

Le Bus PCI.

Alliaume Erwan	alliau_e
Blès François-Jérôme	bles_f
Van Der Linden Jérôme	van-de_j

11 mars 2002

Table des matières

1	Historique.	2
2	Présentation du PCI.	3
2.1	Généralités.	3
2.2	Connectique.	5
3	Fonctionnement du PCI.	7
3.1	Description des signaux.	7
3.2	Transfert des données.	10
3.3	Gestion de l'accès bus.	12
3.4	Gestion des interruptions.	14
4	Évolution du PCI.	15
4.1	Évolution et Évolutivité.	15
4.1.1	Évolution.	15
4.1.2	les variantes.	16
4.2	Normes.	18
4.3	Performances.	20
5	Conclusion.	21

Chapitre 1

Historique.

Anciennement, aucune des applications développées ne posait de problèmes au bus ISA, qui fonctionne à la fréquence de 8.33Mhz, et peut transférer 2 octets par cycle de bus soit une bande passante de 16.7 Mo/s.

Le bus EISA, evolution du précédent, est capable d'effectuer des transferts de 4 octets par cycle, ce qui correspond à une bande passante de 33,3 Mo/s.

Les applications récentes demandant une bande passante très élevée, Intel a défini un nouveau bus, le "Peripheral Component Interconnect bus" (PCI) en 1990. Afin de promouvoir cette nouvelle norme, il décida de mettre dans le domaine public les brevets la concernant. Intel a confié le développement de ce bus à un consortium industriel, le "PCI Special Interest Group". Désormais, la plupart des machines de bureau possèdent un bus PCI, ce qui en fait le standard incontournable du marché.

A l'origine, le bus PCI transférait 32 bits par cycle de bus, et fonctionnait à 33 Mhz, offrant ainsi une bande passante de 133 Mo/s. Il était défini comme un bus local, proche des ressources et du processeur central.

En 1993 naît l'évolution 2.0 qui prévoit des connecteurs d'extensions. Puis en 1995, la version 2.1 apparaît, incorporant les spécifications nécessaires à la migration vers un cadencement à 66 mhz. C'est actuellement la version 2.2 qui est utilisée, incluant un mécanisme de réduction d'alimentation électrique. Le bus PCI fonctionne maintenant à 66 MHz, transfère des données de 64 bits avec une bande passante de 528 mo/s.

Chapitre 2

Présentation du PCI.

2.1 Généralités.

Il existe deux types de bus :

- les bus synchrones : Sur ce type de bus, les opérations sont synchronisés par un signal d'horloge à fréquence fixe. Toute opération dure un nombre entier de période d'horloge appelée cycle de bus.
- les bus asynchrones : Il n'est pas piloté par une horloge mais synchronisé par des signaux d'aquitements.

Le bus PCI est un bus synchrone supportant un multiplexage des signaux d'adressages et des données. Pour économiser le nombre de lignes de bus, les concepteurs ont choisi de mettre en place un bus multiplexé où les adresses et les données partagent les mêmes lignes. Lors du premier cycle d'un transfert, c'est l'adresse qui est placée sur le bus, les données y prenant place ultérieurement.

Il est avant tout prévu pour travailler avec des systèmes 32bits, dans un fonctionnement en échange entre un maître (appelé *initiateur*) et un esclave (appelé *cible*) sous le contrôle d'un arbitre.

Maître : Le maître est le composant du système qui demande et contrôle un transfert. Cela peut être le processeur, mais aussi un contrôleur de disque, une carte d'acquisition, ...

Esclave : L'esclave, lui, fournit ou lit des données à la demande du maître. Cela peut être une carte graphique, un contrôleur de port parallèle,...

Arbitre : Lorsque plusieurs composants réclament le bus, il y a conflit. L'arbitre de bus, analyse alors les demandes de bus et décide qui en sera l'allocataire.

L'interface PCI prévoit l'utilisation d'un minimum de signaux avec un protocole indépendant 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 (voir figure 1). Plus de détails sont fournis au chapitre 3.1.

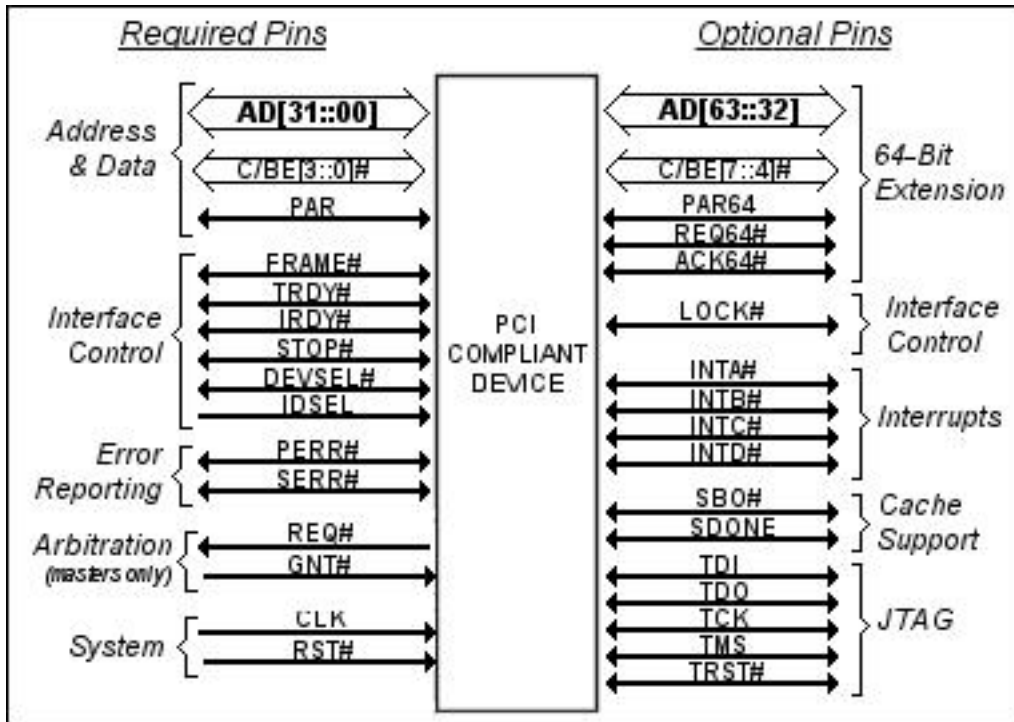


Figure 1 : Liste des signaux du bus PCI

2.2 Connectique.

Bien que la bande passante du bus PCI soit importante, ce bus ne convient pas aux échanges entre processeur et mémoire, ni pour raccorder les anciennes cartes additionnelles conformes au bus ISA.

Pour répondre à ces besoins variés, Intel conçut une architecture générale comprenant plusieurs bus (voir figure 2). On y distingue 3 bus principaux : un bus mémoire qui permet au processeur de dialoguer directement avec la mémoire principale, un bus PCI et un bus ISA pour les dialogues avec les périphériques. Les deux composants clés de cette architecture sont des composants d'interconnexion appelés *ponts* : un *pont PCI*, qui interconnecte le

processeur, la mémoire et le bus PCI; un *pont ISA* qui interconnecte le bus PCI et le bus ISA. Sur le bus PCI, on trouve généralement un ou plusieurs slots d'extension permettant de connecter des périphériques, via des cartes additionnelles.

L'avantage majeur de cette architecture est que la bande passante des bus est adaptée aux besoins des périphériques associés. Le bus PCI se charge des périphériques nécessitant une importante bande passante (cartes graphiques, interface SCSI) et le bus ISA satisfait les besoins des périphériques lents (modems, ports série).

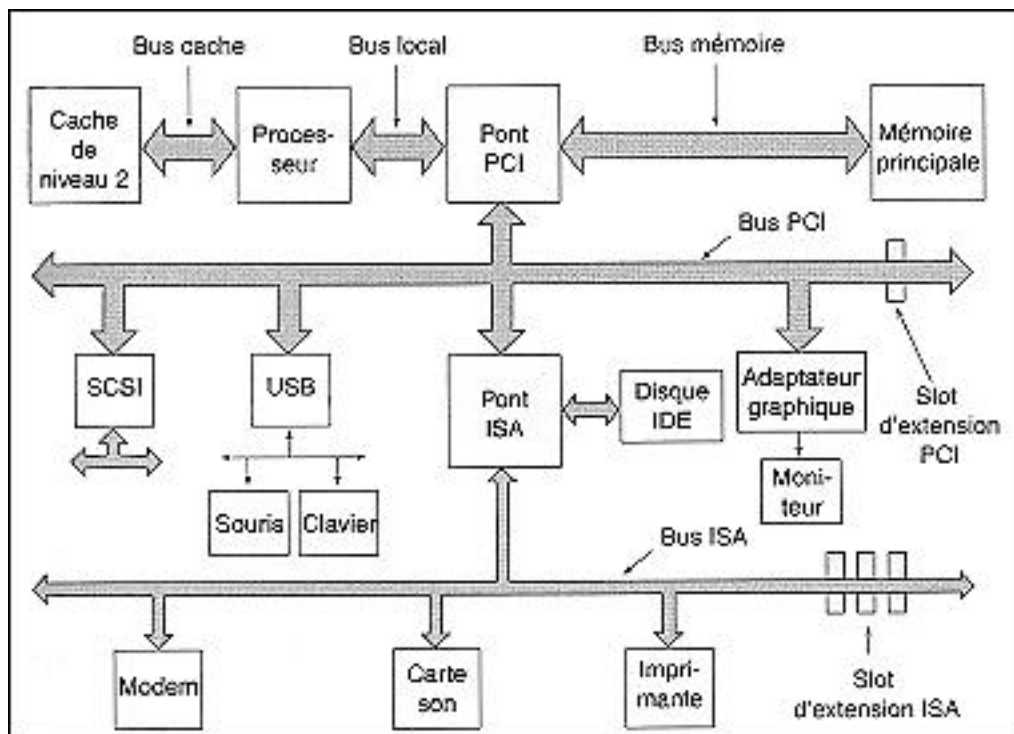


Figure 2 : Schéma d'une architecture PCI

Remarque : Il existe d'autres formes d'architectures comprenant plusieurs bus PCI et/ou ISA. C'est ainsi qu'il existe des ponts PCI/PCI.

Chapitre 3

Fonctionnement du PCI.

3.1 Description des signaux.

Le bus PCI comporte des signaux de base et des signaux optionnels comme le montrent les tableaux suivants. Sur ceux-ci, les colonnes *Maître* (M) et *Esclave* (S) précisent celui qui active le signal lors de la transaction courante. Si le signal est activé par un autre élément (par exemple, l'horloge, CLK), alors ces deux colonnes restent vides.

Signal	L	M	S	OBJET
CLK	1			Horloge (33 ou 66 MHz)
AD	32	X	X	Lignes d'adresses ou de données multiplexées
PAR	1	X		Parité sur AD
C/BE#	4	X		Signaux de commande du bus
FRAME#	1	X		Valide les signaux AD et C/BE
IRDY#	1	X		Lecture : le maître accepte Ecriture : la donnée est présente sur AD
IDSEL	1	X		Accès à l'espace de configuration plutôt qu'à la mémoire
DEVSEL#	1		X	L'esclave a décodé son adresse, il est à l'écoute
TRDY#	1		X	Lecture ; la donnée est présente ; écriture : l'esclave accepte
STOP#	1		X	Arrêt immédiat de transaction
PERR#	1			Erreur de la parité détecté par le récepteur
SERR#	1			Erreur de parité système ou adresse
REQ#	1			Arbitrage : demande de bus
GNT#	1			Arbitrage : allocation du bus
RST#	1			RàZ général du système

Les signaux du bus PCI : signaux de base du bus 32 bits

Signal	L	M	S	OBJET
REQ64#	1	X		Demande de transaction 64 bits
ACK64#	1		X	Permission accordée pour la transaction
AD	32	X		Les 32 lignes supplémentaires du bus
PAR64	1	X		Parité des 32 bits supplémentaire
C/BE#	4	X		4 bits supplémentaires à C/BE de base
LOCK	1	X		Blocage du bus pour transactions multiples
SBO#	1			Découverte d'un cache distant (en multiprocesseur)
SDONE	1			Snooping effectué
INTx	4			Demande d'interruption
JTAG	5			Signaux de tests JTAG
M66EN	1			Sélection de la fréquence d'horloge

Les signaux du bus PCI : signaux optionnels propre au bus 64 bits

Légende : L Lignes
M Maître
S Esclave

Le signal d'horloge, CLK, pilote le fonctionnement du bus. La plupart des signaux du bus sont synchrones à l'horloge. Sur le bus PCI, toute transaction commence sur un front descendant de l'horloge, c'est à dire au milieu d'un cycle de bus, plutôt qu'à son début.

Les 32 signaux AD sont soit les signaux d'adresses, soit les signaux de données (pour une transaction de 32 bits). En général, pendant le cycle 1, l'adresse est positionnée sur les lignes AD. Les données sont placées lors du cycle 3. Le signal PAR est un bit de parité sur les signaux AD.

Le signal C/BE# comprends 4 bits. Il a deux significations. Au cours du cycle 1, il indique le type d'action sur le bus ; c'est le signal de commande de l'échange (par exemple écriture ou lecture d'un octet ou d'un bloc d'octets). Au cours du cycle 2, les 4 bits indiquent quel(s) octet(s) du mot de 32 bits est (sont) valide(s). Ainsi, il est possible de lire (ou d'écrire) un, deux, trois ou quatre octets ou le mot entier. Le signal FRAME# est activé par le maître pour indiquer le début d'une transaction (l'initialisation). Il indique à l'esclave que l'adresse (AD) et les signaux de commande (C/BE#) sont valides. Lors d'une opération de lecture, le signal IRDY# est activé en même temps que FRAME#. Cela indique à l'esclave que le maître est prêt à lire la donnée qu'il lui transmettra. Lors d'une opération d'écriture, IRDY# est activé quelque temps après, lorsque la donnée à écrire se trouve sur le bus.

Tout équipement PCI a un espace de configuration de 256 octets qui peut être lu par n'importe quel autre équipement en activant IDSEL. Cet espace de configuration contient les informations techniques propres au périphérique concerné. Par exemple, le mécanisme *Plug and Play* utilise ce signal pour faire prendre connaissance au nouvel élément de son environnement et pour signaler sa présence aux autres éléments.

Le signal DEVSEL# indique que l'esclave a détecté son adresse sur le bus (lignes AD) et qu'il est en train de réaliser la transaction demandée. Si DEVSEL# n'est pas activé au bout d'un certain temps, le maître suppose que l'esclave est en panne ou qu'il est absent (par exemple qu'il est hors tension).

Le second signal propre à l'esclave est TRDY#. Il est activé par l'esclave lors d'une opération de lecture pour signaler au maître que la donnée sollicitée est sur le bus (lignes AD). Lors d'une opération d'écriture, TRDY# indique au maître que l'esclave est prêt à accepter la donnée qu'il lui envoie.

Les trois autres signaux permettent d'indiquer des erreurs. Il y a tout d'abord $STOP\#$, qui est activé par l'esclave pour mettre fin à la transaction courante pour raison grave. $PERR\#$ indique une erreur de parité sur la donnée pendant le cycle courant. Lors d'une lecture, il est activé par le maître et lors d'une écriture, c'est l'esclave qui s'en charge. C'est au récepteur concerné (maître ou esclave, selon l'opération) de prendre les décisions qui s'imposent. Enfin, $SERR\#$ indique une erreur de parité sur l'adresse.

Les signaux d'arbitrage du bus sont $REQ\#$ (demande de bus) et $GNT\#$ (allocation du bus). Il ne sont pas validés par le maître courant du bus PCI, mais par un nouveau demandeur de bus et par l'arbitre. Le dernier signal est $RESET\#$. Il permet de réinitialiser le système en cas d'erreur grave.

Les signaux optionnels (tableau b) correspondent à l'extension du bus PCI 32 bits à un bus de 64 bits. Les signaux $REQ64\#$ et $ACK64\#$ permettent au maître de demander à l'esclave la réalisation d'une transaction qui porte sur 64 bits, et à l'esclave de répondre s'il est d'accord ou non. Les signaux AD , $PAR64$ et $C/BE\#$ sont les extensions des signaux identiques du bus 32 bits.

Les trois signaux $LOCK$, $SB0\#$ et $SDONE$ ne sont pas présents sur le bus 32 bits. Ils concernent les systèmes multiprocesseurs. $LOCK$ permet de bloquer le bus pour effectuer plusieurs transactions successives. Les deux autres signaux sont utilisés pour régler des problèmes de cohérence de données entre le cache et la mémoire.

Les lignes $INTx$ portent les signaux de demande d'interruption. Une carte d'extension PCI peut comporter et gérer jusqu'à quatre équipements logiques distincts, chacun pouvant être doté d'un niveau d'interruption qui lui est propre. Les signaux $JTAG$ permettent d'effectuer les procédures de tests de la norme IEEE 149.1 $JTAG$. Enfin, le signal $M66EN$ est porté soit au potentiel $+V_{cc}$, soit à la masse, pour définir la fréquence de l'horloge. Il ne peut pas être modifié en cours de fonctionnement.

3.2 Transfert des données.

Le principe de transfert de données sur le bus PCI exploite les échanges par paquets. 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 réalisé avec succès, le maître engage l'accès au 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 prêt pour le transfert, mais surtout termine la phase d'adressage. Les esclaves potentiellement concernés sont alors amenés à répondre au maître consécutivement aux décodages 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 prêt pour la suite en conservant TRDY#. Le maître termine enfin l'échange, ou l'esclave réclame la fin du transfert par STOP#, comme dans la figure suivante.

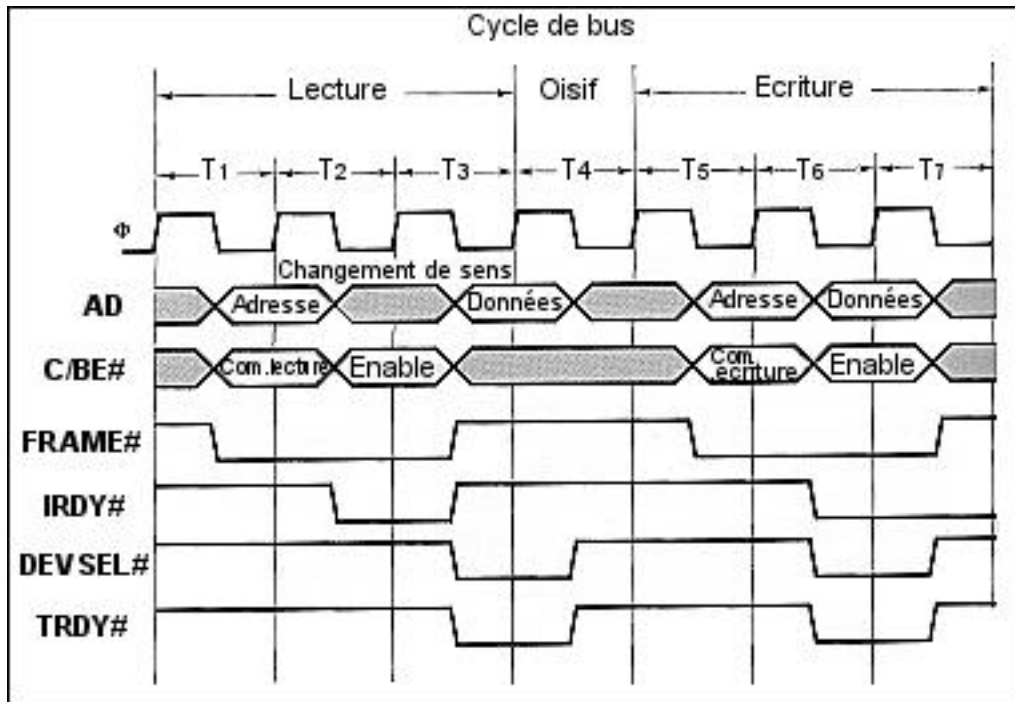


Figure 3 : Exemple de transaction sur un bus PCI

Cette figure illustre une transaction de lecture, suivie d'un cycle sans activité, et enfin d'une transaction d'écriture. Tout ceci est réalisé par le même maître.

Au front descendant du signal d'horloge (CLOCK), pendant T1, le processeur positionne l'adresse mémoire sur les lignes AD et le type d'opération souhaitée sur les lignes C/BE#. Il active ensuite FRAME# pour initialiser

la transaction sur le bus.

Pendant T2, le maître se place en état flottant sur les lignes AD afin de permettre à l'esclave de prendre ces lignes et de les piloter, c'est à dire d'y mettre la donnée désiré, pendant T3. Le maître ajuste également les signaux C/BE# pour indiquer dans le mot de 32 bits quels sont les octets qui l'intéressent.

Au cour du cycle T3, l'esclave active DEVSEL# de façon à ce que le maître sache que l'esclave a reconnu l'adresse et se prépare à répondre à la demande de lecture. Il positionne alors la donnée sur les lignes AD et valide TRDY# pour indiquer que la donnée est sur le bus, qu'il peut la lire. Si l'esclave n'est pas encore prêt à positionner la donnée sur le bus, il maintient actif le signal DEVSEL# mais n'active pas encore le signal TRDY#. Cela permet d'introduire un ou plusieurs états d'attente sur le bus. Lorsque la donnée est prête, qu'elle est sur le bus, l'esclave le signale en activant TRDY#.

Dans cet exemple, comme dans la réalité, le cycle de bus qui suit une transaction terminée est un cycle d'oisiveté (c'est à dire où il ne se produit rien). Lors du cycle T5, une nouvelle transaction commence. Il s'agit d'une opération d'écriture. Elle commence par le positionnement de l'adresse et de la commande de l'opération sur le bus, comme dans toute transaction. Ensuite, lors du second cycle de cette transaction, pendant T6, la donnée est positionnée sur le bus. Comme c'est le même maître qui utilise le bus, il n'est pas nécessaire de lancer une demande de bus. Enfin, pendant T7, la mémoire enregistre la donnée qui se trouve sur le bus.

3.3 Gestion de l'accès bus.

Pour utiliser le bus PCI, un maître potentiel doit tout d'abord le demander et l'obtenir en s'adressant à un arbitre centralisé, comme le montre la figure 4. Sur la plupart des systèmes, l'arbitre du bus est intégré au pont PCI. Chaque équipement relié au bus PCI dispose de deux lignes propres dédiées à l'arbitrage : une ligne de demande du bus, REQ# et une ligne d'allocation du bus, GNT#.

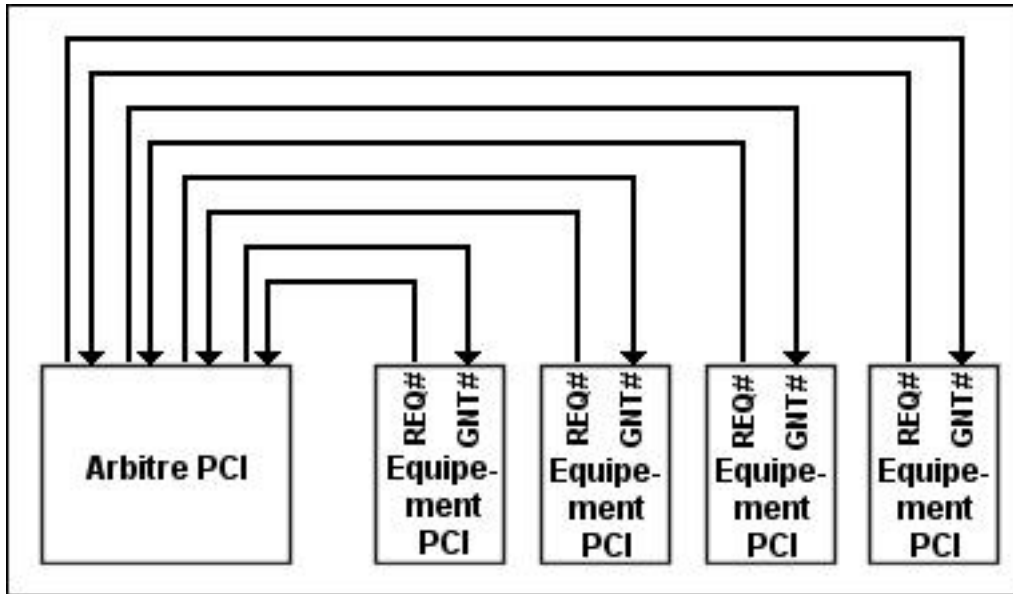


Figure 4 : L'arbitre centralisé d'un bus PCI

Pour demander le bus, un équipement PCI (y compris le processeur) active sa propre ligne REQ# et attend que l'arbitre lui réponde en activant sa ligne GNT#. Suite à cette autorisation, l'équipement peut alors utiliser le bus dès le début du cycle suivant. L'algorithme d'arbitrage n'est pas imposé, ni défini, dans les spécifications techniques du bus PCI. On rencontre les algorithmes classiques à priorité fixe et à priorité tournante, ainsi que d'autres techniques.

L'usage du bus est alloué pour une transaction, même si la durée de cette transaction n'est pas définie. Si un équipement désire commencer une deuxième transaction alors qu'aucun autre demandeur du bus ne s'est manifesté, il peut entreprendre sa seconde transaction. Toutefois, un cycle d'inactivité doit être inséré sur le bus entre les transactions. Dans certaines circonstances, en l'absence de compétition d'accès au bus, les transactions peuvent s'enchaîner sans passer par des états intermédiaires d'inactivité. Si un maître du bus entreprend une longue transaction et que d'autres équipements manifestent leur souhait d'utiliser le bus, l'arbitre décide alors de désactiver la ligne GNT# de l'utilisateur courant. Ce dernier libère le bus dès la fin du cycle en cours. Cela permet à un équipement de faire de longs transferts sur le bus lorsqu'il est seul demandeur, tout en permettant aux autres équipements susceptibles de demander le bus, intempestivement, de l'obtenir rapidement.

3.4 Gestion des interruptions.

Une interruption oblige le processeur à suspendre l'exécution d'une tâche en cours et à entreprendre l'exécution d'une procédure spéciale appelée *procédure d'interruption*. Une telle procédure permet par exemple de gérer les erreurs de programme, d'entreprendre des actions spécifiques ...

Les interruptions sur le PCI sont optionnelles par rapport à la configuration minimale nécessaire. Lorsque'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. 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.

Chapitre 4

Évolution du PCI.

4.1 Évolution et Évolutivité.

4.1.1 Évolution.

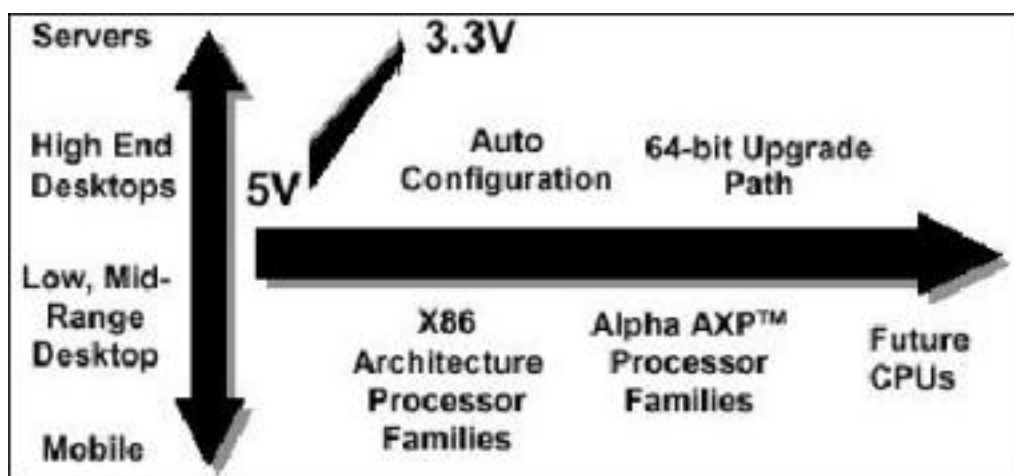


Figure 5 : Possibilités d'évolution du PCI

Le schéma ci dessus montre les possibilités d'évolution du bus PCI suivant 3 axes : La fonction de la machine équipée, L'architecture employée et les caractéristiques électriques adoptées.

4.1.2 les variantes.

Le Compact PCI.

Basé sur une spécification du PCI Industrial Computers Manufacturers Group (PICMG), cette norme est en fait une adaptation du PCI aux besoins industriels. En effet, les possibilités du PCI rendent son usage plus qu'intéressant, mais sa relative fragilité n'en permettait pas l'usage. Le Compact PCI est nettement plus résistant, il est doté d'un connecteur plus large et peu sensible aux impuretés. Malgré cela, il reste électriquement compatible avec le PCI usuel. Conçu pour fonctionner à une tension de 3.3 ou 5V, il est aussi 64bits. Il a aussi intégré la norme Hot Swap, qui permet l'ajout et le retrait des cartes à chaud.

Le Small PCI (SPCI).

Le PCI Sig, qui se charge du développement du PCI, a formé un groupe de travail chargé d'éditer la norme Small PCI. Suite à la demande de nombreux constructeurs, il devenait indispensable d'intégrer l'architecture PCI dans des ordinateurs de petite taille (portables,...). Pour cela, tout le packaging était à revoir, en effet, la taille des slots et des cartes est totalement inadéquat.

Le Small PCI offre les mêmes performances et les mêmes caractéristiques que le PCI standard. Basé sur une fréquence de 33Mhz, il possède une largeur de bus de 32bits et travaille en mode synchrone. Le Bus Mastering et le Bus Concurrency sont supportés. Sa tension est de 3.3V. Un signal, nommé CLKRUN, a même été ajouté. Ce dernier permet au système et au contrôleur de contrôler la fréquence du bus PCI. Cela permet ainsi de la réduire, quand aucune tche n'y est requise, ce qui permet de substantielles économies d'énergie. Le SPCI consomme beaucoup moins d'énergie qu'une carte PCI.

Un périphérique Small PCI se présente sous la forme d'une carte de même forme que les PCMCIA ou encore les Cardbus. Toute confusion est impossible, les cartes SPCI sont dotées d'un détrompeur. Comme le PCI standard, les cartes sont disponibles à trois tensions différentes : 3.3V, 5V et Universal (3.3 et 5V). Le connecteur de base est doté de 108 contacts. Les cartes sont disponibles sous deux formats :

- Style A : Cette carte fait appel à une double rangée de connecteurs.
- Style B : Ce format n'utilise qu'une rangée de connecteur. Il est ainsi possible de l'insérer dans un connecteur Style A. Elle n'utilisera alors que la rangée du haut.

Le Small PCI n'est en aucun cas concurrent du PCMCIA ou du Cardbus. En effet, le SPCI n'est prévu que pour un usage interne. Une carte ne pourra pas être insérée à chaud (Hot Plug). En fait, elle s'installera comme n'importe quelle carte d'extension PCI. De plus, sa surface sera nettement moins résistante que le PCMCIA. Par contre, elle sera nettement plus véloce que le Cardbus. Ces performances seront obtenues grce à des technologies telles que le Bus Mastering et une bande passante plus large. Les premiers systèmes disposants de tels connecteurs sont parus dans le courant de l'été 1998.

Le PCI Hot-Plug.

Cette norme est en réalité une extension de la norme PCI de base. Elle autorise l'ajout et le retrait des cartes PCI à chaud, sans extinction de l'ordinateur. En fait, trois situations sont prévues :

- Hot remplacement Permet le retrait à chaud d'une carte défectueuse et son remplacement par une autre carte, du même modèle.
- Hot upgrade Permet le remplacement d'un adaptateur par une version upgradée, soit plus récent, soit doté de nouvelles fonctions. Cette norme s'applique aussi aux drivers qui peuvent être mis à jour à chaud, sans interruption du système.
- Hot Expansion Permet l'ajout à chaud d'un adaptateur additionnel dans un slot libre

La société Compaq est l'investigatrice de cette norme. Mais, désormais, il s'agit d'un Industry Open standard. Cela sous-entend que n'importe quel constructeur peut utiliser cette norme sans payer de quelconques royalties. De plus, son développement est désormais public et ouvert à tous.

Pour pouvoir utiliser cette norme, il y a certains pré-requis au niveau hardware. Le système doit être capable de stopper l'alimentation électrique d'un slot uniquement, sans pour autant gêner les autres. De plus, il doit être capable de protéger les autres adaptateurs des chocs électriques causés par le Hot-Plug. En effet, il serait inimaginable qu'une autre carte plante ou subisse des micro-coupures lors de ces manipulations. Le système d'exploitation doit pouvoir gérer le retrait et l'ajout de cartes à chaud, tant au niveau hardware qu'au niveau logiciel. Les adaptateurs Hot-Plug et leurs drivers doivent être conçus en ce sens.

Lors de l'insertion d'une carte dans un slot libre, les étapes suivantes vont se produire :

- Installation à chaud et fixation de l'adaptateur
- L'utilisateur doit signaler au système que le slot peut être à nouveau alimenté. A cet effet, il peut le faire de manière logicielle via le système d'exploitation ou de manière hardware. En ce cas, les cartes-mères Hot-Plug sont souvent dotées d'un "interrupteur" situé à côté de chaque slot
- Le système ré-alimente le slot et détecte le composant. Il va alors charger le driver, qu'il possède peut-être. Dans la négative, il sera demandé à l'utilisateur de le fournir.

Le PCIx.

IBM, Hewlett-Packard et Compaq se sont regroupés pour jeter les bases d'une nouvelle spécification PCI destinée aux serveurs. Celle-ci offrira des performances nettement supérieures aux niveaux des performances I/O. Cette spécification, dont le nom de code est Project I, serait le premier pas d'une série de développements conjoints. Le but final étant d'assurer à ces trois sociétés un rôle dominant dans le monde des serveurs basés sur une architecture x86.

Le PCIx sera à même de supporter une fréquence de bus de 133Mhz pour des transferts de 1Go par seconde. Pour mémoire, le PCI actuel est limité à une fréquence de 66Mhz pour 528Mo/s. Les nouveaux composants PCIx offriront une compatibilité avec le matériel PCI actuel.

4.2 Normes.

Dans la définition d'un bus, on peut extraire différentes couches système que la norme peut, ou non spécifier. Le PCI se présente comme un concept global, qui définit l'ensemble des couches interagissant entre elles, comme dans la figure suivante.

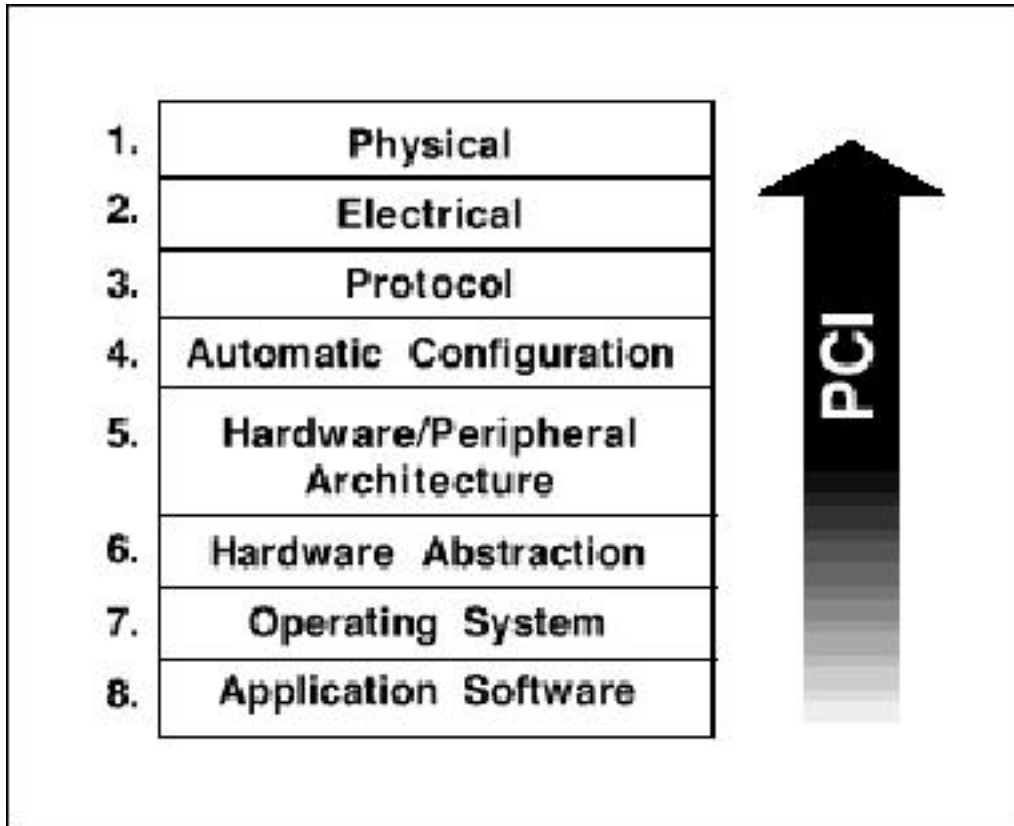


Figure 6 : Modèle de couches d'un système ouvert typique

Dans sa norme, le PCI offre une spécification en se plaçant depuis le logiciel et en descendant vers la couche physique.

4.3 Performances.

Le PCI cadencé à 33 MHz, peut assurer un taux de transfert de 133 Mo/s lorsqu'il est en mode 32 bits. En mode 64 bits, il est cadencé à 66 MHz, avec une bande passante de 528 mo/s. Il faut tout de même modérer ces taux de transfert en pics. À cet effet, INTEL a réalisé des mesures de taux de transfert en déterminant un ratio 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.

En appliquant cette valeur d'efficacité au taux maximum de transfert, on revient vers des valeurs d'environ 80 à 90 Mo/s pour le mode 32 bits et 320 à 360 pour le mode 64 bits.

Le bus PCI intègre des fonctions intelligentes telles que le *plug and play* (dont nous avons parlé avant) qui font de lui un bus très fonctionnel.

Chapitre 5

Conclusion.

Aujourd'hui, le bus PCI équipe la majorité des systèmes. Cette étude nous a permis de découvrir les principales caractéristiques de cet incontournable standard. Il s'impose comme le digne successeur du bus ISA en apportant un nombre non négligeable d'avantages, bien qu'il soit, au premier abord, plus difficile à utiliser. D'ailleurs, avec l'utilisation massive des cartes PCI, le bus ISA est amené à disparaître, puisque la plupart des constructeurs préfèrent utiliser les capacités plus importantes du PCI.

Bibliographie

- [1] Andrew TANENBAUM. *Architecture de l'ordinateur*. Dunod, 2001.
- [2] Tom CHANLEY et Don ANDERSON. *PCI System Architecture*. MindShare, 1995.
- [3] Patrice KADIONIK. *Le Bus Industriel PCI*. Enseirb, 2001.