Attaques DMA peer-to-peer et contre-mesures

Fernand Lone Sang[1], Vincent Nicomette[1],
Yves Deswarte[1], Loïc Duflot[2]

[1]Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)
[2]Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)

SSTIC'2011

LAAS-CNRS

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
Premier ministre
Agence Nationale de la Sécurité des Systèmes d'Information

Contexte et problématique

Des difficultés à protéger efficacement un système informatique :

- complexité croissante
- surface d'attaque de plus en plus large

Les vecteurs d'attaque sur un ordinateur :

- logiciel malveillant s'exécutant sur le processeur
  - exploitation d'une vulnérabilité
    → débordements de tampon, chaînes de format, ...
  - abus de fonctionnalités du système
    → `/dev/kmem`, `/dev/mem`, chargeur de modules noyau, ...
- utilisation des mécanismes d'entrées-sorties (E/S)
  - mécanisme de type *Direct Memory Access* (DMA)
  - autres mécanismes (interruptions, SMBUS, ...)

Contexte et problématique

Un intérêt grandissant pour les attaques depuis les contrôleurs d'E/S :

- USB [Duflot 07, Maynor 05]
- FireWire [Dornseif 04, Becher 05, Boileau 06, Aumaitre 08]
- Ethernet [Triulzi 08, Triulzi 10, Duflot 10, Delugré 10]
- FPGA [Devine 09, Aumaitre 10]

Contexte et problématique

Un intérêt grandissant pour les attaques depuis les contrôleurs d'E/S :

- USB [Duflot 07, Maynor 05]
- FireWire [Dornseif 04, Becher 05, Boileau 06, Aumaitre 08]
- Ethernet [Triulzi 08, Triulzi 10, Duflot 10, Delugré 10]
- FPGA [Devine 09, Aumaitre 10]

Les conditions de mise en œuvre de ces attaques sont

- parfois méconnues
- la plupart du temps spécifiques à un *chipset* particulier

Un état des lieux de ces attaques sur les *chipsets* actuels :

- sur quels *chipsets* peuvent fonctionner ces attaques ?
- quels contrôleurs peuvent être sources ou cibles d'attaque ?
- quelles contre-mesures génériques permettent de s'en protéger ?

Sommaire de la présentation

Sommaire de la présentation

Mécanisme d'accès direct à la mémoire

Qu'est-ce que le mécanisme d'accès direct à la mémoire ?

- *Direct Memory Access* (DMA)
- mécanisme d'entrée-sortie permettant à un contrôleur
  - d'effectuer directement des transferts d'entrée-sortie vers la mémoire
  - de décharger le processeur de ces transferts d'entrée-sortie
- repose sur un moteur DMA (*DMA engine*)

Mécanisme d'accès direct à la mémoire

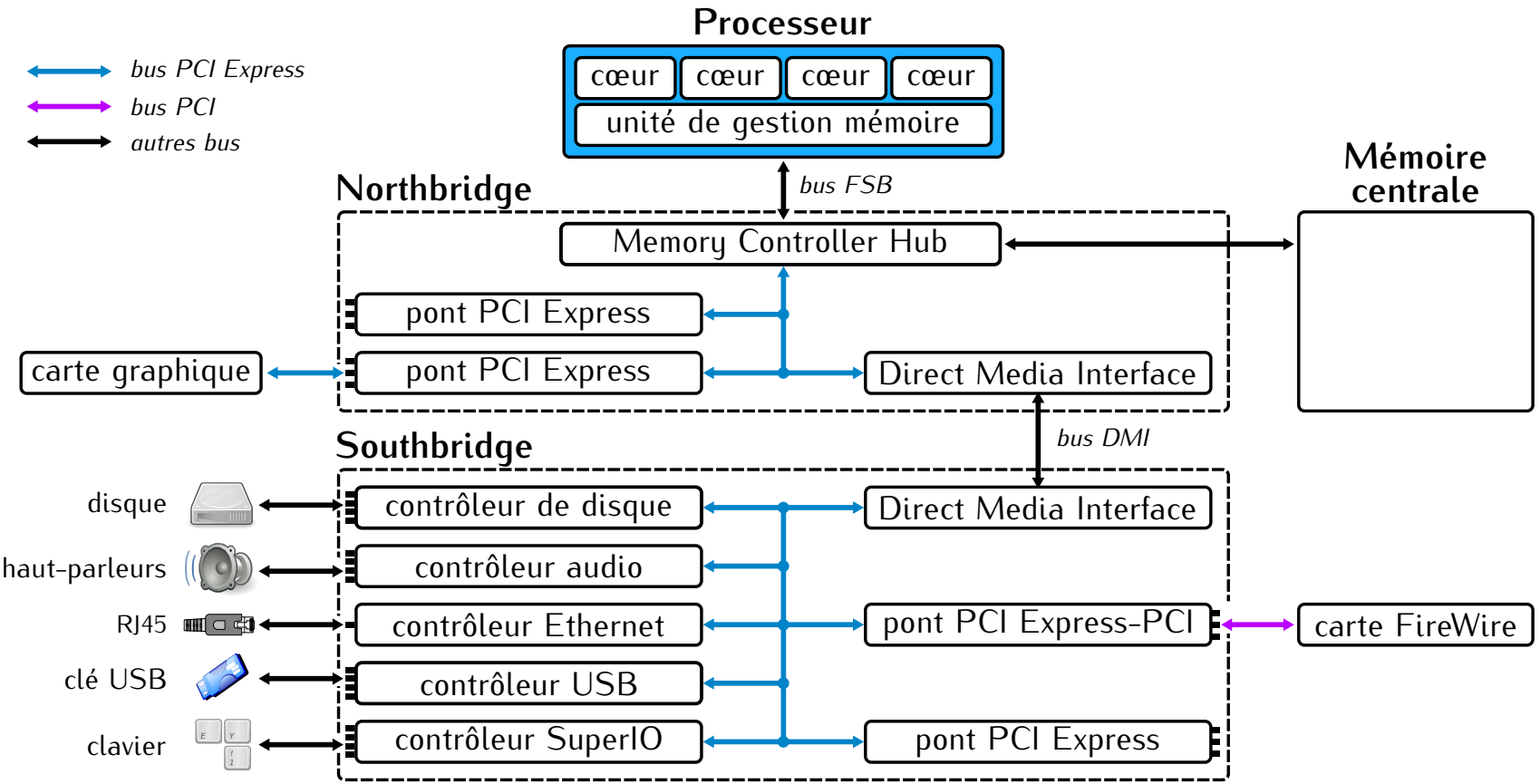
Qu'est-ce que le mécanisme d'accès direct à la mémoire ?

- *Direct Memory Access* (DMA)
- mécanisme d'entrée-sortie permettant à un contrôleur
  - d'effectuer directement des transferts d'entrée-sortie vers la mémoire
  - de décharger le processeur de ces transferts d'entrée-sortie
- repose sur un moteur DMA (*DMA engine*)

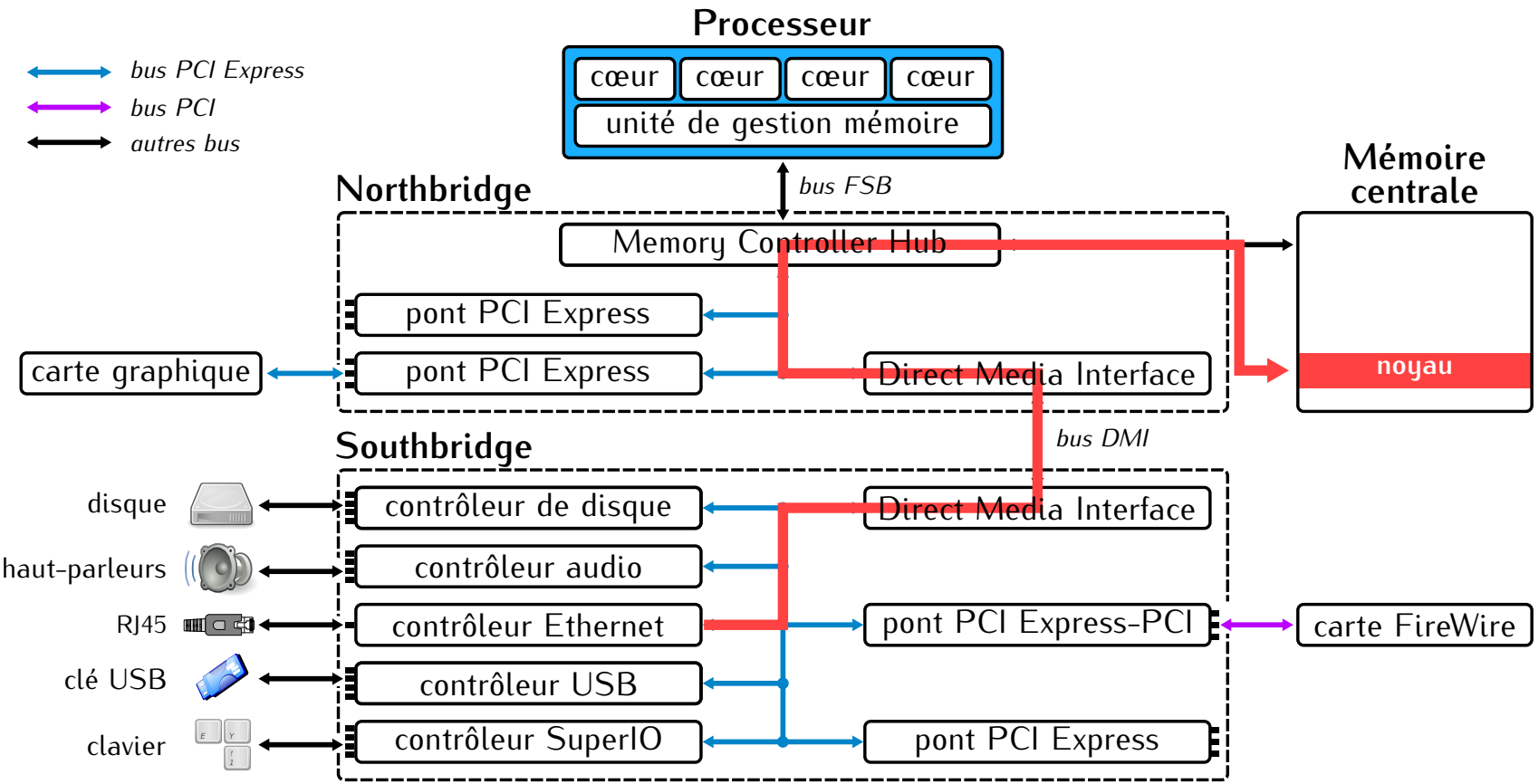
Exemples de contrôleurs utilisant le DMA :

- contrôleurs réseaux (WiFi, Ethernet, ...)
  → par exemple, pour transférer des trames réseaux
- contrôleurs de disques
  → par exemple, pour transférer des données (ex. fichiers)
- contrôleurs graphiques
  → par exemple, pour transférer des textures, des objets graphiques

Mécanisme d'accès direct à la mémoire

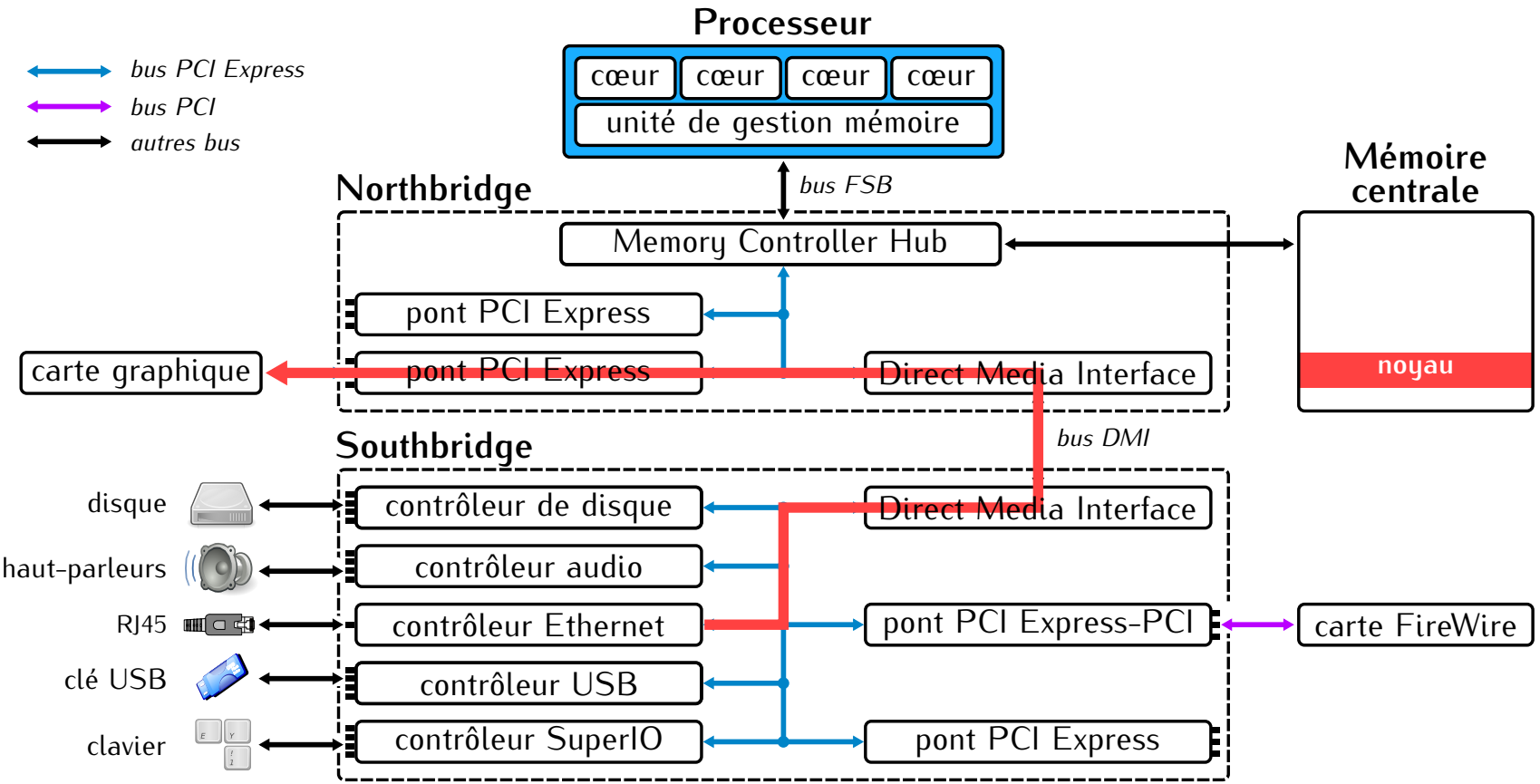


Mécanisme d'accès direct à la mémoire



Attaque DMA ciblant la mémoire centrale

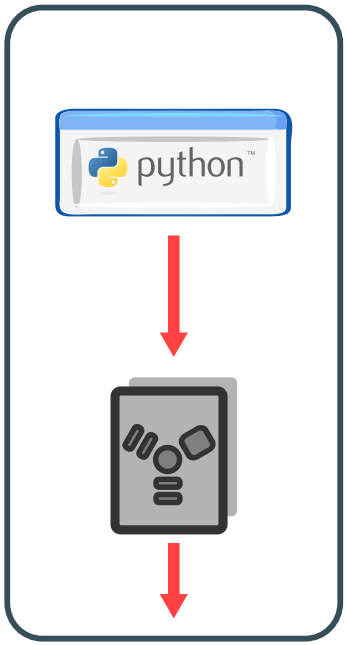
Mécanisme d'accès direct à la mémoire

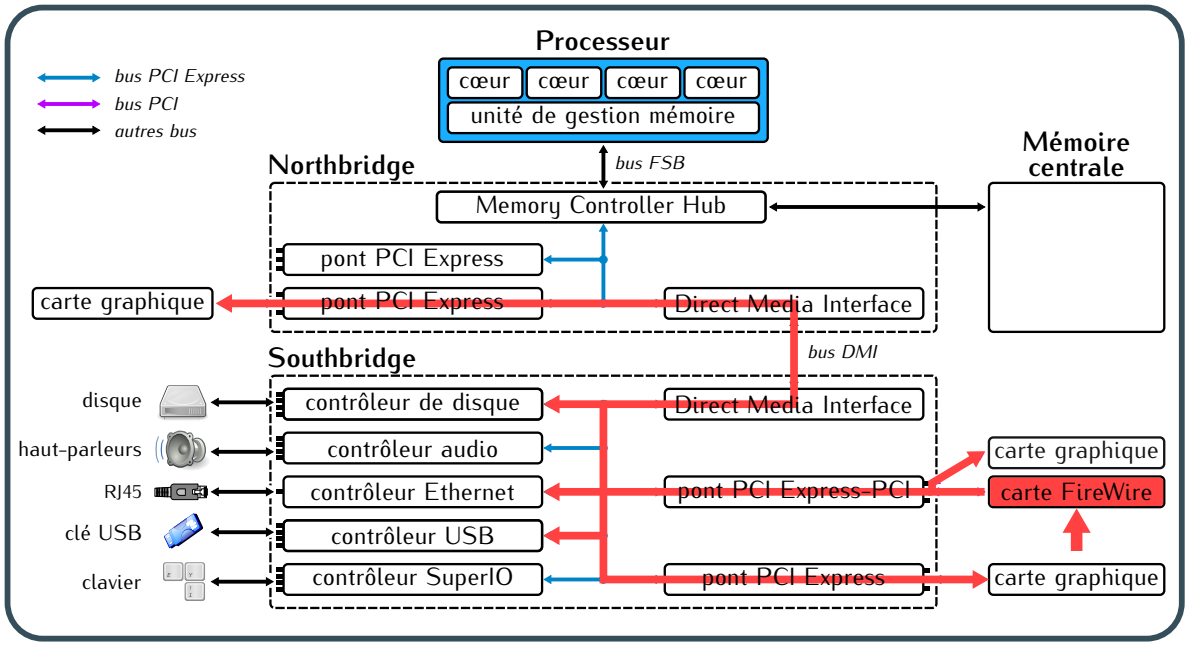


Attaque DMA ciblant la mémoire des contrôleurs
« attaque DMA *peer-to-peer* »

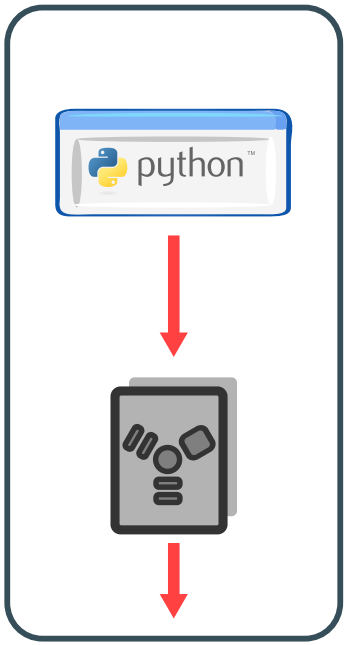
Sommaire de la présentation

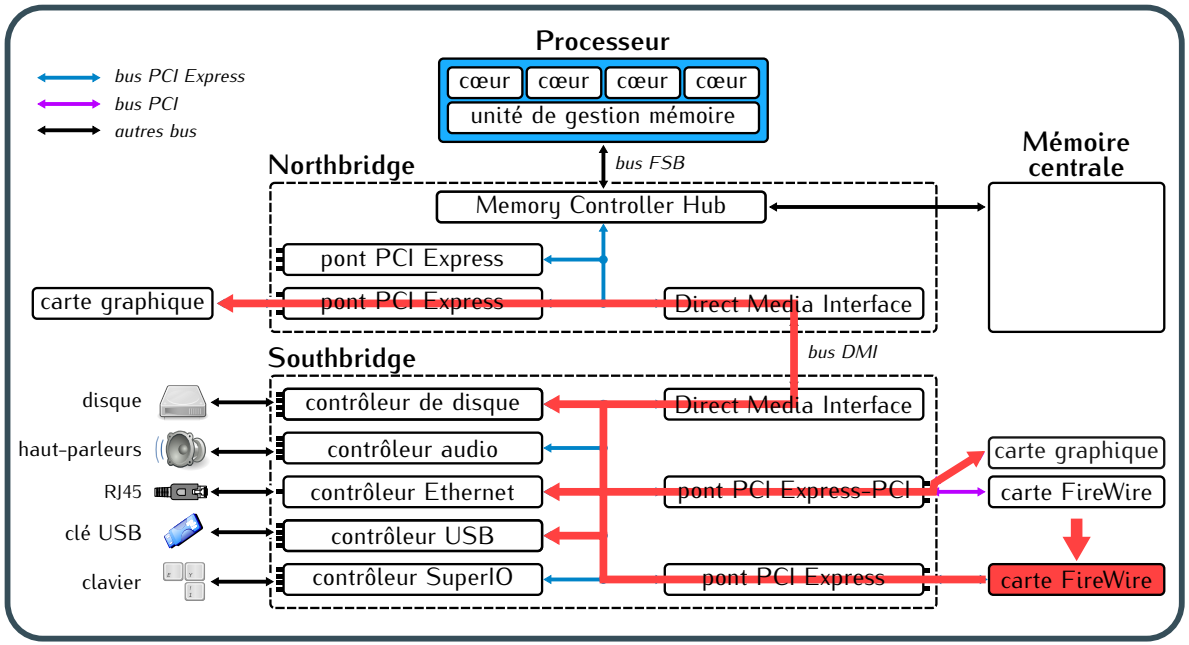
Plateforme d'expérimentation



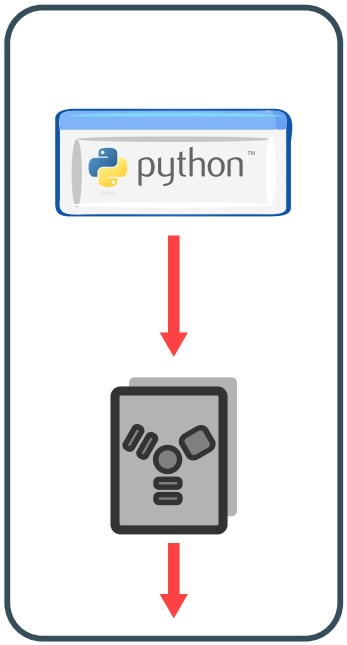


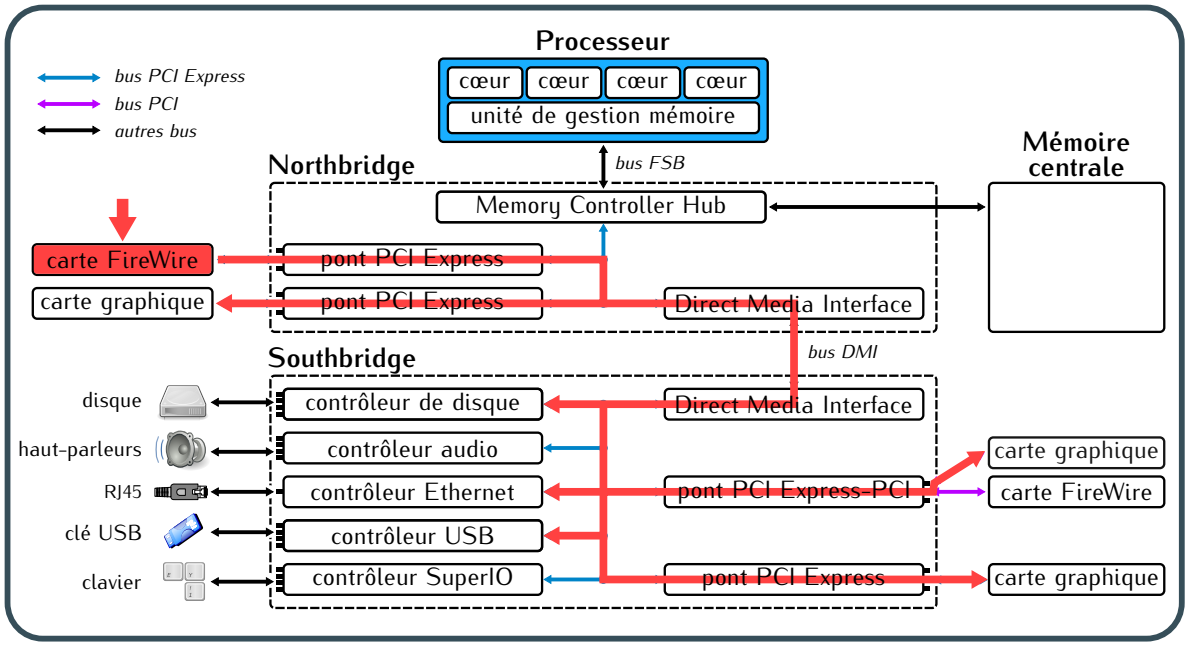
Plateforme d'expérimentation





Plateforme d'expérimentation





Plateforme d'expérimentation

Configuration matérielle des machines cibles :

	Chipset	Northbridge	Southbridge	Famille [1]	Modèle de machine
M0	Intel 945GM	GMCH 945GM	ICH7-M	Lakeport	MacBook Pro A1150 (*Laptop*)
M1	Intel Q45M	GMCH GM45	ICH9-M	Eaglelake	DELL Latitude E6400 (*Laptop*)
M2	Intel Q35	MCH 82Q35	ICH9	Eaglelake	DELL Optiplex 755 (*Desktop*)
M3	Intel Q45	MCH 82Q45	ICH10	Eaglelake	DELL Optiplex 960 (*Desktop*)
M4	Intel x58	IOH x58	ICH10	Tylersburg	Machine assemblée (*Desktop*)

Configuration logicielle des machines cibles :

- système d'exploitation GNU/Linux (64-bit, noyau recompilé)
- utilisation de `dd` pour copier la mémoire interne des contrôleurs
  - permet de vérifier si les accès DMA *peer-to-peer*, effectués par le contrôleur FireWire, ont réussi ou ont échoué

1. `http://ark.intel.com/#codenamesall`

Plateforme d'expérimentation

Configuration matérielle des machines cibles :

	Chipset	Northbridge	Southbridge	Famille [1]	Modèle de machine
M0	Intel 945GM	GMCH 945GM	ICH7-M	Lakeport	MacBook Pro A1150 (*Laptop*)
M1	Intel Q45M	GMCH GM45	ICH9-M	Eaglelake	DELL Latitude E6400 (*Laptop*)
M2	Intel Q35	MCH 82Q35	ICH9	Eaglelake	DELL Optiplex 755 (*Desktop*)
M3	Intel Q45	MCH 82Q45	ICH10	Eaglelake	DELL Optiplex 960 (*Desktop*)
M4	Intel x58	IOH x58	ICH10	Tylersburg	Machine assemblée (*Desktop*)

Configuration logicielle des machines cibles :

- système d'exploitation GNU/Linux (64-bit, noyau recompilé)
- utilisation de `dd` pour copier la mémoire interne des contrôleurs
  - permet de vérifier si les accès DMA *peer-to-peer*, effectués par le contrôleur FireWire, ont réussi ou ont échoué

Configuration logicielle/matérielle de la machine de l'attaquant :

- système d'exploitation GNU/Linux
- carte FireWire connectée à la machine cible par un cable FireWire
  - pour initier des accès DMA *peer-to-peer* sur la machine cible

1. `http://ark.intel.com/#codenamesall`

Résultats expérimentaux

		Eaglelake, Lakeport						Tylersburg					
Source \ Dest.		NB	SB					NB	SB				
		CPe	CU	CD	CP	CR	CPe	CPe	CU	CD	CP	CR	CPe
NB	FireWire PCIe	W	-	-	-	-	-	RW	RW	RW	RW	RW	RW
SB	FireWire PCIe	W	-	-	-	-	X	RW	-	-	-	-	-
	FireWire PCI	W	-	-	RW	-	-	RW	-	-	RW	-	-
	FireWire PCI*	W	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW

NB pour northbridge **SB** pour southbridge **CPe** pour contrôleur PCI Express **CU** pour contrôleur USB

CD pour contrôleur de disque **CP** pour contrôleur PCI **CR** pour contrôleur réseau pour contrôleur intégré

X pour non-applicable car le nombre de *slots* PCIe sont insuffisants **R** pour lecture réussie **W** pour écriture réussie

PCI* pour configuration du pont PCIe-PCI modifiée (bit *Peer Decode Enable* activé)

Résultats expérimentaux

		Eaglelake, Lakeport						Tylersburg					
Source \ Dest.		NB	SB					NB	SB				
		CPe	CU	CD	CP	CR	CPe	CPe	CU	CD	CP	CR	CPe
NB	FireWire PCIe	W	-	-	-	-	-	RW	RW	RW	RW	RW	RW
SB	FireWire PCIe	W	-	-	-	-	X	RW	-	-	-	-	-
	FireWire PCI	W	-	-	RW	-	-	RW	-	-	RW	-	-
	FireWire PCI*	W	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW

NB pour northbridge **SB** pour southbridge **CPe** pour contrôleur PCI Express **CU** pour contrôleur USB
CD pour contrôleur de disque **CP** pour contrôleur PCI **CR** pour contrôleur réseau pour contrôleur intégré
X pour non-applicable car le nombre de *slots* PCIe sont insuffisants **R** pour lecture réussie **W** pour écriture réussie
PCI* pour configuration du pont PCIe-PCI modifiée (bit *Peer Decode Enable* activé)

Analyse des résultats expérimentaux :

- concernant les *chipsets* Eaglelake, Lakeport :
  - dans le *northbridge*, accès *peer-to-peer* en écriture uniquement
  - dans le *southbridge*, accès *peer-to-peer* entre contrôleurs PCI
  - dans le *southbridge*, accès *peer-to-peer* depuis un contrôleur PCI
  - accès *peer-to-peer* en écriture du *southbridge* au *northbridge*

Résultats expérimentaux

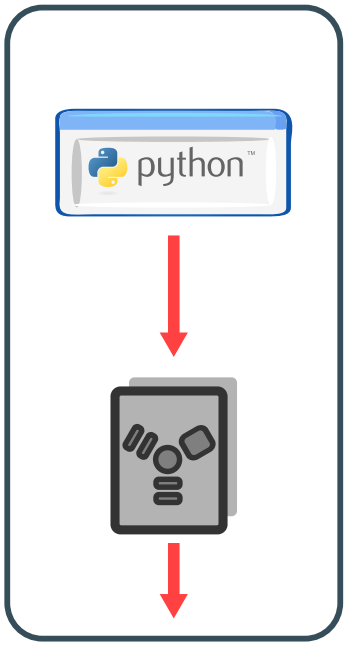
		Eaglelake, Lakeport						Tylersburg					
Source \ Dest.		NB	SB					NB	SB				
		CPe	CU	CD	CP	CR	CPe	CPe	CU	CD	CP	CR	CPe
NB	FireWire PCIe	W	-	-	-	-	-	RW	RW	RW	RW	RW	RW
SB	FireWire PCIe	W	-	-	-	-	X	RW	-	-	-	-	-
	FireWire PCI	W	-	-	RW	-	-	RW	-	-	RW	-	-
	FireWire PCI*	W	RW	RW	RW	-	RW	RW	RW	RW	RW	-	RW

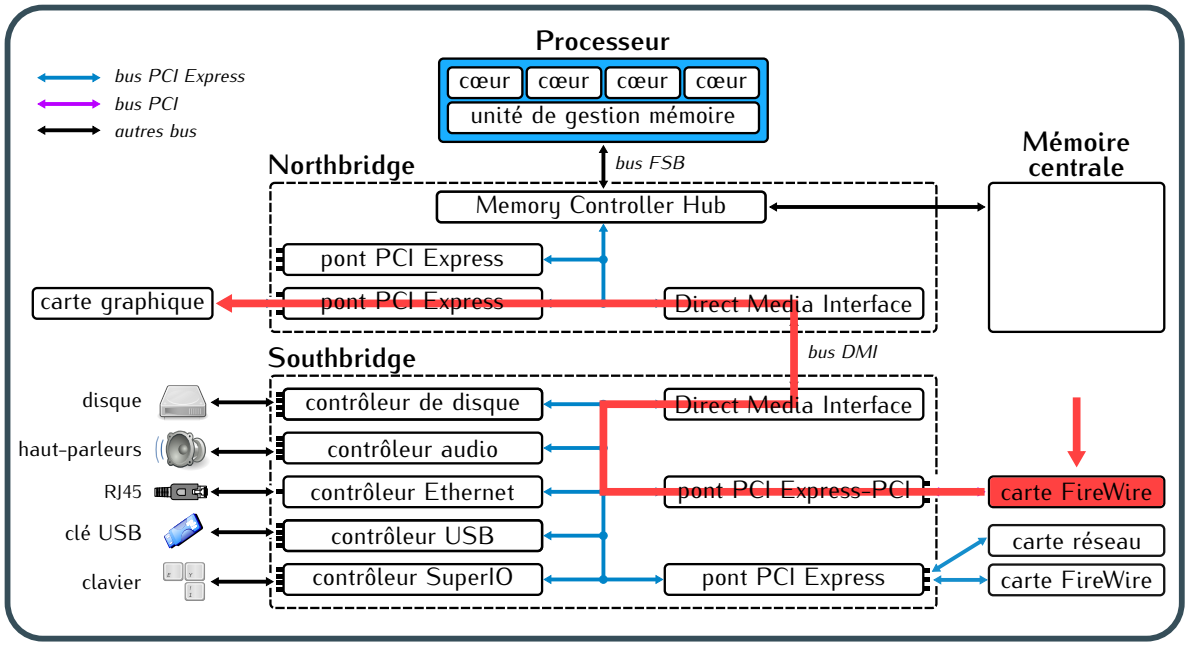
NB pour northbridge **SB** pour southbridge **CPe** pour contrôleur PCI Express **CU** pour contrôleur USB
CD pour contrôleur de disque **CP** pour contrôleur PCI **CR** pour contrôleur réseau pour contrôleur intégré
X pour non-applicable car le nombre de *slots* PCIe sont insuffisants **R** pour lecture réussie **W** pour écriture réussie
PCI* pour configuration du pont PCIe-PCI modifiée (bit *Peer Decode Enable* activé)

Analyse des résultats expérimentaux :

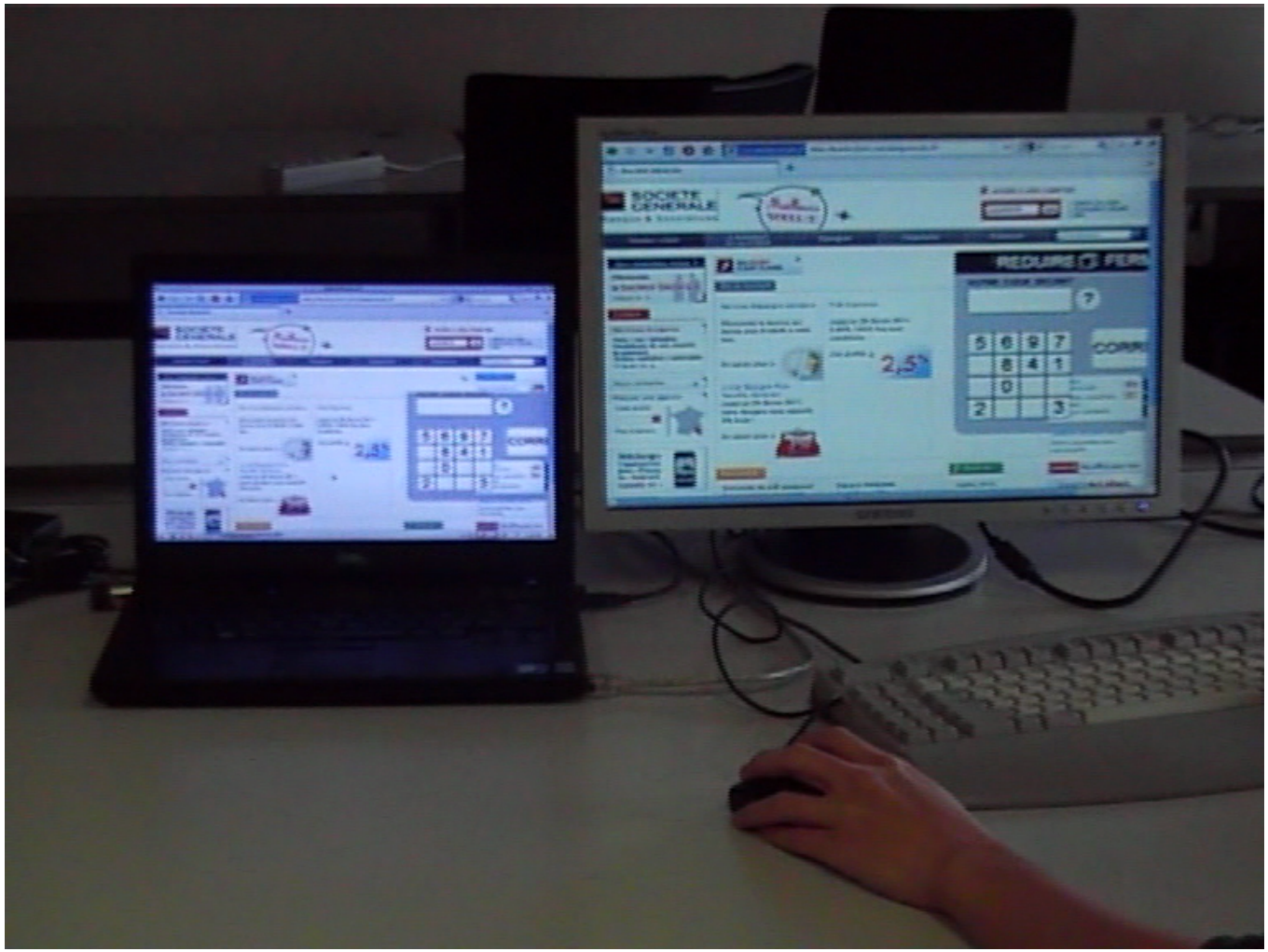
- concernant le *chipset* Tylersburg :
  - idem que les *chipsets* Eaglelake et Lakeport
  - accès *peer-to-peer* en écriture ET en lecture dans le *northbridge*
  - accès *peer-to-peer* bidirectionnel entre le *northbridge* et le *southbridge*

Pour aller plus loin ...





Démonstration



Screen-grabbing depuis le bus FireWire [Lone Sang 11]

Résultats & performances

Résultats obtenus :

- copie de 800x600 pixels à 2 images/seconde (3,84 Mo/seconde)
  - suffisant par exemple pour capturer du texte
  - insuffisant par exemple pour capturer une vidéo

Facteurs impactant les performances :

- l'application de l'attaquant
  - l'application est développée en Python
  - seraient-elles meilleures avec une application dans un langage natif ?
- le type de contrôleur qui effectue le DMA
  - le contrôleur FireWire est de type PCI (par défaut sur les machines)
  - seraient-elles meilleures avec un contrôleur PCIe ?
- le contrôleur ciblé
  - le temps de réponse du contrôleur (ex. accès mémoire)
  - les modes d'accès (octet, mot, bloc, ...) supportés

Sommaire de la présentation

Input/Output Memory Management Unit (I/O MMU)

Qu'est ce qu'une *I/O Memory Management Unit* (I/O MMU) ?

- composant similaire à l'unité de gestion mémoire du processeur
  - virtualise la mémoire principale
  - contrôle les accès à cette mémoire
  - se configure à l'aide de tables de pages en mémoire principale
- unité de gestion mémoire dédiée aux contrôleurs d'E/S

Input/Output Memory Management Unit (I/O MMU)

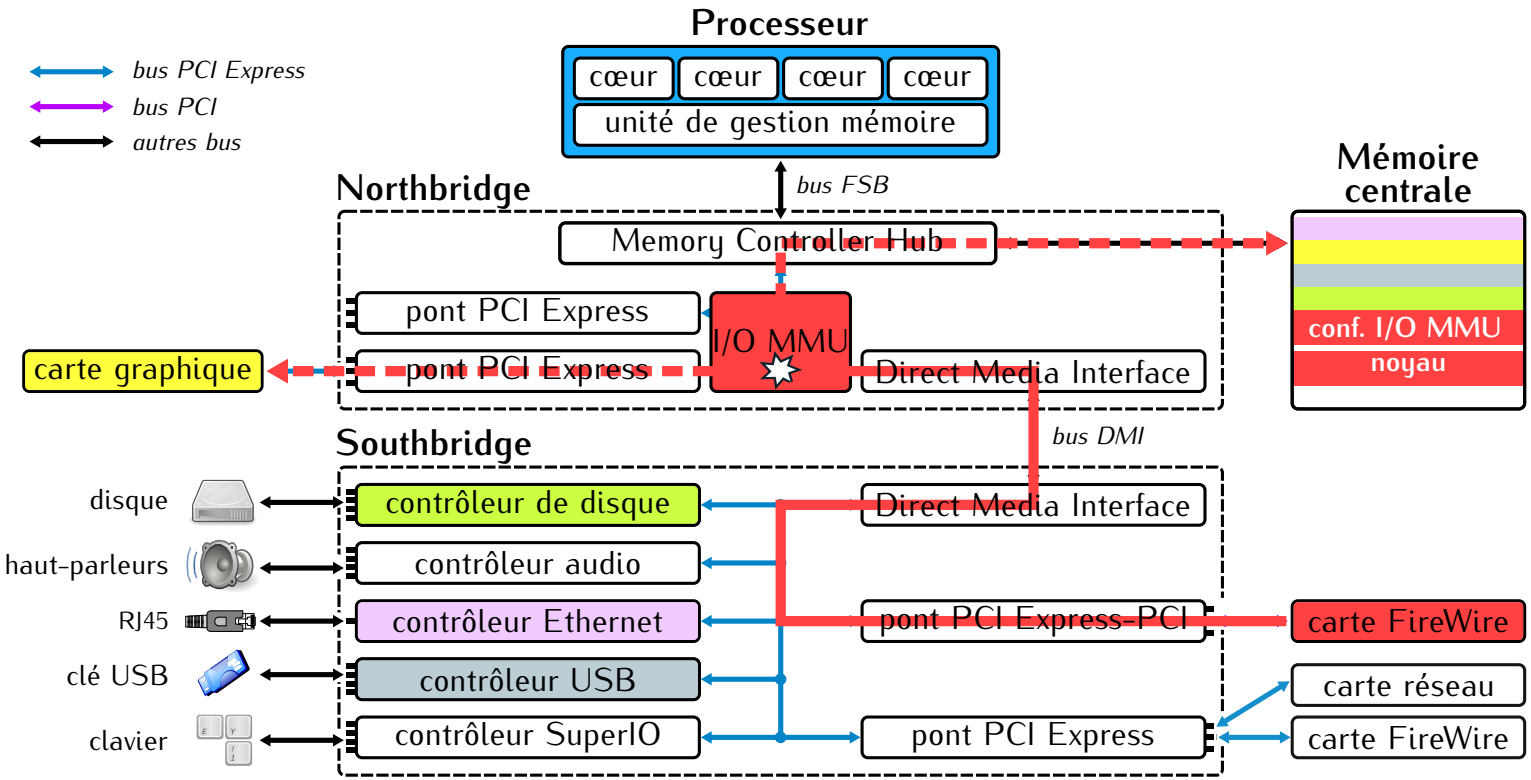
Qu'est ce qu'une *I/O Memory Management Unit* (I/O MMU) ?

- composant similaire à l'unité de gestion mémoire du processeur
  - virtualise la mémoire principale
  - contrôle les accès à cette mémoire
  - se configure à l'aide de tables de pages en mémoire principale
- unité de gestion mémoire dédiée aux contrôleurs d'E/S

Cas d'utilisation d'une *I/O Memory Management Unit* (I/O MMU) :

- étendre l'espace qui peut être adressé depuis un contrôleur
  - remplace les mécanismes de *bounce-buffers* ou de *scatter-gather*
- assurer l'isolation entre les régions de mémoire des contrôleurs
  - associe un domaine et des régions de mémoire à chaque contrôleur
  - restreint l'accès des contrôleurs à leurs domaines respectifs

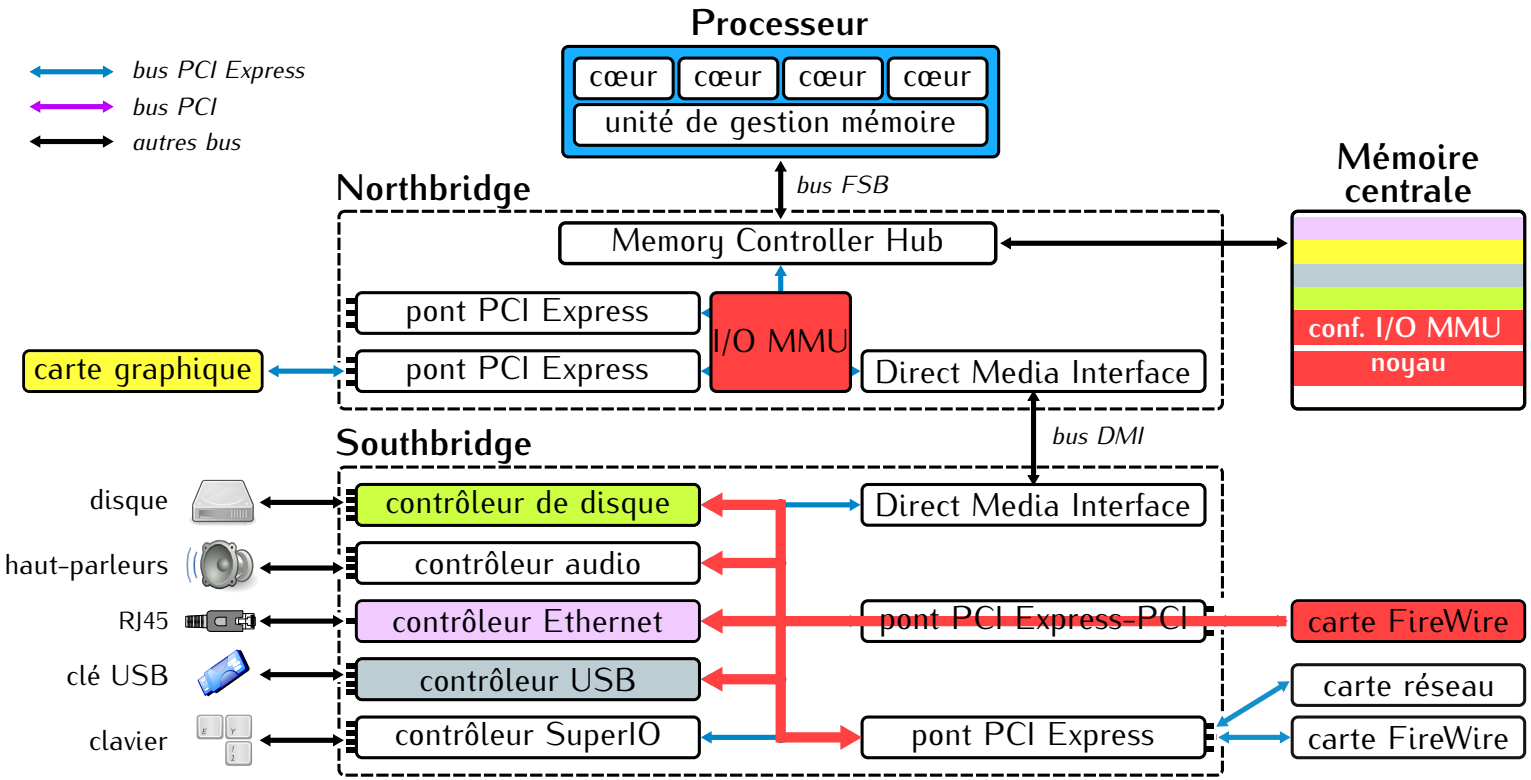
Input/Output Memory Management Unit (I/O MMU)



L'I/O MMU dans les *chipsets* Intel :

- contrôle les accès mémoire vers un composant du *northbridge*

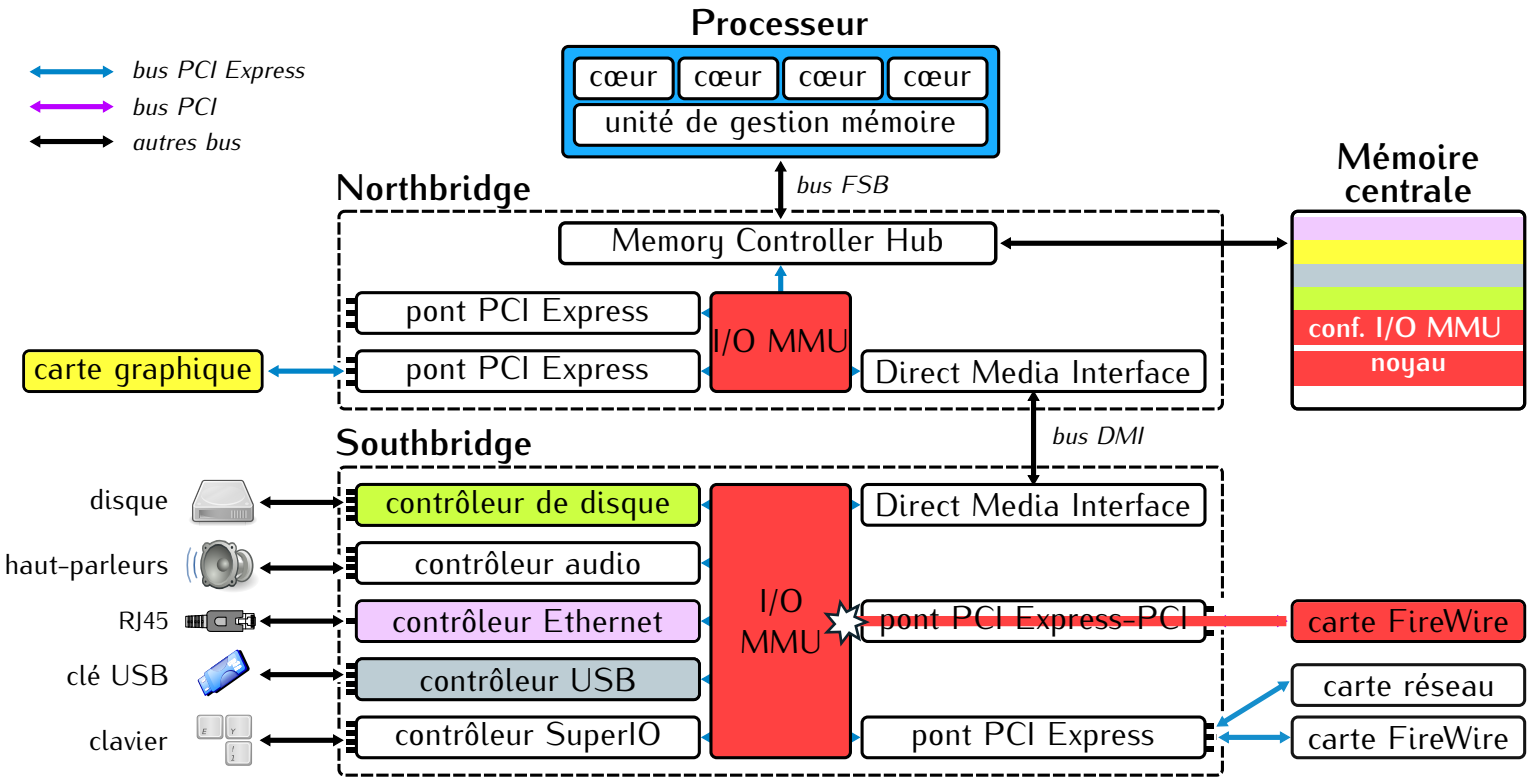
Input/Output Memory Management Unit (I/O MMU)



L'I/O MMU dans les *chipsets* Intel :

- contrôle les accès mémoire vers un composant du *northbridge*
- ne contrôle pas les accès mémoire internes au *southbridge*

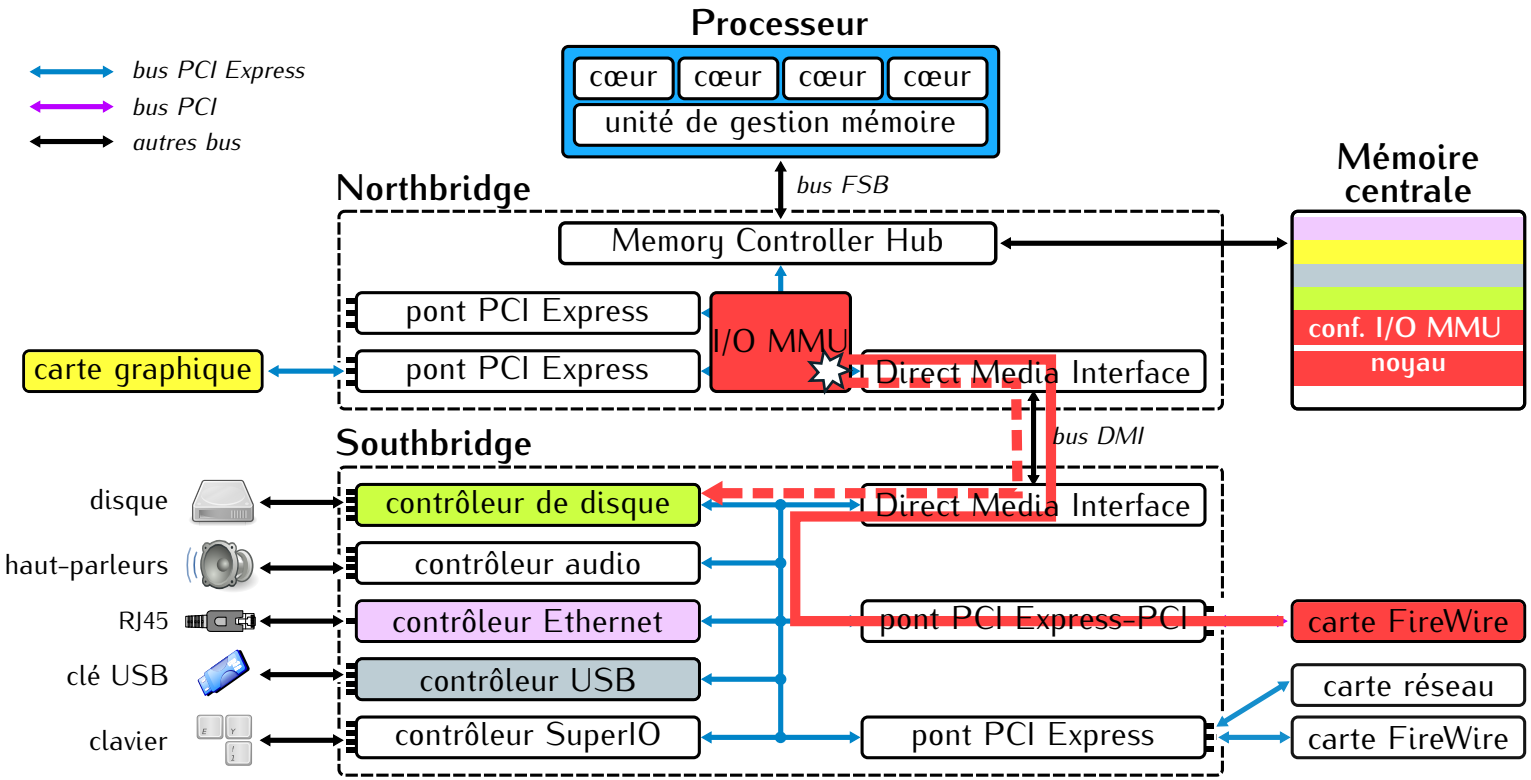
Input/Output Memory Management Unit (I/O MMU)



Les extensions envisageables :

- avoir plusieurs I/O MMUs dans le *chipset* (ex. AMD IOMMU)

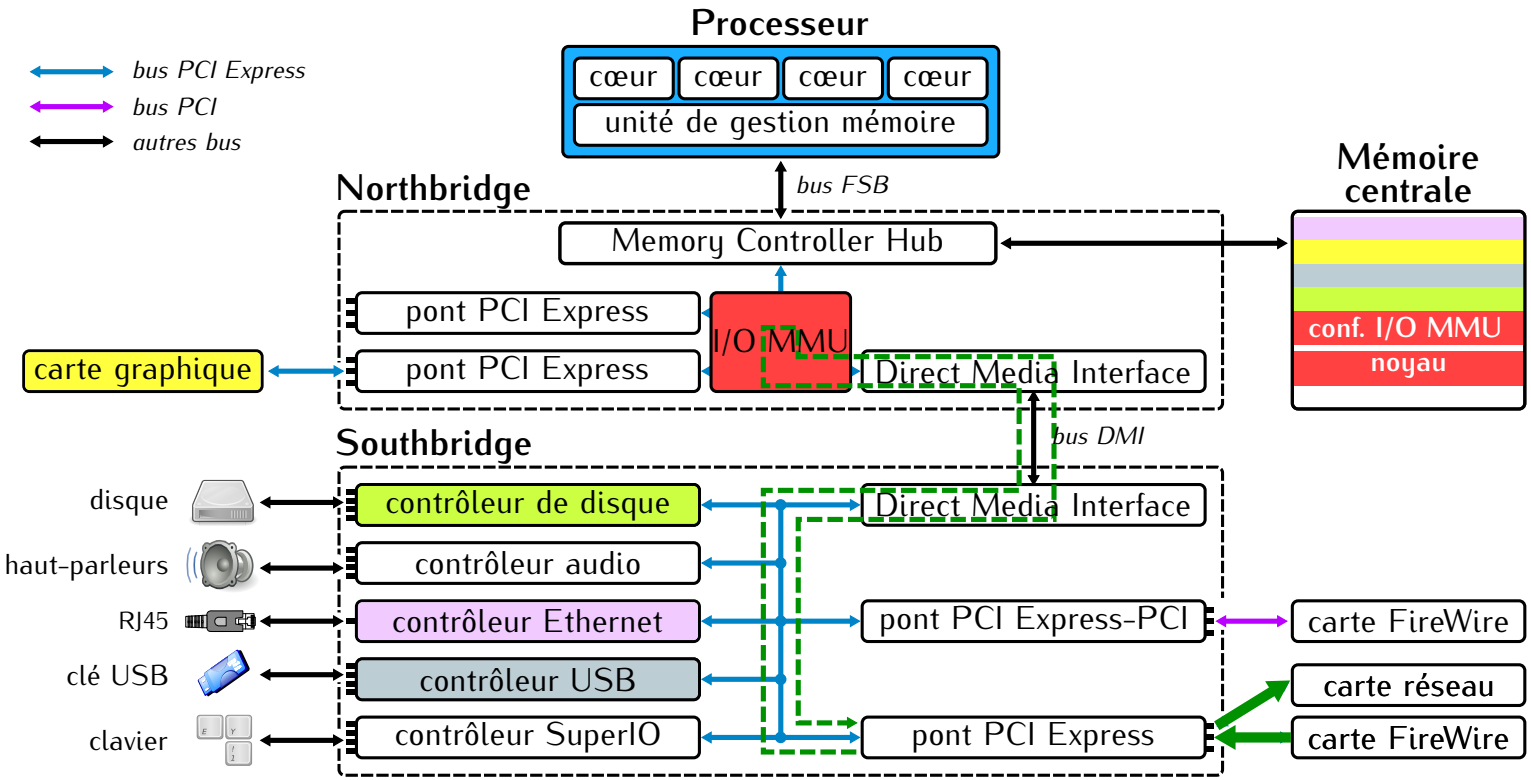
Input/Output Memory Management Unit (I/O MMU)



Les extensions envisageables :

- avoir plusieurs I/O MMUs dans le *chipset* (ex. AMD IOMMU)
- rediriger tous les flux vers le *northbridge*

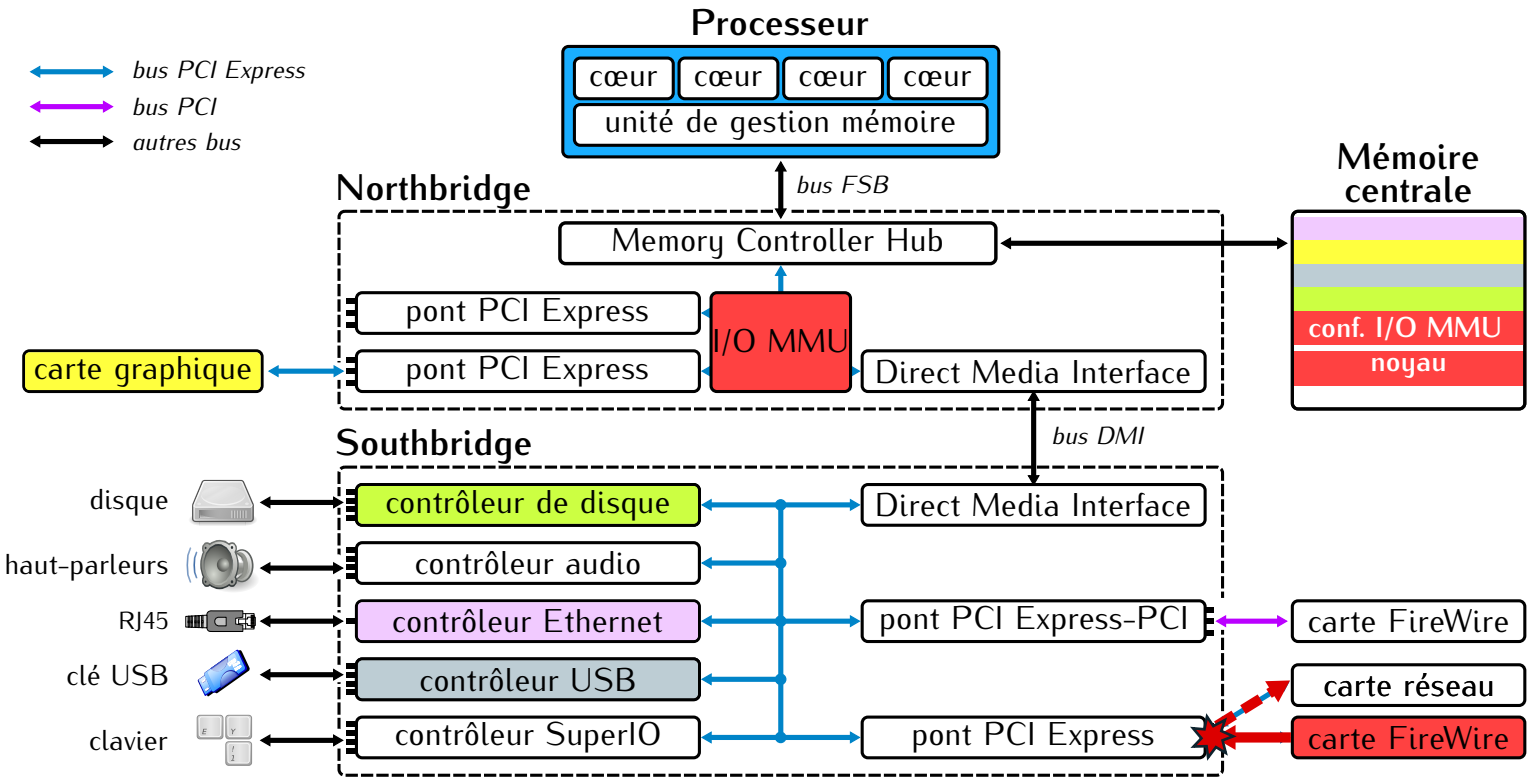
Access Control Services (ACS)



Qu'est ce que les *Access Control Services* (ACS) ?

- définissent des points de contrôle sur les bus d'E/S
- configurent les composants pour effectuer du contrôle d'accès
  - *ACS Upstream Forwarding* (U)

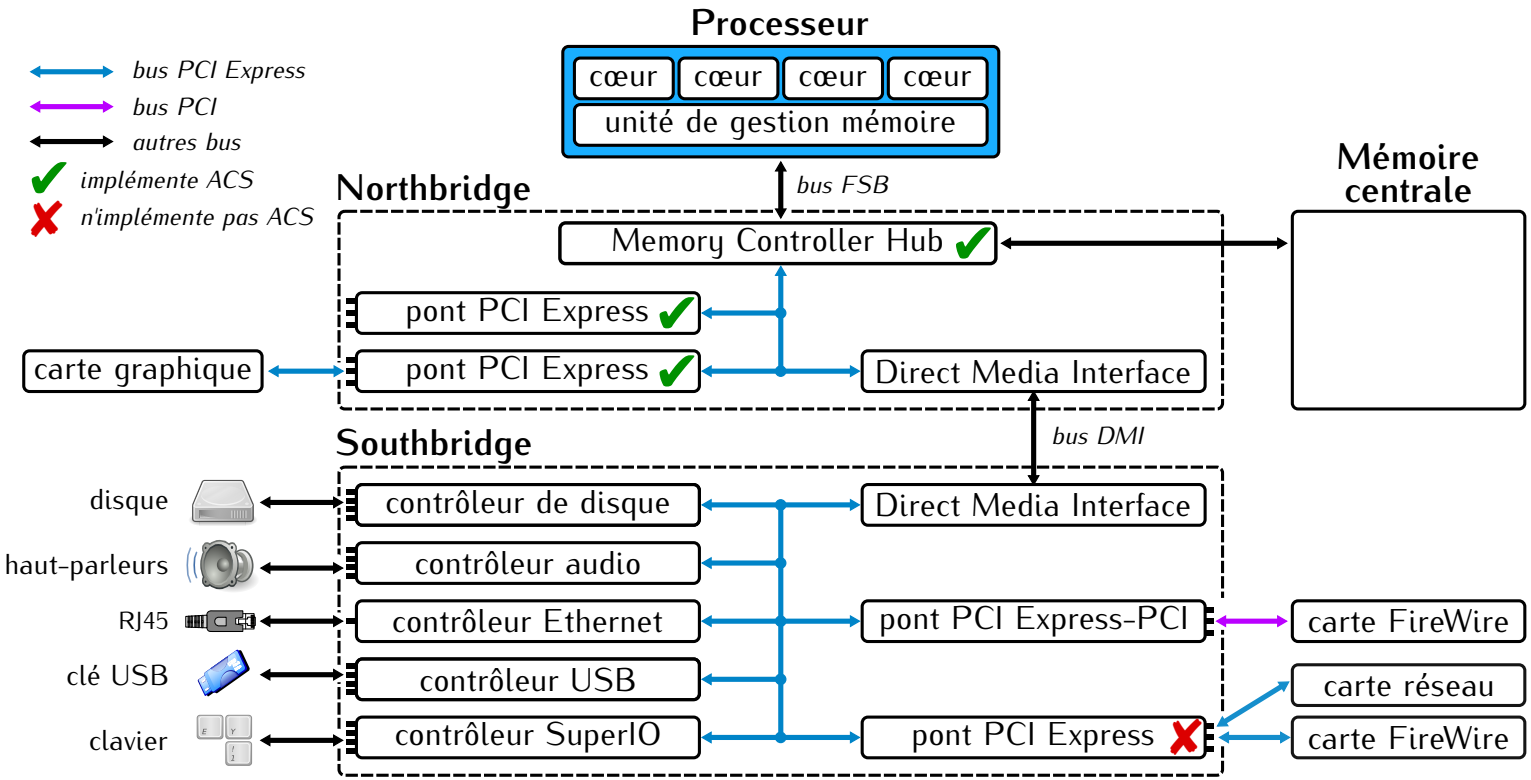
Access Control Services (ACS)



Qu'est ce que les *Access Control Services* (ACS) ?

- définissent des points de contrôle sur les bus d'E/S
- configurent les composants pour effectuer du contrôle d'accès
  - *ACS Upstream Forwarding* (U)
  - *ACS P2P Egress Port* (E)
  - ...

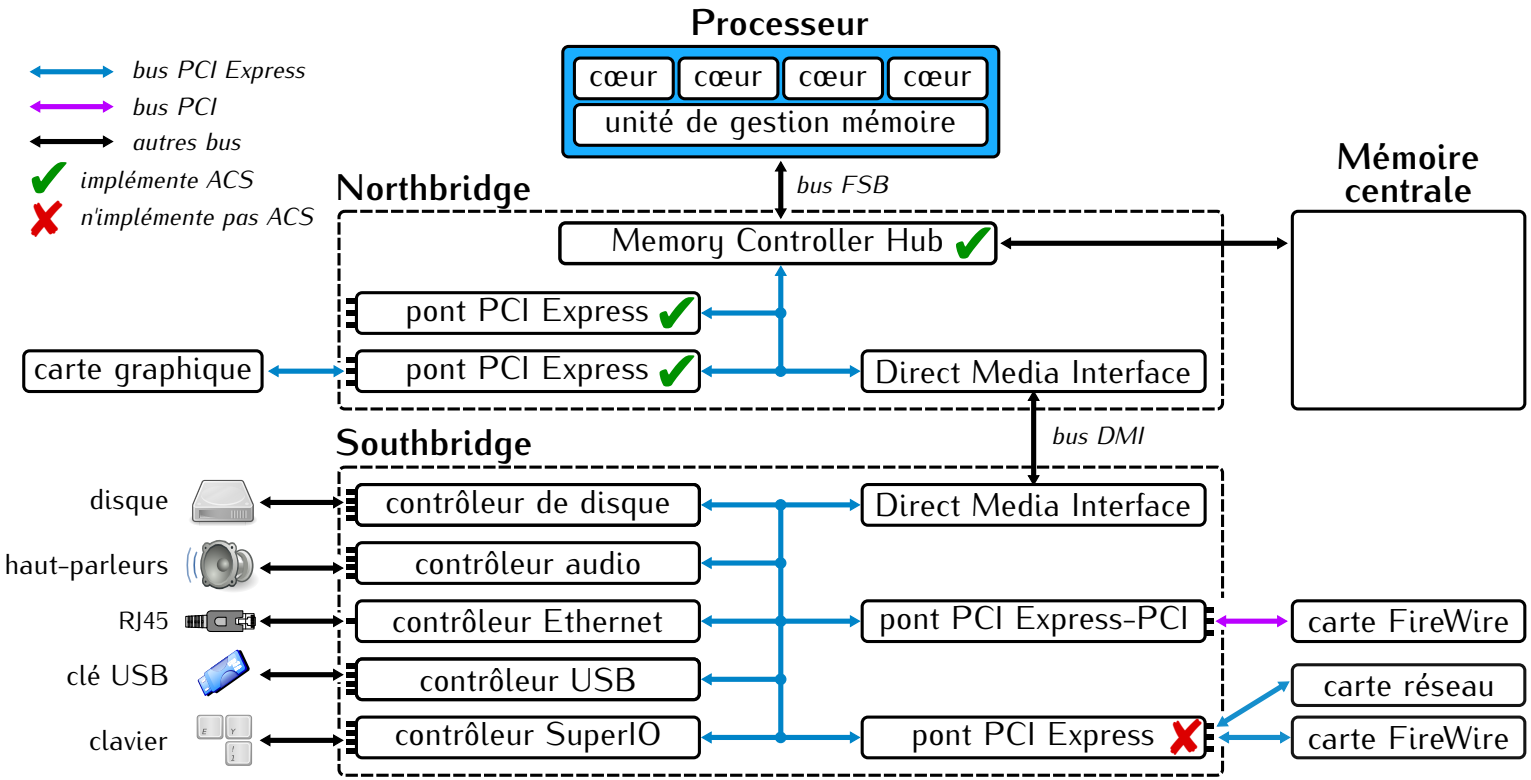
Access Control Services (ACS)



Les ACS dans les *chipsets* Intel :

- apparus récemment (Intel x58), implémentés au sein du *northbridge*
- désactivés par défaut, il faut les configurer manuellement

Access Control Services (ACS)



Quelques remarques quant à leur activation :

- les pilotes ne tiennent pour l'instant pas compte de ces dispositifs
- leur activation peut bloquer le bon fonctionnement des contrôleurs

Sommaire de la présentation

Conclusion

Ce que nous avons abordé dans cette présentation :

- le mécanisme d'accès direct à la mémoire
  - *Direct Memory Access* (DMA) vers la mémoire principale
  - *Direct Memory Access* (DMA) vers les contrôleurs d'E/S
- l'étude des attaques DMA peer-to-peer sur différents *chipsets*
  - accès en écriture possible sur la plupart des *chipsets*
  - accès en lecture possible sur certains *chipsets* récents
  - preuve de concept d'attaque *peer-to-peer* sur une carte graphique
- un aperçu de quelques contre-mesures matérielles
  - *Input/Output Memory Management Unit* (I/O MMU)
  - *Access Control Services* (ACS)

Perspectives à nos travaux

Pour aller plus loin ...

- compléter notre étude avec d'autres *chipsets* récents
- étudier les nouvelles architectures matérielles Intel
  - impact sur les attaques DMA *peer-to-peer*
  - impact sur les contre-mesures telles que l'I/O MMU

Perspectives à nos travaux

Pour aller plus loin ...

- compléter notre étude avec d'autres *chipsets* récents
- étudier les nouvelles architectures matérielles Intel
  - impact sur les attaques DMA *peer-to-peer*
  - impact sur les contre-mesures telles que l'I/O MMU

Pour aller (encore) plus loin ...

- développer une carte pour générer des requêtes sur les bus d'E/S
  - pour étudier exhaustivement les attaques possibles depuis les E/S
  - pour faire du *fuzzing* sur le *chipset*
  - pour du contrôle d'intégrité de *firmware*
- proposer une taxonomie des attaques par entrée-sortie

Merci de votre attention ...

Références 1

Damien Aumaitre.
Voyage au coeur de la mémoire.
In Actes du 6ème Symposium sur la Sécurité des Technologies de l'Information et des Communications (SSTIC 2008), pages 378–437, Rennes, June 2008.
http://actes.sstic.org/SSTIC08/Voyage_Coeur_Memoire/.

Damien Aumaitre & Christophe Devine.
Virtdbg : Un débogueur noyau utilisant la technologie de virtualisation matérielle VT-x.
In Actes du 8ème Symposium sur la Sécurité des Technologies de l'Information et des Communications (SSTIC 2010), pages 74–133, Rennes, June 2010.
http://www.sstic.org/media/SSTIC2010/SSTIC-actes/virtdbg/.

Références 2

Michael Becher, Maximillian Dornseif & Christian N. Klein.
FireWire - all your memory are belong to us.
In CanSecWest/core05, 4-5 May 2005.
http://md.hudora.de/presentations/#firewire-cansecwest.

Adam Boileau.
Hit by a Bus : Physical Access Attacks with FireWire.
In RUXCON 2006, October 2006.
http://www.ruxcon.org.au/files/2006/firewire_attacks.pdf.

Guillaume Delugré.
Closer to metal : reverse-engineering the Broadcom NetExtreme's firmware.
In Hack.lu, Luxembourg, 27-29 October 2010.
http://esec-lab.sogeti.com/dotclear/public/publications/10-hack.lu-nicreverse_slides.pdf.

Références 3


Christophe Devine & Guillaume Vissian.
Compromission physique par le bus PCI.
In Actes du 7ème Symposium sur la Sécurité des Technologies de l'Information et des Communications (SSTIC 2009), pages 169–193, Rennes, June 2009.
http://actes.sstic.org/SSTIC09/Compromission_physique_par_le_bus_PCI/.


Maximillian Dornseif.
0wned by an iPod - Hacking by Firewire.
In PacSec/core04, 11-12 November 2004.
http://md.hudora.de/presentations/#firewire-pacsec.

Références 4

Loïc Duflot.
Contribution à la sécurité des systèmes d'exploitation et des microprocesseurs.
Thèse de doctorat, Université de Paris XI, October 2007.
http://www.ssi.gouv.fr/archive/fr/sciences/fichiers/lti/these-duflot.pdf.

Loïc Duflot, Yves-Alexis Perez, Guillaume Valadon & Olivier Levillain.
Quelques éléments en matière de sécurité des cartes réseau.
In Actes du 8ème Symposium sur la Sécurité des Technologies de l'Information et des Communications (SSTIC 2010), pages 213–235, Rennes, June 2010.
http://www.sstic.org/media/SSTIC2010/SSTIC-actes/Peut_on_faire_confiance_aux_cartes_reseau/.

Références 5

Fernand Lone Sang, Vincent Nicomette & Yves Deswarte.
Démonstration d'une attaque DMA peer-to-peer depuis un périphérique FireWire vers un contrôleur graphique, January 2011.
http://homepages.laas.fr/nicomett/Videos/.

David Maynor.
0wn3d by everything else - USB/PCMCIA Issues.
In CanSecWest/core05, 4-5 May 2005.
http://cansecwest.com/core05/DMA.ppt.

Arrigo Triulzi.
Project Maux Mk.II - « I 0wn the NIC, Now I want a Shell ! ».
In PacSec/core08, 12-13 November 2008.
http://www.alchemistowl.org/arrigo/Papers/Arrigo-Triulzi-PACSEC08-Project-Maux-II.pdf.

Références 6

Arrigo Triulzi.
The Jedi Packet Trick takes over the Deathstar (or : « Taking NIC Backdoors to the Next Level »).
In CanSecWest/core10, 24-26 March 2010.
http://www.alchemistowl.org/arrigo/Papers/Arrigo-Triulzi-CANSEC10-Project-Maux-III.pdf.