toutefois, il est soumis aux divers ralentissements que peuvent engendrer des modules avec de longs temps d'accès. Et, de fait, lorsque qu'un cycle d'accès est engagé, il doit se terminer afin que le bus puisse de nouveau être utilisé, à moins que ne survienne un signal d'erreur de bus, ou que l'arbitre demande expressément la libération du bus.

On peut également regretter que ce bus n'assure pas de lignes de contrôles dédiées aux signaux de données, comme, par exemple, un signal de parité.

Pour sa part, le PCI est conçu avant tout pour fonctionner dans un mode rafale de données. Ainsi, lorsque la phase d'adressage est arrivée à terme, le transfert de données est engagé de manière à faire transiter une donnée au coup d'horloge. Pour une exploitation en mode lecture-modification-écriture, le PCI verra circuler deux accès encadrant l'opération de modification réalisée par le calculateur, ce qui n'est clairement pas optimal. De plus, le principe de réservation de ressource est très succinct, dans le sens où la norme spécifie qu'une réservation de ressource doit s'appliquer sur un minimum de 16 octets de données consécutifs. Dans le respect strict de la norme, cet aspect peut donc sembler plutôt restreint, sachant de plus, que le bus ne supporte qu'une seule réservation de ressource à la fois.

On a finalement bien un bus local, dont la vocation première est d'offrir une possibilité de volume de transfert important.

On retiendra aussi tout particulièrement la différence fondamentale de fonctionnement de ces deux bus, avec une exploitation asynchrone des signaux pour le bus VME, tandis que toute la gestion du bus PCI est liée à une horloge système, et donc, il travaille de manière synchrone.

### ❖ Performances

Comme cela a déjà été exposé, le principe de transfert des données par le VME n'est pas lié à un cadencement global, par contre, la norme spécifie les séquences et les durées des phases de transfert qu'il faut respecter. Sur cette base, il est possible d'établir de calculer un taux de transfert maximum théorique qui peut circuler sur ce bus. Ainsi, en utilisant le DTB en 32 bits, le bus VME offre un débit théorique de 40 Mo/s.

Pour sa part, le PCI cadencé à 33 MHz, peut assurer un taux de transfert de 132 Mo/s, lorsqu'il est exploité en mode 32 bits.

Dans ces deux cas, il convient de modérer ces taux de transfert en pics. A cet effet, notamment, INTEL a réalisé des mesures de taux de transfert en déterminant un ratio normalisé qui valorise l'efficacité du bus. Ainsi, le bus est considéré comme non-efficace lorsqu'il sert à supporter la phase d'adressage, ou qu'il est en état d'attente. A partir de cette première mesure, il convient alors de considérer la taille des rafales d'octets circulant sur le

bus, comme en Figure 7 : Efficacité PCI suivant les tailles de rafale. En appliquant alors cette valeur d'efficacité au taux maximum de transfert, on revient vers des valeurs d'environ 80 à 90 Mo/s.

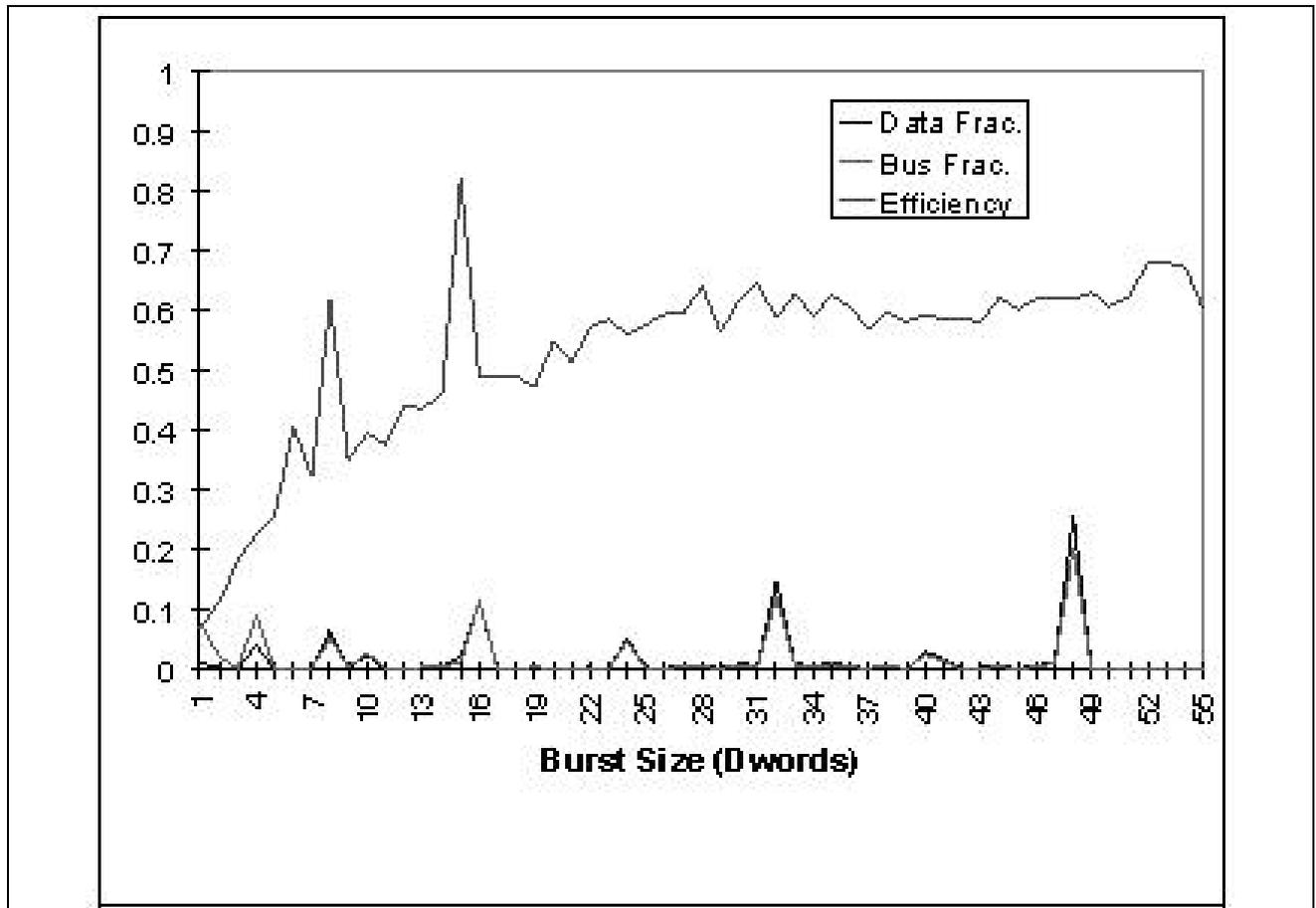


Figure 7 : Efficacité PCI suivant les tailles de rafale

❖ Fonctions intelligentes

Le bus PCI, particulièrement, intègre des fonctions de types intelligentes comme la configuration «plug and play». L'intérêt de cette définition dans la norme est de permettre de gérer l'insertion d'un nouveau module sur le bus, alors que celui-ci est en pleine exploitation. En fait, les arbitres et gestionnaire de bus doivent être capables, non seulement de détecter cette nouvelle interface, mais surtout de pouvoir l'intégrer à part entière dans le système actuel, en modifiant des configurations au besoin. Du point de vue utilisateur, on obtient alors des capacités de mise en œuvre ramenées à quelques minutes, avec la résolution par les ressources déjà en place des diverses possibilités de conflits.

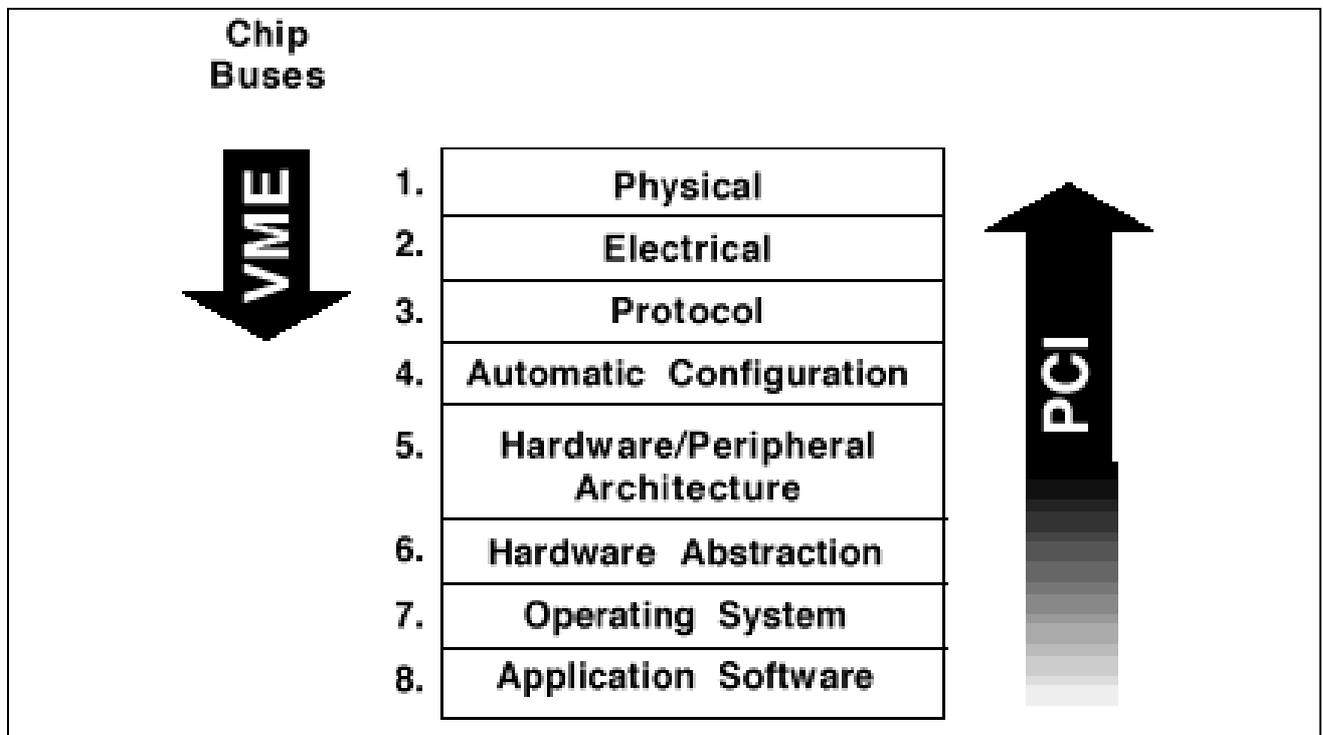
On notera que cette puissante fonctionnalité n'est pas présentée sur le bus VME. Cependant, et comme présenté en □ Nouveaux standards, le bus VXI, descendant du VME, exploite également les fonctionnalités «plug and play». Et, de fait, c'est le bus PCI qui a repris cette spécification pour sa propre norme, sachant que VXI a été spécifié avant l'apparition du PCI. Comme également présenté plus loin, le bus PXI, dont le fonctionnement de base exploite le PCI, va lui aussi exploiter la norme du bus VXI, mais pour sa part, afin de répondre au marché de l'instrumentation.

Les fonctionnalités intelligentes présentées, en ce domaine, par la génération PCI n'ont donc rien de véritablement novateur, la génération VME ayant déjà produit cette étude.

### 2.3. Les Normes

Dans la définition d'un bus, on peut extraire différentes couches système que la norme peut, ou non spécifier. Le bus VME est dans ce sens un standard ouvert qui s'intéresse particulièrement au niveau matériel des ensembles. C'est un aspect qui le rend flexible vis à vis des exploitations que l'on peut en faire. Par contre, les logiciels à mettre en œuvre deviennent alors particulièrement dépendants des applications, et ne sont plus portables. Pour une nouvelle ressource implantée dans le châssis, il faut consacrer un budget à l'adaptation du système d'exploitation.

De son côté, le PCI se présente plus comme un concept global, qui définit l'ensemble des couches interagissant entre elles, comme le montre la Figure 8 : Modèle de couches d'un système ouvert typique.



**Figure 8 : Modèle de couches d'un système ouvert typique**

Dans sa norme, le bus VME définit clairement les 3 premières couches en partant des composants et du bus physique, tandis que le PCI offre une spécification en se plaçant depuis le logiciel d'application et en redescendant vers la couche physique.

## 2.4. Evolutivité

### ❖ Evolution du bus

Alors qu'il a été créé avec la génération de processeurs 680xx, le bus VME souffre aujourd'hui de l'arrivée en fin de vie de cette famille de composant. L'augmentation des cadencement des processeurs, issus de nouvelles familles, est aussi un frein à l'utilisation, comme telle, de cette architecture qui semble très dédiée. Aussi le bus VME évolue pour offrir des tailles de bus et des taux de transfert plus élevés. Cependant, la norme VME reste fixe et ce sont de nouvelles normes qui apparaissent, toutes restant compatibles de la génération existante. Ainsi on trouve les bus VME64, VME64x, VME320, VXI.

La spécification PCI, quant à elle, s'inscrit déjà dans une schématique d'évolution, comme le présente la Figure 9 : Possibilités d'évolutions du PCI.

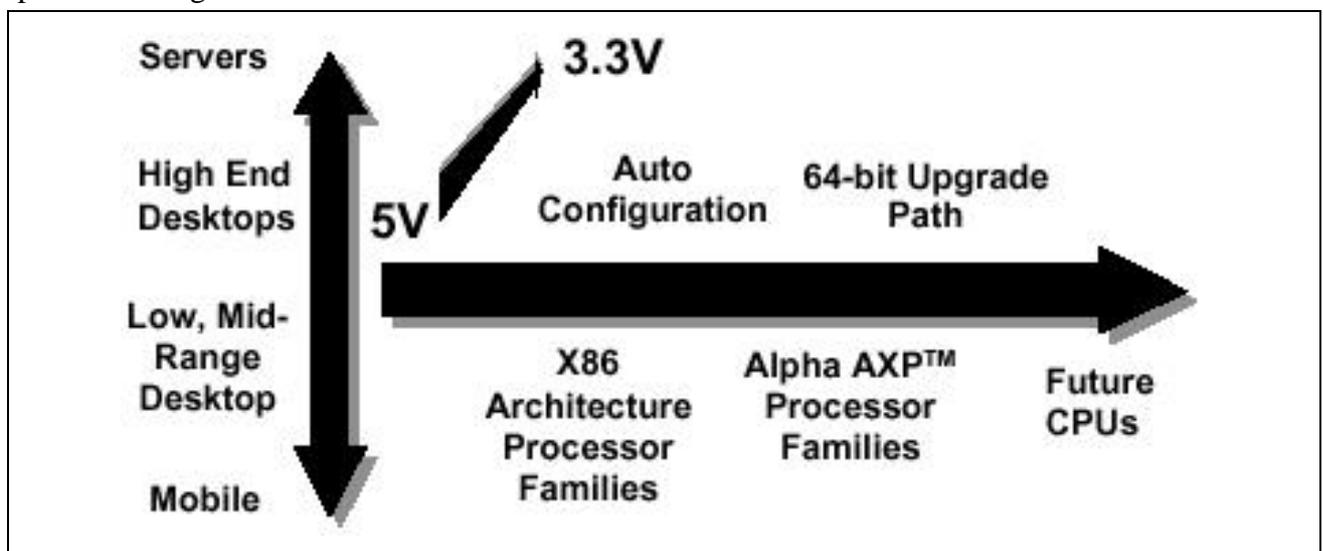


Figure 9 : Possibilités d'évolutions du PCI

### ❖ Nouveaux standards

Pour le bus VME, les voies d'évolutions passent par de nouveaux standards qui reprennent le standard VME existant.

- Le VME64 propose le passage du bus en mode 64 bits pour s'adapter aux besoins des nouvelles générations de processeurs, tout en offrant un taux de transfert doublé à 80 Mo/s en maximum théorique.
- Le VME64x propose également une exploitation du bus en mode 64 bits, mais avec des échantillonnages des signaux de commande sur les deux fronts de commutations pour des débits de 160 Mo/s. Ceci oblige notamment à l'emploi d'une technologie avancée au niveau des pilotes de signaux avec de l'ETL (Enhanced Transceiver Logic). Ce qui le rend donc incompatible du bus d'origine, d'autant que le VME64x nécessite des connecteurs de 160 points.
- Le VXI (VME eXtension for Instrumentation) a lui été développé dans le but d'être utilisé dans le domaine de l'instrumentation. Il utilise notamment l'architecture 64 bits avec en plus des signaux de synchronisations. Par ailleurs, ce bus regroupe les fonctionnalités d'autoconfiguration des cartes avec le «plug and play».

- Le VME320 propose lui des transmissions de données en synchrones, et toujours avec la technologie 64 bits, ce qui l'amène à des débits de 320 Mo/s.

De son côté, le PCI souffre de son incompatibilité avec le milieu industriel. Dans ce sens, de nouveaux standards ont été développés :

- Le CompactPCI est un bus qui reprend les principes électriques de fonctionnement de PCI, tout en y intégrant toute la rigueur du système châssis du VME. De plus, il donne alors la possibilité de connecter plus de cartes entre elles, et même d'utiliser des passerelles PCI/PCI. Il a donc pour vocation de combler le vide d'exploitation laissé par PCI dans le domaine de l'utilisation industrielle.

- Le PXI (PCI eXtension for Instrumentation) est également un bus adapté selon la même réflexion que le CompactPCI, mais il vise plus le domaine de l'instrumentation. Sa norme est plus sévère que celle du CompactPCI, sachant que des capacités logicielles sont aussi spécifiées. Ces spécifications logicielles stipulent que chaque module PXI doit supporter les OS (Operating System) Windows 95 et Windows NT.

### **3. CONCLUSION**

Après examen de ces deux bus, ainsi que des différents standards qui leur font suite, on peut déjà exprimer quelques conclusions.

On constate, en particulier, toute la différence qu'il y a entre un bus pensé industriel comme le VME et bus local comme le PCI. Ils font chacun parti d'un monde qui n'est pas véritablement le même.

Le bus VME est le leader incontestable en termes de cartes. Les standards VME/VXI sont bien établis dans le monde industriel, et normalisés au sein d'une puissante organisation qu'est le VITA. Malgré plus de quinze années d'existence, et des problèmes de spécifications assez lâches au début, les constructeurs ont réussi à réaliser des environnements solides. Les produits disponibles, associés au volume des ventes, représentent toujours 50 % des parts de marchés.

De plus, les compétences acquises et les niveaux de performances atteints tendent à faire hésiter le passage vers de nouveaux standards.

Dans le domaine de l'instrumentation, le VXI, fait lui l'unanimité, tant par ses performances que par ses qualités, notamment avec le «plug and play».

D'un autre côté, il y a le monde de l'informatique dont la puissance provient du développement phénoménal du PC. Mais c'est aussi une toute autre philosophie, les standards sont en permanente évolution, les besoins en puissances sont sans cesse croissants. Le volume de production des cartes est extrêmement important, ce qui en assure la baisse des coûts. Par contre la pérennité des développements et les risques d'obsolescence des matériels ne sont pas supportables pour l'industrie.

Le bus PCI est un bus très rapide mais peu adapté au monde industriel, l'apparition des nouveaux standards comme le CompactPCI peut effectivement apporter une solution PCI pour l'industrie, bien qu'il soit encore trop tôt pour en juger.

Par conséquent, autant les besoins en vitesse et débit sur des bus gravitant autour d'un processeur vont faire appel au PCI, autant les nécessités en rigueur et en performances des

systèmes de gestion de bus font référence au VME. Clairement, il n'est pas possible de dire qu'un de ces deux bus prévaut sur l'autre du fait de ces qualités. Chacun, en fait, comble un certain éventail de besoins. Cette réflexion est d'ailleurs parfaitement illustrée par les constructeurs de carte qui proposent des structures mixtes complémentaires, incluant un bus local PCI et une passerelle PCI/VME pour accéder aux autres modules.

On retiendra également, que les deux familles de standards vont vers des niveaux de définition qui assurent, de plus en plus, des disponibilités en cartes assemblables dans des systèmes très personnalisés. Ainsi, l'utilisateur est libre de constituer son châssis, dans une application d'acquisition, en utilisant des modules cartes aptes à très simplement s'interfacer entre elles. On rejoint ainsi la tendance actuelle, qui vise à apporter à l'exploitant un niveau d'abstraction, toujours croissant, par rapport aux couches de définitions matérielles. La mise en œuvre doit aller en se simplifiant, et l'interface utilisateur système devient plus ergonomique et apparaît facilité.

## **4. BIBLIOGRAPHIE**

- ◆ **PCI Local Bus specification Revision 2.1S**  
PCI Special Interest Group 06/01/1995.
- ◆ **IEEE Standard for VMEbus**  
ANSI/IEEE std 1014-1987.
- ◆ **RASSP architecture guide Revision C : AVY-L-S-00081-101-C**  
Lockheed Sanders, Inc ; Hughes Aircraft ; Motorola ; ISX Corporation.
- ◆ **VME64 Extensions for Physics and Other Applications : VITA 23-1998**  
VMEbus International Physics Association.
- ◆ **www.pcisig.com**  
Site officiel de la spécification PCI.
- ◆ **www.vita.com**  
VITA : VME International Trade Association : Site officiel de la spécification VME.
- ◆ **www.mcg.mot.com**  
Motorola computer group : Applications VME et CPCI.
- ◆ **www.forcecomputers.com**  
Etudes et développements sur bus VME64 et CPCI.
- ◆ **www.gespac.com**  
Articles sur les bus VME, PCI et CPCI.