ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE
M. Ing.

PAR
Steve BEAULIEU

ANALYSE DU DÉTERMINISME ET DE LA FIABILITÉ DU PROTOCOLE PCI EXPRESS DANS UN CONTEXTE DE CERTIFICATION AVIONIQUE

MONTRÉAL, LE 10 FÉVRIER 2012

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Jean-François Boland, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Guy Bois, codirecteur de mémoire
Département de génie informatique et logiciel à l'École Polytechnique de Montréal

M. Michel Kadoch, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Bruno De Kelper, membre du jury
Département de génie électrique à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 18 JANVIER 2012

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

En premier lieu, j'aimerais remercier mon directeur, le professeur Jean-François Boland et mon codirecteur, le professeur Guy Bois pour leur soutien tout au long de mes études de maîtrise.

J'aimerais aussi remercier les organismes qui m'ont supporté financièrement : le Regroupement stratégique en microsystèmes du Québec (ReSMiQ), le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), le Consortium de recherche et d'innovation en aérospatiale au Québec (CRIAQ) et l'École de technologie supérieure.

Pour leurs supports financiers et leurs précieux conseils, je remercie également les deux partenaires industriels impliqués dans mon projet, soient CAE et CMC Électronique.

Finalement, je voudrais remercier ma famille pour son appui inconditionnel. Ainsi, je dédie ce mémoire à ma conjointe Caroline, ainsi qu'à mes enfants, Alyson et Joanie.

ANALYSE DU DÉTERMINISME ET DE LA FIABILITÉ DU PROTOCOLE PCI EXPRESS DANS UN CONTEXTE DE CERTIFICATION AVIONIQUE

Steve BEAULIEU

RÉSUMÉ

L'avionique classique n'a plus à démontrer ses capacités en termes de fiabilité. Cependant, des contraintes économiques ont poussé les avionneurs à moderniser les systèmes existants. Ainsi, les systèmes avioniques modulaires intégrés (IMA) ont vu le jour dans le but de réduire la masse, le volume et la consommation électrique du matériel embarqué.

Pour combler les besoins en communication des systèmes IMA, les bus de données traditionnels sont devenus limitatifs. Bien que les protocoles actuels comme l'AFDX (*Avionics Full Duplex Switched Ethernet*) soient très performants, d'autres alternatives sont envisageables. Par conséquent, ce mémoire propose une étude du protocole PCI Express en vue de son utilisation dans les systèmes avioniques.

La caractérisation des bus de communication a permis d'identifier les éléments pertinents en lien avec les exigences imposées par les autorités de certification avionique. Ainsi, les contraintes relatives au déterminisme et à la fiabilité représentent les éléments principaux de cette recherche. Un travail assidu a permis d'extraire des spécifications tous les mécanismes lui conférant sa fiabilité et son déterminisme.

Ce travail avait comme deuxième objectif le développement d'une plateforme d'expérimentation dans le but de valider les concepts théoriques. Des expérimentations ont ainsi permis de tester les éléments critiques en lien avec les objectifs de cette recherche.

Finalement, mentionnons que les résultats de cette recherche permettent de démontrer que le protocole PCI Express possède les caractéristiques essentielles exigées par les autorités de certification avionique. En effet, en plus de ses nombreux mécanismes de fiabilité, ce protocole possède un mode de fonctionnement entièrement déterministe.

Mots-clés : bus de données, avionique, PCI Express, déterminisme, fiabilité

ANALYSE DU DÉTERMINISME ET DE LA FIABILITÉ DU PROTOCOLE PCI EXPRESS DANS UN CONTEXTE DE CERTIFICATION AVIONIQUE

Steve BEAULIEU

ABSTRACT

Over the years, federated avionics has demonstrated its capabilities in terms of reliability. However, economic constraints have led manufacturers to upgrade existing systems. Thus, the integrated modular avionics (IMA) have emerged in order to reduce the size, weight and power of embedded systems.

To meet the communication needs of IMA systems, new data bus architectures and protocols are required. Although current protocols such as AFDX (*Avionics Full Duplex Switched Ethernet*) are very effective, other alternatives are possible. Therefore, this master's thesis proposes a study of the PCI Express protocol for use in avionics systems.

The characterization of the communication bus has identified the relevant factors in relation to requirements imposed by the avionics certification authorities. Thus, constraints on the determinism and reliability are the main focus of this research. An extensive work has been done to extract from the specification all the mechanisms conferring reliability and determinism.

The second objective of this research was to develop an experimental platform to validate the theoretical concepts. Experiments were conducted to test the critical elements related to the objectives of this research.

Finally, the results of this research demonstrate that the PCI Express protocol possesses the essential characteristics required by the avionics certification authorities. Indeed, in addition to its multiple mechanisms of reliability, the protocol has a deterministic mode of operation.

Keywords : data buses, avionics, PCI Express, determinism, reliability

TABLE DES MATIÈRES

Page

CHAPITRE 1 INTRODUCTION 21
1.1 Problématique 21
1.2 Objectifs 22
1.3 Originalité et contributions 22
1.4 Organisation du mémoire 24

CHAPITRE 2 REVUE DE LA LITTÉRATURE 25
2.1 Survol 25
2.2 Architecture PCI Express 25
2.2.1 Historique 25
2.2.2 Composants 27
2.2.3 Lien PCI Express 29
2.2.4 Topologie 30
2.2.5 Configuration 34
2.2.6 Connexions physiques 36
2.3 Le protocole PCI Express 37
2.3.1 Modèle en couches 37
2.3.2 Catégories de transaction 40
2.3.3 Registres de configuration 41
2.3.4 Canaux virtuels et catégories de trafic 41
2.3.5 Méthodes de routage 43
2.3.6 Bande passante 44
2.4 Systèmes avioniques 46
2.4.1 Architecture des systèmes avioniques 46
2.4.2 Certification des systèmes avioniques 51
2.5 Étude du réseau AFDX 53
2.5.1 Historique 53
2.5.2 Modification du réseau Ethernet 55
2.5.3 Éléments de déterminisme et de fiabilité 56
2.5.4 Certification de l'AFDX 60

CHAPITRE 3 CARACTÉRISATION DES BUS DE COMMUNICATION 63
3.1 Survol 63
3.2 Métriques 63
3.2.1 Utilisation 63
3.2.2 Efficacité 64
3.2.3 Débit 64
3.2.4 Latence 65
3.2.5 Gigue 65
3.3 Bus de données avioniques 67
3.3.1 Besoins des bus de données avioniques 68

3.3.2 Sécurité des bus de données avioniques 69
3.3.3 Exigences des bus de données avioniques 73

CHAPITRE 4 FIABILITÉ DU PROTOCOLE PCI EXPRESS 77
4.1 Survol 77
4.2 Mécanismes de fiabilité 77
4.2.1 Contrôle de flux 78
4.2.2 CRC de bout en bout (ECRC) 79
4.2.3 CRC de liaison (LCRC) 79
4.2.4 Numéro de séquence 79
4.2.5 Encodage 8b/10b 80
4.3 Gestion des erreurs 80
4.4 Classification des erreurs 81
4.4.1 Erreur corrigible 81
4.4.2 Erreur incorrigible non fatale 81
4.4.3 Erreur incorrigible fatale 81
4.5 Signalement des erreurs 82
4.6 Enregistrement des erreurs 84
4.7 Interruption 84
4.8 Support pour le branchement à chaud (hot-plug) 84
4.9 Synthèse 85

CHAPITRE 5 DETERMINISME DU PROTOCOLE PCI EXPRESS 87
5.1 Survol 87
5.2 Éléments affectant la performance 88
5.2.1 Largeur du lien 88
5.2.2 Surdébit 88
5.2.3 Taille maximale de la charge utile 92
5.2.4 Taille maximale d'une requête de lecture 94
5.2.5 Disponibilité des crédits du contrôle de flux 96
5.2.6 Taille du tampon de retransmission 98
5.2.7 Nombre d'étiquettes 100
5.3 Mode isochrone 100
5.3.1 Transactions isochrones 100
5.3.2 Contrat isochrone 102
5.3.3 Paramètres isochrones 102
5.4 Synthèse 105

CHAPITRE 6 PLATEFORME D'EXPÉRIMENTATION 107
6.1 Survol 107
6.2 Méthodologie 107
6.2.1 Méthode expérimentale 107
6.2.2 Matériel utilisé 110
6.2.3 Logiciels utilisés 111
6.2.4 Architecture de tests 111
6.2.5 Scénarios de tests 113

6.3 Environnement de simulation....115
6.3.1 Résultats de simulation 115
6.3.2 Chronogrammes de simulation 116
6.4 Environnement expérimental 116
6.4.1 Résultats expérimentaux 117
6.4.2 Chronogrammes expérimentaux 117
6.4.3 Signaux de tests.... 118

CHAPITRE 7 PRÉSENTATION ET INTERPRÉTATION DES RÉSULTATS.....119
7.1 Survol 119
7.2 Résultats de simulation....120
7.2.1 Surdébit (*Overhead*).... 120
7.2.2 Taille maximale de la charge utile (MPS : *Maximum Payload Size*) 122
7.2.3 Disponibilité des crédits du contrôle de flux 124
7.2.4 Taille du tampon de retransmission 126
7.2.5 Nombre d'étiquettes (*TAG*).... 127
7.3 Résultats expérimentaux sur plateforme matérielle 128
7.3.1 Largeur du lien.... 128
7.3.2 Taille maximale d'une requête de lecture.... 129
7.3.3 Insertion d'erreurs.... 132
7.4 Présentation du programme....137
7.4.1 Démonstration.... 139

CONCLUSION....141

RECOMMANDATIONS 145

BIBLIOGRAPHIE....147

LISTE DES TABLEAUX

Page

Tableau 2.1 Types de transaction 40

Tableau 2.2 Niveaux de criticité DO-178B 52

Tableau 3.1 Classification des défaillances (Moir et Seabridge, 2008) 70

Tableau 3.2 Critères d'évaluation des bus de données avioniques (Zalewski et al., 2005) 71

Tableau 4.1 Résumé des classes d'erreurs PCI Express 82

Tableau 4.2 Messages d'erreur envoyés au Root Complex 83

Tableau 4.3 Ordre de préséance des erreurs détectées par la couche transaction 83

Tableau 5.1 Bande passante PCI Express (génération 1) 89

Tableau 5.2 Efficacité du TLP 93

Tableau 5.3 Bande passante maximale théorique 95

Tableau 7.1 Paramètres communs 120

Tableau 7.2 Nombre de paquets 122

Tableau 7.3 Bande passante selon la taille du tampon de retransmission 126

Tableau 7.4 Bande passante atteinte 131

Tableau 7.5 Latence aller-retour pour chacun des scénarios 136

Tableau 7.6 Démonstration du contrôleur DMA 139

LISTE DES FIGURES

Page

Figure 2.1	Réseau PCI Express typique	27
Figure 2.2	Commutateur à trois ports	29
Figure 2.3	Lien à une voie	30
Figure 2.4	Pont non transparent séparant deux domaines	31
Figure 2.5	Exemple de configuration PCI Express	35
Figure 2.6	Modèle en couches du protocole PCI Express	38
Figure 2.7	Liaison point à point VS bout en bout	39
Figure 2.8	Association TCs/VCs	43
Figure 2.9	Exemple d'architecture fédérée	47
Figure 2.10	Exemple d'architecture IMA	49
Figure 2.11	Exemple de réseau AFDX	55
Figure 2.12	Mécanisme du BAG	57
Figure 2.13	Gigue sur un réseau AFDX	58
Figure 2.14	Redondance du réseau AFDX	59
Figure 4.1	Mécanismes de fiabilité du protocole PCI Express	78
Figure 5.1	Structure du TLP	90
Figure 5.2	Taille maximale de la charge utile	92
Figure 5.3	Efficacité du TLP selon la valeur de MPS	94
Figure 5.4	Processus du contrôle de flux	96
Figure 5.5	Latence d’acquittement faible	98
Figure 5.6	Latence d'acquittement élevée	99
Figure 5.7	Types de liaison isochrone	101

Figure 6.1 Carte PCI Express Altera Arria II GX 110
Figure 6.2 Architecture de tests 112
Figure 6.3 Exemple de résultats de simulation 115
Figure 6.4 Exemple de chronogramme de simulation 116
Figure 6.5 Exemple de chronogramme en temps réel 117
Figure 7.1 Effet du surdébit sur la bande passante 121
Figure 7.2 Effet de la taille maximale de la charge utile sur la bande passante 123
Figure 7.3 Bande passante selon le nombre de crédits disponibles 125
Figure 7.4 Effet du nombre d'étiquettes sur la bande passante 127
Figure 7.5 Bande passante selon le nombre de voies par lien 129
Figure 7.6 Requête d'écriture dans la mémoire principale 130
Figure 7.7 Réponse d'une requête de lecture dans la mémoire principale 130
Figure 7.8 Test sur une liaison PCI Express 132
Figure 7.9 Chronogramme du scénario 2 133
Figure 7.10 Chronogramme du scénario 3 135
Figure 7.11 Menu principal du programme 137

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ACK	Acknowledged
AFDX	Avionics Full Duplex Switched Ethernet
ARINC	Aeronautical Radio, Incorporated
ASIC	Application Specific Integrated Circuit
BAG	Bandwidth Allocation Gap
BAR	Base Address Register
BER	Bit Error Rate
COTS	Commercial Off-The-Shelf
CRC	Cyclic Redundancy Check
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DLLP	Data Link Layer Packet
DMA	Direct Memory Access
FC	Flow Control
FPGA	Field Programmable Gate Array
IEEE	Institute of Electrical and Electronics Engineers
IMA	Integrated Modular Avionics
LRM	Line Replaceable Module
LRU	Line Replaceable Unit
MMU	Memory Management Unit
NAK	Not Acknowledged
NTB	Non-Transparent Bridge
PCI	Peripheral Component Interconnect

PCI-X	PCI Extended
PLD	Programmable Logic Device
PLP	Physical Layer Packet
QoS	Quality of Service
RAID	Redundant Array of Independent Disks
RCB	Read Completion Boundary
RTCA	Radio Technical Commission for Aeronautics
SKP	Skip Ordered-Set
SWaP	Size, Weight and Power
TC	Traffic Class
TLP	Transaction Layer Packet
VC	Virtual Channel
VHDL	Very high speed integrated circuit Hardware Description Language
VL	Virtual Link

CHAPITRE 1

INTRODUCTION

1.1 Problématique

De nos jours, prendre l'avion est une activité qui procure une grande tranquillité d'esprit. En effet, les normes strictes imposées aux avionneurs font de l'avion l'un des moyens de transport les plus sûrs. Ceci est dû, en grande partie, à la fiabilité des systèmes embarqués. Ayant fait ses preuves, l'avionique fédérée classique n'a plus à démontrer ses capacités en terme de fiabilité. Cependant, des contraintes économiques ont poussé les avionneurs à moderniser les systèmes existants. Ainsi, les nouveaux appareils tels l'Airbus A380 et le Boeing 787 ont vu leurs systèmes embarqués s'intégrer à de nouvelles plateformes de type modulaire. Ces plateformes, connues sous l'appellation de systèmes IMA (*Integrated Modular Avionics*), ont été conçues dans le but de réduire l'espace, le poids et la consommation électrique des systèmes avioniques (Wilson et Preyssler, 2009).

Pour combler les besoins en communication des nouveaux appareils utilisant les systèmes IMA, les bus de données traditionnels sont devenus limitatifs. Ainsi, un nouveau standard a vu le jour. Il s'agit de l'AFDX (*Avionics Full Duplex Switched Ethernet*), une version déterministe et fiabilisée du réseau Ethernet (Alena et al., 2007). Cependant, dans la catégorie des protocoles de haute performance, il n'y a pas que l'Ethernet et ses variantes. Certains protocoles, dont le PCI Express, sont de plus en plus présents dans les systèmes qui requièrent une performance élevée. À titre d'exemple, pour sa nouvelle architecture Atom, Intel a remplacé son bus propriétaire par un bus PCI Express pour relier le processeur aux autres circuits intégrés (*chipset*) (Coupé, 2011). Rendu à sa troisième génération, ce protocole procure une performance élevée en plus d'offrir une grande fiabilité. Étant destinée à l'origine au marché grand public, la technologie PCI Express demeure très accessible.

Grâce à ses caractéristiques, il n'est pas étonnant que la technologie PCI Express suscite l'intérêt des concepteurs de systèmes avioniques. En effet, le protocole PCI Express possède

plusieurs fonctionnalités de fiabilité, de disponibilité et de facilité de maintenance communément appelées RAS (*Reliability, Availability, Serviceability*) (Krig, 2003). La fiabilité fait référence à la capacité du protocole à se conformer à ses spécifications dans des conditions d'utilisation normale. La disponibilité est le rapport entre le temps qu'un système est fonctionnel et le temps total nécessaire pour exécuter ses fonctions. La facilité de maintenance, quant à elle, concerne la capacité de remplacer ou réparer un système en limitant les impacts sur sa fonctionnalité.

En plus d'être fiable, un protocole à haut débit doit être performant. De plus, si ce protocole est utilisé dans un environnement où la sécurité est critique, le déterminisme devient une priorité. Ainsi, les paramètres de déterminisme, de fiabilité et de performance sont au cœur de ce travail de recherche portant sur le protocole PCI Express.

1.2 Objectifs

Ce travail de recherche est issu du projet CRIAQ AVIO-509 qui traite de l'exploration architecturale des systèmes avioniques modulaires intégrés. Plus spécifiquement, ce mémoire est consacré à l'étude du protocole PCI Express en vue de son utilisation dans les systèmes avioniques. L'objectif principal étant de caractériser le protocole en considérant les exigences imposées par l'industrie avionique. Parmi ces exigences, mentionnons les plus importantes qui sont le déterminisme et la fiabilité.

Le second objectif de ce travail est le développement d'une architecture de tests dont le but est de pouvoir valider expérimentalement les concepts théoriques.

1.3 Originalité et contributions

Une recherche exhaustive a permis de caractériser le protocole PCI Express dans un contexte de certification avionique. Au meilleur de notre connaissance, aucune recherche en ce sens n'a été effectuée antérieurement. Il s'agit donc d'une contribution importante pour le domaine aérospatial.

À l'heure actuelle, ce protocole n'est pas utilisé dans les systèmes avioniques. Donc, dans un premier temps, il a fallu étudier les systèmes en place afin de les caractériser. En étudiant les bus de données avioniques et plus particulièrement l'AFDX, les éléments critiques ont été identifiés. Ainsi, le déterminisme et la fiabilité représentent les éléments fondamentaux de tous bus de données avioniques.

Le déterminisme procure une garantie qu'un message envoyé sera reçu dans une limite temporelle prévisible. La fiabilité, quant à elle, est de deux types. Au niveau système, la fiabilité représente la capacité d'un réseau à résister aux défaillances. Au niveau message, la fiabilité procure plutôt une garantie que l'information sera livrée sans erreur (Federal Aviation Administration, 2005a).

L'étude du protocole PCI Express a été réalisée en considérant ces critères fondamentaux qui sont le déterminisme et la fiabilité. Les spécifications, relativement complexes, contiennent beaucoup d'informations dont les thèmes principaux sont l'architecture système, le modèle en couches et la gestion de l'énergie. Parmi ces informations, plusieurs sont de moindre importance pour la recherche qui nous concerne. Ainsi, l'avantage de cette recherche est d'avoir pu identifier seulement les éléments pertinents en lien avec les exigences prescrites par les autorités de certification avionique. Mentionnons que cette recherche a été effectuée en considérant la version 2.0 des spécifications. Lorsque cette recherche a été effectuée, les spécifications de la troisième génération n'étaient pas encore disponibles. Cependant, des vérifications ultérieures ont permis de constater que les sujets abordés ici sont encore valables dans les spécifications de la troisième génération.

La seconde contribution a été le développement d'une architecture de tests, dont le principal objectif était de pouvoir valider de façon expérimentale les concepts théoriques. Il s'agit d'une architecture polyvalente conçue sur un réseau prédiffusé programmable (FPGA). Un circuit FPGA offre plusieurs avantages. En effet, la flexibilité et la modularité que procure cette technologie permettent un développement rapide et adapté à des besoins spécifiques. De plus, le FPGA choisi contient un module PCI Express intégré paramétrable selon les tests

désirés. Enfin, grâce à sa grande flexibilité, cette architecture pourra servir ultérieurement dans d'autres travaux de recherche.

Finalement, mentionnons que cette recherche a permis de démontrer que le protocole PCI Express possède les caractéristiques essentielles exigées par les autorités de certification avionique. En effet, ce protocole possède plusieurs mécanismes de fiabilité en plus de fournir une configuration déterministe.

1.4 Organisation du mémoire

Ce mémoire est organisé comme suit. Le chapitre 2 est consacré à la revue de la littérature. La technologie PCI Express y est présentée ainsi que les systèmes avioniques. Ce chapitre se termine par une étude du réseau AFDX. Le chapitre 3 caractérise les bus de communication. Les métriques y sont présentées, suivi des bus de communication avionique. Le chapitre 4 identifie les mécanismes de fiabilité du protocole PCI Express. Le déterminisme, quant à lui, est présenté au chapitre 5. Le chapitre 6 est consacré à la plateforme d'expérimentation. La méthodologie y est présentée de même que les environnements de simulation et d'expérimentation. Le chapitre 7 expose les résultats obtenus et présente brièvement le programme ayant servi à l'exécution des tests. Finalement, la conclusion fait la synthèse des principaux éléments du protocole PCI Express en lien avec les exigences de l'industrie avionique. Ce mémoire se termine par des recommandations.

CHAPITRE 2

REVUE DE LA LITTÉRATURE

2.1 Survol

Les deux premières sections de la revue de la littérature présentent respectivement l'architecture et le protocole PCI Express. Ensuite, les systèmes avioniques sont présentés dans leur généralité. La dernière section, quant à elle, est consacrée à l'étude du réseau AFDX (*Avionics Full Duplex Switched Ethernet*).

2.2 Architecture PCI Express

Après un bref historique, cette section présente l'architecture PCI Express. Dans l'ordre, cette section décrit les types de composant, les liens, les topologies, les configurations du réseau et les types d'interconnexion physique.

2.2.1 Historique

Au début des années 90, le bus PCI fait son apparition dans les architectures PC. Avec une fréquence de fonctionnement de 33 MHz, le bus PCI comblait tout les besoins en bande passante des périphériques de l'époque. En 1995, pour accommoder les nouveaux périphériques plus rapides, la fréquence du bus PCI passe à 66 MHz. Cependant, durant la dernière décennie, la fréquence de fonctionnement des processeurs et des périphériques a considérablement augmenté. Par conséquent, le bus PCI devient rapidement limitatif pour la performance des systèmes. Par exemple, les nouvelles technologies telles le Gigabit Ethernet et le 1394b (FireWire), monopolisent presque entièrement le bus PCI (Wilen, Schade et Thornburg, 2003). Rapidement, il devient évident qu'une nouvelle technologie doit remplacer le bus PCI.

Le PCI Express a été introduit sur le marché en 2004. Il s'agit de la troisième génération des bus d'interconnexion de haute performance. La première génération concerne les bus ISA et EISA alors que la deuxième génération fait référence aux bus PCI, PCI-X et AGP.

Dans la littérature, le terme PCI Express est souvent réduit à PCIe. Cependant, on ne doit pas confondre PCI-X et PCI Express. Le bus PCI-X a été développé pour les serveurs dans les années 90. Il s'agit d'une évolution du bus PCI standard. Le bus PCI-X propose des fréquences d'horloge plus élevées que le bus PCI.

Le bus PCI Express possède l'avantage d'être rétrocompatible avec les systèmes PCI et PCI-X. Par conséquent, les systèmes d'exploitation existants ont pu faire la transition vers le PCI Express sans aucune modification au niveau des programmes et des pilotes.

2.2.2 Composants

Il existe quatre types de composants PCI Express, soit le Root Complex, l'Endpoint, le commutateur et le pont. La Figure 2.1 représente un réseau PCI Express typique.

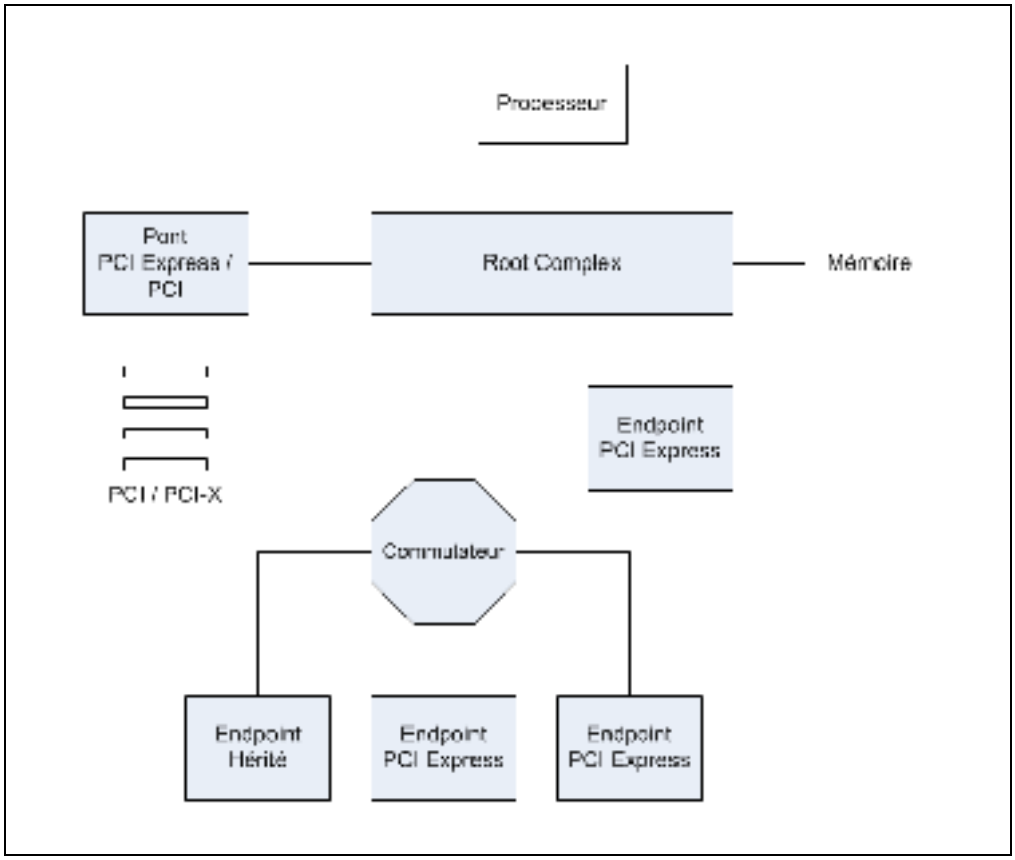

Figure 2.1 Réseau PCI Express typique

Voici une description sommaire de chacun des composants.

Root Complex

Le Root Complex est le composant principal du système. Il permet de relier les entrées et les sorties au processeur et à la mémoire. Le Root Complex possède un ou plusieurs ports PCI Express. Chacun de ses ports représente un domaine hiérarchique différent. Un domaine

hiérarchique peut contenir un seul Endpoint ou plusieurs Endpoints reliés par des commutateurs.

Endpoint

Un Endpoint est un composant périphérique telle une carte graphique ou une carte Ethernet. Il existe deux types d'Endpoint, soit l'Endpoint PCI Express et l'Endpoint hérité (*Legacy Endpoint*). Ce dernier répond aux besoins de compatibilité avec les logiciels PCI et PCI-X. Autrement dit, un Endpoint hérité est un composant PCI qui a été adapté à la technologie PCI Express.

Commutateur

Un commutateur permet de relier plusieurs composants au domaine hiérarchique. La tâche du commutateur est de transférer les paquets de données entre ses différents ports. Des règles de routage permettent d'acheminer les données vers le bon port. Un commutateur possède un seul port ascendant (*Upstream Port*) et au moins deux ports descendants (*Downstream Port*). Le port ascendant est le port qui se dirige vers le Root Complex. À l'interne, des ponts PCI permettent de relier entre eux les différents ports. La Figure 2.2 illustre l'intérieur d'un commutateur à trois ports.

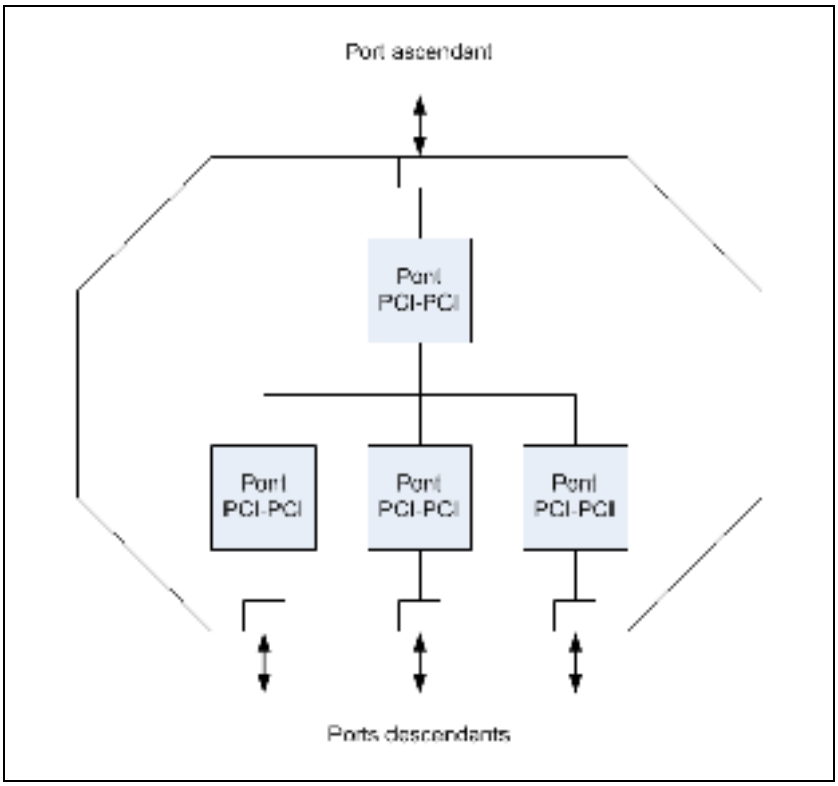

Figure 2.2 Commutateur à trois ports

Pont PCI Express / PCI

Le pont PCI Express / PCI possède un port PCI Express et une ou plusieurs interfaces PCI. Ce pont permet la coexistence des technologies PCI et PCI-X au sein d'un système PCI Express.

2.2.3 Lien PCI Express

La technologie du bus PCI Express est très différente de celle employée par son prédécesseur. En effet, le PCI est un bus parallèle partagé par plusieurs composants. Le bus PCI Express, quant à lui, implémente une liaison série, point à point, entre deux composants.

Une voie PCI Express est définie par une paire de signaux différentiels. La Figure 2.3 illustre un lien à une voie entre deux composants.

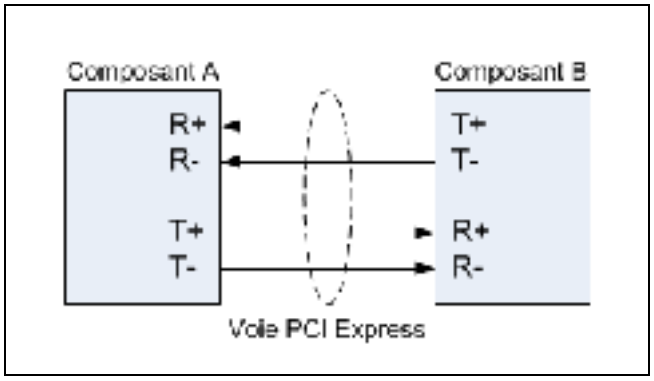

Figure 2.3 Lien à une voie

Un lien PCI Express peut être composé de plusieurs voies. On retrouve des liens à 1, 2, 4, 8, 12, 16 et 32 voies. La largeur du lien se note avec un « x » devant le nombre de voies. Par exemple, un lien x4 identifie un lien à quatre voies.

2.2.4 Topologie

On distingue deux grands types de topologie PCI Express. Il s'agit des topologies à Root Complex unique et à Root Complex multiple.

La grande majorité des systèmes PCI Express contient un Root Complex, des composants Endpoint et des commutateurs. Dans certains systèmes on retrouve aussi un pont PCI Express / PCI pour relier les composants PCI et PCI-X. Ces systèmes sont dits à Root Complex unique.

La technologie PCI Express n'autorise qu'un seul Root Complex dans son architecture. En effet, à la mise sous tension ou après une remise à zéro du système, le Root Complex procède à l'énumération du système en entier. Si plus d'un Root Complex est présent, il y aura des conflits et le système sera non fonctionnel (Budruk, Anderson et Shanley, 2004). Il existe

cependant un mécanisme qui permet à plusieurs Root Complex de coexister au sein d'un même système. Il s'agit du pont non transparent (NTB : *Non-Transparent Bridge*). Grâce à ce type de pont, les systèmes à Root Complex multiple sont permis.

Le pont non transparent procure une isolation logique entre deux domaines distincts (Gudmundson, 2004). Chacun de ces domaines contient un processeur et son Root Complex. De plus, les composants appartenant à un domaine n'ont aucune visibilité sur l'autre côté du pont. En effet, les ports du pont non transparent sont de type 0, c'est-à-dire de type Endpoint (voir section 2.3.3). Pendant l'énumération, si le Root Complex découvre un pont non transparent, ce dernier est identifié comme étant un Endpoint. Ainsi, ce qui se trouve de l'autre côté du pont n'est pas connu de ce Root Complex. La Figure 2.4 illustre un pont non transparent séparant deux domaines.

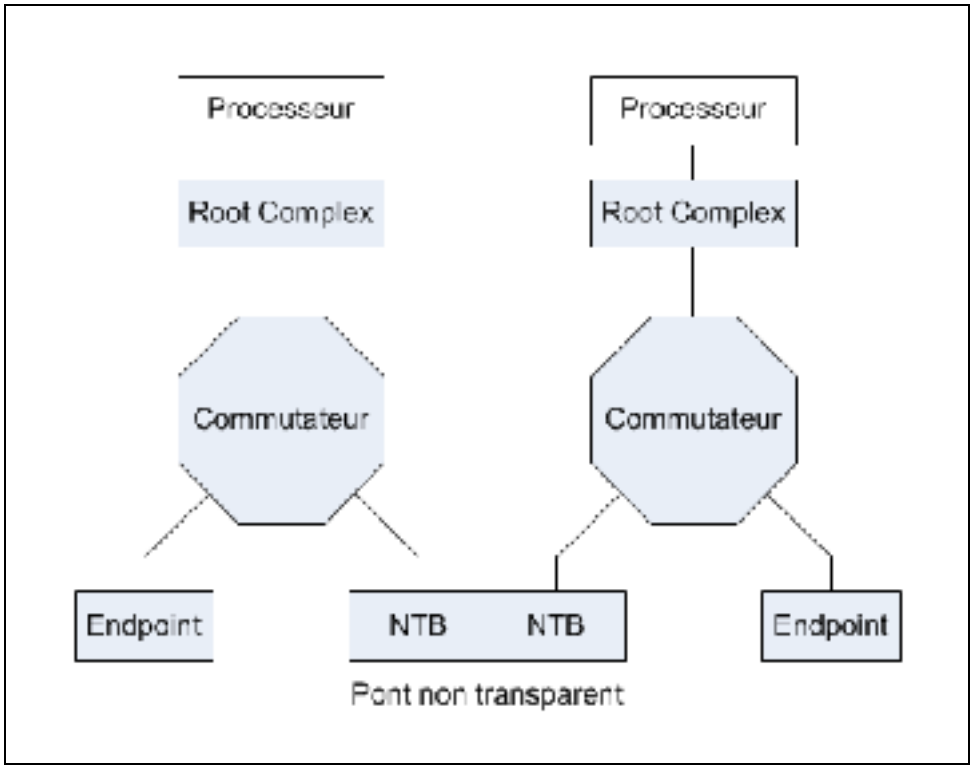

Figure 2.4 Pont non transparent séparant deux domaines

La communication entre les deux domaines est possible grâce aux registres de translation. Les ports du pont non transparent possèdent tous des registres d'adresse de base (BARs). Ces derniers sont utilisés pour définir les adresses de translation permettant d'accéder à l'espace mémoire de l'autre côté du pont. Une transaction peut ainsi passer d'un domaine à l'autre grâce à ce mécanisme de translation d'adresses.

Le pont non transparent ne fait pas partie des spécifications PCI Express (Gudmundson, 2004). Par conséquent, pour assurer la compatibilité avec le protocole, des mécanismes spécifiques à ce composant ont été créés (PLX Technology, 2004). Il s'agit des registres bloc-notes, des registres sonnette et des messages battement de cœur.

Registres bloc-notes (*Scratchpad Registers*)
Ces registres sont accessibles en lecture et en écriture des deux côtés du pont non transparent. Ils servent à transférer des informations de contrôle entre les deux domaines.

Registres sonnette (*Doorbell Registers*)
Ces registres servent à envoyer des interruptions d'un domaine à l'autre.

Messages battement de cœur (*Heartbeat Messages*)
Des messages battement de cœur sont envoyés périodiquement entre les deux domaines. Ces messages donnent une indication sur l'état du domaine voisin. Si un hôte cesse de recevoir ces messages, c'est qu'il y a une défaillance dans l'autre domaine.

De nos jours, les plateformes multiprocesseurs sont de plus en plus présentes. Grâce au pont non transparent, la technologie PCI Express peut maintenant faire partie de ces plateformes. On retrouve le pont non transparent au niveau des composants intelligents et des systèmes multi-hôtes.

Un composant intelligent est une carte PCI Express munie de son propre processeur et de son Root Complex. À titre d'exemple, certains contrôleurs RAID de disques durs sont des composants intelligents. Les systèmes multi-hôtes, quant à eux, sont de trois types : multiprocesseur, redondant et de protection (Budruk, Anderson et Shanley, 2004).

Configuration multiprocesseur

Plusieurs processeurs travaillent en parallèle ou se partagent les tâches.

Configuration redondante

Des systèmes d'architecture identique exécutent les mêmes tâches dans le but d'augmenter la tolérance aux fautes.

Configuration de protection

Un système en veille surveille l'état du système principal et prend la relève en cas de défaillance.

En avionique, la fiabilité est un élément essentiel pour un bus de données. Pour certaines configurations, un pont non transparent PCI Express représente un élément de fiabilité. En effet, les configurations redondantes et de protection augmentent la fiabilité de ce protocole déjà fiable par ses mécanismes spécifiques (voir CHAPITRE 4).

2.2.5 Configuration

Pour connaître la configuration du réseau, le logiciel de configuration doit effectuer un balayage du système. Ce processus s'appelle l'énumération. Pendant l'énumération, le logiciel assigne des numéros aux bus, aux composants et aux fonctions. Les numéros assignés sont sauvegardés dans les registres de configuration des composants.

Bus

Il existe trois types de bus dans un système PCI Express :

- bus primaire : bus directement relié au port ascendant d'un pont PCI-PCI,
- bus secondaire : bus directement relié au port descendant d'un pont PCI-PCI,
- bus subordonné : dernier bus accessible par un port descendant.

Composant

Parmi les composants, on retrouve les Endpoints, les ponts PCI Express / PCI et les ponts à l'intérieur des commutateurs et du Root Complex. Un composant relié à un port descendant externe porte toujours le numéro 0 (*Device 0*).

Fonction

Tout composant possède une ou plusieurs fonctions. Un composant PCI Express peut posséder jusqu'à huit fonctions, numérotées de 0 à 7. Par exemple, la fonction d'une carte Ethernet est d'effectuer des échanges de données sur un réseau. Pour une carte graphique, la fonction est d'afficher des images sur un moniteur. Les cartes graphiques qui ont deux sorties possèdent deux fonctions. Chacune de ces fonctions gère son propre affichage.

Un exemple de configuration est illustré par la Figure 2.5. Le numéro des bus, des composants et des fonctions ont été assignés par le logiciel de configuration pendant le processus d'énumération.

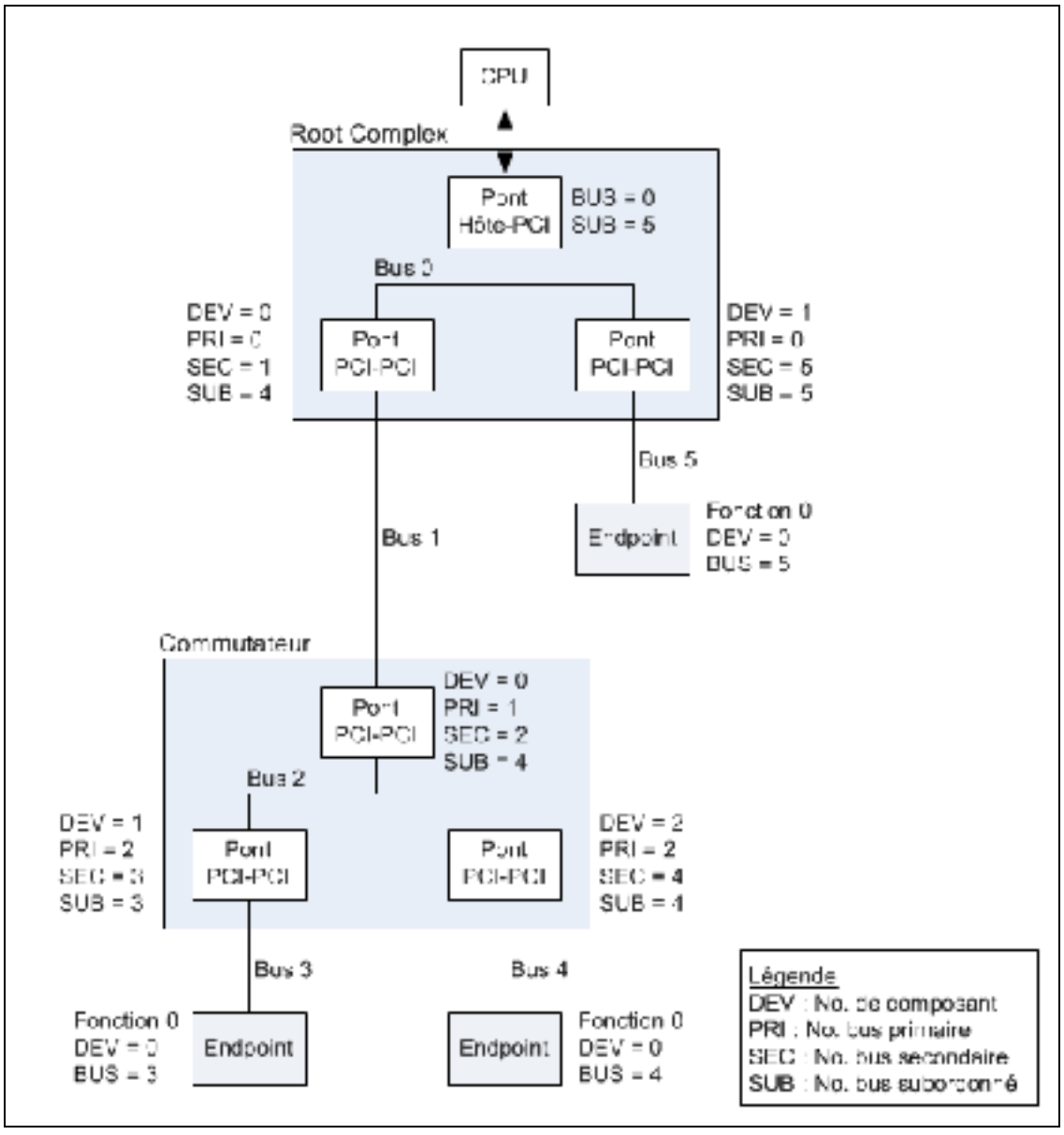

Figure 2.5 Exemple de configuration PCI Express

Comme nous pouvons le remarquer sur cette figure, seuls les Endpoints possèdent un numéro de fonction. Les numéros de composant et de bus sont assignés à tous les composants, c'est-à-dire aux Endpoints et aux ponts internes du Root Complex et des commutateurs.

2.2.6 Connexions physiques

La technologie PCI Express offre différents types de connexions physiques :

- circuit intégré à circuit intégré (*IC-to-IC*),
- carte à carte (*Board-to-Board*),
- par câble (*PCI Express over cable*).

Circuit intégré à circuit intégré (*IC-to-IC*)

De plus en plus, le PCI Express est utilisé pour la communication entre les circuits intégrés (Coupé, 2011). Plusieurs composants sur un circuit imprimé peuvent ainsi bénéficier des avantages de ce protocole à haut débit. Comme il a été mentionné dans l'introduction, Intel utilise ce protocole pour sa nouvelle architecture Atom.

Carte à carte (*Board-to-Board*)

Ce type d'interconnexion est le plus commun. C'est celui que l'on retrouve dans les ordinateurs. Les cartes périphériques sont reliées à la carte mère via ce type d'interconnexion.

Par câble (*PCI Express over cable*)

Il est possible d'utiliser la technologie PCI Express pour relier entre eux différents appareils. En effet, en 2007, le PCI-SIG (*PCI Special Interest Group*) a approuvé le câblage externe par PCI Express. Ainsi, on retrouve le PCI Express dans des applications tels la vidéo à haute vitesse, le stockage externe et les systèmes d'imagerie médicale (Cooper, 2008).

2.3 Le protocole PCI Express

L'architecture d'un réseau PCI Express a été présentée dans la section précédente. Voici maintenant les détails concernant le protocole (PCI-SIG, 2006). Les sujets qui sont abordés sont les suivants :

- modèle en couches,
- catégories de transaction,
- registres de configuration,
- canaux virtuels et catégories de trafic,
- méthodes de routage,
- bande passante.

2.3.1 Modèle en couches

Le protocole PCI Express est construit sur un modèle en couches. Il s'agit des couches transaction (*Transaction Layer*), liaison (*Data Link Layer*) et physique (*Physical Layer*). Chacune de ces couches est divisée en deux sections, soient les sections transmission (*Tx*) et réception (*Rx*). La Figure 2.6 représente le modèle en couches de deux composants reliés entre eux par un lien PCI Express.

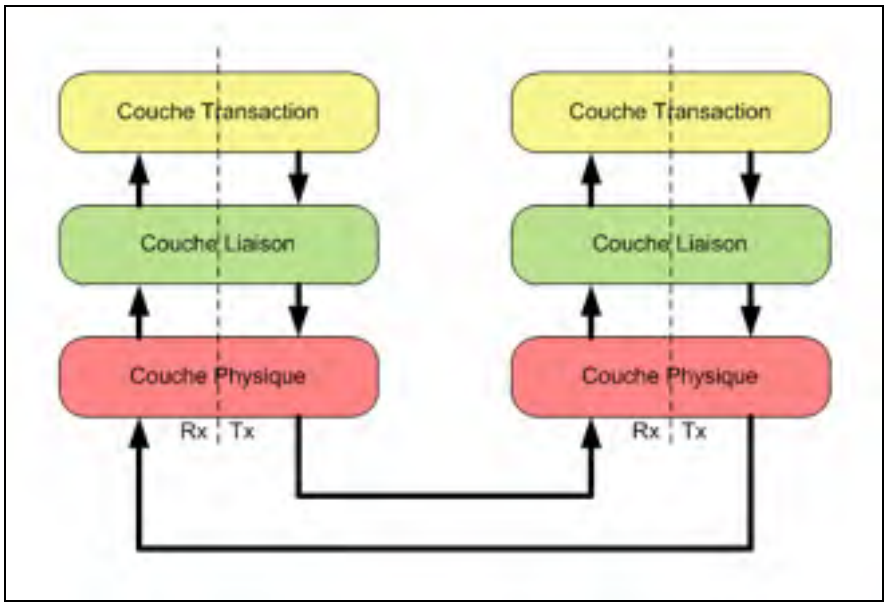

Figure 2.6 Modèle en couches du protocole PCI Express

Couche transaction

Sur un réseau PCI Express, les transactions sont effectuées via des paquets que l'on nomme TLPs (*Transaction Layer Packets*). C'est la couche transaction qui est responsable de l'assemblage et du désassemblage des TLPs. Les TLPs sont principalement utilisés pour des transactions de lecture et écriture.

Couche liaison

La couche liaison est située entre la couche transaction et la couche physique. Les principales tâches de cette couche sont la gestion du lien et la vérification de l'intégrité des données. La gestion du lien s'effectue par des paquets nommés DLLPs (*Data Link Layer Packets*). Ces paquets sont produits à l'intérieur même de la couche liaison.

Les DLLPs sont utilisés pour :

- la gestion de l'énergie,
- l'échange d'information du contrôle de flux,
- les accusés de réception des TLPs (protocole ACK/NAK).

Contrairement aux TLPs, les DLLPs ne passent pas au travers des commutateurs. En effet, les DLLPs sont utilisés pour des liaisons de point à point entre deux composants et non pour des liaisons de bout en bout. La Figure 2.7 illustre la différence entre les deux types de liaison.

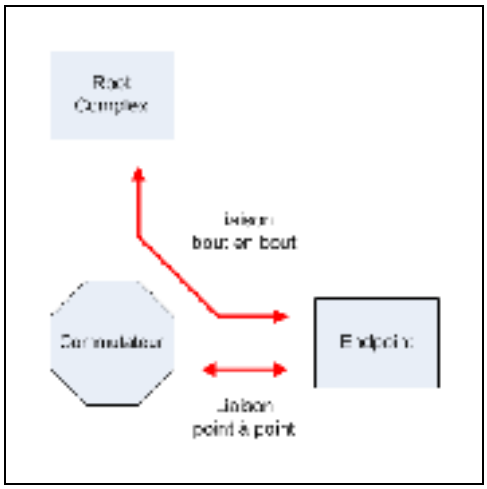

Figure 2.7 Liaison point à point VS bout en bout

Couche physique

La couche physique est responsable de l'envoi des données sur le réseau. Dans un premier temps, la couche physique encadre les TLPs et les DLLPs provenant de la couche liaison. Il s'agit de symboles identifiant le début et la fin du paquet. Ensuite, le paquet est encodé dans le format 8b/10b défini par le standard ANSI X3.230-1994 (Budruk, Anderson et Shanley, 2004; Krig, 2003). Le but premier de cet encodage est de créer suffisamment de transitions

de bits pour faciliter la récupération de l'horloge à la réception. Les paquets envoyés par la couche physique se nomment PLPs (*Physical Layer Packets*).

2.3.2 Catégories de transaction

Dans un réseau PCI Express, les transactions (TLPs) se regroupent en quatre catégories :

- mémoire : lecture et écriture de données en mémoire,
- entrée/sortie : lecture et écriture de données aux entrées/sorties,
- configuration : configuration des composants du réseau,
- message : messagerie et signalement d'événements.

Une transaction est définie comme étant une transmission de paquets entre un demandeur (*Requester*) et un répondeur (*Completer*). Les transactions sont de deux types, soit posté et non-posté. Pour une transaction non-postée, le répondeur doit retourner un paquet de réponses au demandeur. Une transaction postée, quant à elle, ne requiert aucun paquet de réponses de la part du répondeur. Le Tableau 2.1 identifie les catégories de transaction et leur type.

Tableau 2.1 Types de transaction

Catégorie de transaction	**Type**
Mémoire – lecture	Non-postée
Mémoire – écriture	Postée
Mémoire – lecture verrouillée	Non-postée
Entrée/Sortie – lecture	Non-postée
Entrée/Sortie – écriture	Non-postée
Configuration – lecture	Non-postée
Configuration – écriture	Non-postée
Message	Postée

2.3.3 Registres de configuration

Dans un système PCI Express, chacune des fonctions possède un ensemble de registres de configuration qui occupe 4096 octets. Les 256 premiers octets de cet espace sont utilisés pour la compatibilité avec les systèmes PCI. L'espace restant est réservé aux registres de configuration propres au protocole PCI Express. Un composant qui utilise seulement les 256 premiers octets de l'espace de configuration est un composant PCI Express hérité (*Legacy*).

Seul le Root Complex peut accéder à ces registres grâce aux transactions de configuration. Les registres de configuration ont plusieurs fonctions dont voici les principales :

- la détection des composants PCI Express,
- l'identification des fonctions des composants,
- l'assignation des ressources système aux composants.

C'est dans l'espace de configuration que se trouvent les registres d'adresse de base, ou BARs (*Base Address Registers*). Ces registres sont utilisés pour allouer à chaque fonction des plages d'adresses mémoire et des plages d'adresses d'entrée/sortie. Ce sont des registres de 32 bits, mais ils peuvent être agencés en groupe de deux si l'on désire utiliser des adresses de 64 bits.

Le protocole PCI Express définit deux types de configuration, nommés type 0 et type 1. Un Endpoint a un espace de configuration de type 0. Le type 1 fait référence aux autres composants PCI Express, c'est-à-dire le Root Complex, le commutateur et le pont.

2.3.4 Canaux virtuels et catégories de trafic

La qualité de service (QoS) permet une différenciation des services offerts aux applications. Rappelons que la qualité de service est un élément essentiel pour un bus de données avionique. Dans un réseau de communication, les différentes applications n'ont pas toutes les mêmes besoins en termes de performance. Pour répondre aux besoins des applications, le

protocole PCI Express fournit un mécanisme de canaux virtuels associé à des catégories de trafic.

Les catégories de trafic (TCs : *Traffic Classes*) permettent une différenciation des services. Ainsi, il est possible de prioriser certains flux de données par rapport à d'autres. Prenons, par exemple, deux types de transfert : des données provenant d'Internet et des données provenant d'une caméra vidéo. Pour préserver l'aspect temps réel de la vidéo, les données provenant de la caméra doivent être priorisées par rapport aux données de l'Internet. Les catégories de trafic procurent ainsi une qualité de service en fonction des besoins des applications. Tout lien doit avoir au moins une catégorie de trafic (TC0). Cependant, un lien peut posséder jusqu'à huit catégories de trafic (TC0 à TC7).

Les canaux virtuels (VCs : *Virtual Channels*) représentent des chemins indépendants au sein d'un même lien. Ceux-ci fournissent une indépendance au niveau des ressources telles les files d'attente, les tampons et le contrôle de flux. Tout lien doit posséder au moins un canal virtuel (VC0). Cependant, il est possible pour un lien de posséder jusqu'à huit canaux virtuels (VC0 à VC7). Lorsque plusieurs canaux virtuels empruntent le même lien, ceux-ci sont multiplexés. Il faut préciser qu'il n'y a aucune relation entre le nombre de canaux virtuels et le nombre de voies d'un lien. Par exemple, un lien à une voie pourrait posséder quatre canaux virtuels alors qu'un lien à quatre voies pourrait posséder un seul canal virtuel.

Toutes les catégories de trafic doivent être associées à un canal virtuel. TC0 est toujours associé à VC0. Pour ce qui est des autres associations, le protocole permet beaucoup de flexibilité. Il y a cependant certaines règles à respecter :

- plusieurs catégories de trafic peuvent être associées à un seul canal virtuel,
- une catégorie de trafic ne peut être associée à plus d'un canal virtuel,
- l'association TCs/VCs doit être identique des deux côtés d'un lien.

La Figure 2.8 illustre un exemple d'association TCs/VCs. Le port descendant du commutateur possède un seul canal virtuel (VC0), alors que le port ascendant en contient deux (VC0 et VC1). Ici, les catégories de trafic 0 à 4 sont utilisées. On peut remarquer que TC4 emprunte un canal virtuel différent entre le commutateur et le Root Complex.

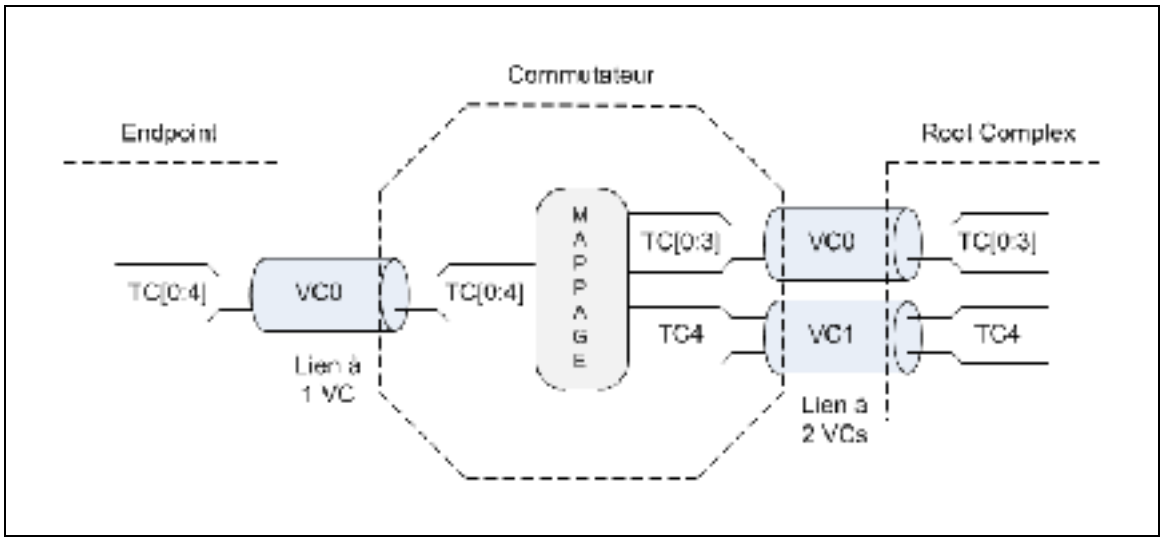

Figure 2.8 Association TCs/VCs

Pour toute transaction, la catégorie de trafic est identifiée dans l'en-tête du TLP. Par contre, un TLP ne contient aucune indication sur le canal virtuel associé. L'association TCs/VCs est connue au niveau des ports seulement.

2.3.5 Méthodes de routage

Le routage est un mécanisme permettant d'acheminer les paquets de données vers le bon destinataire. Il existe trois méthodes de routage :

- routage par adresse,
- routage par identifiant (ID),
- routage implicite.

Routage par adresse

Le routage par adresse est utilisé pour les transactions mémoires et les transactions d'entrée/sortie. Pour ce type de routage, un composant compare ses registres avec l'adresse de destination contenue dans l'en-tête des TLPs.

Routage par identifiant (ID)

Le routage par identifiant est utilisé pour les transactions de configuration, pour certaines transactions de messages ainsi que pour les réponses (*Completion*). Ce type de routage utilise les numéros de bus, de composants et de fonctions pour spécifier la destination des TLPs.

Routage implicite

Le routage implicite est utilisé pour les transactions ne contenant aucune indication d'adresse ou d'identifiant. Il s'agit des transactions de message suivantes :

- routé vers le Root Complex,
- diffusion générale du Root Complex (*Broadcast*),
- local (se termine au récepteur),
- recueilli et routé vers le Root Complex,
- réservé (se termine au récepteur).

2.3.6 Bande passante

Le protocole PCI Express est maintenant rendu à sa troisième génération. La première génération spécifie un taux de transfert de 2.5 Gbits/s. La deuxième génération double ce taux à 5 Gbits/s.

Les deux premières générations utilisent un encodage 8b/10b. Avec ce type d'encodage, 10 bits sont nécessaires pour encoder un octet. Par conséquent, 20 % de la bande passante est

utilisée par l'encodage. La bande passante effective est donc de 2 Gbits/s pour la première génération et de 4 Gbits/s pour la seconde génération.

En ce qui concerne la troisième génération du protocole, l'objectif était de doubler la bande passante de la génération précédente. Cependant, pour des raisons d'intégrité du signal, le taux de transfert de la troisième génération est de 8 Gbits/s, soit un gain de 60 % par rapport à la seconde génération. Par contre, la troisième génération du protocole utilise un encodage différent, soit l'encodage 128b/130b. Puisque 130 bits sont requis pour encoder 128 bits de données, cet encodage utilise à peine 1.5% de la bande passante. Par conséquent, la bande passante effective est de 7.9 Gbits/s, c'est-à-dire presque le double de la génération précédente (Kazmi, 2009).

2.4 Systèmes avioniques

Après avoir vu la théorie concernant la technologie PCI Express, voici maintenant un survol des systèmes avioniques. Cette section présente les architectures des systèmes avioniques et la certification de ces systèmes. Il s'agit d'une brève introduction sur les systèmes avioniques dans le but de mettre en contexte les bus de données avioniques présentés au chapitre suivant.

L'avionique est l'ensemble des équipements et systèmes informatiques et électroniques qu'on retrouve à bord d'un avion. Ceci comprend les capteurs et actionneurs, les ordinateurs et logiciels associés ainsi que les bus de données assurant la communication entre tous ces éléments.

2.4.1 Architecture des systèmes avioniques

Il existe différentes architectures des systèmes avioniques. Dans les architectures classiques, chacun des systèmes électroniques est dédié à une tâche spécifique. Par conséquent, la partie logicielle est fortement liée à la partie matérielle qu'elle commande. On parle alors d'une architecture fédérée.

Dans les architectures modernes, on assiste à une dissociation des parties logicielles et matérielles. On y retrouve les mêmes fonctions que sur les architectures classiques. Cependant, les ressources sont partagées sur une nouvelle plateforme matérielle de type modulaire. Il s'agit de l'architecture IMA (*Integrated Modular Avionics*). Voici une description de ces deux types d'architecture.

Système avionique fédéré

Dans l'architecture classique des systèmes avioniques, chaque fonction de l'appareil est exécutée par son propre matériel et logiciel associé. Toutes les ressources d'une fonction sont ainsi regroupées dans un seul module physique que l'on nomme LRU (*Line Replaceable*

Unit) (Watkins et Walter, 2007). De cette façon, il n'y a aucun partage de ressources entre les différentes fonctions. Ceci procure l'avantage de créer une barrière contre la propagation d'erreurs. Ainsi, une défaillance locale n'influencera pas le comportement des autres fonctions du système (Wolfig et Jakovljevic, 2008). Un exemple d'architecture fédérée est illustré par la Figure 2.9.

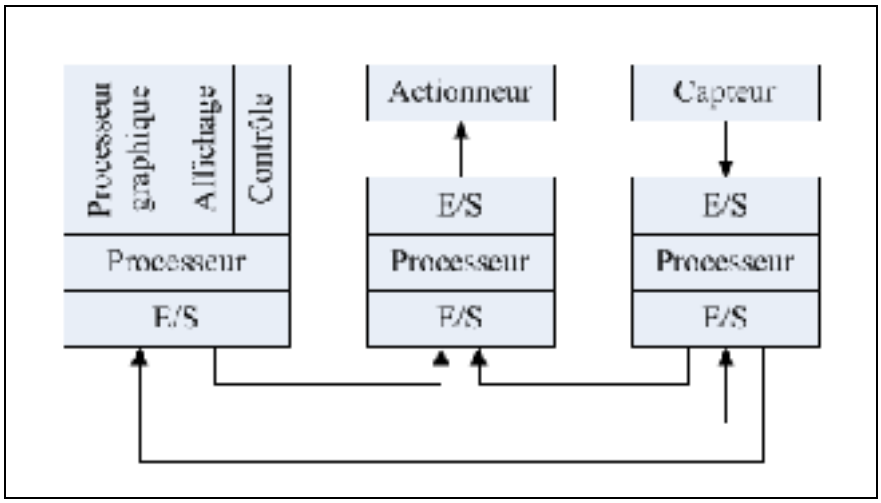

Figure 2.9 Exemple d'architecture fédérée
Tirée de Watkins et Walter (2007) avec la permission de Christopher Watkins

Bien qu'un LRU soit facilement remplaçable en cas de défaillance, une architecture fédérée comporte certains désavantages. En effet, étant donné l'absence d'interopérabilité entre les différentes fonctions, cette architecture souffre d'un manque de flexibilité. Chaque LRU a été conçu pour répondre à un besoin très spécifique et ne peut être interchangé avec un autre.

La mise à jour des fonctions représente un autre désavantage important. Puisque la durée de vie moyenne d'un appareil est de 30 ans, le matériel et les logiciels doivent s'adapter aux avancés technologiques au fil du temps (Wolfig et Jakovljevic, 2008). Le moindre changement apporté à une fonction a pour conséquence d'entrer à nouveau dans le processus de certification. Ainsi, même si un appareil est en fonction depuis plusieurs années, des coûts énormes sont associés à la mise à jour des systèmes en place.

Système avionique modulaire intégré

Pour moderniser les architectures fédérées, un nouveau concept a été introduit au milieu des années 90. Il s'agit de l'avionique modulaire intégrée (IMA : *Integrated Modular Avionics*) (Lopez et al., 2008). La norme DO-297 définit l'IMA comme suit :

« *IMA est un ensemble matériel et logiciel flexible, réutilisable et interopérable, qui, lorsqu'intégré, forme une plateforme qui fournit à des applications avioniques des services conçus et vérifiés selon des exigences de sécurité et de performance* » (RTCA, 2005).

L'architecture physique de l'IMA est décrite par la norme ARINC 651 (Bluff, 1999). Les unités LRU de l'architecture fédérée ont été remplacées par des modules génériques communs appelés LRM (*Line Replaceable Module*).

Dans cette architecture, les processeurs sont partagés de même que l'infrastructure associée (système d'alimentation, mécanismes de refroidissement, etc.). Les canaux de communication dédiée de l'architecture fédérée ont été remplacés par un canal commun. Finalement, les entrées/sorties sont maintenant regroupées dans une interface commune (Watkins et Walter, 2007). La Figure 2.10 illustre un exemple typique de l'architecture IMA.

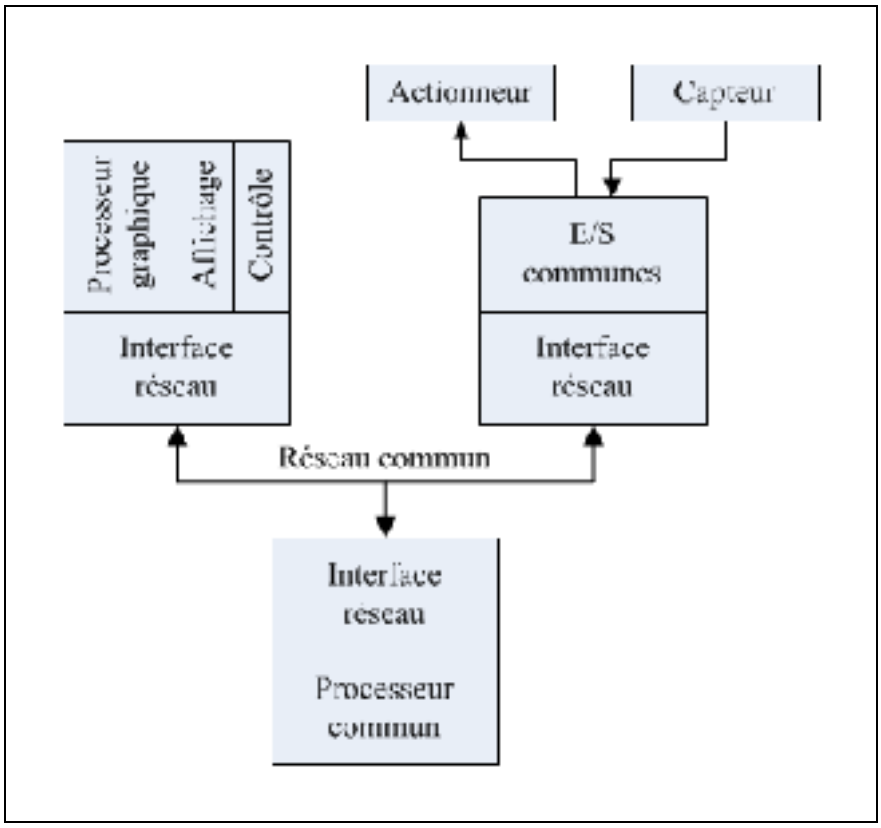

Figure 2.10 Exemple d'architecture IMA
Tirée de Watkins et Walter (2007) avec la permission de Christopher Watkins

L'architecture logicielle de l'IMA est décrite par la norme ARINC 653 (Prisaznuk, 2008). Puisque les ressources sont partagées entre les applications, un partitionnement robuste est nécessaire au niveau logiciel. Le partitionnement est de deux types, soit spatial et temporel (Laarouchi, 2009). Le partitionnement spatial concerne les ressources matérielles, par exemple la gestion de la mémoire. Le partitionnement temporel, quant à lui, fait référence aux ressources temporelles gérées par le processeur, comme l'ordonnancement des tâches.

L'architecture IMA comporte plusieurs avantages (Grieu, 2004).

Interopérabilité

Si un module subit une défaillance, il est toujours possible de reconfigurer une application pour qu'elle s'exécute sur un autre module.

Réutilisation

L'architecture IMA supporte la réutilisation des modules. Ceci a un impact direct sur les coûts de développements des systèmes avioniques.

Adaptabilité

La modularité de cette architecture permet de reconfigurer un appareil pour l'adapter à une mission en particulier.

Maintenance

La standardisation du matériel électronique simplifie la maintenance.

Réduction de la masse et du volume et économie d'énergie

Puisque les ressources matérielles sont partagées, moins de matériel électronique est nécessaire pour exécuter toutes les fonctions de l'appareil. Ceci se traduit par une diminution de la masse et du volume du matériel embarqué. De plus, le partage des ressources matérielles implique nécessairement une économie d'énergie. Dans le jargon aérospatial, ceci fait référence à la réduction SWaP (*Size, Weight and Power*) (Wilson et Preyssler, 2009).

2.4.2 Certification des systèmes avioniques

Un système avionique qui ne fonctionne pas correctement peut occasionner des dommages matériels importants et des pertes de vies. Par conséquent, il est essentiel que l'industrie avionique soit normalisée (Kornecki et Zalewski, 2008). Il existe plusieurs normes en avionique. Cependant, deux des plus importantes concernent la conception des produits logiciel et matériel. Les normes DO-178B et DO-254 font référence aux parties logicielle et matérielle respectivement. Celles-ci définissent les objectifs qu'un produit doit rencontrer pour obtenir la certification.

Norme DO-178B

La norme DO-178B (*Software Considerations in Airborne Systems and Equipment Certification*) a été développée par l'industrie avionique commerciale et le RTCA (*Radio Technical Commission for Aeronautics*) pour permettre aux développeurs de logiciels embarqués d'obtenir la certification (Kornecki, Butka et Zalewski, 2008). Les objectifs définis par cette norme concernent les conditions de sécurité applicables aux logiciels des systèmes critiques.

La norme spécifie différents modes de défaillance et les classe selon cinq niveaux suivant une échelle de sévérité. Le niveau A est le plus critique. Ce dernier est identifié comme étant catastrophique, car il entraîne la destruction de l'appareil et la perte de vies humaines. Le niveau E, quant à lui, n'a aucun effet notable sur l'appareil et les personnes à bord.

Le Tableau 2.2 présente les cinq niveaux de criticité énoncés par la norme.

Tableau 2.2 Niveaux de criticité DO-178B
Tiré de Kornecki, Butka et Zalewski (2008)

Niveau	**Description**
A	Catastrophique : Destruction de l'appareil et pertes de vies humaines.
B	Sévère : Dommages matériels pouvant causer des pertes de vies humaines.
C	Majeur : Dommages matériels pouvant causer des blessures graves.
D	Mineur : Dommages matériels pouvant causer des blessures mineures.
E	Sans effet : Aucun effet sur l'appareil et les personnes à bord.

Norme DO-254

La norme DO-254 (*Design Assurance Guidance for Airborne Electronic Hardware*) concerne la conception du matériel électronique complexe pour les systèmes avioniques. Ce matériel inclut les composants personnalisables suivants :

- PLD (Programmable Logic Device),
- FPGA (Field Programmable Gate Array),
- ASIC (Application Specific Integrated Circuit).

Cette norme ne donne aucun détail sur la façon de concevoir le matériel. Elle donne des recommandations pour que le design atteigne les objectifs prescrits par les autorités de certification (Gagea et Rajkovic, 2008).

De plus, la norme suppose que les parties matérielle et logicielle fonctionnent de façon complémentaire. D'ailleurs, on y retrouve les mêmes niveaux de criticité que ceux de la norme DO-178B (Tableau 2.2).

2.5 Étude du réseau AFDX

Il existe plusieurs standards concernant les réseaux avioniques, mais deux d'entre eux ont dominé le marché. Il s'agit des normes ARINC 429 pour l'industrie commerciale et MIL-STD-1553 pour l'industrie militaire (Alena et al., 2007). Les avancées technologiques des dernières années ont poussé les concepteurs à développer un nouveau standard plus performant pour répondre aux besoins croissants des systèmes avioniques. Ce nouveau standard s'appelle AFDX (*Avionics Full-Duplex Switched Ethernet*). L'AFDX définit une version déterministe et fiabilisée du réseau Ethernet (IEEE 802.3) pour des applications temps-réel (Schuster et Verma, 2008). L'AFDX a vu le jour grâce aux initiatives de l'avionneur Airbus pour équiper son nouvel appareil A380. Par la suite, Boeing a adopté ce standard pour ses nouveaux appareils, dont le 787 Dreamliner.

2.5.1 Historique

Étant une technologie mature, le réseau Ethernet représentait un choix attrayant pour l'industrie avionique. En effet, l'utilisation d'un produit sur étagère (COTS : *commercial off-the-shelf*) permet de réduire les coûts et le temps de développement (Actel Corporation, 2005). Cependant, ce réseau ne pouvait être utilisé dans sa version d'origine. Des ajustements s'imposaient pour respecter les contraintes relatives au déterminisme et à la fiabilité des réseaux avioniques. Les modifications apportées au réseau Ethernet ont conduit à la création de l'AFDX dont la description se retrouve dans la norme ARINC 664.

Un réseau AFDX est composé de commutateurs et d'équipements terminaux appelés *End Systems*. Le réseau possède une topologie en étoile et chacun des commutateurs peut être relié à un maximum de 24 *End Systems*. Cependant, il est possible de mettre en cascade

plusieurs commutateurs pour construire de plus grands réseaux. Au niveau de la performance, le débit sur un réseau AFDX peut atteindre 1 Gb/s. Grâce à ses caractéristiques, l'AFDX s'adapte parfaitement au concept des systèmes IMA.

La Figure 2.11 illustre un exemple typique de réseau AFDX. Deux des ordinateurs possèdent un sous-système avionique pouvant représenter, par exemple, un système de contrôle de vol ou un système de positionnement global (GPS). Quant au troisième ordinateur, il fait office de passerelle vers le monde extérieur via le réseau Internet.

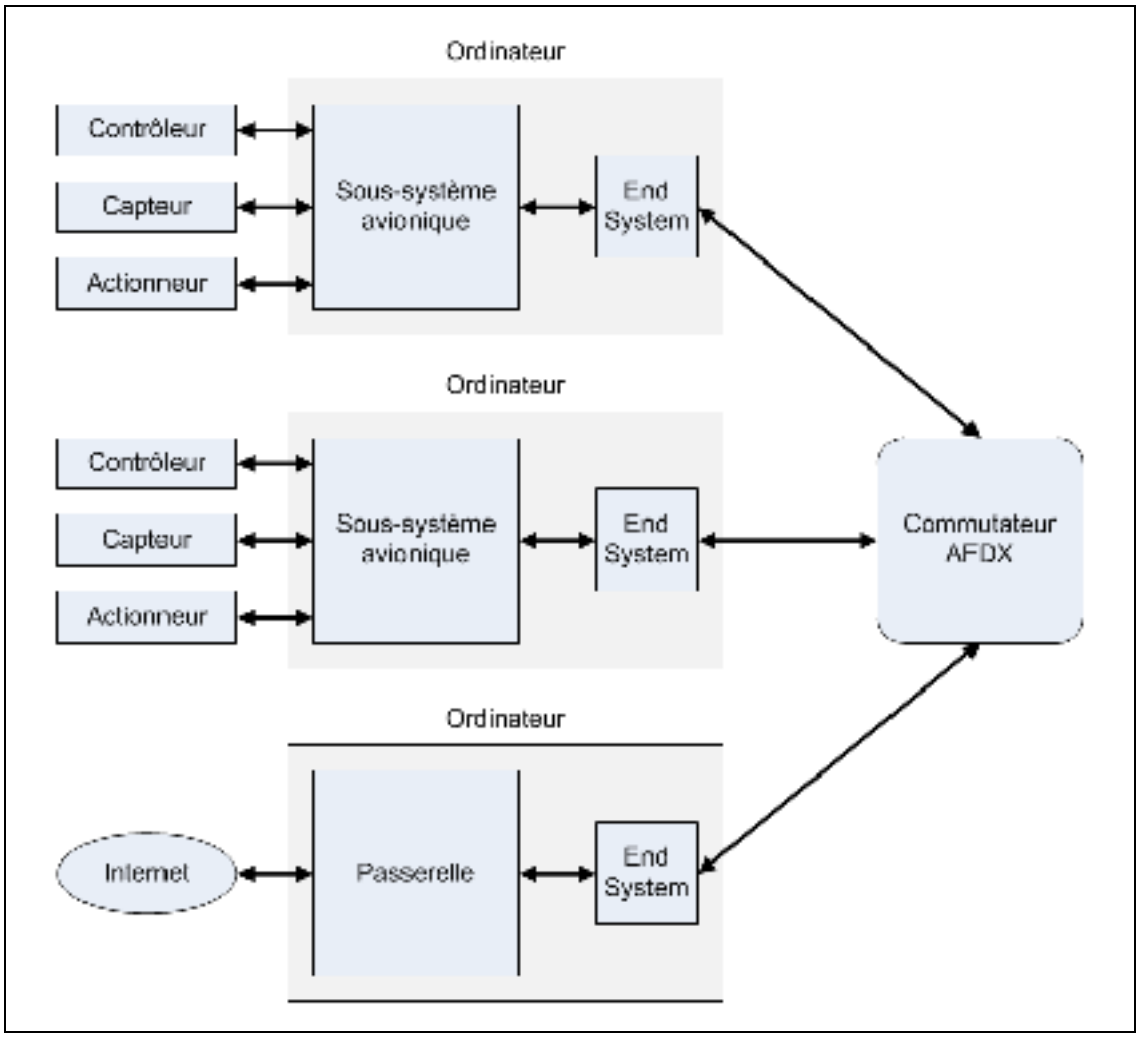

Figure 2.11 Exemple de réseau AFDX

2.5.2 Modification du réseau Ethernet

Dans un réseau Ethernet classique, la méthode d'accès au support physique représente la principale source d'indéterminisme (Condor Engineering Inc, 2005). En effet, puisque le support est partagé, les accès doivent être contrôlés et les collisions doivent être détectées. Cette gestion est assurée par le protocole CSMA/CD (*Carrier Sense Multiple Access / Collision Detection*). Avant d'émettre, une station vérifie toujours si le support est libre. S'il n'y a pas de transmission en cours, la station peut émettre son message. Mais lorsque deux

stations émettent simultanément, il se produit une collision. Dans ce cas, les stations doivent attendre un délai aléatoire avant de retransmettre leur message. Ce type de protocole est dit probabiliste, car le temps d'accès au support demeure imprévisible.

Pour empêcher les accès simultanés au support, la solution a été de relier chaque équipement directement au commutateur. Cette configuration représente un réseau Ethernet commuté. Dans ce type de réseau, les seules collisions possibles se retrouvent au niveau des liaisons point à point, c'est-à-dire entre l'équipement et le commutateur. Pour éviter ces collisions, la solution a été d'utiliser des liens bidirectionnels. En utilisant un canal différent pour la transmission et la réception, les collisions sont éliminées et le protocole CSMA/CD n'est plus nécessaire. L'AFDX représente ainsi un réseau Ethernet commuté *Full Duplex*.

2.5.3 Éléments de déterminisme et de fiabilité

Pour obtenir la certification, un réseau avionique doit prouver son déterminisme et sa fiabilité. Pour satisfaire aux exigences relatives au déterminisme, des mécanismes spécifiques à l'AFDX ont été créés. Grâce à ces mécanismes, la bande passante est garantie, la latence maximale de bout en bout est connue et la gigue est connue. Quant à la fiabilité du réseau, elle est obtenue par sa redondance physique.

Bande passante

Dans un réseau AFDX, le contrôle du débit est assuré par les liens virtuels (VL : *Virtual Link*) (Grieu, 2004). Sur un lien virtuel, la bande passante est contrôlée grâce à un mécanisme que l'on appelle BAG : *Bandwidth Allocation Gap*. Le BAG constitue le temps minimum entre chaque trame de données consécutive. Le concept du BAG est illustré par la Figure 2.12.

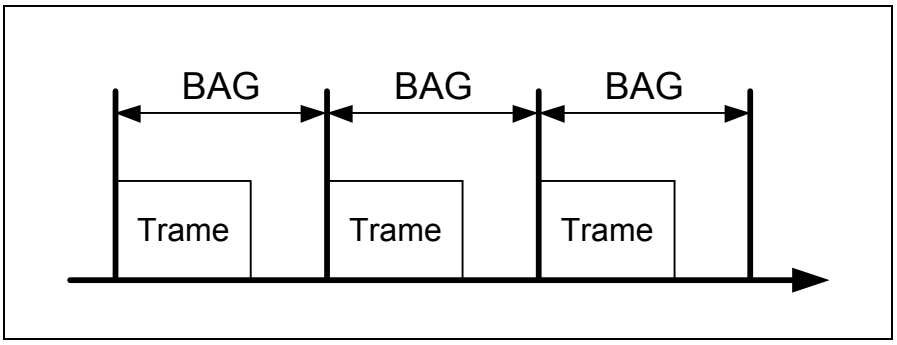

Figure 2.12 Mécanisme du BAG

Entre l'envoi de chacune des trames de données, il existe un temps fixe. Cette valeur temporelle est déterminée par l'intégrateur de systèmes pour combler les besoins des applications et des équipements (Actel Corporation, 2005). Ainsi, pour un lien virtuel en particulier, le BAG peut prendre une valeur dans un intervalle de 1 à 128 ms (BAG = 1 ms x 2^k, où k = 0 à 7). Grâce à ce mécanisme, la bande passante est garantie.

Latence maximale de bout en bout

La latence de bout en bout dépend du délai de transmission des liens et de la latence des commutateurs traversés (Bauer, Scharbarg et Fraboul, 2010). Par conséquent, la latence maximale dépend de la taille du réseau. Il faut noter que les spécifications ne précisent pas de maximum pour la latence du système. Tout fournisseur est tenu de préciser une limite supérieure de latence pour tous les équipements du réseau. Ainsi, pour un réseau AFDX en particulier, la latence maximale de bout en bout est connue (Actel Corporation, 2005).

Gigue

Le traitement des trames de données par un commutateur peut introduire une gigue sur le canal de communication. La gigue représente l'intervalle de temps qui sépare le commencement du BAG et l'envoie du premier bit. Pour assurer le déterminisme du système, cette gigue doit être connue et ne doit jamais dépasser 500 µs (Actel Corporation, 2005). La Figure 2.13 illustre le phénomène de la gigue.

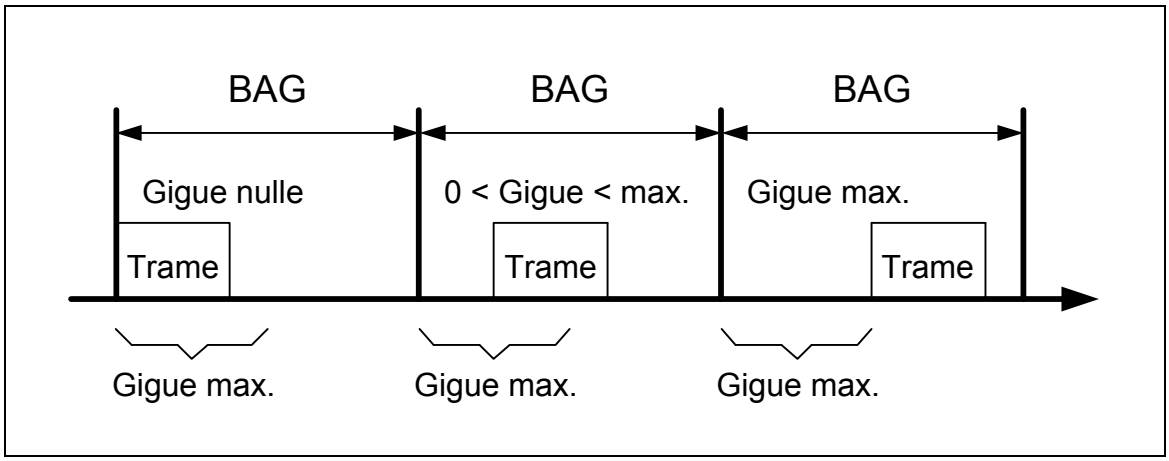

Figure 2.13 Gigue sur un réseau AFDX

Redondance

Un réseau AFDX est physiquement redondant. Dans le but d'augmenter la fiabilité, les données sont envoyées sur deux canaux différents (Actel Corporation, 2005). La Figure 2.14 représente un réseau AFDX. Tous les *End Systems* sont connectés aux deux réseaux. Ainsi, en cas de défaillance d'un réseau, les données peuvent quand même se rendre à destination. Cette redondance prévient aussi les erreurs de transmission.

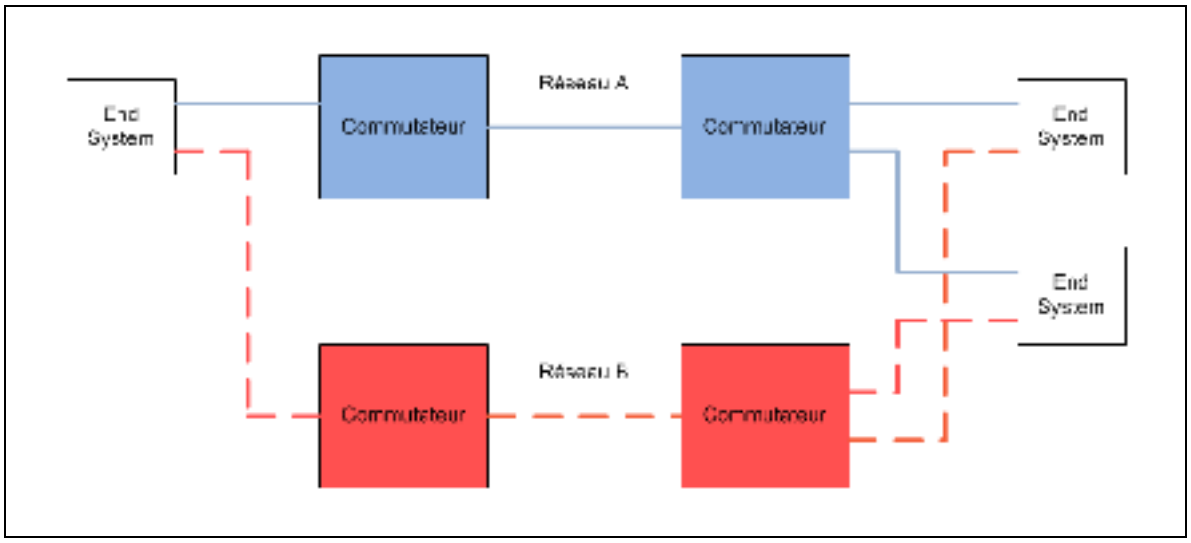

Figure 2.14 Redondance du réseau AFDX

2.5.4 Certification de l'AFDX

Un réseau commuté *Full Duplex* possède l'avantage d'éliminer l'indéterminisme causé par le temps d'accès au support physique. Cependant, un problème subsiste au niveau du commutateur. La latence à l'intérieur d'un commutateur est très variable dû à la confluence des flux asynchrones (Bauer, Scharbarg et Fraboul, 2010). Si trop de données se dirigent simultanément vers le même port de sortie, il peut se produire un débordement des files d'attente. En effet, les commutateurs AFDX sont conformes à la norme IEEE 802.1D, avec laquelle il est possible de perdre des trames par congestion. Par conséquent, des précautions doivent être prises pour éviter tous problèmes de congestion.

Le réseau AFDX possède des mécanismes spécifiques pour assurer le déterminisme (Bauer, Scharbarg et Fraboul, 2010). Cependant, il doit être prouvé que la latence de bout en bout et la gigue possède une valeur finie et bornée. Cette preuve est apportée par l'analyse du trafic dans le réseau. Pour évaluer les délais dans un réseau, deux types d'approches existent : stochastique et déterministe (Martin, 2004).

Approches stochastiques

En étudiant le comportement général du réseau, cette approche fournit des délais de bout en bout probabilistes.

Approches déterministes

En étudiant le comportement du réseau par une analyse « pire cas », cette approche donne des temps de réponse « pire cas » de bout en bout.

Puisque les autorités de certification imposent de connaître le comportement du réseau dans les pires conditions d'utilisation, les approches déterministes sont utilisées. La plus connue de ces approches est celle du calcul réseau (*Network Calculus*) introduit par Cruz en 1991 (Grieu, 2004).

L'approche par calcul réseau permet de calculer les bornes sur les temps de traversée du réseau et sur les gigues. Le calcul réseau permet aussi de dimensionner les files d'attente dans le but d'éviter les débordements. Grâce à cette approche, il a été possible de prouver que le réseau AFDX répond bien aux exigences prescrites par les autorités de certification.

À titre d'exemple, prenons une file d'attente de types FIFO (*First In First Out*). Les premières données qui arrivent sont les premières à repartir. Quelle taille doit avoir cette file d'attente pour ne perdre aucune donnée ? Pour pouvoir répondre à cette question, il faut connaître à chaque instant le taux d'entrée et le taux de sortie des données dans la file. Dans ses travaux, Cruz propose une méthode pour calculer la borne maximale de la taille d'une file d'attente (Grieu, 2004). Pour y arriver, il borne les quantités de données qui entrent et ressortent de la file. Ainsi, grâce à la courbe d'arrivée et à la courbe de sortie des données, il est possible de déterminer le temps maximal qu'une donnée passera dans la file d'attente.

CHAPITRE 3

CARACTÉRISATION DES BUS DE COMMUNICATION

3.1 Survol

La revue de la littérature a présenté la technologie PCI Express, une introduction aux systèmes avioniques, ainsi que les détails concernant le réseau AFDX.

Ce chapitre est consacré aux bus de communication en lien avec le sujet de cette recherche. Dans un premier temps, les métriques sont présentées avec certaines précisions concernant le protocole PCI Express. Finalement, ce chapitre se termine par la caractérisation des bus de communication avionique.

3.2 Métriques

Pour caractériser la performance d'un réseau à commutation de paquets, trois paramètres doivent être définis : l'utilisation, l'efficacité et le débit (Agilent Technologies, 2006). Les autres métriques importantes sont la latence et la gigue.

3.2.1 Utilisation

Dans un réseau, les données sont transmises par des paquets. Cependant, il n'y a pas que des paquets de données qui transitent sur un réseau. À titre d'exemple, un accusé de réception est un type de paquets servant à confirmer réception des données. L'utilisation correspond au nombre de bits transmis associé aux paquets de données par rapport au nombre total de bits transmis.

$$Utilisation = \frac{Nombre\ de\ bits\ associé\ aux\ paquets\ de\ données}{Nombre\ total\ de\ bits} \tag{3.1}$$

Comme il a été mentionné à la section 2.3.1, les données du protocole PCI Express sont transmises via des paquets que l'on nomme TLPs : *Transaction Layer Packets*. D'autres types de paquets sont transmis par les autres couches. L'utilisation correspond au nombre de symboles transmis associé aux TLPs par rapport au nombre total de symboles transmis. Ici, le terme « symbole » fait référence au groupe de 10 bits issus du processus d'encodage 8b/10b. Le taux d'utilisation dépend de la façon dont le composant a été configuré. Par exemple, les DLLPs de la couche liaison sont utilisés principalement pour la mise à jour du nombre de crédits disponibles et pour les accusés de réception. La fréquence d'envoi de ce type de paquets dépend du design du composant et influence directement le taux d'utilisation.

3.2.2 Efficacité

En plus de la charge utile, c'est-à-dire les données, un paquet contient d'autres informations (en-tête, CRC, etc.). L'efficacité correspond à la taille de la charge utile par rapport à la taille du paquet.

$$Efficacité = \frac{Taille\ de\ la\ charge\ utile}{Taille\ du\ paquet} \tag{3.2}$$

3.2.3 Débit

Le débit correspond à la quantité d'information transmise par unité de temps. Dans la littérature, le débit est souvent remplacé par le terme « bande passante ».

$$Débit = \frac{Nombre\ de\ bits}{Seconde} \tag{3.3}$$

Pour le calcul du débit réel de données, nous devons tenir compte des paramètres d'utilisation et d'efficacité.

$$Débit\ réel = Débit \times Utilisation \times Efficacité \tag{3.4}$$

3.2.4 Latence

La latence est définie par le temps que prend un paquet de données pour aller d'un point à un autre du réseau. Bien qu'il existe différents types de latence, les plus importantes sont la latence de bout en bout et la latence aller-retour.

Latence de bout en bout (*end-to-end latency*)

Il s'agit du délai de transmission d'un paquet de données entre la source et la destination.

Latence aller-retour (*round-trip latency*)

Il s'agit du délai entre une requête et la réponse à cette requête. La latence aller-retour détermine la réactivité d'un système.

3.2.5 Gigue

Tel que décrit à la section 2.5.3, la gigue est définie comme étant le décalage d'un signal par rapport à une position temporelle idéale (Agilent Technologies, 2008). Puisque cette variation du signal peut causer des erreurs de transmission, la gigue représente une métrique importante pour un réseau de communication à haute vitesse. La gigue peut être de deux types, soit déterministe et aléatoire. La somme de ces deux types de gigue représente la gigue totale du signal.

Gigue déterministe

La gigue déterministe est bornée. Par conséquent, les limites de la variation de phase peuvent être mesurées. Les principales causes de la gigue déterministe sont les suivantes (Li, 2005) :

- interférence inter-symbole,
- diaphonie sur le circuit imprimé (*Crosstalk*),
- radiation électromagnétique,
- bruit provenant de la source d'alimentation,
- limitation de la bande passante de la ligne de transmission,
- impédance mal équilibrée.

Gigue aléatoire

La gigue aléatoire est non bornée et possède une distribution gaussienne. Dû à sa nature aléatoire, ce type de gigue est plus difficile à évaluer. Sur un circuit imprimé, les principales causes de la gigue aléatoire sont les suivantes (Li, 2005) :

- bruit de grenaille (*Shot Noise*),
- bruit de scintillement (*Flicker Noise*),
- bruit thermique (*Thermal Noise*).

Le bruit de grenaille est causé par la fluctuation du courant à l'intérieur des semi-conducteurs. Le bruit de scintillement, quant à lui, est dû à un phénomène de piégeage des porteurs entre la grille et le canal d'un transistor. Ce bruit est inversement proportionnel à la fréquence. Finalement, le bruit thermique est associé au flux d'électron dans les conducteurs. Ce bruit croît avec la bande passante et la température.

La gigue présente sur un signal peut causer des erreurs dans la transmission et ce taux d'erreurs peut être évalué. En effet, le taux d'erreur binaire (BER : *Bit Error Rate*) est une mesure indiquant la probabilité d'erreur lors du transfert d'un bit. Par exemple, un BER de

10^{-3} nous indique qu'il y a une chance sur 1000 qu'un bit soit interprété incorrectement à la réception.

Le taux d'erreur binaire est directement relié à la gigue aléatoire présente sur un signal (Coleman et al., 2004). Puisque la gigue aléatoire est non bornée, il n'est pas possible de concevoir un système à l'épreuve des erreurs. Par conséquent, un protocole à haute vitesse doit être tolérant à la gigue.

Les spécifications du protocole PCI Express imposent un BER total maximal de 10^{-12} (Coleman et al., 2004). Tout concepteur utilisant la technologie PCI Express doit s'assurer que le taux d'erreur binaire de son système ne dépasse pas cette valeur. Cette valeur de BER est conforme aux exigences de l'industrie avionique. En effet, pour les bus de données avioniques, le taux d'erreur binaire se situe dans la plage de 10^{-6} à 10^{-15} (Federal Aviation Administration, 2009). D'ailleurs, pour le réseau AFDX, la valeur de BER est de 10^{-12} (TechSAT, 2009).

3.3 Bus de données avioniques

Cette section caractérise les bus de données avioniques. Dans un premier temps, nous verrons les besoins des bus de données avioniques. Ensuite, les critères de sécurité seront exposés. Finalement, les exigences de ce type de bus seront détaillées.

3.3.1 Besoins des bus de données avioniques

Les systèmes avioniques doivent répondre à des besoins de plus en plus nombreux et diversifiés (contrôle de l'appareil, communications, divertissement, etc.). Puisque ces systèmes sont reliés à un réseau, les bus de données ont des demandes élevées en termes de performance. Mais il n'y a pas que la performance qui justifie le choix d'un bus de données. En effet, plusieurs critères doivent être pris en considération (Schuster et Verma, 2008) :

- performance,
- fiabilité,
- sécurité et capacité à être certifié,
- coûts,
- prise en charge,
- flexibilité et capacité d'évoluer.

Performance

Pour satisfaire aux critères de performance, un réseau doit tenir compte de caractéristiques telles la vitesse de transmission, le débit, la latence et la qualité de service.

Fiabilité

La fiabilité est la capacité du réseau à demeurer opérationnel et sans défaillance pendant un temps déterminé. Les exigences en termes de fiabilité sont assurées par le type de topologie et le protocole utilisé.

Sécurité et capacité à être certifié

Un réseau avionique est un système critique qui doit répondre aux exigences de sécurité établies par les autorités de certification. Le développement des logiciels a un impact important sur la sécurité du réseau et sa capacité à être certifié.

Coûts

Les coûts d'un réseau impliquent des coûts de développement et des coûts d'acquisition. Les principales caractéristiques qui influencent les coûts d'un réseau sont la disponibilité des composants sur étagère (COTS : *commercial off-the-shelf*) et le développement des logiciels.

Prise en charge

Durant la durée de vie d'un appareil, un réseau avionique doit être pris en charge de façon rentable avec un minimum de ressources. Ceci concerne l'entretien du réseau et la mise à jour des systèmes en place. La disponibilité des composants sur étagère et les taux de défaillance logicielle et matérielle ont un impact direct sur la prise en charge.

Flexibilité et capacité d'évoluer

L'utilisation des composants sur étagère demeure importante pour la flexibilité d'un réseau et sa capacité d'évoluer.

3.3.2 Sécurité des bus de données avioniques

Un système avionique est conçu pour opérer dans un environnement où la sécurité est critique. Un système critique sur le plan de la sécurité est un système dont la défaillance implique de graves conséquences. En effet, le dysfonctionnement d'un tel système peut mettre en danger les personnes et causer des dommages importants au matériel et à l'environnement.

Comme tout système avionique, un bus de données doit se comporter selon ses spécifications. Cependant, il y a toujours un certain risque associé à l'utilisation d'un système critique. Concernant les bus de données avioniques, trois aspects importants doivent être considérés dans le processus d'évaluation des risques (Zalewski et al., 2005) :

- analyse des risques fonctionnels (FHA : *Functional Hazard Analysis*),
- analyse de la sécurité du système (SSA : *System Safety Analysis*),
- analyse des causes communes (CCA : *Common Cause Analysis*).

Analyse de risques fonctionnels

Cette analyse identifie les défaillances du système et leurs effets. Les défaillances sont classées selon un niveau de criticité établie par la norme DO-178B (voir section 2.4.2). À chacun des niveaux sont associé un objectif de sécurité et une probabilité d'occurrence. Le Tableau 3.1 identifie les conditions de défaillance et leur objectif de sécurité.

Tableau 3.1 Classification des défaillances
Tiré de Moir et Seabridge (2008)

Condition de défaillance	**Niveau de criticité**	**Objectif de sécurité**	**Objectif quantitatif requis (probabilité par heure de vol)**
Catastrophique	A	Requis	$< 1 \text{ X } 10^{-9}$
Sévère	B	Peut être requis	$< 1 \text{ X } 10^{-7}$
Majeur	C	Peut être requis	$< 1 \text{ X } 10^{-5}$
Mineur	D	Non requis	Aucun
Sans effet	E	Non requis	Aucun

Par exemple, une défaillance ayant un effet catastrophique sera de niveau A. Le système de contrôle de vol d'un appareil appartient à ce niveau. Il doit être démontré que la probabilité qu'une défaillance survienne est inférieure à $1x10^{-9}$. Autrement dit, la défaillance en question doit survenir moins d'une fois pour un milliard d'heures de vol.

Analyse de la sécurité du système

Cette analyse permet de démontrer que les exigences de sécurité d'un système sont satisfaites. Dans le cycle de conception d'un système, l'analyse de la sécurité est un préalable au processus de certification (Moir et Seabridge, 2008).

Comme il a été mentionné précédemment, l'industrie aéronautique est régie par un processus de certification qui impose des contraintes sévères à tout système avionique. En ce qui concerne les bus de données, plusieurs critères sont à considérer par les fabricants d'avions (Rierson et Lewis, 2003). Les principaux critères d'évaluation des bus de données avioniques sont indiqués dans le Tableau 3.2. Ces critères sont détaillés dans le document original (Certification Authorities Software Team (CAST), 2003).

Tableau 3.2 Critères d'évaluation des bus de données avioniques
Tiré de Zalewski et al. (2005)

Critères	**Facteurs d'évaluation**
Sécurité	Fiabilité et disponibilité du bus. Partitionnement du bus. Détection et gestion des défaillances. Défaillances de causes communes. Reconfiguration du réseau. Stratégie d'expansion du bus. Gestion de la redondance.
Intégrité des données	Taux d'erreur maximum. Détection et récupération d'erreurs. Analyse des limitations du bus. Capacité du bus. Débordement des tampons. Saturation des commutateurs.

Critères	**Facteurs d'évaluation**
Performance	Vitesse d'opération du bus. Bande passante. Ordonnancement des messages. Interopérabilité du système. Capacité de retransmission. Latence et efficacité.
Assurance de développement	Matériel conforme à la norme DO-254. Logiciel conforme à la norme DO-178B.
Compatibilité électromagnétique	Vitesse de commutation. Temps de montée et descente d'une impulsion. Immunité à la foudre et aux radiations. Câblage et blindage.
Vérification et validation	Tests de fonctionnalité du bus. Vérification et validation des opérations du bus. Tests de défaillance.
Gestion de la configuration	Contrôle des modifications. Spécification des normes. Documentation.
Maintien de la navigabilité	Dégradation physique des composants. Modifications en service. Maintenance.

Ces critères d'évaluation ne proviennent pas officiellement des autorités de certification. Ils ont été établis par des spécialistes en certification logicielle (Certification Authorities Software Team (CAST), 2003). De plus, cette liste de critères d'évaluation n'est pas exhaustive. Il s'agit plutôt d'un guide minimal à considérer si l'on désire faire certifier un bus de données avionique (Zalewski et al., 2005).

Analyse de causes communes

Cette analyse permet de minimiser les risques reliés aux causes communes. Grâce à cette analyse, le concepteur peut s'orienter vers des stratégies permettant d'éviter les défaillances de causes communes (Moir et Seabridge, 2008). Voici quelques exemples de ce type de défaillances :

- erreurs de conception matérielle,
- défaillance des composants,
- erreurs de conception logicielle,
- défauts des outils logiciels,
- erreurs de maintenance.

3.3.3 Exigences des bus de données avioniques

En avionique, les systèmes temps réel à sécurité critique requièrent des exigences particulières quant au déterminisme et à la fiabilité. En effet, ces critères demeurent les plus importants pour un réseau de communication avionique.

Le déterminisme

Le déterminisme est défini comme étant la capacité du réseau à garantir la livraison d'un message dans un temps prévisible. Dans un réseau déterministe :

- la bande passante est garantie,
- la latence maximale de bout en bout est connue,
- la gigue est connue.

Les facteurs affectant le déterminisme d'un réseau sont les suivants (Federal Aviation Administration, 2005b) :

- la bande passante disponible,
- le nombre de nœuds connecté au réseau,
- le taux moyen du flux de trafic et les rafales de données,
- le protocole utilisé et les mécanismes de résolution de conflits,
- les taux de transmission et de réception des nœuds individuels,
- le matériel d'interconnexion utilisé.

La fiabilité

Au niveau système, la fiabilité représente la capacité du réseau à résister aux défaillances. Au niveau message, la fiabilité procure plutôt une garantie que l'information sera livrée sans erreur (Federal Aviation Administration, 2005a). La fiabilité d'un réseau donne une indication sur son degré de tolérance aux fautes. Le but de la tolérance aux fautes est d'empêcher la défaillance malgré la présence de fautes.

Dans un réseau, la tolérance aux fautes est généralement apportée par différentes techniques de redondance (Federal Aviation Administration, 2005a) :

- redondance physique,
- redondance d'information,
- redondance temporelle.

La redondance physique permet d'envoyer les données sur plusieurs réseaux simultanément. Ceci représente une protection contre les pertes de données dues à une défaillance d'un lien ou d'un équipement. Ce type de redondance prévient aussi les pertes de données causées par une erreur dans la transmission.

Avec la redondance d'information, des données supplémentaires sont ajoutées à la transmission afin de faciliter la détection des erreurs et de les corriger. À titre d'exemple, le contrôle de redondance cyclique (CRC) est très utilisé sur certains réseaux.

La redondance temporelle, quant à elle, consiste à retransmettre les mêmes données plus d'une fois.

CHAPITRE 4

FIABILITÉ DU PROTOCOLE PCI EXPRESS

4.1 Survol

Dans le chapitre précédent, les caractéristiques des bus de communication ont été présentées. Voici maintenant les mécanismes assurant la fiabilité du protocole PCI Express et le traitement des erreurs. De plus, ce chapitre présente le support pour le branchement à chaud.

C'est au CHAPITRE 4 que débute les contributions en termes d'éléments inédits. Rappelons que l'un des objectifs de cette recherche était d'identifier les éléments de fiabilité que procure la technologie PCI Express. Par conséquent, un travail assidu a été nécessaire afin d'extraire des spécifications tous les mécanismes de fiabilité qu'offre le protocole.

4.2 Mécanismes de fiabilité

La notion de fiabilité est souvent associée à une analyse du temps moyen entre les pannes pouvant survenir dans un système (MTBF : *Mean Time Between Failures*). Ce type d'analyse fait référence au matériel utilisé. Cependant, cette recherche concerne seulement le protocole PCI Express et non le matériel l'utilisant. Par conséquent, puisque le matériel avionique utilisant la technologie PCI Express est encore inexistant, aucune analyse MTBF n'a été faite au cours de cette recherche.

Pour les applications temps réel critiques, un protocole de communication doit être fiable et tolérant aux fautes. La fiabilité et la tolérance aux fautes ont été présentées à la section 3.3.3 qui traite des exigences des bus de données avioniques. Pour être fiable, un protocole de communication doit posséder des mécanismes efficaces de gestion d'erreurs.

Des mécanismes assurant la fiabilité du protocole PCI Express sont présents au sein de ses trois couches fonctionnelles (Figure 4.1).

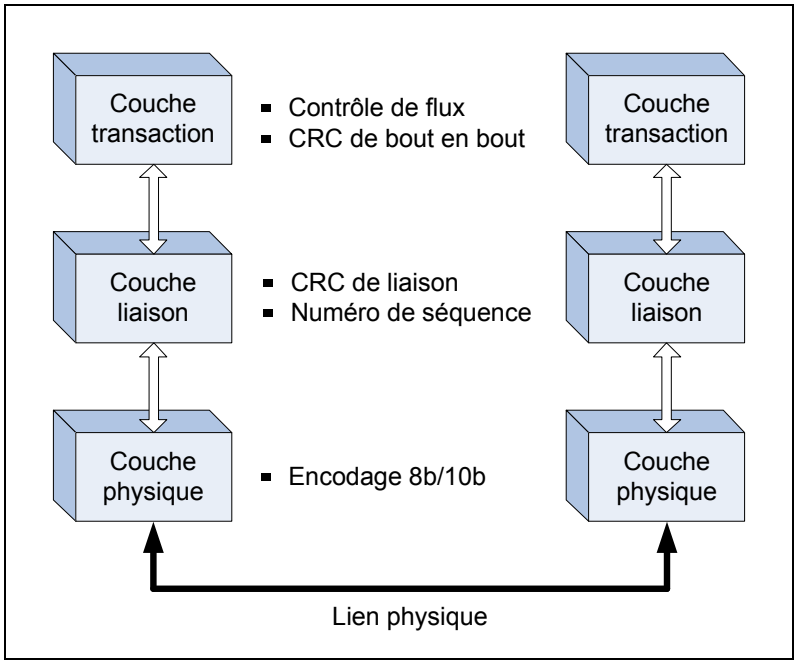

Figure 4.1 Mécanismes de fiabilité du protocole PCI Express

Voici la description détaillée des mécanismes de fiabilité du protocole PCI Express.

4.2.1 Contrôle de flux

La taille de la mémoire tampon des ports de réception de tous les composants PCI Express est quantifiée par une unité que l'on nomme crédit. Ainsi, pour prévenir les débordements dans les tampons de réception, la couche transaction utilise un mécanisme de contrôle de flux basé sur les crédits. Pour chacun des types de transaction, un composant indique le nombre de crédits disponibles. Avant d'envoyer un paquet de données sur un canal virtuel, un transmetteur vérifie toujours le nombre de crédits disponible au récepteur. Si le nombre de crédits est insuffisant, le paquet ne sera pas envoyé. S'il n'y a pas d'erreurs de transmission, ce mécanisme représente une garantie que les données envoyées seront reçues.

Le contrôle de flux s'exécute sur une liaison de point à point, entre les ports de chaque composant. De plus, le contrôle de flux est indépendant pour chacun des canaux virtuels. L'information concernant les crédits disponibles est transmise périodiquement par les DLLPs de la couche liaison. Ces DLLPs se nomment *Flow Control Packets*.

4.2.2 CRC de bout en bout (ECRC)

Il est possible de configurer la couche transaction pour appliquer un CRC de bout en bout de 32 bits (ECRC) à chacun des TLPs transmis. Contrairement au CRC de la couche liaison (voir 4.2.3), le ECRC demeure inchangé entre le transmetteur et le récepteur.

Le ECRC est uniquement appliqué aux champs invariables d'un paquet. Ainsi, le ECRC demeure inchangé lorsque le paquet traverse un commutateur. Ceci représente une protection supplémentaire contre les erreurs.

4.2.3 CRC de liaison (LCRC)

Pour chacun des TLPs transmis, la couche liaison applique un CRC de 32 bits (LCRC). Celui-ci est valide uniquement pour une liaison point à point. Lorsqu'un paquet transite sur le réseau, un commutateur peut changer certains champs de contrôle dans le paquet. Par conséquent, le commutateur doit régénérer un nouveau LCRC avant de retransmettre le paquet. Si un commutateur introduit une erreur dans les données, le nouveau LCRC masquera cette erreur. D'où l'importance d'utiliser conjointement le CRC de bout en bout (ECRC).

4.2.4 Numéro de séquence

La couche liaison applique un numéro de séquence à tous les TLPs transmis. Ceci représente une protection pour éviter la perte de paquets. En effet, le récepteur avertira le transmetteur s'il détecte un numéro manquant dans la séquence.

4.2.5 Encodage 8b/10b

La couche physique est responsable de l'encodage des données dans le format 8b/10b défini dans le standard ANSI X3.230-1994 (Krig, 2003). Le but premier de cet encodage est d'incorporer le signal d'horloge dans le flot de données. Cependant, cet encodage permet aussi d'inclure des symboles spéciaux aux paquets provenant de la couche liaison. Par exemple, ces symboles permettent d'identifier le début et la fin des paquets.

Pour encoder les données, une table de correspondance permet de transformer une série de 8 bits en un symbole de 10 bits. Le choix du symbole permet d'équilibrer le nombre de 1 et de 0 transmis sur le lien et limite ainsi le nombre de bits identiques consécutifs. C'est une fonction de disparité qui permet de choisir le bon symbole dans la table.

Cet encodage augmente le niveau de fiabilité du protocole, car il permet de détecter les erreurs de transmission. En effet, le récepteur fait le travail inverse lors du décodage. Si la fonction de disparité décode un mauvais symbole, la couche physique informe la couche liaison d'une erreur de transmission.

4.3 Gestion des erreurs

Le protocole PCI Express définit deux mécanismes de gestion d'erreurs. Un mécanisme de base répond aux besoins minimums et est requis pour tous les composants PCI Express. Pour une gestion d'erreurs plus robuste, un mécanisme avancé est disponible (AER : *Advanced Error Reporting*). Ce mécanisme avancé est optionnel et spécifique à chacun des types de composants.

Pour des applications en avionique, le mécanisme avancé est préférable. En effet, le CRC de bout en bout (ECRC) fait partie du mécanisme avancé et procure plus de fiabilité au protocole. Cependant, l'utilisation du ECRC a un impact négatif sur l'efficacité, car l'en-tête des TLPs contient 32 bits de plus.

4.4 Classification des erreurs

Le protocole PCI Express spécifie deux classes d'erreurs : les erreurs corrigibles et les erreurs incorrigibles. Parmi les erreurs incorrigibles, on distingue les erreurs fatales et les erreurs non fatales.

4.4.1 Erreur corrigible

Ce type d'erreurs n'affecte pas l'intégrité des données, mais a un impact sur la performance du système. Les erreurs corrigibles ne requièrent aucune intervention au niveau logiciel. En effet, ces erreurs sont entièrement corrigées au niveau matériel. Par contre, la détection d'une telle erreur ne représente pas une garantie que celle-ci sera corrigée. Si cette erreur ne peut être corrigée, elle sera rapportée comme étant une erreur incorrigible.

4.4.2 Erreur incorrigible non fatale

Les erreurs incorrigibles ont un impact sur la fonctionnalité de l'interface. Il n'existe aucun mécanisme PCI Express qui permet de corriger ce type d'erreurs. Une erreur non fatale ne met pas en cause l'intégrité du système. Il s'agit d'une erreur de transaction que le récepteur ne peut corriger. La récupération d'une telle erreur est toujours possible, mais cela dépend du logiciel associé au composant qui a initié cette transaction.

4.4.3 Erreur incorrigible fatale

Si une erreur incorrigible est rapportée comme étant fatale, le lien concerné est identifié comme étant non fiable. Par conséquent, la responsabilité de l'intervention en revient au logiciel système. La plupart du temps, une remise à zéro des composants du lien sera nécessaire pour un retour aux conditions normales.

Le Tableau 4.1 résume les classes d'erreurs PCI Express.

Tableau 4.1 Résumé des classes d'erreurs PCI Express

Type d'erreur	**Responsabilité**	**Description**
Corrigible	Matériel	N'affecte pas l'intégrité des données, mais a un impact sur la performance du système.
Incorrigible non fatale	Logiciel composant	Implique des pertes de données, mais l'intégrité du système est maintenue.
Incorrigible fatale	Logiciel système	Implique des pertes de données et une défaillance du système.

4.5 Signalement des erreurs

Il existe trois mécanismes complémentaires de signalement des erreurs :

- état des demandes (*Completion Status*),
- messages d'erreur (*Error Messages*),
- acheminement d'erreur (*Error Forwarding*).

État des demandes

L'en-tête d'un paquet de réponses (*Completion Header*) contient un champ sur l'état de la demande (*Completion Status Field*). Un répondeur peut ainsi aviser le demandeur qu'une erreur est survenue lors d'une transaction.

Message d'erreur

Des messages d'erreurs sont envoyés au Root Complex pour signaler la détection d'une erreur. Le Tableau 4.2 identifie les types de messages d'erreur selon la sévérité.

Tableau 4.2 Messages d'erreur envoyés au Root Complex

Message d'erreur	**Description**
ERR_COR	Détection d'une erreur corrigible
ERR_NONFATAL	Détection d'une erreur incorrigible non fatale
ERR_FATAL	Détection d'une erreur incorrigible fatale

Des erreurs peuvent être détectées au niveau des trois couches fonctionnelles du protocole. Pour éviter de multiples messages d'erreur pour une même transaction, la détection d'une erreur doit être isolée dès sa première occurrence. Par exemple, quand la couche liaison détecte une erreur, cette dernière ne doit pas se propager à la couche transaction. De plus, il existe un ordre de préséance pour les erreurs détectées au sein d'une même couche fonctionnelle. Il est recommandé de ne rapporter qu'une seule erreur pour une transaction donnée. Le Tableau 4.3 indique l'ordre de préséance des erreurs détectées par la couche transaction.

Tableau 4.3 Ordre de préséance des erreurs détectées par la couche transaction

Ordre	**Type d'erreur**
1	Débordement du récepteur
2	Erreur du contrôle de flux
3	Échec de vérification du CRC de bout en bout (ECRC)
4	TLP malformé
5	Requête non supportée, Avortement du répondeur, Réponse inattendue
6	TLP erroné

Acheminement d'erreur

L'acheminement d'erreur est une autre méthode pour rapporter une erreur. Contrairement au mécanisme de l'état des demandes, l'acheminement d'erreur peut être utilisé soit par une

requête ou bien par une réponse qui contient des données. Une erreur est signalée par le champ EP (*Error Forwarding*) de l'en-tête du TLP.

4.6 Enregistrement des erreurs

L'enregistrement des erreurs est spécifique au mécanisme de gestion des erreurs. Les fonctions qui ne supportent pas la gestion avancée des erreurs (AER) peuvent seulement indiquer dans le registre d'état (*Device Status register*) qu'une erreur a été détectée.

Les fonctions supportant la gestion avancée des erreurs peuvent préciser la nature de l'erreur détectée. Pour chaque type d'erreur correspond un bit particulier dans les registres d'état des erreurs (*Uncorrectable Error Status register* et *Correctable Error Status register*). Ces informations sont utilisées par la partie logicielle pour déterminer la nature et la sévérité des erreurs.

4.7 Interruption

Le Root Complex a la capacité de générer une interruption système en réponse à un message d'erreur (ERR_COR, ERR_NONFATAL, ERR_FATAL). Pour chacun des trois types de message, les interruptions peuvent être activées ou désactivées via le registre *Root Error Command*.

4.8 Support pour le branchement à chaud (hot-plug)

La facilité de maintenance (*Serviceability*) est une propriété importante pour un système utilisant le protocole PCI Express. En effet, le protocole permet de remplacer à chaud un composant du réseau. Ainsi, lorsqu'une défaillance survient, le système peut demeurer opérationnel pendant le remplacement du composant défectueux. Cette capacité de branchement à chaud est basée sur un modèle standard de l'industrie nommé *Standard Hot Plug Controller* (SHPC) (Krig, 2003).

Cette propriété est très intéressante pour les systèmes avioniques. En effet, lorsqu'un composant subit une défaillance, il est possible de le remplacer sans affecter le reste du système.

4.9 Synthèse

Ce chapitre a permis d'identifier tous les mécanismes de fiabilité présents au sein du protocole PCI Express. Voici un bref résumé de ces mécanismes.

Le contrôle de flux par des crédits donne une assurance que les données envoyées seront reçues par le récepteur. Ce mécanisme est très intéressant, car il permet d'éviter les problèmes de débordement des tampons de réception.

Le contrôle de redondance cyclique (ECRC et LCRC) est un mécanisme efficace pour contrôler l'intégrité des données. D'ailleurs, plusieurs protocoles de communication utilisent le CRC pour détecter les erreurs. Bien que le ECRC soit optionnel, son utilisation est recommandée pour une fiabilité accrue.

Le numéro de séquence ajoute une protection contre la perte de paquets. Puisque tous les paquets possèdent un numéro, le récepteur pourra identifier facilement la perte d'un paquet durant un transfert de données.

L'encodage 8b/10b offre aussi une protection supplémentaire, car il permet de détecter des erreurs. Lors du décodage, le récepteur peut identifier une altération au niveau des paquets.

Tous ces mécanismes font du PCI Express un protocole fiable. Les systèmes à sécurité critique, tels les systèmes embarqués avioniques, peuvent tirer avantages de cette grande fiabilité. De plus, le support pour le branchement à chaud représente un avantage certain pour le domaine avionique. Côté fiabilité, ce protocole représente un choix intéressant pour l'industrie aérospatiale.

CHAPITRE 5

DETERMINISME DU PROTOCOLE PCI EXPRESS

5.1 Survol

Le chapitre précédent a présenté en détail les mécanismes de fiabilité du protocole. En plus d'être fiable, un protocole à haut débit doit être performant. De plus, si ce protocole est utilisé dans un environnement où la sécurité est critique, le déterminisme devient une priorité. En avionique, par exemple, le déterminisme représente un facteur de haute importance. Le déterminisme des bus de données avioniques a été présenté à la section 3.3.3.

Pour qu'un réseau soit considéré comme étant déterministe, son comportement doit être prévisible en tout temps. Dans un réseau déterministe :

- la bande passante est garantie,
- la latence maximale de bout en bout est connue,
- la gigue est connue.

Ce chapitre est consacré au déterminisme du protocole PCI Express. Dans un premier temps, les éléments affectant la performance sont détaillés. Ensuite, le mode isochrone est présenté.

Rappelons que cette recherche avait comme objectif d'extraire des spécifications les éléments affectant le déterminisme du protocole. Nous pourrons ainsi déterminer si le protocole PCI Express rencontre les exigences prescrites par les autorités de certification avionique.

5.2 Éléments affectant la performance

Plusieurs paramètres du protocole PCI Express ont une influence sur la performance du système. Certains de ces paramètres, s'ils sont mal ajustés, ont un impact direct sur le déterminisme. Les paramètres de performance sont les suivants :

- la largeur du lien,
- le surdébit (*Overhead*),
- la taille maximale de la charge utile (*Maximum Payload Size*),
- la taille maximale d'une requête de lecture,
- la disponibilité des crédits du contrôle de flux,
- la taille du tampon de retransmission,
- le nombre d'étiquettes (*TAG*).

5.2.1 Largeur du lien

Un lien PCI Express peut être composé de plusieurs voies. On retrouve des liens à 1, 2, 4, 8, 12, 16 et 32 voies. Plus le lien est large, moins de cycles d'horloge seront nécessaires pour transmettre un paquet.

5.2.2 Surdébit

Dans un réseau PCI Express, le trafic est constitué de données utiles et d'autres informations permettant la gestion des transferts. Cette information supplémentaire se nomme surdébit (*Overhead*). Voici les différentes catégories de surdébit qui affectent la performance du système :

- encodage par symboles,
- structure du TLP,
- contrôle du trafic.

Encodage par symboles

Comme il a été mentionné dans la section 2.3.1, les données sont encodées dans le format 8b/10b. Avec cet encodage, 10 bits sont requis pour transmettre un octet. Ceci représente une perte de 20% de la bande passante. Par conséquent, la bande passante effective se calcule ainsi :

$$Bande\ passante\ effective\ (octets) = \frac{2.5\ Gbits/s \times Nb.\ de\ voies}{10\ bits/octet} \tag{5.1}$$

Dans cette formule, la valeur de 2.5 Gbits/s représente la bande passante brute, c'est-à-dire le taux auquel les bits sont transmis sur le lien.

Le Tableau 5.1 identifie la bande passante par direction selon le nombre de voies par lien.

Tableau 5.1 Bande passante PCI Express (génération 1)

Nombre de voies	**x1**	**x2**	**x4**	**x8**	**x12**	**x16**	**x32**
Bande passante brute (Gbits/s)	2.5	5	10	20	30	40	80
Bande passante effective (Mo/s)	250	500	1000	2000	3000	4000	8000

Structure du TLP

La structure d'un paquet de données est variable. La composition du TLP est représentée par la Figure 5.1.

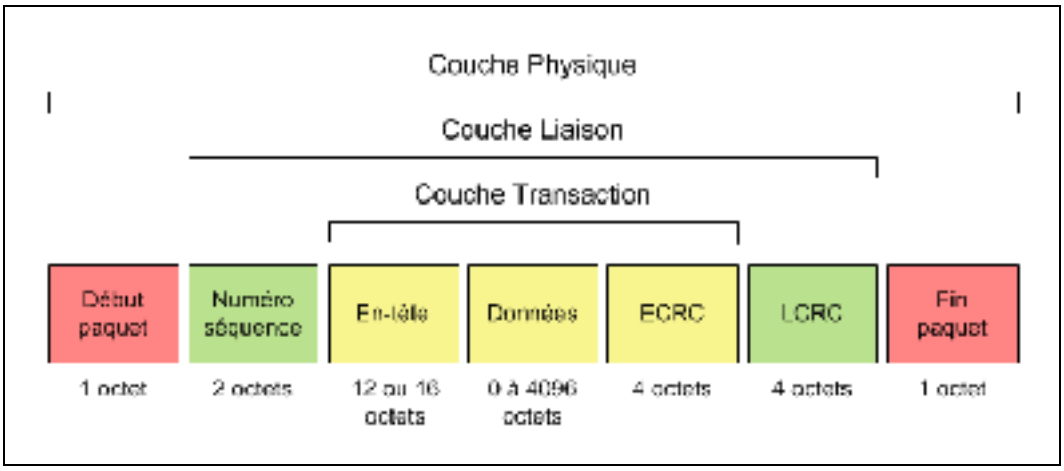

Figure 5.1 Structure du TLP

Chacune des trois couches fonctionnelles ajoute une certaine quantité d'information au paquet de données. L'en-tête contient 12 ou 16 octets selon que l'on utilise un adressage de 32 bits ou 64 bits. Le CRC de bout en bout (ECRC) est optionnel et a une taille de 4 octets. Par conséquent, le surdébit associé au TLP varie entre 20 et 28 octets.

Contrôle du trafic

Pour un contrôle efficace du trafic, la couche physique et la couche liaison introduisent dans le réseau une certaine quantité d'information. Ce surdébit affecte directement le paramètre d'utilisation du lien.

De façon périodique, la couche physique envoie sur le réseau des paquets PLPs (*Physical Layer Packets*) de types SKP (*Skip Ordered-Set*). Ces paquets ont pour but de compenser la variation des fréquences d'horloge entre le transmetteur et le récepteur. Les paquets SKP ont une taille de 4 octets et sont transmis dans un intervalle qui varie entre 1180 et 1538 symboles.

L'objectif principal de la couche liaison est d'assurer la fiabilité de la liaison entre deux points du réseau. Pour gérer les transferts de données, la couche liaison utilise des paquets que l'on nomme DLLPs (*Data Link Layer Packets*). Il existe différents types de DLLPs. Ceux ayant le plus d'effet sur la performance du système sont les suivants (Goldhammer et Ayer Jr, 2008) :

- acquittement (ACK),
- non-acquittement (NAK),
- contrôle de flux (FC).

Chacun des TLPs transmis sur le réseau doit être acquitté par le récepteur. Si le TLP est reçu avec succès, le récepteur envoie un DLLP de type ACK. Dans le cas contraire, c'est un DLLP de type NAK qui est envoyé. L'envoi constant de paquets d'acquittements a pour effet de réduire la bande passante disponible. Pour contrer cet effet, le protocole permet d'acquitter plusieurs TLP avec le même DLLP. Cependant, ce mécanisme est spécifique au design et dépend de la façon dont le composant a été configuré.

Comme il a été mentionné dans la section 4.2.1, pour éviter le débordement des tampons de réception, le protocole utilise un mécanisme de contrôle de flux basé sur des crédits. Le nombre de crédits est proportionnel à la taille de la mémoire tampon des ports de réception. Chacun des composants du réseau doit constamment mettre à jour la quantité de crédits disponibles de ces ports de réception. Cette information est partagée par l'envoi de DLLPs de type FC (*Flow Control*). La fréquence d'envoi de ce type de paquets dépend du design du composant.

5.2.3 Taille maximale de la charge utile

La taille maximale de la charge utile (MPS : *Maximum Payload Size*) est un paramètre ajustable pour chacun des composants du réseau. Selon la capacité du composant, ce paramètre peut prendre les valeurs suivantes : 128, 256, 512, 1024, 2048 ou 4096 octets. Cependant, tous les éléments du réseau doivent être ajustés à la même valeur. Cette valeur sera choisie pour accommoder le composant ayant la plus faible valeur de MPS. Par exemple, sur la Figure 5.2, la valeur MPS du système sera ajustée à 128 octets pour accommoder le composant Endpoint numéro 3 (Goldhammer et Ayer Jr, 2008). Pour toute transaction sur le réseau, la taille des données ne doit jamais surpasser la valeur de MPS. Pour une performance optimale, il faut donc éviter d'inclure dans le réseau des composants ayant une faible valeur de MPS.

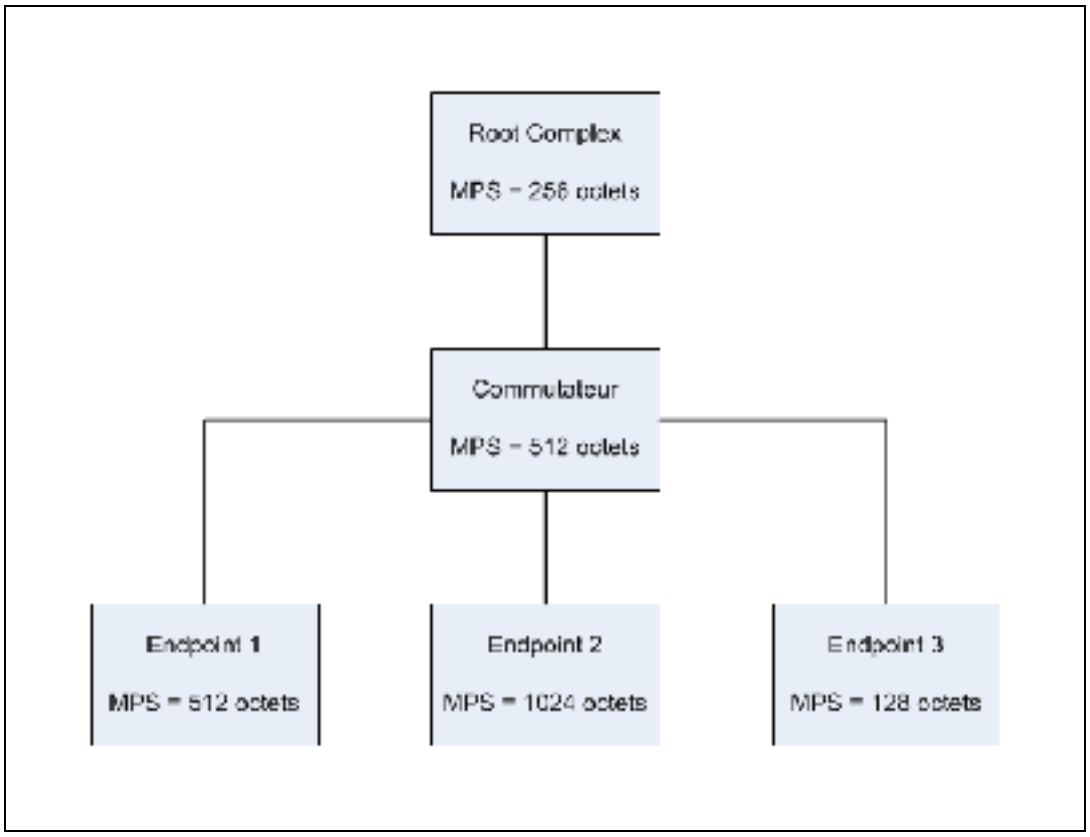

Figure 5.2 Taille maximale de la charge utile

La valeur de MPS a un impact direct sur la performance du système. En effet, cette valeur détermine le nombre de TLPs requis pour transmettre une certaine quantité de données. Plus la valeur de MPS sera faible, plus de TLPs seront nécessaires pour transmettre la même quantité d'information. Par conséquent, le surdébit associé à chacun des TLPs fera diminuer l'efficacité du transfert.

L'efficacité associée au TLP se calcul avec la formule suivante :

$$Efficacité\ du\ TLP = \frac{MPS}{MPS + Surdébit} \tag{5.2}$$

Le Tableau 5.2 identifie l'efficacité du TLP selon les différentes valeurs de MPS. Dans cet exemple de calcul, le surdébit associé au TLP est de 24 octets (adressage de 32 bits et utilisation du CRC de bout en bout : ECRC).

Tableau 5.2 Efficacité du TLP

MPS (octets)	**Efficacité du TLP (%)**
128	84.2
256	91.4
512	95.5
1024	97.7
2048	98.8
4096	99.4

Les résultats du Tableau 5.2 sont représentés graphiquement par la Figure 5.3.

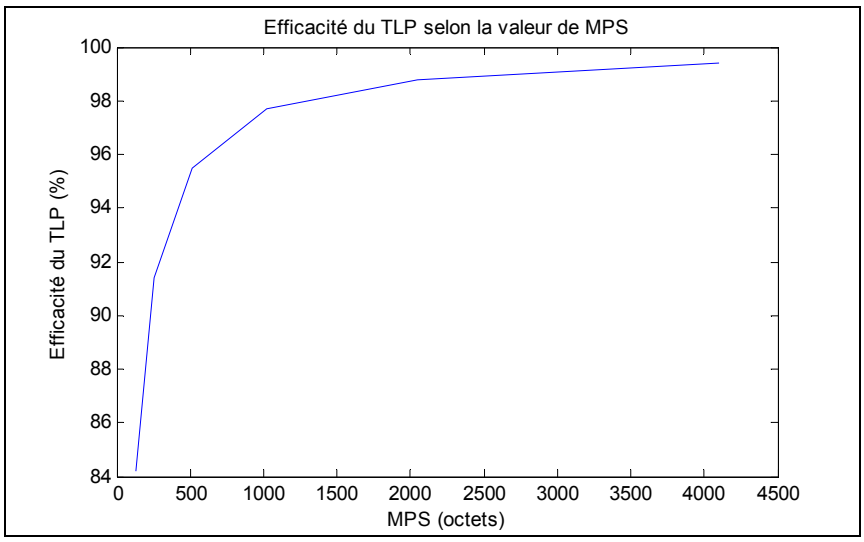

Figure 5.3 Efficacité du TLP selon la valeur de MPS

Pour transférer de gros blocs de données, il est préférable d'utiliser une valeur élevée de MPS. Cependant, la plupart des composants sur le marché présentement ont une valeur de MPS de 128 ou 256 octets (Goldhammer et Ayer Jr, 2008).

5.2.4 Taille maximale d'une requête de lecture

Pour éviter de monopoliser la bande passante, les requêtes de lecture sont limitées en taille. Pour chacun des composants du réseau, ce paramètre est ajusté durant la phase de configuration. La taille maximale possible pour une requête de lecture est de 4096 octets. Il est à noter que ce paramètre est indépendant de la taille maximale de la charge utile (MPS). Par conséquent, la taille d'une requête de lecture peut être plus élevée que la valeur de MPS. Dans ce cas, plusieurs paquets de réponses seront nécessaires pour faire une lecture.

Lorsqu'un composant fait une requête de lecture, les données sont retournées dans des paquets de réponses (*TLP Completion*). Pour un système donné, la taille des paquets de réponses est fixe. Dû à l'alignement des données avec les adresses mémoires, les paquets de réponses contiennent toujours 64 ou 128 octets de données. Il s'agit du paramètre RCB : *Read Completion Boundary*. La plupart des Root Complex retournent les données par paquets de 64 octets (PCI-SIG, 2006). Ainsi, une simple requête de lecture peut demander plusieurs paquets de réponses.

L'efficacité de la communication est directement affectée par la taille des paquets de réponses. À titre d'exemple, prenons un système ajusté selon les paramètres suivants :

- taille maximale de la charge utile (MPS) : 128 octets,
- taille des paquets de réponses (RCB) : 64 octets,
- surdébit associé au TLP : 24 octets (adressage 32 bits et ECRC).

En écriture, l'efficacité du TLP sera : 128 / (128 + 24) = 84 %. La bande passante maximale sera donc à 84 % de la bande passante effective de 250 Mo/s, soit 210 Mo/s.

En lecture, l'efficacité du TLP sera : 64 / (64 + 24) = 73 %. La bande passante maximale sera donc à 73 % de la bande passante effective de 250 Mo/s, soit 182 Mo/s.

Le Tableau 5.3 identifie la bande passante maximale théorique en lecture et en écriture pour une voie et une direction.

Tableau 5.3 Bande passante maximale théorique

Bande passante théorique (Mo/s)	**Bande passante maximale théorique en écriture (Mo/s)**	**Bande passante maximale théorique en lecture (Mo/s)**
250	210	182

Ceci représente des valeurs maximales théoriques. En réalité, ces valeurs ne sont jamais atteintes. En effet, pour le calcul de la valeur réelle, il faut considérer le surdébit introduit par la couche physique (PLPs) et la couche liaison (DLLPs).

5.2.5 Disponibilité des crédits du contrôle de flux

Pour chaque type de transaction et pour chacun des canaux virtuels, le récepteur maintient le compte des crédits disponibles. Périodiquement, le récepteur doit envoyer des DLLPs (*Data Link Layer Packets*) de type FC (*Flow Control*) pour informer le transmetteur du nombre de crédits disponibles. La Figure 5.4 illustre le processus de contrôle de flux.

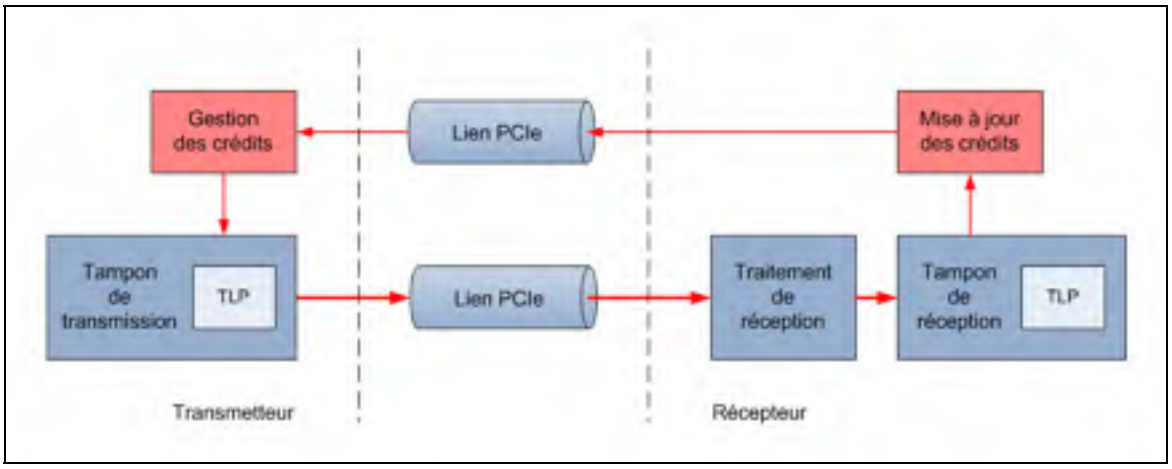

Figure 5.4 Processus du contrôle de flux

La fréquence d'envoi des DLLPs de type FC affecte directement la performance du système. Si la fréquence de mise à jour des crédits disponibles est trop élevée, de la bande passante est consommée inutilement. D'un autre côté, si cette fréquence est trop faible, le transmetteur ne peut atteindre son débit maximal. En effet, si la mise à jour des crédits disponibles n'est pas assez fréquente, le transmetteur doit continuellement attendre que des ressources se libèrent avant de transmettre de nouveau. La fréquence de mise à jour des crédits est ajustée pendant la phase de design du composant.

Le nombre de crédits disponibles est directement relié à la taille de la mémoire tampon des ports de réception. Pour une performance optimale, les crédits disponibles doivent être suffisants pour masquer la latence introduite par le mécanisme du contrôle de flux (Granovsky et Perlin, 2007). Lorsque cette latence est masquée, un débit constant est possible.

5.2.6 Taille du tampon de retransmission

Lorsqu'un composant transmet un TLP, une copie de ce paquet est mise dans un tampon temporaire. Il s'agit du tampon de retransmission (*Retry Buffer*). Le paquet demeure dans ce tampon jusqu'à ce qu'il soit acquitté. Si le paquet est acquitté négativement (NAK), il devra être retransmis de nouveau. Par contre, si l'acquittement est positif (ACK), le paquet sera libéré de la mémoire tampon.

La taille du tampon de retransmission a un impact sur la performance du système. En effet, un tampon trop petit peut devenir saturé si les paquets ne sont pas acquittés assez rapidement. Lorsque le tampon de retransmission est plein, le transmetteur ne peut plus transmettre. Les deux figures suivantes illustrent l'effet de la latence d'acquittement sur le tampon de retransmission. Le composant A transmet des TLPs au composant B qui doit acquitter les paquets reçus. Pour cet exemple, le tampon de retransmission du composant A peut contenir 4 TLPs.

La Figure 5.5 représente le cas où le temps d'acquittement est inférieur à 4 unités de temps de transmission d'un TLP ($t_{ACK} < 4*t_{TLP}$). Puisque le tampon de retransmission à une taille de 4 TLPs, le composant A peut transmettre à un débit constant.

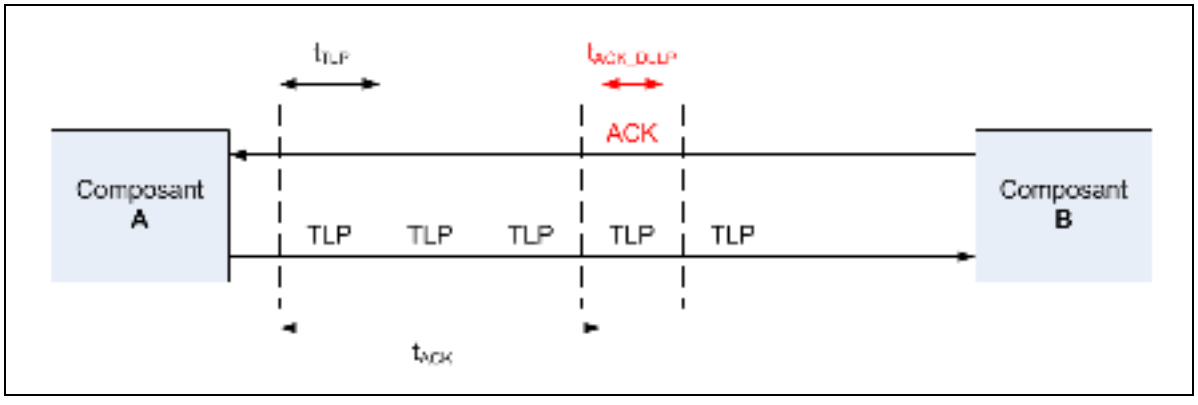

Figure 5.5 Latence d'acquittement faible

Pour la Figure 5.6, le temps d'acquittement est supérieur à 4 unités de temps de transmission d'un TLP ($t_{ACK} > 4*t_{TLP}$). Par conséquent, le composant A ne peut transmettre de façon continue. Lorsque le composant A transmet 4 TLPs, son tampon de retransmission est plein. Il doit attendre un acquittement de la part du composant B avant de pouvoir continuer à transmettre.

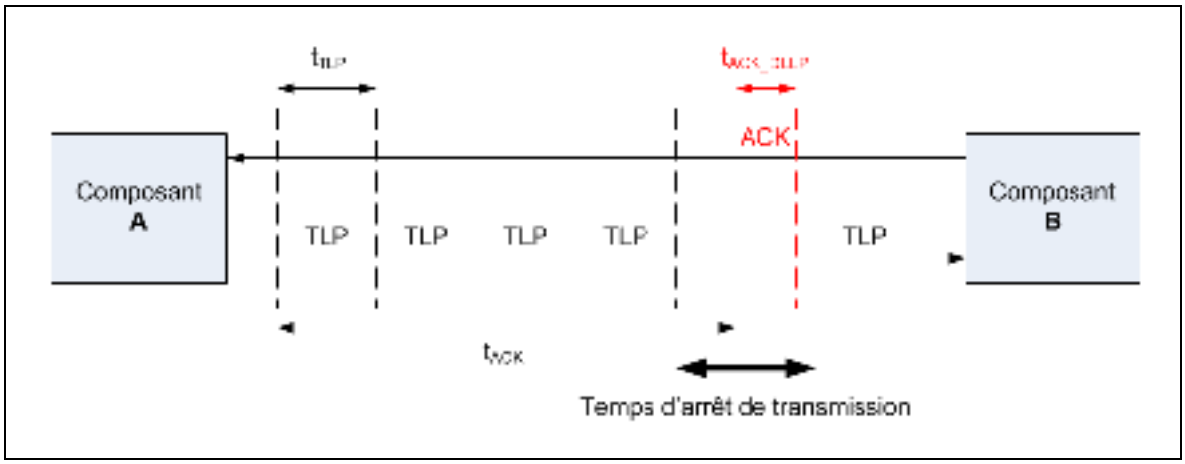

Figure 5.6 Latence d'acquittement élevée

Pour une performance optimale, la taille du tampon de retransmission doit être suffisante pour masquer la latence introduite par le processus d'acquittement (Yogendhar, Thyagarajan et Swaminathan, 2007). Pour une application avionique, ce critère est très important. En effet, pour préserver le caractère déterministe de la transmission, des pauses imprévisibles ne peuvent être tolérées.

5.2.7 Nombre d'étiquettes

Pour toute requête non postée, une étiquette (*TAG*) est mise dans l'en-tête du TLP. Il s'agit d'un numéro identifiant la requête. Cette même étiquette fera partie de l'en-tête de la réponse. Lorsqu'un composant fait plusieurs requêtes une à la suite de l'autre, les étiquettes permettent d'associer les réponses aux requêtes correspondantes.

La performance du système dépend du nombre d'étiquettes disponibles. En effet, un composant ne peut effectuer plus de requêtes « simultanées » que le nombre d'étiquettes disponibles. Si le nombre d'étiquettes est trop faible, un composant devra attendre l'arrivée des réponses pour effectuer de nouvelles requêtes. Le nombre d'étiquettes doit être suffisant pour qu'un composant puisse soumettre des requêtes en continu. Ce paramètre est ajustable durant la phase de conception.

5.3 Mode isochrone

Dans un réseau de communication, la qualité de service (QoS : *Quality of Service*) permet d'offrir aux applications critiques une certaine garantie en termes de performance, déterminisme et fiabilité. Plusieurs caractéristiques d'un réseau sont directement affectées par la qualité de service, dont la bande passante, la latence, la gigue et le taux de perte des données. En ce qui concerne le protocole PCI Express, la qualité de service est offerte grâce à des mécanismes que l'on nomme isochrones (PCI-SIG, 2006). Un exemple typique de composant pouvant bénéficier du mode isochrone est une caméra vidéo munie d'une interface PCI Express (Budruk, Anderson et Shanley, 2004). Les images non compressées et le son peuvent être transmis en temps réel à un taux constant.

5.3.1 Transactions isochrones

Les mécanismes isochrones du protocole supportent deux types de liaisons. Le premier type permet de relier un Endpoint au Root Complex. Le second type de liaison, quant à lui, permet

de relier deux Endpoints entre eux. Dans ce cas il s'agit d'une liaison pair-à-pair (P2P : *Peer-to-Peer*). La Figure 5.7 illustre les deux types de liaison isochrone.

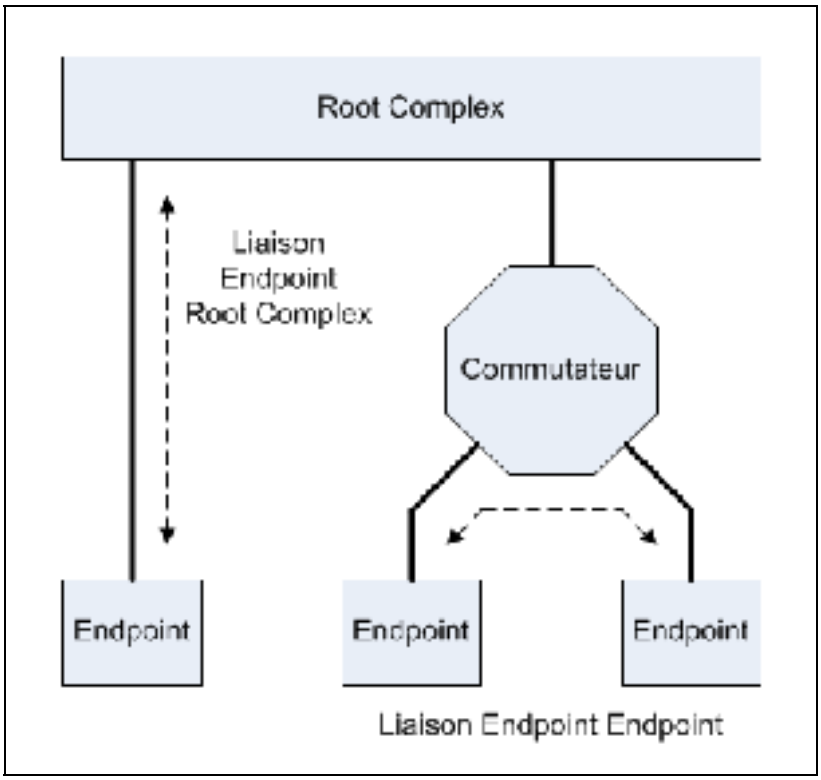

Figure 5.7 Types de liaison isochrone

En configurant adéquatement le réseau, il est possible d'avoir une bande passante garantie et une latence déterministe. On doit cependant éviter de combiner des transactions isochrones et non isochrones dans le même canal virtuel (VC). Toutes les ressources d'un canal virtuel en particulier doivent être allouées au trafic isochrone de ce canal.

Il est à noter que plusieurs flux isochrones peuvent emprunter le même canal virtuel. Dans ce cas, des précautions doivent être prises au niveau logiciel pour ne pas excéder les valeurs permises en ce qui concerne la bande passante et la latence. Finalement, précisons qu'il est possible de partager plusieurs flux isochrones parmi plusieurs canaux virtuels.

5.3.2 Contrat isochrone

Pour établir une liaison isochrone, un contrat est nécessaire entre tous les composants impliqués par cette liaison. Ceci comprend non seulement les composants terminaux (demandeur et répondeur), mais aussi les composants intermédiaires, c'est-à-dire les commutateurs. Ce contrat permet de réserver les ressources et de réguler le trafic pour la liaison isochrone. Sans ce type de contrat, des problèmes de surcharge et de congestion sont à prévoir sur le réseau. Par conséquent, la régulation du trafic est essentielle pour permettre au trafic isochrone et non isochrone de circuler sur le même réseau tout en respectant les spécifications au niveau de la latence.

La régulation du trafic isochrone inclut des mécanismes de contrôle d'admission et de service de discipline. Grâce à ces mécanismes, un composant ne peut initier une transaction isochrone tant que les ressources n'ont pas été allouées au niveau logiciel.

La latence est un autre paramètre important du contrat isochrone. Une transaction isochrone est toujours complétée à l'intérieur d'une limite temporelle. Il s'agit de la latence maximale permise. Lorsqu'un contrat isochrone est établi sur une liaison, la latence et la bande passante deviennent déterministes sur cette liaison (PCI-SIG, 2006).

5.3.3 Paramètres isochrones

Voici les paramètres du mode isochrone du protocole.

Créneau temporel virtuel (*Virtual Timeslot*)

Une période temporelle isochrone (T) est divisée en créneaux temporels virtuels (t) de façon uniforme. Pour préserver la garantie de la bande passante, seulement un paquet de données par créneau temporel virtuel peut être transmis. La valeur du créneau temporel virtuel est spécifiée dans les propriétés du canal virtuel.

Taille de la charge utile isochrone

Pour que la latence demeure un paramètre déterministe, la charge utile de données ne doit jamais excéder la valeur maximale spécifiée pendant la phase de configuration (MPS, voir section 5.2.3).

Bande passante isochrone

Pour le calcul de la bande passante isochrone maximale, nous posons la taille de la charge utile de données égale à la valeur maximale (MPS). Par conséquent, la valeur maximale de la bande passante est spécifiée par la formule suivante :

$$Bande\ passante\ isochrone\ maximale = \frac{MPS}{t} \tag{5.3}$$

Latence de transaction isochrone

La latence introduite par chaque élément du réseau définit ce que l'on appelle la latence de transaction. Cette latence est mesurée en unités de créneaux temporels virtuels (t). Lorsque les bornes de la latence sont connues, nous pouvons déterminer la taille des tampons du demandeur.

Pour un demandeur, la latence isochrone maximale (L_{Max}) est donnée par la formule suivante (PCI-SIG, 2006) :

$$L_{Max} = L_{Fabric} + L_{Completer} \tag{5.4}$$

Ici, L_{Fabric} représente la latence maximale entre le demandeur et le répondeur et $L_{Completer}$ représente la latence maximale du répondeur.

La valeur de L_{Fabric} dépend de la topologie du système et s'applique à tous les liens PCI Express et à tous les points d'arbitrage entre un demandeur et un répondeur. La latence d'un lien PCI Express dépend des facteurs suivants :

- les délais de pipeline,
- la largeur du lien,
- la fréquence d'opération du lien,
- la transmission des signaux électriques sur le support,
- la latence d'éveil du composant,
- les délais causés par le mécanisme de retransmission.

En ce qui concerne la valeur de $L_{Completer}$, celle-ci dépend des facteurs suivants :

- la technologie de la mémoire,
- la configuration de la mémoire,
- les politiques d'arbitrage.

Plusieurs facteurs influencent la latence. Celle-ci doit être calculée précisément pour chaque liaison du réseau. Cette valeur calculée représente une borne supérieure et ne sera jamais dépassée.

En terminant, mentionnons que le mode isochrone est très intéressant pour l'industrie avionique. En effet, il s'agit d'un mode de fonctionnement qui procure une bande passante et une latence déterministe.

5.4 Synthèse

Ce chapitre a permis d'identifier les éléments qui affectent la performance dans un réseau PCI Express. Parmi ces éléments, certains affectent le déterminisme du système. Par conséquent, si la technologie PCI Express est utilisée dans un système à sécurité critique, des ajustements judicieux doivent être faits pour conserver le caractère déterministe de la communication. Des précisions concernant ces ajustements sont données au CHAPITRE 7 (Présentation et interprétation de résultats). Voici un bref résumé des éléments affectant le déterminisme dans un réseau PCI Express.

La fréquence de réception des acquittements et la taille du tampon de retransmission sont interdépendantes et ont un effet sur le déterminisme du système. Si ces paramètres sont mal ajustés, un composant ne peut transmettre en continu. En effet, si la fréquence de réception des acquittements n'est pas assez élevée, le tampon de retransmission peut se remplir et le transfert est interrompu. D'un autre côté, un tampon de retransmission de grande taille peut absorber la latence introduite par le mécanisme d'acquittement. Par conséquent, il est très important, pour le concepteur du système, de bien ajuster ces deux paramètres si le déterminisme est une priorité.

Le nombre de crédits disponibles et la fréquence de mise à jour de cette information ont aussi un effet sur le déterminisme du système. Un nombre de crédits insuffisants peut produire des interruptions durant un transfert de données. De plus, si au récepteur, la fréquence de mise à jour des crédits disponibles n'est pas assez élevée, le transmetteur sera interrompu. Ces paramètres doivent être ajustés de façon adéquate.

Le déterminisme est aussi affecté par le nombre d'étiquettes dont dispose un composant. Un nombre insuffisant d'étiquettes empêchera un composant d'envoyer des requêtes en continu. Le composant devra attendre l'arrivée des paquets de réponses pour réutiliser les mêmes numéros d'étiquette. Un concepteur doit donc tenir compte de ce paramètre pour conserver le déterminisme dans le réseau.

En terminant, rappelons que le mode isochrone garantit un comportement déterministe du protocole PCI Express. Cela ne veut pas dire qu'aucun déterminisme n'est possible sans l'utilisation de ce mode. En effet, la section 5.2 a présenté les paramètres critiques du protocole ayant un effet sur le déterminisme. Si chacun de ses paramètres est ajusté adéquatement, un fonctionnement déterministe est possible.

Le mode isochrone demeure encore marginal. Ce mode de fonctionnement n'est pas supporté actuellement dans les architectures PC. Par conséquent, le mode isochrone du protocole n'a pu être testé au cours de cette recherche. Nous y reviendrons à la section des recommandations.

CHAPITRE 6

PLATEFORME D'EXPÉRIMENTATION

6.1 Survol

Pour valider les concepts théoriques vus précédemment, des expérimentations sont nécessaires. Certains résultats ont été validés sur une plateforme matérielle, alors que d'autres l'ont été par simulation. Ce chapitre présente, dans un premier temps, la méthodologie employée au cours de cette recherche, suivi des environnements de simulation et d'expérimentation.

6.2 Méthodologie

Après avoir décrit la méthode expérimentale, cette section présente le matériel et les logiciels utilisés dans le cadre de cette recherche. Ensuite, l'architecture de tests est présentée. Finalement, cette section se termine par la description des scénarios de tests.

6.2.1 Méthode expérimentale

Comme il a été mentionné précédemment, nous avons choisi d'utiliser une carte de développement muni d'un FPGA. Le principal avantage d'un FPGA est la grande flexibilité que procure cette technologie. En effet, il est possible de configurer le matériel selon les exigences des tests. De plus, la programmation du composant en langage VHDL permet une grande modularité au niveau du design.

Bien qu'il existe plusieurs compagnies de composants FPGA, notre choix s'est arrêté sur la compagnie Altera. Ce choix est justifié par deux raisons principales. Premièrement, le FPGA choisi contient un module PCI Express intégré (*Hard Core*). Ce dernier peut être configuré en Endpoint ou en Root Complex. L'intégration matérielle de ce module procure l'avantage

de ne consommer aucune ressource logique programmable. De plus, il faut noter qu'il est possible d'incorporer au design d'autres modules PCI Express programmables (*Soft Core*).

La deuxième raison justifiant l'utilisation d'un composant Altera est son processeur embarqué NIOS_II. Altera propose une version à sécurité critique de son processeur. Il s'agit du NIOS_II_SC, conforme à la norme RTCA DO-254. La compagnie Thales utilise déjà ce processeur embarqué dans leurs produits (Altera Corporation, 2010). Bien que le présent projet ne fasse pas usage de ce processeur, ce dernier demeure très intéressant pour des projets futurs.

La plateforme matérielle ne sert pas uniquement à valider les résultats théoriques. En effet, un des objectifs du projet était de concevoir une plateforme pouvant être utilisée dans d'autres projets de recherche. La plateforme développée pour cette recherche se limite à des applications pouvant effectuer des échanges d'informations entre un Endpoint et un Root Complex. Ceci comprend non seulement des échanges de données sporadiques, mais également des transferts de grandes quantités d'information via un contrôleur DMA (*Direct Memory Access*). De plus, la flexibilité et la modularité que procure cette plateforme lui permettent de s'adapter facilement à d'autres types de projets. À titre d'exemples, voici le rôle que pourrait jouer cette plateforme dans des projets futurs.

Exemple 1 : Étude de l'effet de la congestion sur le déterminisme.

Lorsque plusieurs composants Endpoint communiquent simultanément, il peut se produire une congestion au niveau du commutateur. Pour provoquer une congestion, nous pourrions utiliser plusieurs cartes Endpoint. Chacune de ces cartes utiliserait son contrôleur DMA simultanément. Nous pourrions ainsi évaluer l'effet de la congestion sur le déterminisme.

Exemple 2 : Conception d'un système multiprocesseur.

En utilisant plusieurs cartes de développement, il serait possible de concevoir un système multiprocesseur. En effet, le module PCI Express utilisé peut se configurer en Root Complex. Nous pourrions ainsi inclure dans le FPGA un processeur Nios_II et un système d'exploitation tel MicroC/OS-II. Une telle architecture permettrait d'étudier le comportement du protocole PCI Express dans un système se rapprochant d'une application avionique.

6.2.2 Matériel utilisé

Pour effectuer les expérimentations, le matériel suivant a été nécessaire :

- ordinateur PC,
- kit de développement PCI Express.

Le kit de développement est une carte PCI Express de la compagnie Altera : *Arria II GX FPGA Development Kit*. Le FPGA de cette carte contient un module matériel PCI Express (*Hard Core*) de première génération. Ce dernier peut être configuré en Endpoint ou en Root Complex. La carte utilisée est illustrée par la Figure 6.1.

Figure 6.1 Carte PCI Express Altera Arria II GX

Bien que cette carte soit de première génération, les tests effectués avec celle-ci seraient aussi valables pour les générations ultérieures. En effet, avec du matériel de deuxième ou troisième génération, les conclusions seraient identiques. Seules les valeurs de débit différeraient.

6.2.3 Logiciels utilisés

Dans le cadre de cette recherche, les logiciels suivants ont été utilisés.

Altera Quartus II (version 10.0)
Il s'agit d'un environnement de programmation et de synthèse pour les composants programmables de la compagnie Altera. Ce logiciel a permis de programmer le FPGA de la carte de développement. Le langage VHDL a été utilisé.

ModelSim SE (version 6.3c)
Ce logiciel a permis d'effectuer toutes les simulations nécessaires pour ce travail de recherche.

Jungo WinDriver (version 10.10)
Il s'agit d'un environnement de développement de pilotes pour ordinateur PC. Ce logiciel a été utilisé pour créer le pilote PCI Express de la carte de développement.

PCITree (version 2.04c)
PCITree est un utilitaire très convivial permettant de visualiser l'architecture PCI Express d'un ordinateur PC. Cet utilitaire permet aussi de lire et éditer les registres d'un composant PCI Express.

Microsoft Visual Studio (version 2008)
Environnement de développement intégré de Microsoft. L'application PC a été conçue en langage C avec ce logiciel.

6.2.4 Architecture de tests

L'architecture de tests est la même pour les expérimentations matérielles et pour les simulations. Cette architecture inclut un Endpoint et un Root Complex. Tous les tests ont été

effectués en considérant la première génération du protocole. Cette limitation provient du matériel utilisé. Sauf avis contraire, le lien PCI Express est toujours constitué d'une seule voie (x1). Le nombre de voies influence seulement le débit. La Figure 6.2 représente l'architecture de tests.

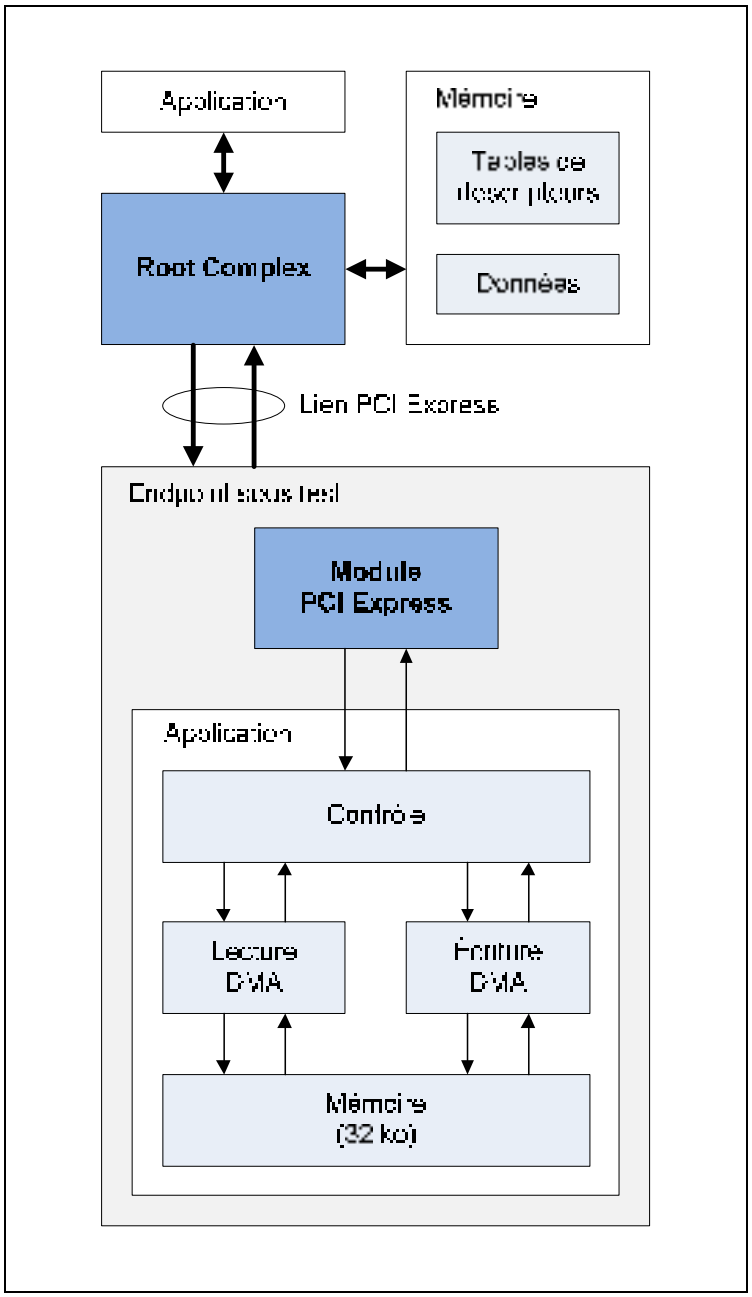

Figure 6.2 Architecture de tests

L'architecture de tests contient deux applications. L'une d'entre elles gère les actions du Root Complex alors que l'autre fait partie de l'Endpoint.

L'application de l'Endpoint est un contrôleur DMA (*Direct Memory Access*). Celle-ci permet de lire et écrire directement dans la mémoire système.

L'application du Root Complex, quant à elle, fournit les informations relatives aux transferts DMA. Ces informations sont inscrites dans des tables de descripteurs de la mémoire système. Une table de descripteur contient les informations suivantes :

- taille du transfert,
- adresse de la source,
- adresse de la destination,
- bits de contrôle.

Pour qu'un transfert de données ait lieu, l'application du Root Complex doit programmer les registres de contrôle du contrôleur DMA. Ces registres indiquent le nombre total de tables de descripteur ainsi que l'adresse de la première table. Après avoir programmé les registres de contrôle, le contrôleur DMA récupère les tables de descripteur et exécute les transferts de données.

6.2.5 Scénarios de tests

Des scénarios de tests sont utilisés pour valider les concepts théoriques du protocole PCI Express. Il faut noter que les tests pouvant être effectués avec la plateforme demeurent limités. Bien que reconfigurable, cette plateforme permet d'évaluer seulement quelques éléments relatifs à la performance, au déterminisme et à la fiabilité. Pour des tests plus élaborés, un équipement spécialisé est nécessaire. La section sur les recommandations présente les détails concernant cet équipement.

Les tests effectués dans l'environnement de simulation sont les suivants :

- le surdébit (*Overhead*),
- la taille maximale de la charge utile (*Maximum Payload Size*),
- la disponibilité des crédits du contrôle de flux,
- la taille du tampon de retransmission,
- le nombre d'étiquettes (*TAG*).

Dû à des limitations du banc d'essai, certains tests ne peuvent pas être effectués dans l'environnement de simulation. Les tests qui sont effectués sur la plateforme matérielle sont les suivants :

- la largeur du lien,
- la taille maximale d'une requête de lecture,
- l'insertion d'erreurs.

6.3 Environnement de simulation

Les simulations sont effectuées avec le logiciel ModelSim. Un banc d'essai permet de programmer les tables de descripteur.

6.3.1 Résultats de simulation

Lorsque la simulation est terminée, les informations relatives à la performance sont inscrites dans un fichier de résultats. La Figure 6.3 illustre un exemple de résultats fourni par le banc d'essai. Sur cet exemple on peut lire que le transfert DMA s'est effectué à un taux de 210 Mo/s.

```
INFO:       54618 ns PERF: Sample Duration: 18880 ns
INFO:       54618 ns PERF:       Tx Packets: 31
INFO:       54618 ns PERF:         Tx Bytes: 3968
INFO:       54618 ns PERF:    Tx MByte/sec: 210
INFO:       54618 ns PERF:     Tx Mbit/sec: 1681
INFO:       54618 ns PERF:       Rx Packets: 0
INFO:       54618 ns PERF:         Rx Bytes: 0
INFO:       54618 ns PERF:    Rx MByte/sec: 0
INFO:       54618 ns PERF:     Rx Mbit/sec: 0
```

Figure 6.3 Exemple de résultats de simulation

6.3.2 Chronogrammes de simulation

Le banc d'essai fournit aussi des chronogrammes utiles pour visualiser les échanges d'informations entre les composants PCI Express. La Figure 6.4 représente un exemple de chronogramme. Il s'agit des signaux entrants du côté de l'Endpoint.

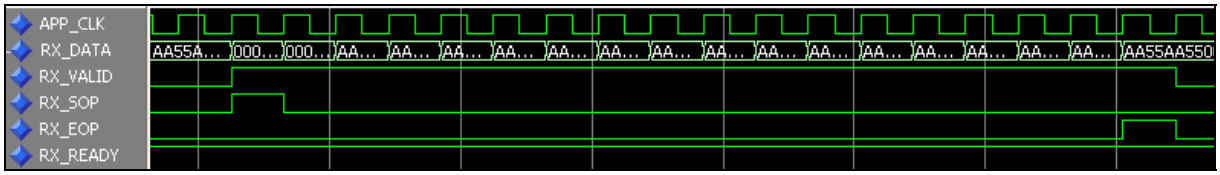

Figure 6.4 Exemple de chronogramme de simulation

Voici la description des noms de signaux du chronogramme :

- APP_CLK horloge du côté application de l'Endpoint,
- RX_DATA données,
- RX_VALID indique quand les données sont valides,
- RX_SOP indique le début du paquet (SOP : *Start of Packet*),
- RX_EOP indique la fin du paquet (EOP : *End of Packet*),
- RX_READY indique que l'application est prête à recevoir des données.

6.4 Environnement expérimental

Pour que l'application puisse communiquer avec le matériel, un pilote est nécessaire. Le pilote WinDriver (Jungo Ltd, 2009a) de la compagnie Jungo est utilisé dans le cadre de cette recherche. De plus, plusieurs fonctions de la librairie Jungo (Jungo Ltd, 2009b) sont utilisées dans l'application du Root Complex. Les fonctions sur la gestion des adresses ont été très utiles lors de la conception du programme. En effet, dû au système d'exploitation, une application PC manipule seulement des adresses virtuelles. Un composant PCI Express, quant à lui, manipule seulement des adresses physiques. L'application PC qui gère les transferts DMA doit nécessairement connaître les adresses physiques correspondantes aux

adresses virtuelles. Dans un ordinateur, la correspondance entre les adresses physiques et virtuelles se fait par le MMU (*Memory Management Unit*). Une application n'a pas accès à cette information. Cependant, certaines fonctions de la librairie Jungo permettent de faire la correspondance entre les deux types d'adresses. Grâce à ces fonctions, il a été possible de faire communiquer entre elles les applications du Root Complex et de l'Endpoint par l'entremise d'une mémoire partagée.

6.4.1 Résultats expérimentaux

Le temps d'exécution d'un transfert DMA est mesuré par l'application de l'Endpoint. Au terme du transfert, cette valeur temporelle est sauvegardée dans un registre de la mémoire partagé. L'application du Root Complex peut ainsi récupérer cette valeur pour afficher le taux auquel s'est exécuté le transfert.

6.4.2 Chronogrammes expérimentaux

Afin de visualiser les signaux en temps réel, il est possible d'incorporer un analyseur logique au design du FPGA. En effet, l'outil *SignalTap II Logic Analyser* permet d'afficher les chronogrammes des signaux internes du FPGA. La Figure 6.5 représente un exemple de chronogramme en temps réel. La description des noms de signaux est la même que pour les chronogrammes de simulation (voir section 6.3.2).

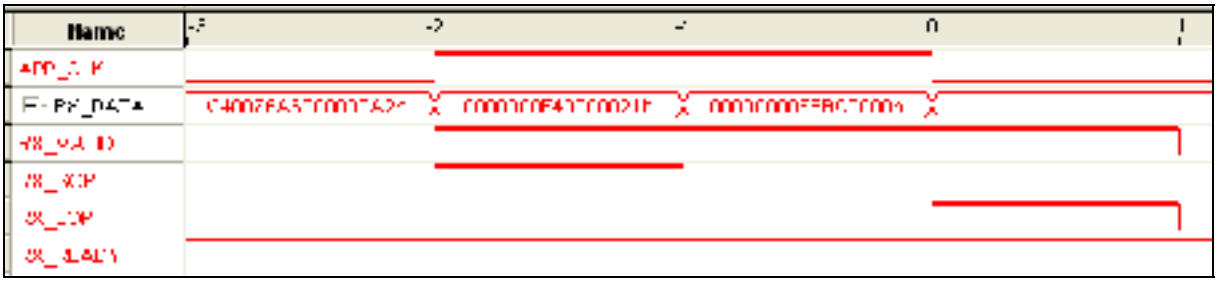

Figure 6.5 Exemple de chronogramme en temps réel

6.4.3 Signaux de tests

Des signaux de tests sont fournis avec le module PCI Express. Il s'agit des signaux **test_in** et **test_out**. Ces derniers permettent une observabilité et un contrôle en temps réel sur le module PCI Express. Par exemple, le signal test_in permet d'injecter certains types d'erreurs dans la transmission. Le signal test_out, quant à lui, permet d'observer la progression des données à travers les trois couches fonctionnelles du protocole.

CHAPITRE 7

PRÉSENTATION ET INTERPRÉTATION DES RÉSULTATS

7.1 Survol

Comme il a été mentionné dans le chapitre précédent, les résultats sont de deux types. Il s'agit des résultats de simulation et des résultats expérimentaux sur une plateforme matérielle. Ce chapitre présente les résultats des tests effectués dans le cadre de cette recherche. Le programme ayant servi à faire les tests expérimentaux est présenté à la fin de ce chapitre.

L'objectif de ces tests est de valider les concepts théoriques. Les chapitres 4 et 5 ont présenté respectivement la fiabilité et le déterminisme du protocole PCI Express. Plusieurs mécanismes du protocole ont un impact direct sur la fiabilité et le déterminisme que procure cette technologie. Ainsi, ces tests permettent d'observer l'effet des éléments critiques dont tout concepteur doit tenir compte dans la réalisation d'un système avionique. À la lumière de ces résultats, nous pourrons recommander ou non l'utilisation de la technologie PCI Express dans les systèmes à sécurité critique tels les systèmes avioniques.

7.2 Résultats de simulation

Dans un premier temps, l'environnement de simulation a été créé avec le logiciel Quartus II de la compagnie Altera. Le langage VHDL a été utilisé. Ensuite, les simulations ont été effectuées avec le logiciel ModelSim de Mentor Graphics. Le Tableau 7.1 présente la description des paramètres communs pour tous les tests de simulation.

Tableau 7.1 Paramètres communs
des tests de simulation

Paramètre	**Description**
FPGA	Arria II GX
Configuration	Endpoint
Module	PCI Express Soft IP
Version	PCI Express 1.1

7.2.1 Surdébit (*Overhead*)

Il existe différentes catégories de surdébit (voir section 5.2.2). Le CRC de bout en bout (ECRC) représente un surdébit associé à la structure du TLP. Lorsqu'il est utilisé, le ECRC ajoute 32 bits à chacun des TLPs transmis. Le ECRC a été utilisé pour analyser l'effet du surdébit sur la bande passante. Deux séries de tests ont été effectuées. Une série de tests inclut le ECRC alors que pour la deuxième série, le ECRC est absent. Les tests ont été effectués en faisant varier la charge utile de données dans les paquets. La Figure 7.1 illustre l'effet du surdébit (ECRC) sur la bande passante.

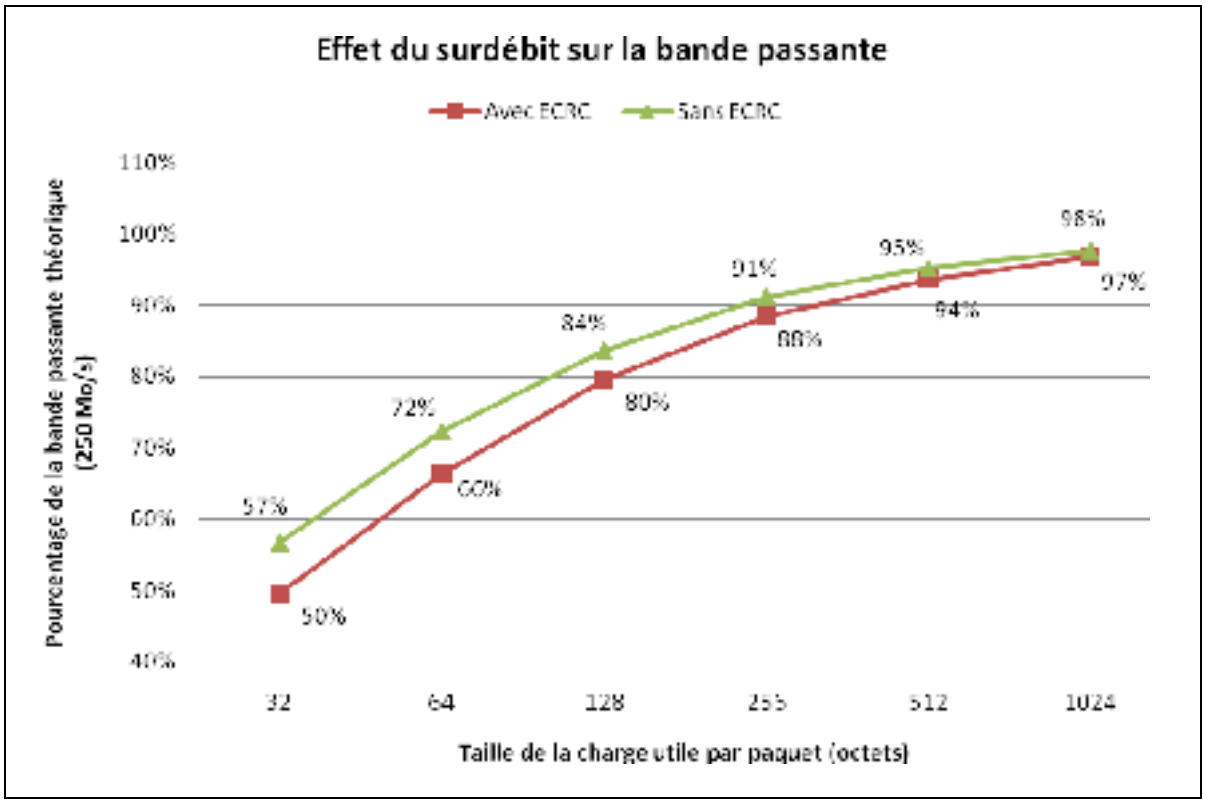

Figure 7.1 Effet du surdébit sur la bande passante

L'ajout du ECRC a un impact négatif sur la performance. En effet, plus la taille de la charge utile par paquet est faible, plus la performance diminue. Pour une bonne performance, la taille de la charge utile de données doit être égale à la valeur maximale permise (MPS : *Maximum Payload Size*).

Comme il a été mentionné dans la section 5.2.2, pour la première génération du protocole PCI Express, la bande passante théorique est de 2,5 Gbits/s (250 Mo/s). Cela représente deux fois et demie la bande passante offerte actuellement pour un réseau AFDX (section 2.5.1). Ainsi, même si certains éléments affectent la performance d'un réseau PCI Express, la bande passante atteinte peut demeurer acceptable, selon les exigences désirées.

7.2.2 Taille maximale de la charge utile (MPS : *Maximum Payload Size*)

La taille maximale de la charge utile détermine le nombre de paquets nécessaires pour transmettre une certaine quantité d'information. Pour ce test, l'Endpoint fait l'écriture de 16 ko de données dans la mémoire principale. D'une simulation à l'autre, nous faisons varier la valeur MPS de 128 à 2048 octets. Plus la taille des paquets est petite, plus élevé sera le nombre de paquets nécessaire pour transférer la totalité des données (16 ko). Le Tableau 7.2 identifie les valeurs MPS et le nombre correspondant de paquets nécessaires pour transmettre 16 ko de données (Nombre de paquets * MPS = 16 ko).

Tableau 7.2 Nombre de paquets
selon la valeur MPS

MPS (octets)	**Nombre de paquets**
128	128
256	64
512	32
1024	16
2048	8

La Figure 7.2 illustre graphiquement l'effet de la taille maximale de la charge utile sur la bande passante.

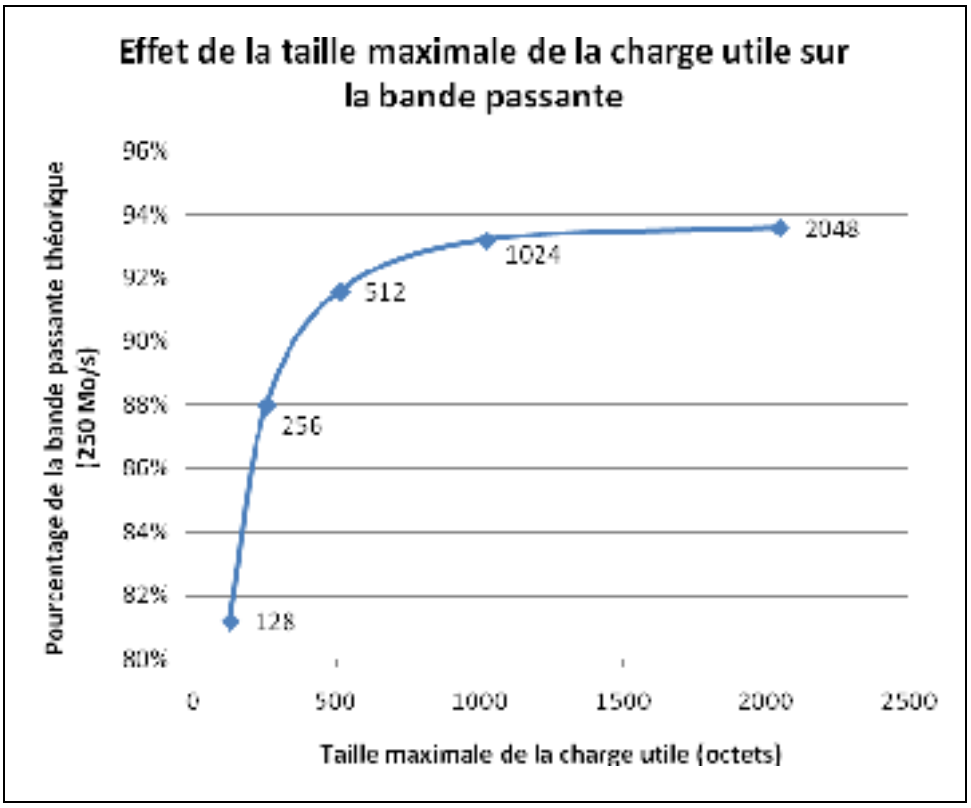

Figure 7.2 Effet de la taille maximale de la charge utile sur la bande passante

Plus la valeur MPS est faible, plus de paquets seront nécessaires pour transmettre la même quantité de données. Par conséquent, le surdébit associé à chacun des TLPs fera diminuer l'efficacité du transfert. Pour une performance optimale, il est préférable d'utiliser la valeur MPS la plus élevée possible.

7.2.3 Disponibilité des crédits du contrôle de flux

À la section 5.2.5, il a été mentionné que le nombre de crédits disponibles est directement relié à la taille de la mémoire tampon des ports de réception. Pour ne pas nuire au déterminisme, les crédits disponibles doivent être suffisants pour masquer la latence introduite par le mécanisme du contrôle de flux.

Ce test a été effectué en faisant varier le nombre de crédits de réponse (*Completion Data Credit*). Le module PCI Express utilisé permet peu de flexibilité au niveau de l'ajustement du nombre de crédits. À la création du module PCI Express, nous pouvons sélectionner l'un des trois choix prédéfinis. Le choix est fait selon la performance désirée. En ce qui concerne les crédits de réponse, les choix sont les suivants :

- faible : 24 crédits,
- moyen : 51 crédits,
- élevé : 112 crédits.

Sur la Figure 7.3 nous pouvons observer l'effet du nombre de crédits de réponse sur la performance.

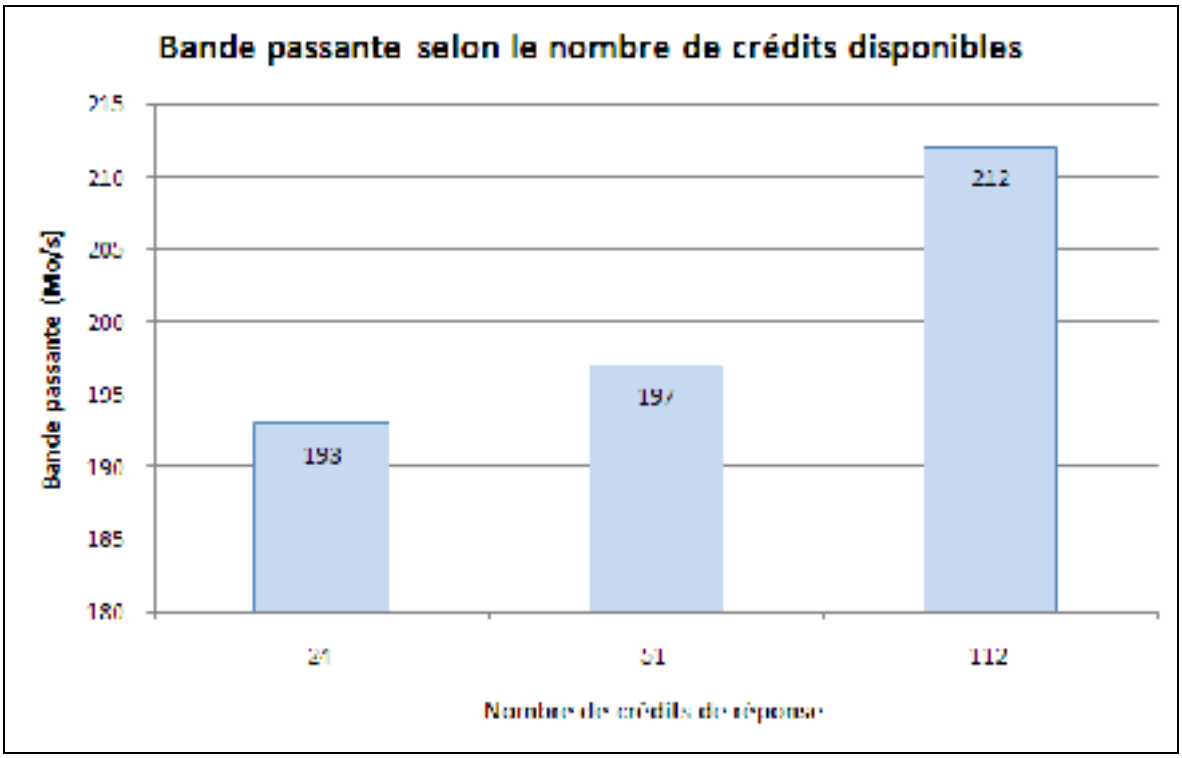

Figure 7.3 Bande passante selon le nombre de crédits disponibles

Un nombre insuffisant de crédits ne permet pas d'atteindre la performance optimale. Pour ce test, la bande passante maximale est de 212 Mo/s. Selon la performance désirée, l'ajustement de ce paramètre doit être fait de façon judicieuse pendant la phase de création du composant. Par exemple, un concepteur devrait toujours choisir le nombre maximal de crédits que la taille de la mémoire le permet. Dans le cas qui nous concerne, cet ajustement serait de 112 crédits.

Ce paramètre a un impact au niveau du déterminisme. En effet, le mécanisme de contrôle de flux introduit une certaine latence dans le transfert de données entre deux points du réseau. Le nombre de crédits disponibles doit toujours être suffisant pour masquer cette latence. Il en revient au concepteur de prendre les précautions nécessaires pour que le nombre de crédits disponibles soit toujours suffisant.

7.2.4 Taille du tampon de retransmission

La taille du tampon de retransmission a un impact sur la performance du système. En effet, un tampon trop petit ne permet pas d'atteindre le maximum de la bande passante. Ce test a été effectué avec le contrôleur DMA en écriture. En considérant des paquets de 128 octets, nous pouvons ajuster la taille du tampon de retransmission de 256 octets à 16 ko. Le Tableau 7.3 montre les résultats obtenus.

Tableau 7.3 Bande passante selon la taille du tampon de retransmission

Taille du tampon de retransmission (octets)	Bande passante (Mo/s)
256	102
512	203

Les tests de simulation ont montré que la bande passante atteint sa valeur maximale lorsque le tampon de retransmission à une taille de 512 octets. Ces résultats peuvent varier d'un système à l'autre. En effet, la vitesse avec laquelle les paquets sont acquittés est variable. L'étude du protocole ACK/NAK du système permettra de faire le bon choix quant à la taille du tampon de retransmission.

Ici encore, la taille du tampon de retransmission a un impact sur le déterminisme du système. Un tampon trop petit peut provoquer des pauses durant un transfert de données. Si les données envoyées ne sont pas acquittées assez rapidement, le tampon devient saturé et la transmission ne peut plus continuer. Le concepteur du système doit bien ajuster ce paramètre pour conserver le déterminisme de la communication. À titre d'exemple, pour un système similaire à notre architecture de tests, un concepteur n'a pas intérêt à prendre un tampon dont la taille est supérieure à 512 octets. En effet, de la mémoire serait consommée inutilement.

7.2.5 Nombre d'étiquettes (*TAG*)

La performance du système dépend aussi du nombre d'étiquettes disponibles. Comme il a été mentionné dans la section 5.2.7, un composant ne peut effectuer plus de requêtes simultanées que le nombre d'étiquettes disponibles. Le contrôleur DMA en lecture a été utilisé pour effectuer ce test. La Figure 7.4 illustre les résultats obtenus.

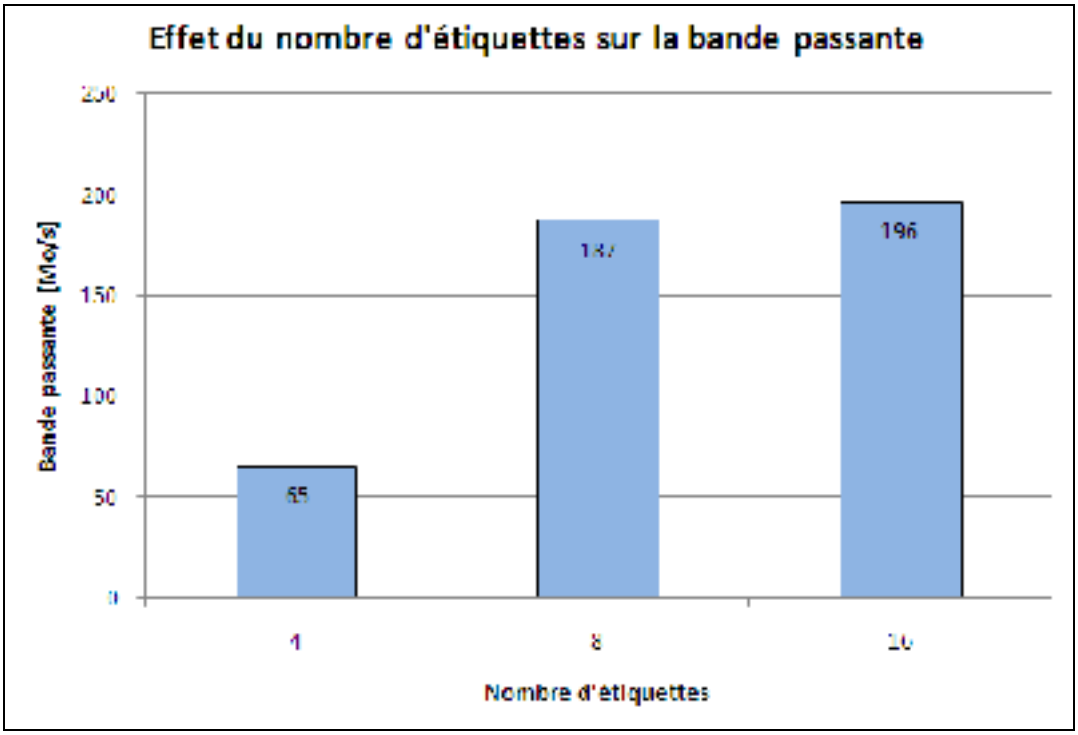

Figure 7.4 Effet du nombre d'étiquettes sur la bande passante

Un nombre insuffisant d'étiquettes ne permet pas d'atteindre la bande passante maximale. Par exemple, si quatre étiquettes sont utilisées, un composant ne peut faire plus de quatre requêtes simultanées. Le composant devra attendre l'arrivée des paquets de réponses pour réutiliser les mêmes numéros d'étiquettes. Le nombre d'étiquettes doit être suffisant pour qu'un composant puisse soumettre des requêtes en continu.

Comme pour la taille du tampon de retransmission, le nombre d'étiquettes a un impact sur le déterminisme. Un nombre insuffisant d'étiquettes provoquera des pauses durant un transfert de données. Le nombre d'étiquettes doit toujours être suffisant pour permettre un transfert de données en continu. Il en revient au concepteur de prendre les précautions nécessaires pour ne pas nuire au caractère déterministe du système. Pour une architecture similaire à la nôtre, un nombre de 16 étiquettes est suffisant.

7.3 Résultats expérimentaux sur plateforme matérielle

Pour les simulations de la section précédente, c'est le module PCI Express *Soft IP* qui a été utilisé, car celui-ci offre une certaine flexibilité au niveau de la configuration. Pour la partie expérimentale, c'est le module *Hard IP* qui a été utilisé. Ce dernier fait partie intégrante du FPGA. Par conséquent, le module *Hard IP* utilise peu de ressources en termes d'éléments logiques. Cependant, le module *Hard IP* ne permet pas autant de flexibilité que le module *Soft IP*.

7.3.1 Largeur du lien

Un lien PCI Express peut être composé de plusieurs voies. On retrouve des liens à 1, 2, 4, 8, 12, 16 et 32 voies. Le FPGA utilisé peut être configuré pour des liens à 1, 4 et 8 voies. Pour effectuer ce test, le contrôleur DMA en écriture a été utilisé. Sur la Figure 7.5, nous pouvons observer la bande passante selon le nombre de voies par lien.

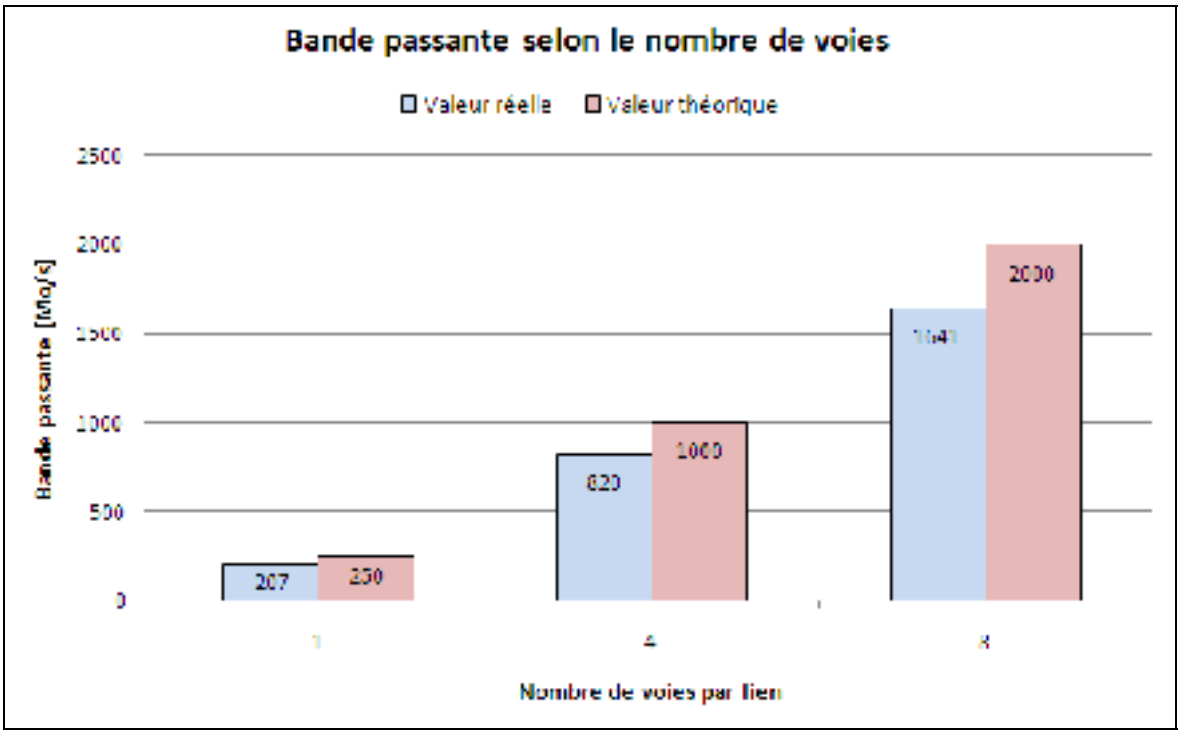

Figure 7.5 Bande passante selon le nombre de voies par lien

Sur le graphique, nous remarquons un écart entre les valeurs théoriques et réelles. La valeur réelle de la bande passante ne peut jamais égaler la valeur théorique. En effet, certains éléments comme le surdébit (DLLPs et PLPs) ont un effet sur la bande passante. Les éléments affectant la performance sont identifiés dans la section 5.2.

7.3.2 Taille maximale d'une requête de lecture

La taille maximale d'une requête de lecture ne dépend pas du composant qui fait la requête. En effet, le paramètre RCB (*Read Completion Boundary*) dépend du répondeur. Lorsque le répondeur est le Root Complex, cette valeur est toujours de 64 ou 128 octets (voir section 5.2.4).

L'appareil sur lequel les tests ont été effectués possède un Root Complex dont le RCB est de 64 octets. Pour ce test, le contrôleur DMA a été utilisé. Lorsque l'Endpoint fait une écriture

dans la mémoire principale, les paquets de données ont une taille maximale de 128 octets. Il s'agit de la valeur MPS (*Maximum Payload Size*). Par contre, lorsque l'Endpoint fait une lecture dans la mémoire principale, la taille maximale des paquets de données est de 64 octets. Il s'agit de la valeur RCB du Root Complex. La Figure 7.6 et la Figure 7.7 illustrent respectivement une requête d'écriture et la réponse d'une requête de lecture dans la mémoire principale. Ces chronogrammes proviennent de l'analyseur logique *Signal Tap II*.

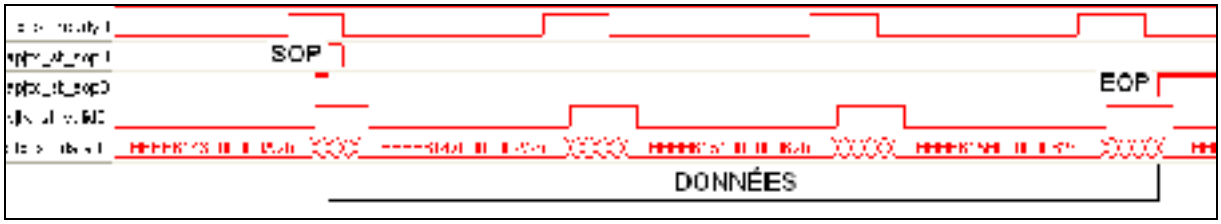

Figure 7.6 Requête d'écriture dans la mémoire principale

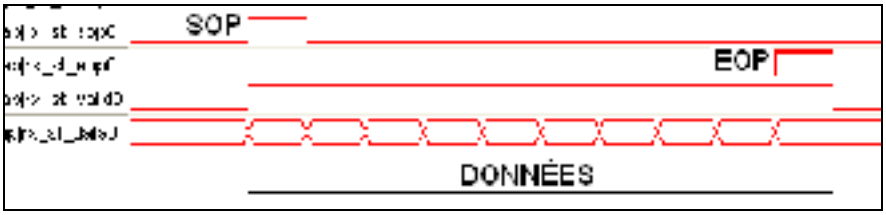

Figure 7.7 Réponse d'une requête de lecture dans la mémoire principale

Sur les deux chronogrammes, nous pouvons observer les signaux SOP (*Start Of Packet*) et EOP (*End Of Packet*) qui délimitent le paquet de données. En agrandissant l'image adéquatement, il est possible de voir toutes les données des paquets. Pour la requête d'écriture, le paquet contient 128 octets de données. En ce qui concerne la réponse de la requête de lecture, le paquet contient 64 octets de données.

Le Tableau 7.4 indique la bande passante atteinte par le contrôleur DMA.

Tableau 7.4 Bande passante atteinte
par le contrôleur DMA

Type de requête	**Bande passante (Mo/s)**
Lecture	179
Écriture	207

L'efficacité du TLP dépend de la taille des paquets. Plus la taille d'un paquet est petite, plus l'efficacité diminue. Ceci explique la différence de performance entre une lecture et une écriture en mémoire principale. En augmentant la valeur maximale de la charge utile de données, nous pourrions améliorer la performance en écriture. Cependant, la performance en lecture ne peut être augmentée, car la taille des paquets de réponses demeure fixe à 64 octets (RCB).

7.3.3 Insertion d'erreurs

Il existe différents types d'erreurs pouvant survenir lors d'un transfert de données. Afin d'étudier l'effet des erreurs sur une liaison PCI Express, le CRC de liaison (LCRC) sera mis à contribution. Puisqu'un LCRC en particulier est valide seulement sur une liaison point à point, ce test est effectué sur la liaison entre l'Endpoint et le commutateur (voir Figure 7.8).

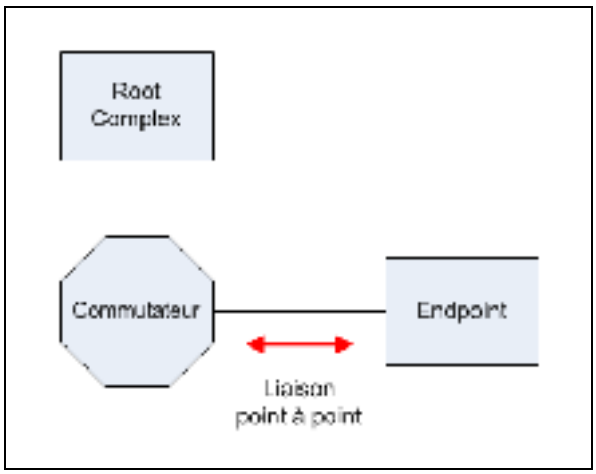

Figure 7.8 Test sur une liaison PCI Express

L'insertion d'erreurs permet d'étudier les mécanismes de fiabilité du protocole. Pour ce test, l'Endpoint fait une requête de lecture au Root Complex. Trois scénarios sont exécutés.

Scénario 1

L'Endpoint fait une requête de lecture au Root Complex. L'endpoint reçoit la réponse du Root Complex sans aucune erreur.

Scénario 2

Une erreur de type LCRC est injectée dans la requête de l'Endpoint. Lorsque le commutateur détecte cette erreur, il en avertit l'Endpoint. La requête est envoyée de nouveau par l'Endpoint.

Pour injecter une erreur de type LCRC à la transmission, il faut activer le bit 23 du signal test_in. Le chronogramme de la Figure 7.9 identifie la séquence des événements rapportée par le signal test_out.

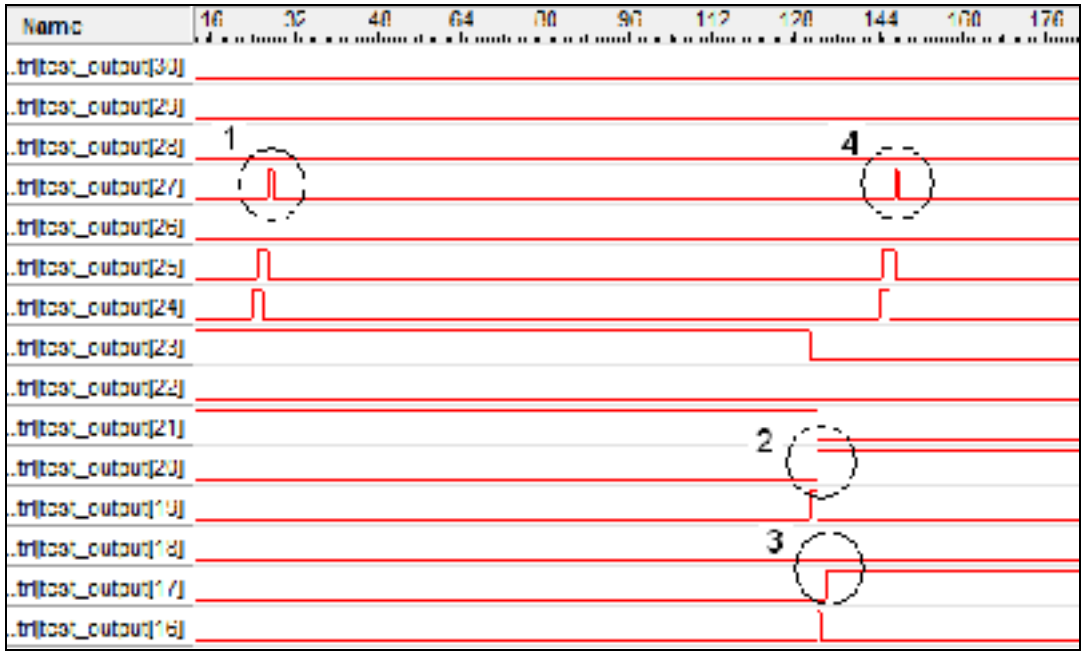

Figure 7.9 Chronogramme du scénario 2

Sur le chronogramme, la séquence des événements est identifiée par un numéro.

1) Le bit 27 indique l'ajout du LCRC au paquet de données à transmettre. Ensuite, le TLP est transféré à la couche physique qui transmet le paquet sur le réseau.

2) Le bit 20 indique la réception d'un DLLP de type NAK (*Not Acknowledged*). Ce signal indique à l'Endpoint que le commutateur a détecté une erreur dans le paquet. L'Endpoint doit donc retransmettre son paquet.

3) Les bits 17 et 18 sont les bits d'un compteur qui incrémente à chaque tentative de transmission.

4) Le bit 27 indique l'ajout du LCRC au paquet de données à transmettre. Puisque le LCRC contenait une erreur, on doit recommencer à cette étape.

Scénario 3

L'Endpoint fait une requête de lecture au Root Complex. Lorsque la réponse revient, une erreur de type LCRC est détectée par l'Endpoint. Ce dernier avise le commutateur qui envoie la réponse de nouveau.

Pour forcer la détection d'une erreur de type LCRC à la réception, il faut activer le bit 16 du signal test_in. Le chronogramme de la Figure 7.10 identifie la séquence des événements rapportée par le signal test_out.

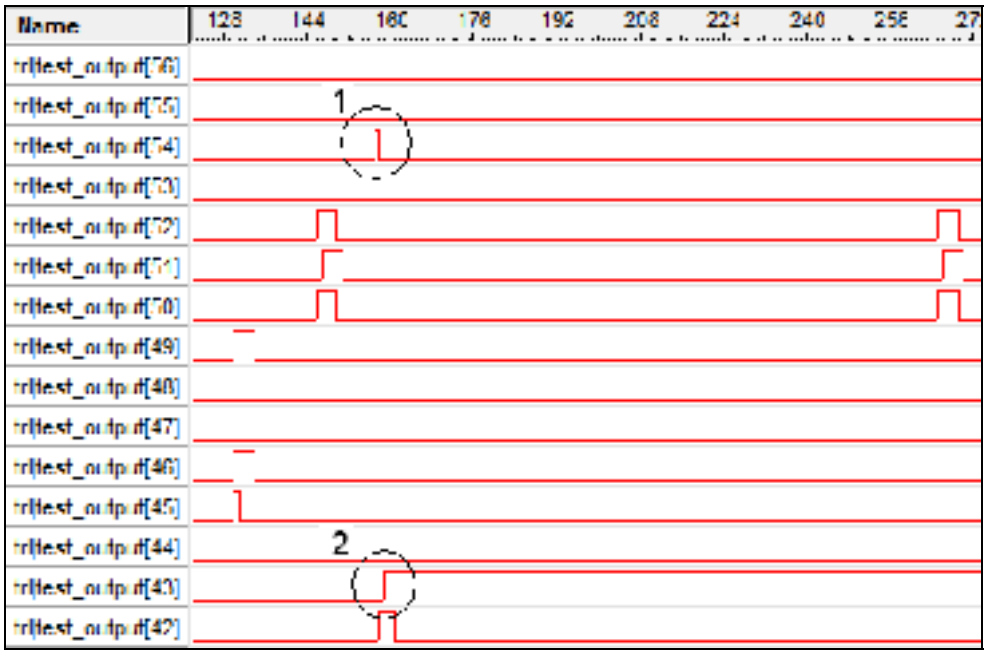

Figure 7.10 Chronogramme du scénario 3

Sur le chronogramme, la séquence des événements est identifiée par un numéro.

1) Le bit 54 indique la réception d'un TLP avec une erreur de type LCRC.

2) Le bit 43 indique l'envoi d'un DLLP de type NAK au commutateur. Ce dernier devra retransmettre son paquet.

Le Tableau 7.5 identifie la latence aller-retour pour chacun des scénarios. Il s'agit du délai entre l'envoi de la requête de lecture au Root Complex et la réception de la réponse par l'Endpoint. La latence est le résultat de la moyenne de cinq essais.

Tableau 7.5 Latence aller-retour pour chacun des scénarios

Scénario	**Latence aller-retour (ns)**
1 : Sans erreur	1296
2 : Erreur LCRC à la transmission	2296
3 : Erreur LCRC à la réception	2240

Lorsqu'une erreur est détectée, un délai supplémentaire de 1000 ns environ (voir Tableau 7.5) est nécessaire pour la correction de cette erreur. De plus, pour une erreur de type LCRC à la transmission ou à la réception, la latence est sensiblement la même.

Ce test permet de valider le bon fonctionnement du mécanisme de détection d'erreurs du protocole. Les mécanismes de détection d'erreurs font partie des exigences des bus de données avioniques.

7.4 Présentation du programme

Cette section présente le programme ayant servi pour exécuter les tests sur la plateforme matérielle. Ce programme a été conçu en langage C dans l'environnement Visual Studio de Microsoft.

Au démarrage, le programme charge le pilote PCI Express de la carte et affiche un menu de sélection sur le moniteur. La Figure 7.11 illustre le menu principal du programme.

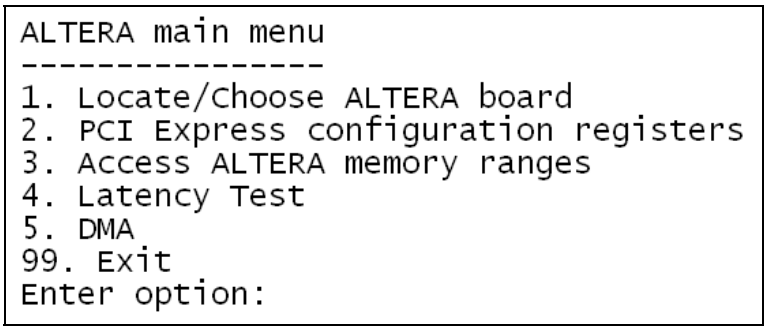

Figure 7.11 Menu principal du programme

Voici une brève description de chacun des choix du menu principal.

Locate/Choose ALTERA board

Permets de rechercher une carte dans le système, car il est possible d'avoir plus d'une carte Altera simultanément. En entrant les numéros de *Vendor ID* et *Device ID*, le programme se « connecte » à la carte dans le but d'effectuer les tests demandés.

PCI Express configuration registers

Permets d'afficher les informations relatives aux registres de configuration de la carte. Par exemple, les numéros de *Vendor ID* et *Device ID* sont contenus dans ces registres, de même que les adresses de registres de base (BAR).

Access ALTERA memory ranges

Permets de lire et écrire dans la mémoire de l'Endpoint. Il est possible de choisir l'espace mémoire désiré (BAR) et un décalage (*offset*).

Latency Test

Exécute un test de latence aller-retour (*round-trip latency*) vers l'Endpoint. Nous pouvons sélectionner l'un des trois choix suivants :

- opération normale,
- insertion d'une erreur LRCR à la transmission,
- force la détection d'une erreur LCRC à la réception.

À la suite de l'opération, le temps de latence s'affiche en nanosecondes.

DMA

Exécute un transfert DMA entre l'Endpoint et le Root Complex. Un sous-menu permet de sélectionner l'opération désirée : lecture seule, écriture seule, lecture puis écriture, écriture puis lecture et finalement lecture et écriture simultanément. Ensuite, nous devons choisir la longueur du transfert (octets) et le nombre d'itérations. Au terme de l'opération, le débit s'affiche sur le moniteur (en Mo/s).

7.4.1 Démonstration

Voici une démonstration de résultats obtenus avec le programme. Ce test permet de constater que le type d'opération effectué sur un lien influence grandement la performance. Pour cette démonstration, le contrôleur DMA a été utilisé pour transférer 16 ko de données. Dans un premier temps, les opérations de lecture et écriture ont été effectuées l'une à la suite de l'autre. Ensuite, les deux opérations ont été effectuées simultanément. Les résultats obtenus sont indiqués dans le Tableau 7.6.

Tableau 7.6 Démonstration du contrôleur DMA

Opération	**Débit en lecture (Mo/s)**	**Débit en écriture (Mo/s)**
Lecture puis écriture	171	205
Lecture et écriture simultanée	143	185

Bien que la lecture et l'écriture se fassent sur deux liens distincts, nous pouvons remarquer une diminution de performance lorsque les deux opérations sont effectuées simultanément. Ceci est dû à la présence de surdébit associé aux opérations. En effet, l'Endpoint et le Root Complex s'échangent continuellement des DLLPs pendant le transfert. Certains DLLPs sont utilisés par les acquittements. D'autres DLLPs servent à mettre à jours le nombre de crédits disponibles de part et d'autre de la liaison. Ce surdébit, combiné aux paquets de données, fait diminuer la performance.

CONCLUSION

Rappelons que l'objectif principal de cette recherche était de caractériser le protocole PCI Express en considérant les exigences imposées par l'industrie avionique. Les systèmes temps réel à sécurité critique et leurs bus de données requièrent des exigences particulières quant au déterminisme et à la fiabilité. Par conséquent, ces critères fondamentaux ont été au cœur de ce travail de recherche.

Comme il a été mentionné précédemment, le déterminisme est défini comme étant la capacité du réseau à garantir la livraison d'un message dans un temps prévisible. Ceci implique que la bande passante soit garantie et que la latence et la gigue soient connues et bornées. Grâce au mode isochrone du protocole, le PCI Express offre une bande passante garantie et une latence déterministe. Quant à la gigue, le taux d'erreur binaire qui lui est associé doit avoir une valeur maximale de 10^{-12}. Il a été mentionné à la section 3.2.5 que cette valeur est conforme aux exigences de l'industrie avionique.

Cette recherche a fait ressortir plusieurs paramètres ayant un impact sur la performance du protocole PCI Express. Ces paramètres sont :

- la largeur du lien,
- le surdébit (*Overhead*),
- la taille maximale de la charge utile (*Maximum Payload Size*),
- la taille maximale d'une requête de lecture,
- la disponibilité des crédits du contrôle de flux,
- la taille du tampon de retransmission,
- le nombre d'étiquettes (*TAG*).

Pour garantir le déterminisme tout en préservant une performance optimale, chacun de ces paramètres doit être ajusté judicieusement. Le CHAPITRE 7 a démontré de façon expérimentale l'importance de bien ajuster ces paramètres.

Concernant la fiabilité, un protocole de communication avionique doit être fiable et tolérant aux fautes. Ceci implique de savoir gérer efficacement les erreurs. Comme il a été mentionné au CHAPITRE 4, le protocole PCI Express définit deux mécanismes de gestion d'erreurs. Un mécanisme de base répond aux besoins minimums. Pour une plus grande fiabilité, un mécanisme avancé est disponible (AER : *Advanced Error Reporting*). Ces mécanismes de gestion d'erreurs incluent les éléments ci-dessous.

Contrôle de flux
Permets de prévenir les débordements dans les tampons de réception. Représente une garantie que les données envoyées seront reçues.

CRC de bout en bout (ECRC)
CRC optionnel, mais représente une protection supplémentaire contre les erreurs. Disponible avec le mécanisme avancé seulement.

CRC de liaison (LCRC)
CRC obligatoire pour tous les TLPs transmis.

Numéro de séquence
Représente une protection pour éviter la perte de paquets.

Encodage 8b/10b
Permets de détecter les erreurs de transmission.

Pour être fiable, un protocole de communication avionique doit être tolérant aux fautes. Cette recherche a présenté différentes techniques de redondance assurant la tolérance aux fautes. Le protocole PCI Express procure une redondance d'information grâce au CRC de liaison et au CRC de bout en bout. Une redondance physique est aussi disponible. En effet, la configuration redondante du pont non transparent (NTB) permet d'envoyer les mêmes

informations sur plus d'un réseau simultanément. Ici, nous pouvons faire un parallèle avec le protocole AFDX, avec lequel les données sont toujours envoyées sur deux réseaux différents.

Concernant le pont non transparent, celui-ci peut aussi être utilisé dans une configuration de protection. Rappelons que dans cette configuration, un système en veille observe l'état du système principal et prend la relève en cas de défaillance. Ainsi, même s'il ne fait pas partie des spécifications PCI Express, le pont non transparent représente un élément de fiabilité supplémentaire.

Le branchement à chaud est une autre propriété intéressante de la technologie PCI Express. En effet, lorsqu'une défaillance survient, le système peut demeurer opérationnel pendant le remplacement du composant défectueux.

Rappelons que le processus de certification de l'industrie avionique impose des contraintes sévères à tout système avionique. D'où l'importance de considérer les critères d'évaluation des bus de données avioniques présentés au Tableau 3.2. Cette liste de critères, bien que non exhaustive, est un bon point de départ dans une démarche visant à faire certifier un bus de données avionique.

Au terme de cette recherche, nous pouvons constater que le protocole PCI Express répond bien aux besoins des bus de données avioniques, tels que mentionnés à la section 3.3.1. Rappelons qu'il n'y a pas que la performance qui justifie le choix d'un bus de données. En plus des éléments déjà mentionnés (fiabilité et déterminisme), d'autres critères sont à considérer :

- coût,
- prise en charge,
- flexibilité et capacité d'évoluer.

Ces critères sont tous dépendants de la disponibilité des composants sur étagère (COTS). La technologie PCI Express, toujours en plein essor, offre de plus en plus de produits sur le marché. D'ailleurs, la troisième génération du PCI Express, qui vient tout juste de voir le jour, est une indication claire sur l'avenir prometteur de ce protocole. L'intégration du protocole PCI Express dans les systèmes avioniques représente une innovation technologique majeure pour le domaine aéronautique.

Ce travail avait comme objectif secondaire le développement d'une plateforme expérimentale dans le but de valider les concepts théoriques. Ainsi, plusieurs éléments concernant le déterminisme et la fiabilité ont pu être testés de façon expérimentale. Nous pouvons conclure que tous les tests se sont avérés conformes à la théorie. Ces tests démontrent l'importance d'effectuer des choix judicieux pendant la phase de conception d'un système utilisant la technologie PCI Express. Pour un système à sécurité critique, tel un système avionique, le bon ajustement de chacun des paramètres permettra de se conformer aux exigences des autorités de certification avionique.

En terminant, rappelons que la plateforme de développement a été conçue en considérant l'aspect de continuité de ce projet de recherche. En effet, la grande flexibilité de cette plateforme lui permettra de s'adapter facilement à d'autres projets ayant un objectif similaire, soit l'intégration de la technologie PCI Express dans les systèmes embarqués avioniques.

RECOMMANDATIONS

En considérant les exigences imposées par les autorités de certification du domaine aérospatial, le protocole PCI Express possède les qualités nécessaires pour être utilisé comme bus de données avionique. Cependant, l'intégration de cette technologie en aéronautique demande une investigation plus profonde. En effet, pour effectuer des tests exhaustifs, du matériel spécialisé est nécessaire.

Pour des résultats complets conformes aux spécifications PCI Express, un environnement de vérification est essentiel. La suite de logiciels Denali PureSuite (maintenant appartenant à Cadence) est un environnement de vérification fonctionnelle pour la technologie PCI Express. PureSuite contient une gamme complète de tests spécifiques aux fonctionnalités PCI Express. Ceci inclut tous les tests définis dans les documents de conformité du *PCI Special Interest Group* (PCI-SIG).

Concernant les appareils de tests et de mesures, les compagnies LeCroy et Agilent Technologies offrent des produits spécifiques à la technologie PCI Express. Pour une installation idéale en laboratoire, les produits suivants sont indispensables.

Analyseur de protocole PCI Express
Permets de capturer les transactions en temps réel pour en faire l'analyse.

Brouilleur PCI Express
Se positionne entre le Root Complex et l'Endpoint. Permets d'injecter des erreurs dans les transactions.

Exerciseur PCI Express
Carte permettant d'émuler un Endpoint ou un Root Complex.

Enfin, pour une conception personnalisée, plusieurs compagnies offrent des circuits intégrés PCI Express. Les compagnies les plus connues sont PLX Technology et IDT (Integrated Device Technology). Ces dernières offrent des circuits de commutation à plusieurs ports dont certains intègrent un pont non transparent (NTB).

En terminant, rappelons que cette étude s'est concentrée au niveau de la couche protocole uniquement. Il serait intéressant de poursuivre cette recherche à un niveau supérieur en incluant des processeurs et un système d'exploitation temps réel. Une application réelle de l'avionique pourrait s'exécuter sur la plateforme expérimentale. Nous pourrions ainsi étudier le comportement du protocole PCI Express dans un environnement représentatif des systèmes avioniques. Enfin, il serait pertinent de tester le mode isochrone avec le matériel adéquat.

BIBLIOGRAPHIE

Actel Corporation. 2005. « Developing AFDX Solutions », En ligne. <www.actel.com/documents/AFDX_Solutions_AN.pdf>. Consulté le 9 mars 2011.

Agilent Technologies. 2006. « PCI Express Performance Measurements », En ligne. <http://cp.literature.agilent.com/litweb/pdf/5989-4076EN.pdf>. Consulté le 2 octobre 2009.

Agilent Technologies. 2008. « Measuring Jitter in Digital Systems », En ligne. <http://cp.literature.agilent.com/litweb/pdf/5988-9109EN.pdf>. Consulté le 30 mars 2011.

Alena, R. L., J. P. Ossenfort, K. I. Laws, A. Goforth et F. Figueroa. 2007. « Communications for Integrated Modular Avionics ». In *Aerospace Conference, 2007 IEEE* (3-10 March 2007). p. 1-18.

Altera Corporation. 2010. « Altera's DO-254/ED-80 Certifiable Nios II Processor Leveraged in Thales Safety-Critical Avionics System Certified by EASA », En ligne. San Jose (USA): <http://www.altera.com/corporate/news_room/releases/2010/products/nr-do-254.html>. Consulté le 25 juillet 2011.

Bauer, H., J. L. Scharbarg et C. Fraboul. 2010. « Worst-case end-to-end delay analysis of an avionics AFDX network ». In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010* (8-12 March 2010). p. 1220-1224.

Bluff, R. J. 1999. « Integrated modular avionics: system modelling ». *Microprocessors and Microsystems,* vol. 23, n° 7, p. 435-448.

Budruk, Ravi, Don Anderson et Tom Shanley. 2004. *PCI express system architecture*. Boston: Addison-Wesley, 1049 pages.

Certification Authorities Software Team (CAST). 2003. *Position Paper CAST-16, Databus Evaluation Criteria*. 8 pages.

Coleman, Dave, Scott Gardiner, Mohammad Kolbehdari et Stephen Peters. 2004. *PCI Express Electrical Interconnect Design : Practical Solutions for Board-level Integration and Validation*. USA: Intel Press, 217 pages.

Condor Engineering Inc. 2005. *AFDX / ARINC 664 Tutorial*. Santa Barbara (USA), 47 pages.

Cooper, Steve. 2008. « PCIe over cable for high-speed I/O, bus expansion and networking », En ligne. <http://rtcmagazine.com/articles/view/101003>. Consulté le 13 août 2011.

Coupé, Cheryl. 2011. « Feel the Need for Speed ? : PCI Express performance enhancements offer tradeoffs for increased complexity ». *Engineers' Guide to PCI Express Solutions*, p. 6-7.

Federal Aviation Administration. 2005a. *Handbook for Ethernet-Based Aviation Databases: Certification and Design Considerations*. Washington DC (USA), 55 pages.

Federal Aviation Administration. 2005b. *Safety and Certification Approaches for Ethernet-Based Aviation Databuses*. Washington DC (USA), 124 pages.

Federal Aviation Administration. 2009. *Data Network Evaluation Criteria Report*. Washington DC (USA), 234 pages.

Gagea, L., et I. Rajkovic. 2008. « Designing devices for avionics applications and the DO-254 guideline ». In *Semiconductor Conference, 2008. CAS 2008. International* (13-15 Oct. 2008). Vol. 2, p. 377-380.

Goldhammer, Alex, et John Ayer Jr. 2008. « Understanding Performance of PCI Express Systems », En ligne. <http://www.xilinx.com/support/documentation/white_papers/wp350.pdf>. Consulté le 2 octobre 2009.

Granovsky, Ilya, et Elchanan Perlin. 2007. « Integration and System Verification of PCI Express IP », En ligne. <http://www.pcisig.com/developers/main/training_materials/get_document?doc_id=0d44b05e096b74159b568f22a6d2bf0ed9aa9ef8>. Consulté le 8 octobre 2009.

Grieu, Jérôme. 2004. « Analyse et évaluation de techniques de commutation Ethernet pour l'interconnexion des systèmes avionique ». Thèse de doctorat, Toulouse, Institut National Polytechnique de Toulouse, 145 pages.

Gudmundson, John. 2004. « Enabling Multi-Host System Designs with PCI Express Technology », En ligne. <http://www.plxtech.com/app/webroot/files/pdf/technical/expresslane/RTC_Enabling%20MulitHostSystemDesigns.pdf>. Consulté le 31 mars 2011.

Jungo Ltd. 2009a. « WinDriver PCI/ISA/CardBus User's Manual », En ligne. <http://www.jungo.com/st/support/documentation/windriver/10.3.1/wdpci_manual.pdf>. Consulté le 6 janvier 2011.

Jungo Ltd. 2009b. « WinDriver PCI/PCMCIA/ISA Low-Level API Reference », En ligne. <http://www.jungo.com/st/support/documentation/windriver/10.3.1/wdpci_low_level_api_ref.pdf>. Consulté le 6 janvier 2011.

Kazmi, Akber. 2009. « PCI Express Gen 3 Simplified », En ligne. <http://www.eetimes.com/design/communications-design/4008241/PCI-Express-Gen-3-Simplified>. Consulté le 18 novembre 2010.

Kornecki, A., B. Butka et J. Zalewski. 2008. « Software Tools for Safety-Critical Systems According to DO-254 ». *Computer,* vol. 41, n° 12, p. 112-115.

Kornecki, A., et J. Zalewski. 2008. « Software certification for safety-critical systems: A status report ». In *Computer Science and Information Technology, 2008. IMCSIT 2008. International Multiconference on* (20-22 Oct. 2008). p. 665-672.

Krig, Steve. 2003. « Initiatives and Technologies : PCI Express Provides Enterprise Reliability, Availability, and Serviceability », En ligne. <http://www.intel.com/technology/pciexpress/devnet/docs/RASWhitePaper.pdf>. Consulté le 18 mai 2009.

Laarouchi, Youssef. 2009. « Sécurités (immunité et innocuité) des architectures ouvertes à niveaux de criticité multiples : application en avionique ». Thèse de doctorat, Toulouse, Université de Toulouse, 181 pages.

Li, Mike. 2005. « Signal Integrity Compliance and Diagnostic Tests for PCI Express », En ligne. <http://www.pcisig.com/developers/main/training_materials>. Consulté le 30 mars 2011.

Lopez, J., P. Royo, C. Barrado et E. Pastor. 2008. « Modular avionics for seamless reconfigurable UAS missions ». In *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th* (26-30 Oct. 2008). p. 1.A.3-1-1.A.3-10.

Martin, Steven. 2004. « Maîtrise de la dimension temporelle de la qualité de service dans les réseaux ». Thèse de doctorat, Paris, Université Paris XII, 198 pages.

Moir, Ian, et Allan Seabridge. 2008. *Aircraft systems: mechanical, electrical, and avionics subsystems integration*, Third Edition. Chichester (England): John Wiley & Sons Ltd, 504 pages.

PCI-SIG. 2006. *PCI Express Base Specification Revision 2.0*. Oregon (USA): PCI-SIG, 608 pages.

PLX Technology. 2004. « Non-Transparent Bridging Simplified », En ligne. <http://www.plxtech.com/app/webroot/files/pdf/technical/expresslane/NTB_Brief_April-05.pdf>. Consulté le 31 mars 2011.

Prisaznuk, P. J. 2008. « ARINC 653 role in Integrated Modular Avionics (IMA) ». In *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th* (26-30 Oct. 2008). p. 1.E.5-1-1.E.5-10.

Rierson, L., et J. Lewis. 2003. « Criteria for certifying databuses on civil aircraft ». In *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd* (12-16 Oct. 2003). Vol. 1, p. 1.A.2-1.1-9 vol.1.

RTCA. 2005. *DO-297: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations*. Washington, DC: RTCA.

Schuster, T., et D. Verma. 2008. « Networking concepts comparison for avionics architecture ». In *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th* (26-30 Oct. 2008). p. 1.D.1-1-1.D.1-11.

TechSAT. 2009. *AFDX / ARINC 664 Tutorial*. Poing (Germany), 30 pages.

Watkins, C. B., et R. Walter. 2007. « Transitioning from federated avionics architectures to Integrated Modular Avionics ». In *Digital Avionics Systems Conference, 2007. DASC '07. IEEE/AIAA 26th* (21-25 Oct. 2007). p. 2.A.1-1-2.A.1-10.

Wilen, Adam, Justin Schade et Ron Thornburg. 2003. *Introduction to PCI Express: A Hardware and Software Developer's Guide*. USA: Intel Press, 325 pages.

Wilson, A., et T. Preyssler. 2009. « Incremental certification and Integrated Modular Avionics ». *Aerospace and Electronic Systems Magazine, IEEE,* vol. 24, n^{o} 11, p. 10-15.

Wolfig, R., et M. Jakovljevic. 2008. « Distributed IMA and DO-297: Architectural, communication and certification attributes ». In *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th* (26-30 Oct. 2008). p. 1.E.4-1-1.E.4-10.

Yogendhar, K, Vidhya Thyagarajan et Sriram Swaminathan. 2007. « Realizing the Performance Potential of a PCI-Express IP », En ligne. <http://www.design-reuse.com/articles/15900/realizing-the-performance-potential-of-a-pci-express-ip.html>. Consulté le 12 octobre 2009.

Zalewski, J., D. Trawczynski, J. Sosnowski, A. Kornecki et M. Sniezek. 2005. « Safety issues in avionics and automotive databuses ». In *IFAC World Congress*. Prague Czech Repuplic.