IronHide
Plate-forme d'attaques par entrées-sorties

Fernand Lone Sang, Vincent Nicomette et Yves Deswarte

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)
Groupe Tolérance aux fautes et Sûreté de Fonctionnement informatique (TSF)

Symposium sur la Sécurité des Technologies de l'Information et des Communications
SSTIC'2012 – Rennes

LAAS-CNRS

Contexte et problématique (1)

Des difficultés à protéger efficacement les systèmes informatiques :

→ Complexité croissante

→ Surface d'attaque de plus en plus large

Deux principaux vecteurs d'attaques sur un ordinateur :

- logiciel malveillant reposant uniquement sur le processeur
  - → exploitation d'une vulnérabilité (ex. débordements de tampon)
  - → abus de fonctionnalités du système (ex. `/dev/mem`)
- logiciel malveillant impliquant d'autres composants matériels
  - → altération de la structure du composant matériel (ex. *firmware*)
  - → abus de fonctionnalités des composants matériels (ex. DMA)

Contexte et problématique (2)

Le spectre de nos travaux de recherche :

- étude des attaques « hybrides » impliquant à la fois le logiciel et le matériel, en particulier les attaques par entrées-sorties
- proposition de contre-mesures génériques à ces attaques

Approches possibles pour étudier ces attaques :

- utiliser un contrôleur d'entrées-sorties sur-étagère
  - ✖ interfaces spécifiques et actions malveillantes limitées
- développer un contrôleur d'entrées-sorties (ex. avec un FPGA)
  - ✓ interfaces spécifiques mais possibilités infinies
  - → générer tout type de requêtes (valides et invalides) sur les bus
  - → identifier et démontrer de nouvelles vulnérabilités
  - ex. [Petroni *et al.* 04, Carrier & Grand 04, Devine & Vissian 09]

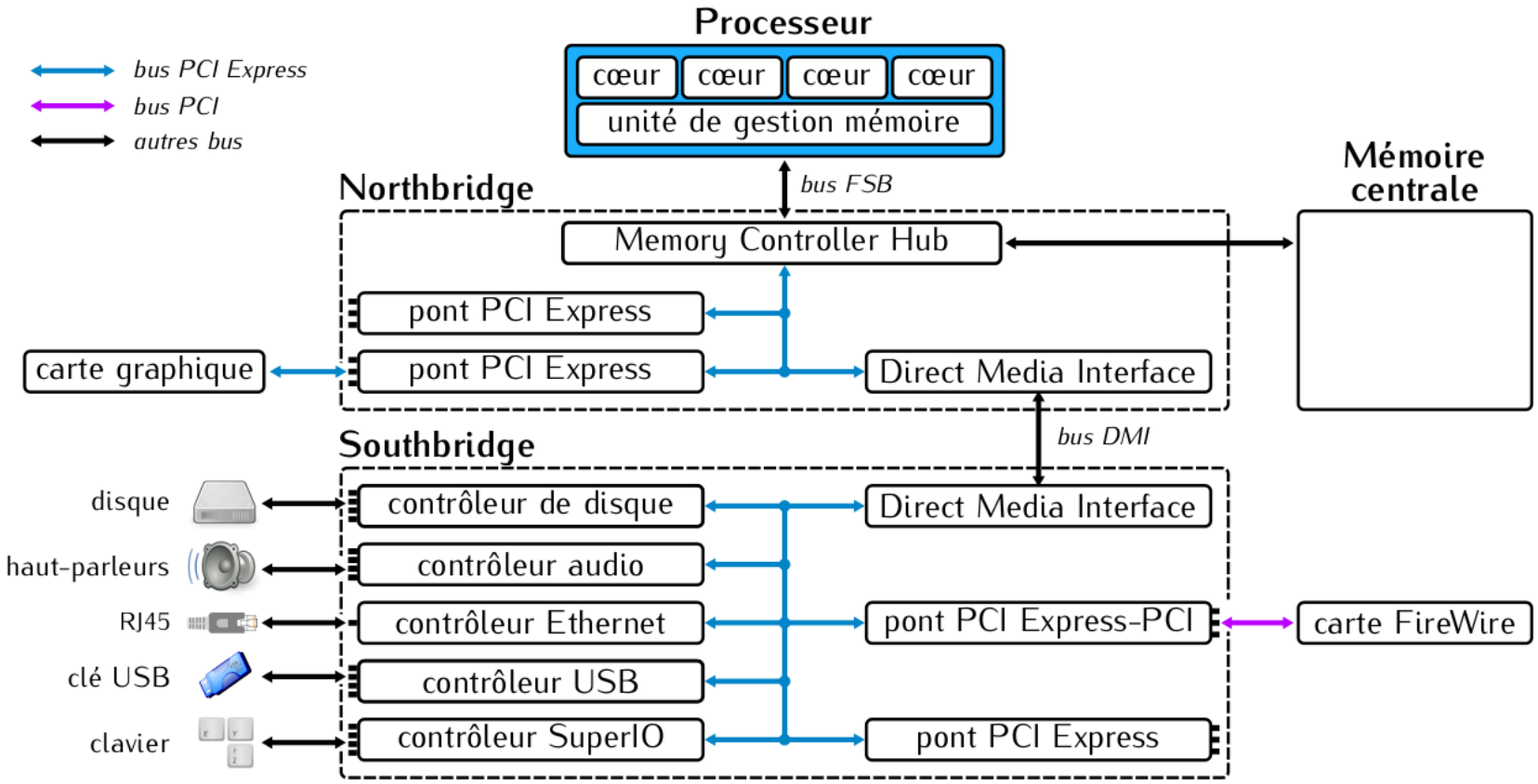
Sommaire de la présentation

1. Élements d'architecture
2. IronHide : contrôleur d'entrées-sorties générique
3. Quelques résultats d'expérimentations
4. Conclusion & perspectives

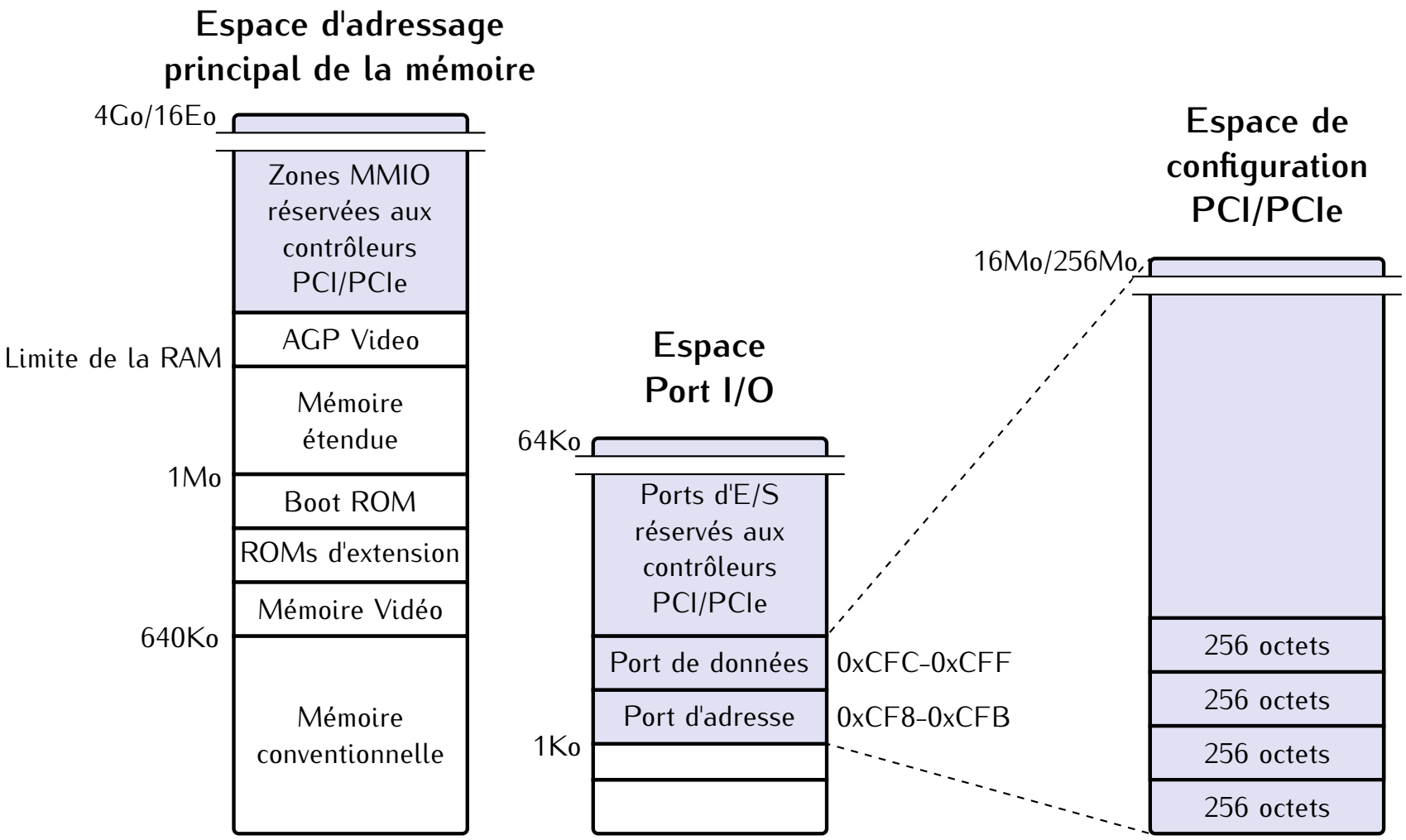
Sommaire de la présentation

1. Élements d'architecture
2. IronHide : contrôleur d'entrées-sorties générique
3. Quelques résultats d'expérimentations
4. Conclusion & perspectives

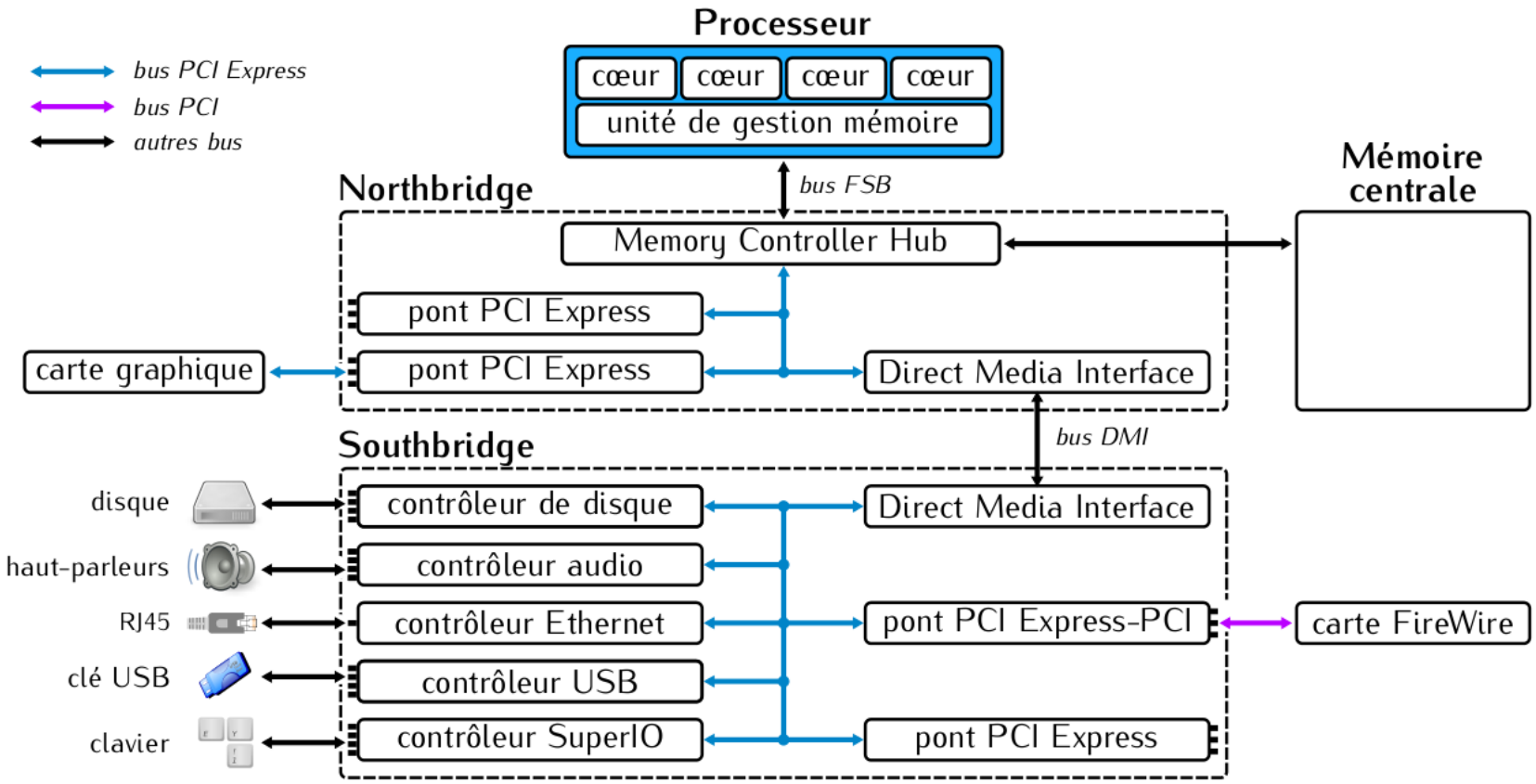
Architecture matérielle PC



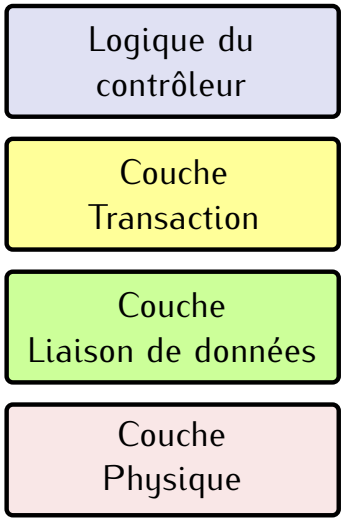
Espaces d'adressage du processeur (1)



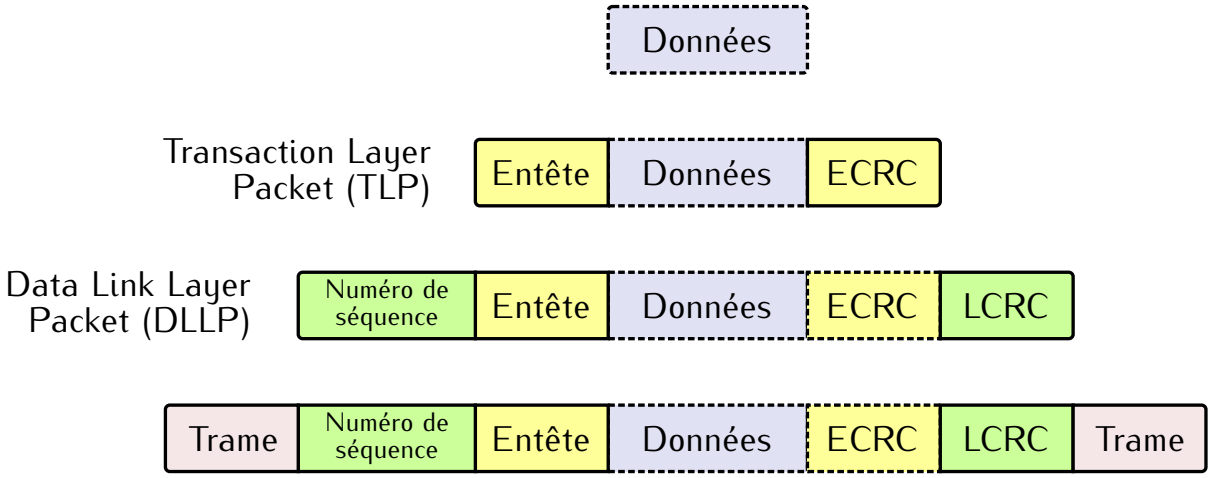
Espaces d'adressage du processeur (2)



Protocole de bus PCI Express (1)

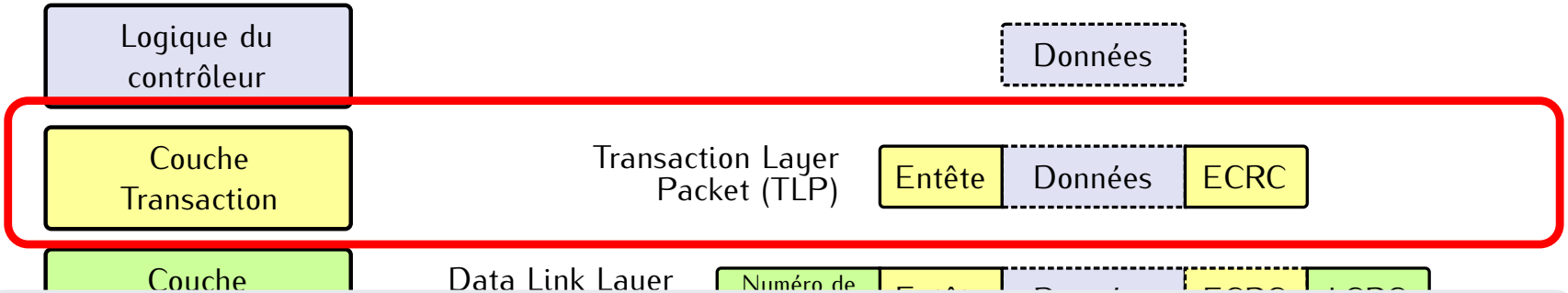


Couches protocolaires



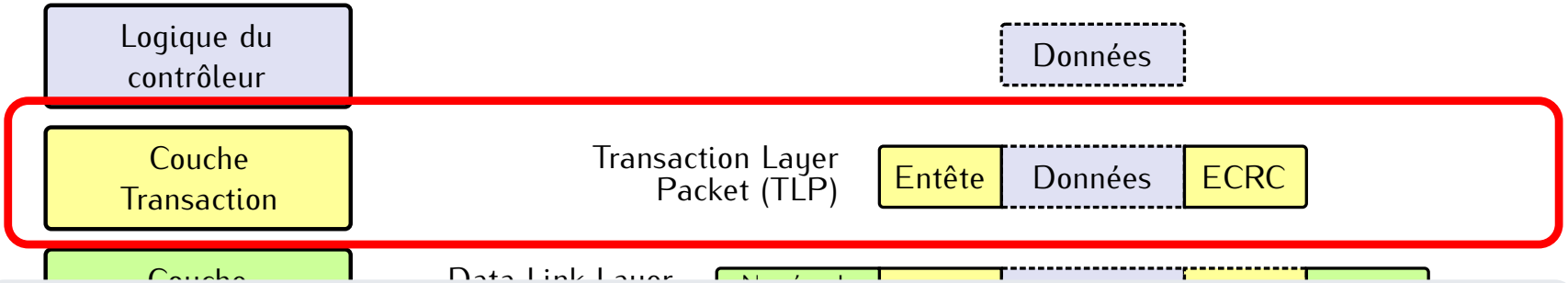
Format des paquets

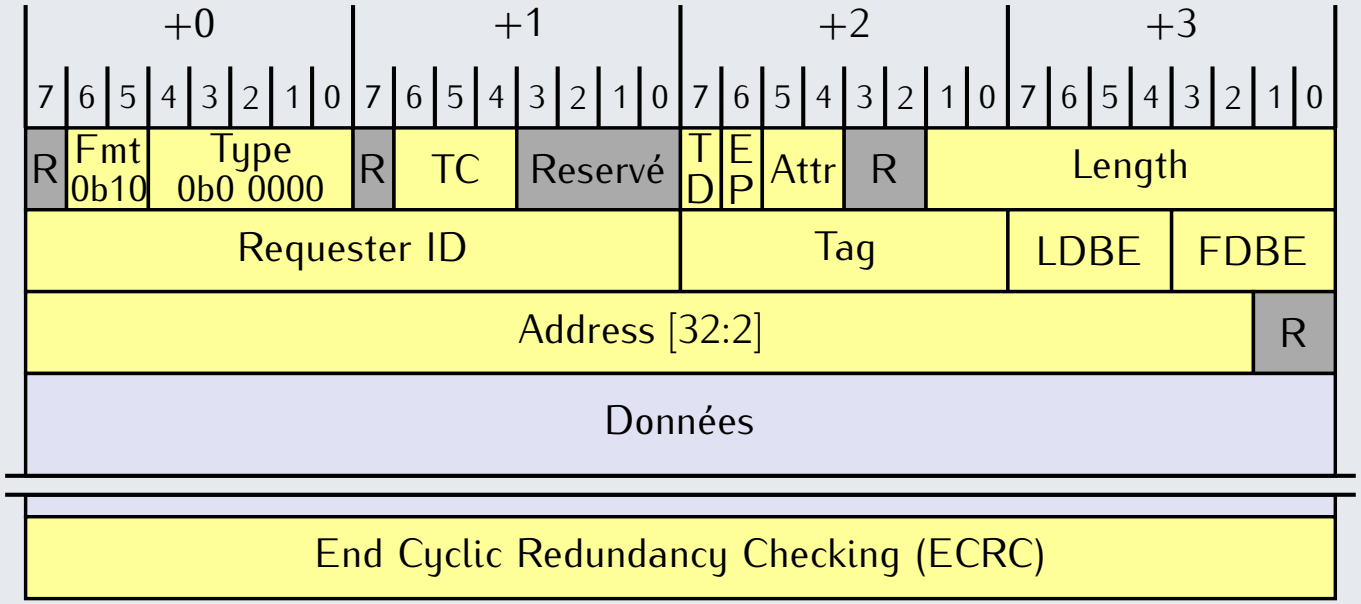
Protocole de bus PCI Express (2)



Catégories	Transactions	*Completion*
Memory	Memory Read	oui
	Memory Write	non
	Memory Read Lock	oui
I/O	IO Read	oui
	IO Write	oui
Configuration	Configuration Read (Type 0 et Type 1)	oui
	Configuration Write (Type 0 et Type 1)	oui
Message	Message	non

Protocole de bus PCI Express (3)



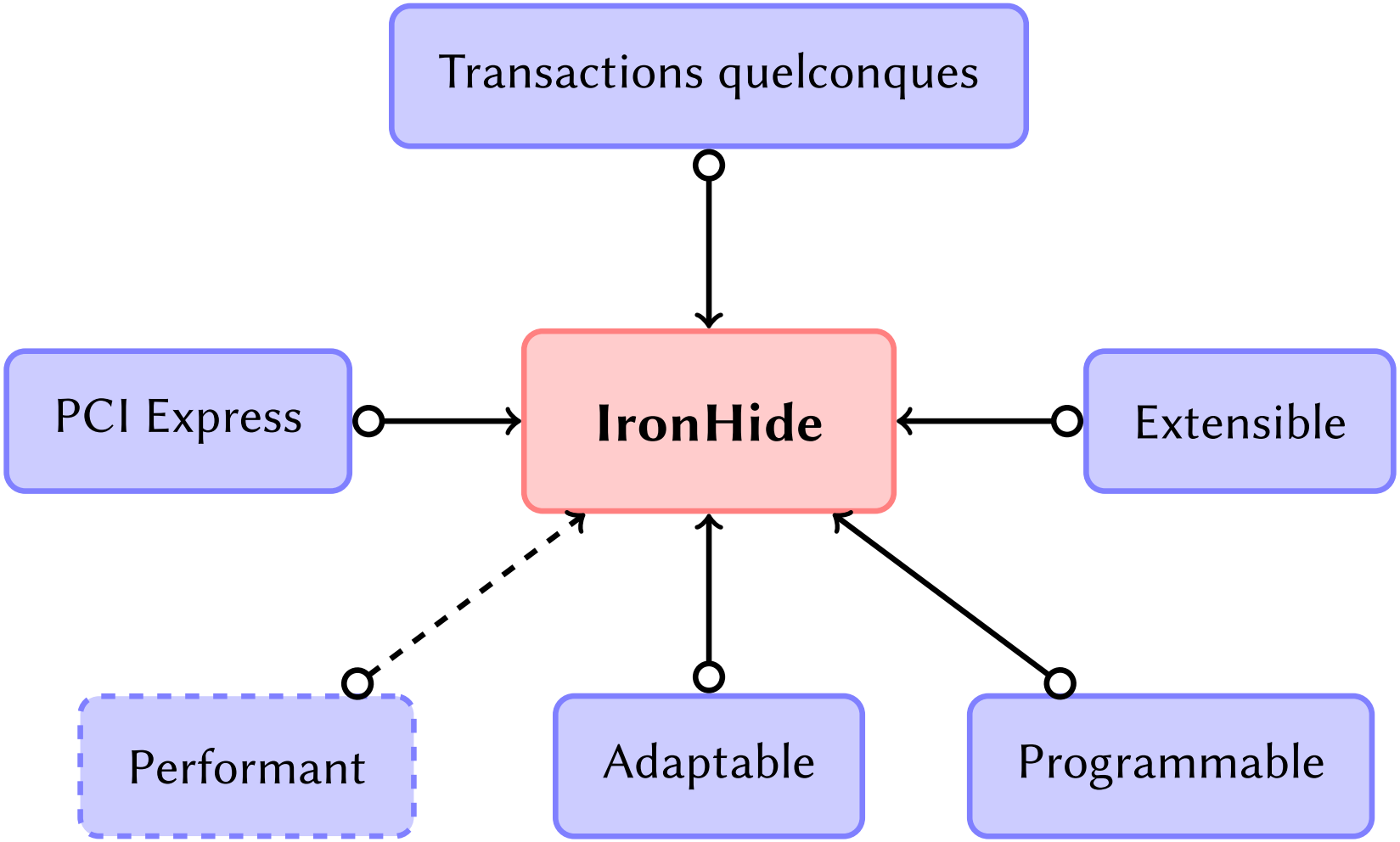


Exemple de transaction pour une écriture en mémoire

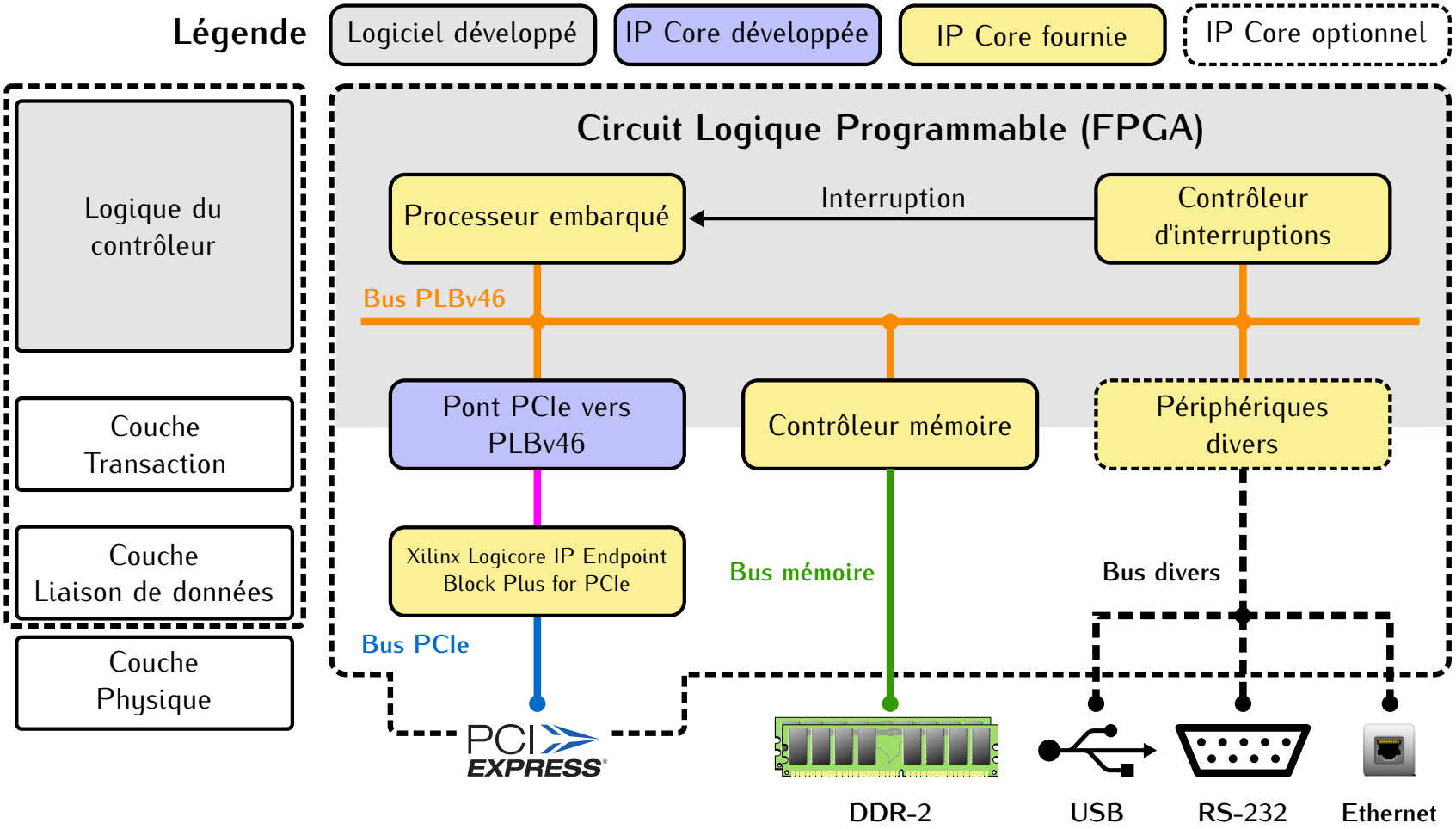
Sommaire de la présentation

1. Élements d'architecture
2. IronHide : contrôleur d'entrées-sorties générique
3. Quelques résultats d'expérimentations
4. Conclusion & perspectives

Caractéristiques souhaitées pour IronHide (1)



Architecture interne d'IronHide



Architecture matérielle

État d'avancement du prototype (1)

Détails sur l'implémentation actuelle :

- repose sur des IP-Cores propriétaires, mais libres d'utilisation
- peut générer tout type de paquets PCI Express
  - → paquets valides, pour utiliser (ou abuser) de fonctionnalités
  - → paquets invalides, pour évaluer les implémentations matérielles
- a un comportement entièrement programmable
  - → *firmware* écrit en langage C

Détails sur les cartes de développement utilisées :

- Pico E-17 (Express Card, Xilinx Virtex-5, PowerPC)
- ModularLogix MLX-1000 (PCIe, Xilinx Virtex-5, Microblaze)

État d'avancement du prototype (2)

Quelques pistes d'améliorations pour les futurs prototypes :

- Améliorer les performances actuelles de notre implémentation
- Déléguer certaines opérations à la partie matérielle
  - → Vérification de l'intégrité du paquet
  - → Filtrage des requêtes que l'on ne souhaite pas analyser
  - → Renvoi automatique d'une *completion* pour certaines requêtes
- Utiliser un système d'exploitation sur-étagère
  - → Linux embarqué
  - → Système temps réel (ex. VxWorks)

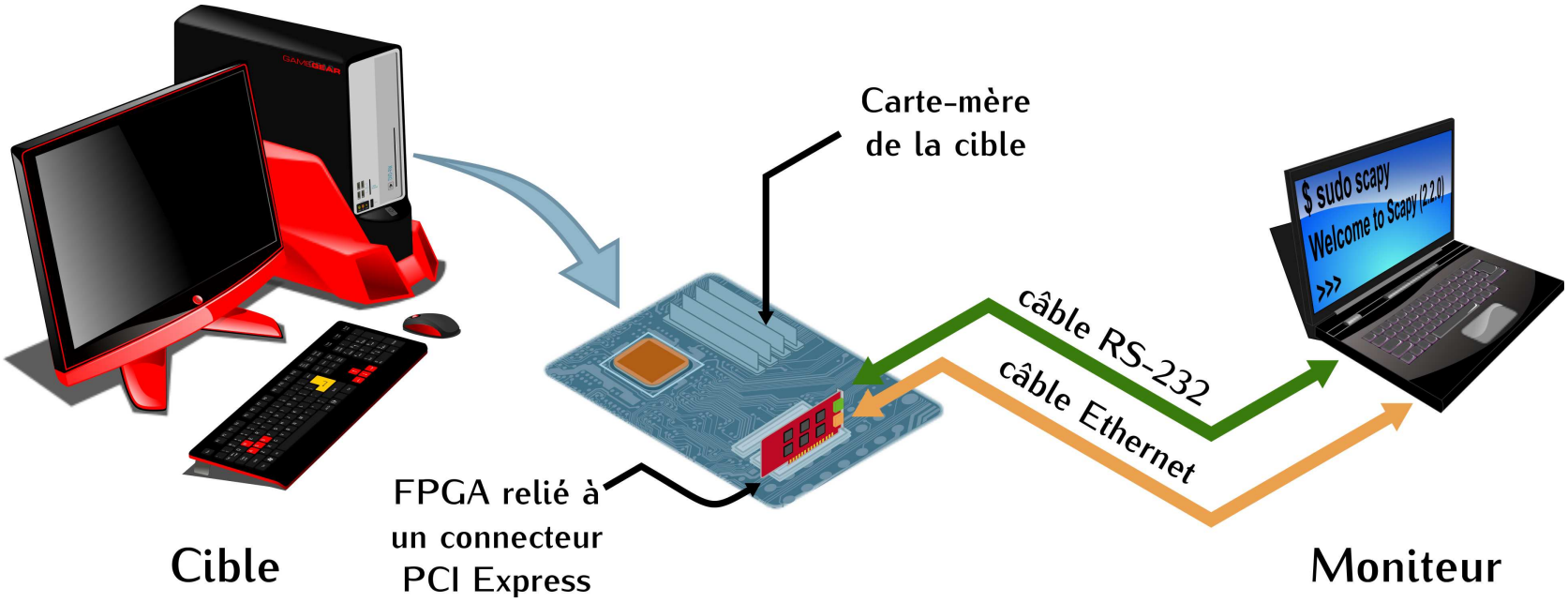
Carte Pico E-17

Carte ModularLogix MLX-1000-XC5V

Sommaire de la présentation

1. Élements d'architecture
2. IronHide : contrôleur d'entrées-sorties générique
3. Quelques résultats d'expérimentations
4. Conclusion & perspectives

Description de l'environnement d'expérimentation (1)



Description de l'environnement d'expérimentation (2)

Détection et configuration d'IronHide sous Linux

```
user@cible$ sudo lspci -v
00:00.0 Host bridge: Intel Corporation 5520/5500/X58 I/O Hub
    Subsystem: Intel Corporation Device 4f53
    Flags: fast devsel, IRQ 11
    Capabilities: [40] #00 [0000]
[...]
01:00.0 Memory controller: Pico Computing Device 0e17 (rev 01)
    Subsystem: Device 4658:0000
    Flags: bus master, fast devsel, latency 0, IRQ 255
    [virtual] Expansion ROM at d3600000 [disabled] [size=1M]
    Capabilities: [40] Power Management version 3
    Capabilities: [48] MSI: Enable- Count=1/1 Maskable- 64bit+
    Capabilities: [60] Express Endpoint, MSI 00
    Capabilities: [100] Device Serial Number 00-00-00-01-01-00-0a-35
```

Exemple d'analyse de trafic (1)

00 00 00 01 00 08 10 01 d0 00 00 00

Memory32	
fmt	3DW HDR no data
type	MRd32
rsvd0	0x0L
tc	0x0L
rsvd1	0x0L
attr	0L
rsvd2	0x0L
th	0L
td	0L
ep	0L
attr2	0L
at	0L
length	0x1L
reqID	(0, 1, 0)
tag	0x10
ldbe	0x0L
fdbe	0x1L
address	0xd0000000L
rsvd3	0x0L
data	0

Exemple d'analyse de trafic (2)

Configuration de la machine d'expérimentation [1]

```
user@cible$ sudo lspci -v -s 02:00.0
02:00.0 VGA compatible controller: nVidia Corporation G92
    Subsystem: nVidia Corporation Device 0719
    Flags: bus master, fast devsel, latency 0, IRQ 16
    Memory at d2000000 (32-bit, non-prefetchable) [size=16M]
    Memory at e0000000 (64-bit, prefetchable) [size=256M]
    Memory at d0000000 (64-bit, non-prefetchable) [size=32M]
    I/O ports at 3000 [size=128]
    Expansion ROM at <ignored> [disabled]
[...]
```

[1] Ordinateur de bureau assemblé en 2010 avec un *chipset* x58

Exemple d'analyse de trafic (3)

00 00 00 01 00 00 11 0f f1 e0 00 00

Memory32	
fmt	3DW HDR no data
type	MRd32
rsvd0	0x0L
tc	0x0L
rsvd1	0x0L
attr	0L
rsvd2	0x0L
th	0L
td	0L
ep	0L
attr2	0L
at	0L
length	0x1L
reqID	(0, 0, 0)
tag	0x11
ldbe	0x0L
fdbe	0xfL
address	0xf1e00000L
rsvd3	0x0L
data	0

Exemple d'analyse de trafic (4)

Memory32

fmt 3DW HDR no data

00 00 00 01 00 00 11 0f f1 e0 00 00

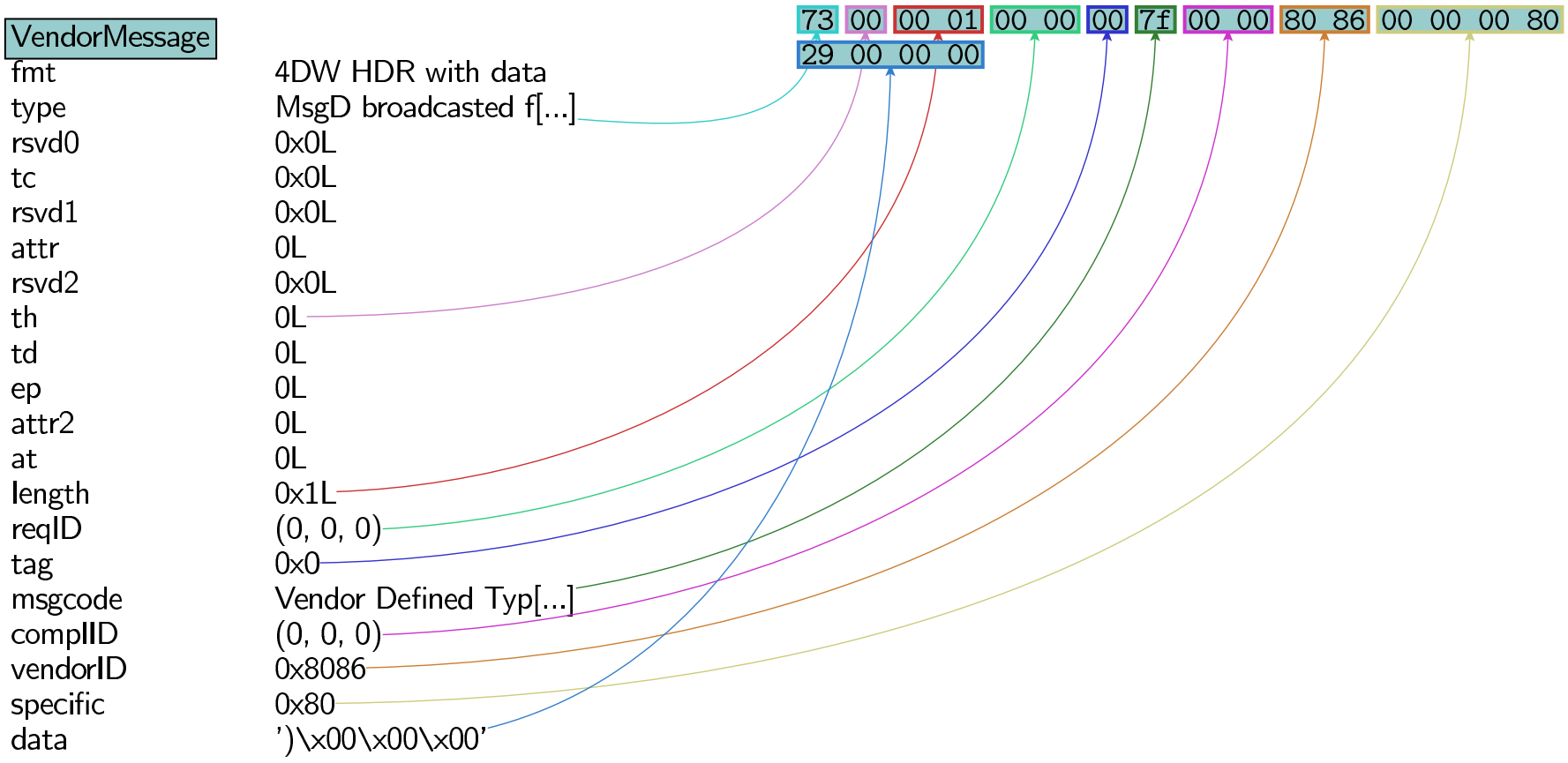
Configuration de la machine d'expérimentation [2]

```
user@cible$ sudo lspci -v -s 0e:00.0
0e:00.0 Memory controller: Pico Computing Device 0e17 (rev 01)
    Subsystem: Device 4658:0000
    Flags: bus master, fast devsel, latency 0, IRQ 10
    Expansion ROM at f1e00000 [disabled] [size=1M]
    Capabilities: [40] Power Management version 3
    Capabilities: [48] MSI: Enable- Count=1/1 Maskable- 64bit+
    Capabilities: [60] Express Endpoint, MSI 00
[...]
```

[2] Ordinateur portable DELL Latitude E6400 de 2009 avec un *chipset* GMQ45

data 0

Exemple d'analyse de trafic (5)



Exemple d'analyse de trafic (6)

Quelques détails sur le paquet reçu

- Paquet PCI-Express de type « Message »
  - → utilisé pour les interruptions, la gestion d'énergie, ...
  - → format défini par le constructeur car `msgcode` à `0x7f`
  - → spécifique aux *chipsets* Intel car `vendorID` à `0x8086`
- Type d'information véhiculé non-identifié
  - → format non-documenté dans les spécifications
  - → généré lors des événements du clavier ou de la souris

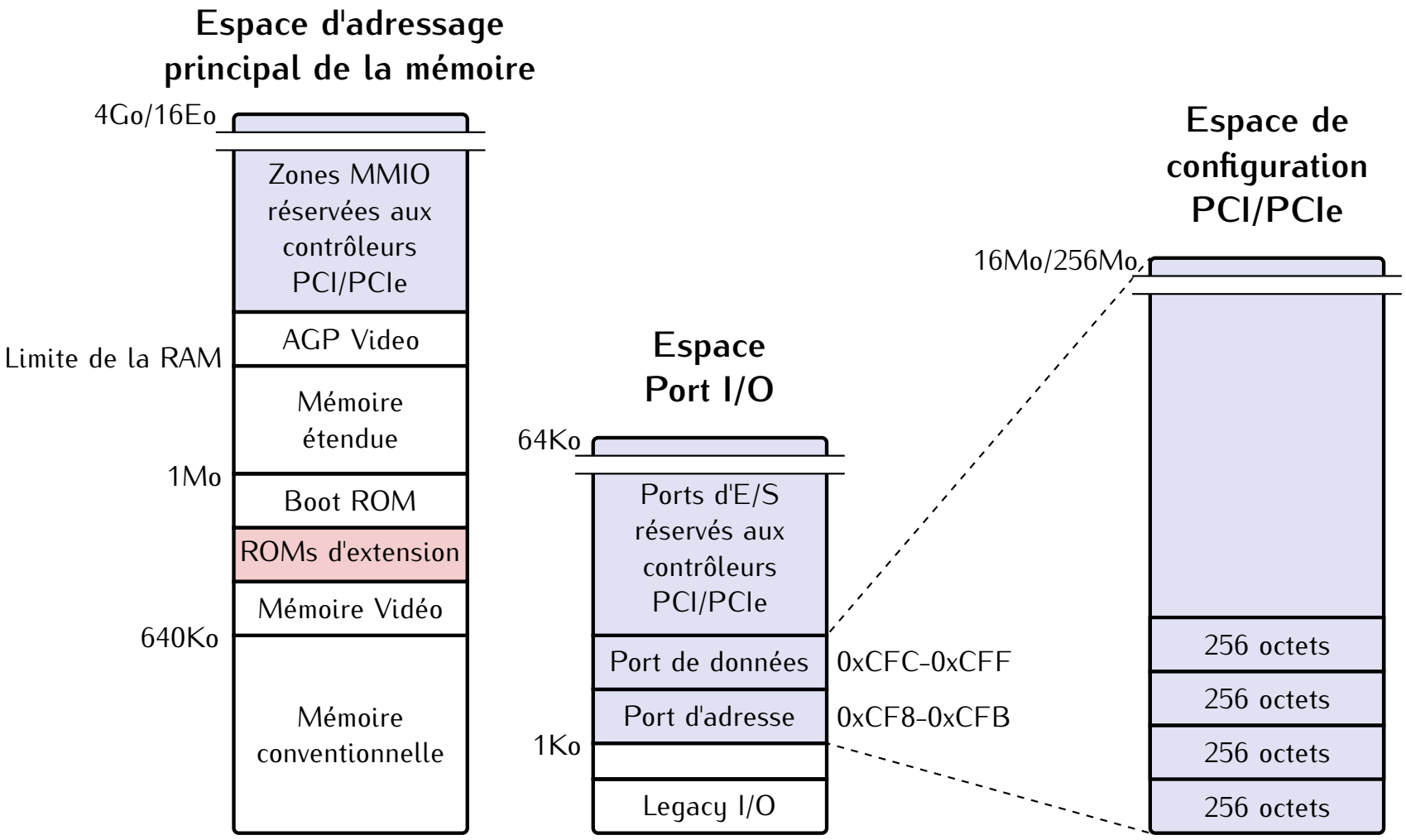
Un *System Management Interrupt* (SMI) dûe à la fonctionnalité d'*USB Legacy Support* (émulation des périphériques PS/2) ?

Accès à l'espace d'adressage mémoire principale (1)

Désactivation de la fonctionnalité de *bus-mastering*

```
user@cible$ sudo setpci -s 0e:00.0 COMMAND.W=0

user@cible$ sudo lspci -v -s 0e:00.0
0e:00.0 Memory controller: Pico Computing Device 0e17 (rev 01)
    Subsystem: Device 4658:0000
    Flags: fast devsel, latency 0, IRQ 10
    Expansion ROM at f1e00000 [disabled] [size=1M]
    Capabilities: [40] Power Management version 3
    Capabilities: [48] MSI: Enable- Count=1/1 Maskable- 64bit+
    Capabilities: [60] Express Endpoint, MSI 00
    Capabilities: [100] Device Serial Number 00-00-00-01-01-00-0a-35
[...]
```

Accès à l'espace d'adressage mémoire principale (2)

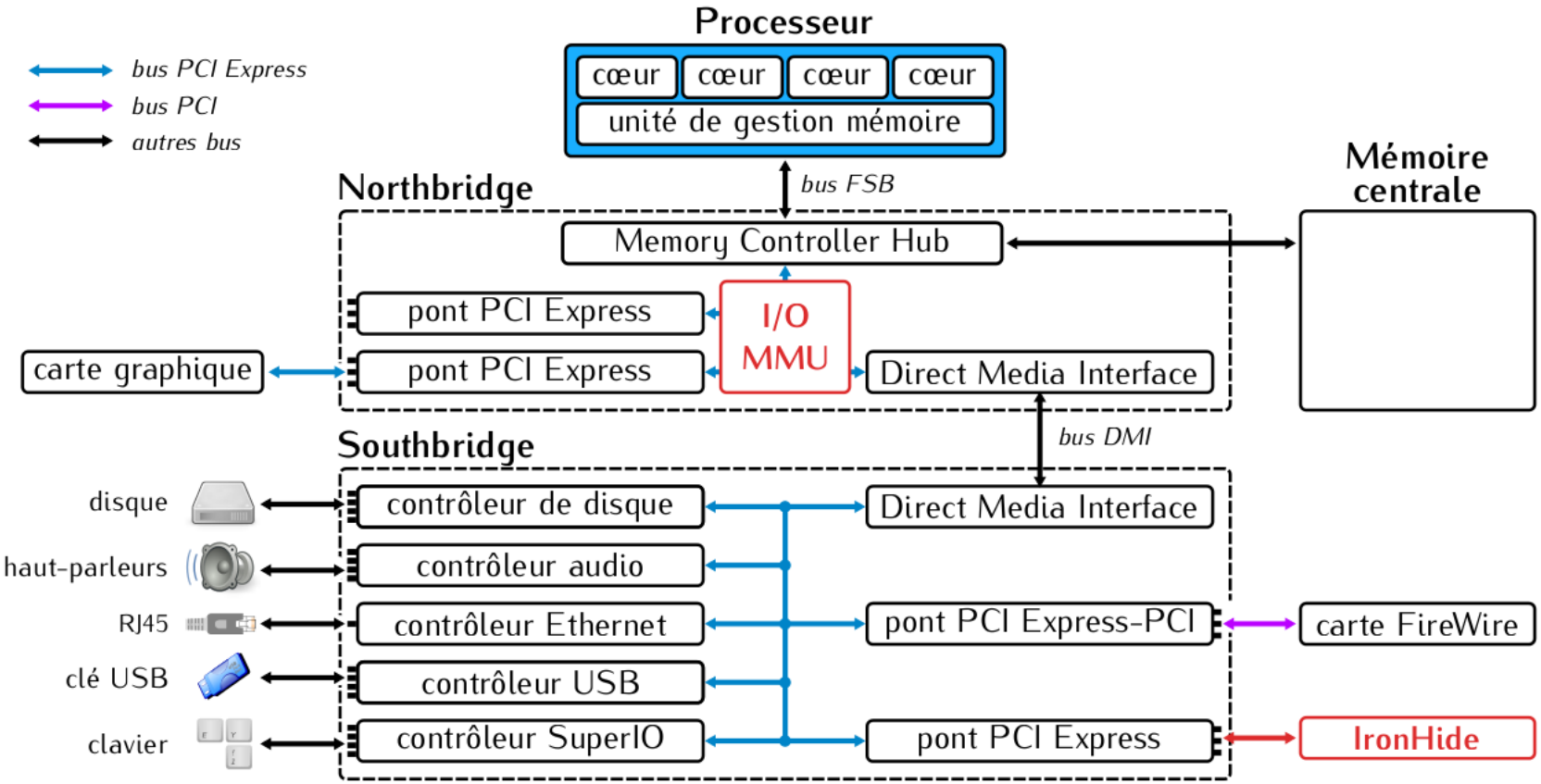


Accès à l'espace d'adressage mémoire principale (3)

Image de la ROM d'extension de la carte graphique

```
user@moniteur$ sudo ./ironhide-memread.py vbios.bin 0xc0000 0xce7ff
[+] Connected to IronHide
[+] Dumping remote memory from 0xc0000 to 0xce7ff to file vbios.bin

user@moniteur$ file vbios.bin
vbios.bin: BIOS (ia32) ROM Ext. IBM comp. Video (116*512)

user@moniteur$ xxd -g1 vbios.bin
0000000: 55 aa 74 eb 4b 37 34 30 30 e9 4c 19 77 cc 56 49
0000010: 44 45 4f 20 0d 00 00 00 f0 01 3a 1d 00 00 49 42
0000020: 4d 20 56 47 41 20 43 6f 6d 70 61 74 69 62 6c 65
0000030: 01 00 00 00 80 00 30 bd 30 36 2f 32 36 2f 30 38
0000040: 00 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00
0000050: e9 f6 34 00 28 10 fe 01 ff 3f fe 7f 00 00 00 00
[...]
```

Accès à l'espace d'adressage mémoire principale (4)

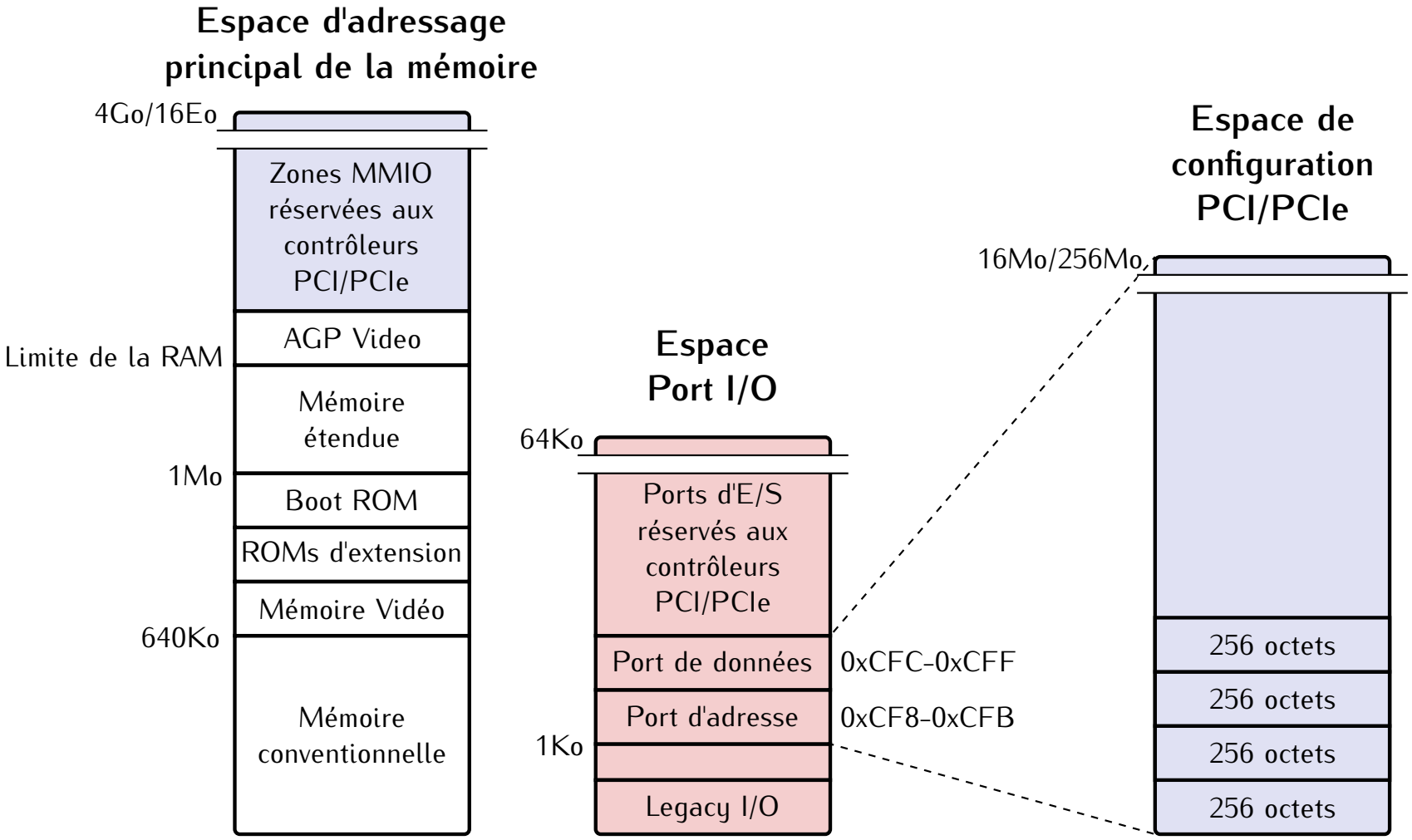


Accès à l'espace d'adressage mémoire principale (5)

		Intel VT-d	Intel VT-d2 [1]
Paquet valide	Respect du protocole (ex. `TC`)	Déni de service	-
	Non-respect du protocole (ex. `ReqID`)	Déni de service	Détecté
Paquet invalide	*Buffer underflow*	-	-
	Buffer overflow	Déni de service	Détecté
	Drapeaux incohérents	-	Détecté
	Champs réservés	-	Détecté

[1] Dans les *chipsets* Intel supportant les processeurs Nehalem et ses successeurs

Accès à l'espace des ports d'entrées-sorties (1)



Accès à l'espace des ports d'entrées-sorties (2)

Image de l'espace des ports d'entrées-sorties

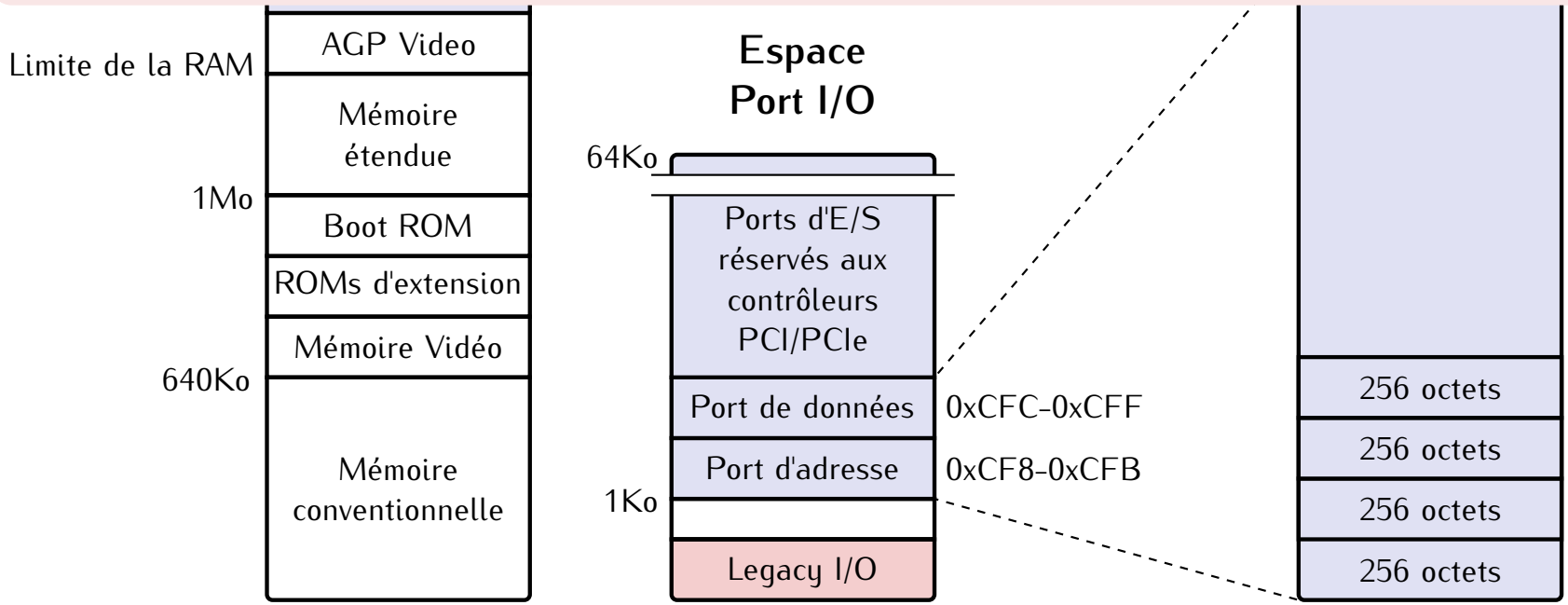
```
user@moniteur$ sudo ./ironhide-ioread.py pio.bin 0x0 0xff
[+] Connected to IronHide
[+] Dumping remote I/O from 0x0000 to 0x00ff to file pio.bin

user@moniteur$ xxd -g1 pio.bin
00000000 8c 0a 0f af 24 92 04 ff 00 ff ff ff ff ff ff 00
*
00000020 01 ff ff ff 01 ff ff ff 01 ff ff ff 01 ff aa 00
00000030 01 ff ff ff 01 ff ff ff 01 ff ff ff 01 ff ff ff
00000040 98 10 ac ff ff ff ff ff ff ff ff ff ff ff ff ff
00000050 d0 06 03 ff ff ff ff ff ff ff ff ff ff ff ff ff
00000060 fe 2c ff ff 74 ff ff ff ff ff ff ff ff ff ff ff
00000070 ff 02 7d 01 0b 02 7d 01 ff ff ff ff ff ff ff ff
00000080 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
[...]
```

Accès à l'espace des ports d'entrées-sorties (3)

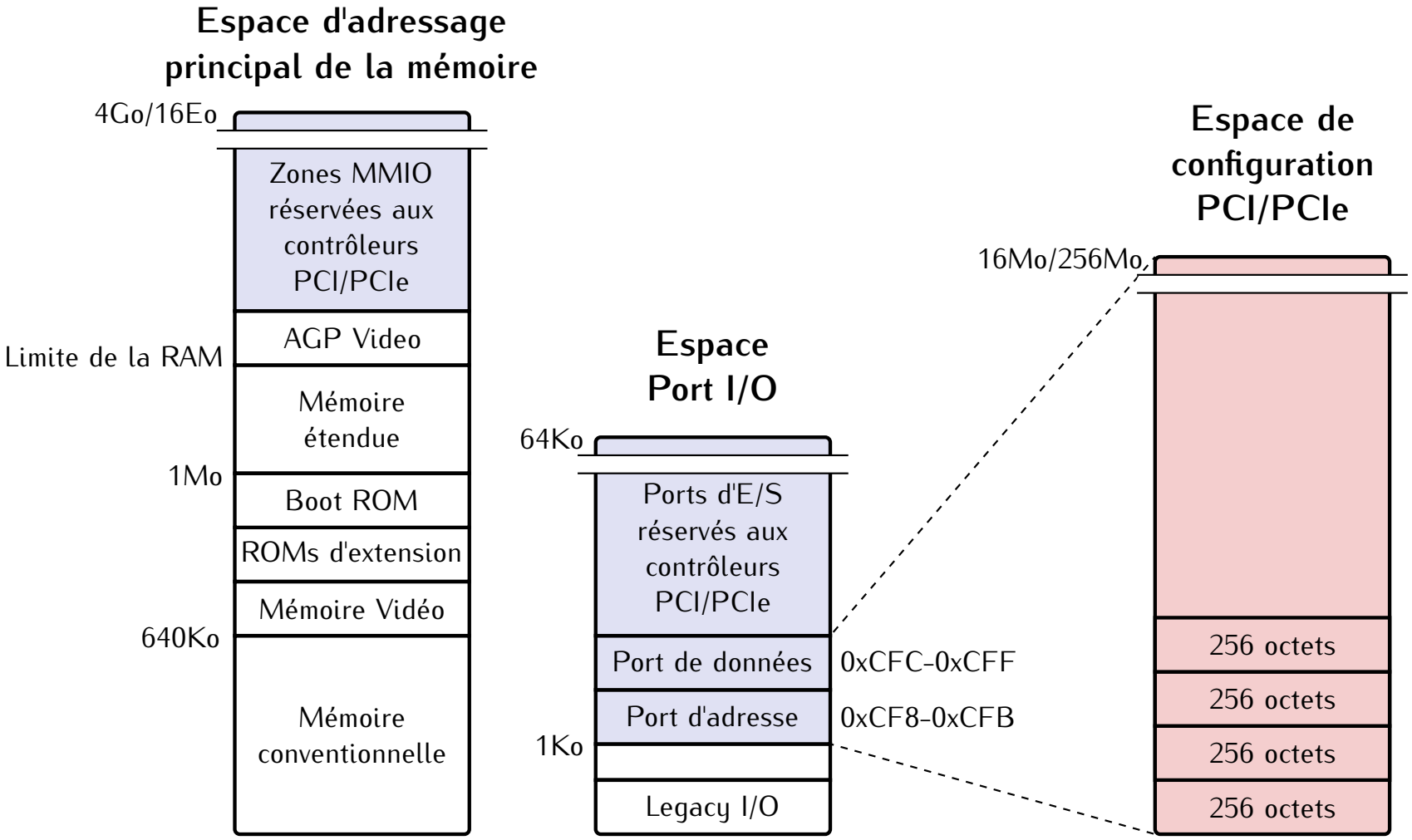
Démonstration avec le contrôleur de clavier

Ports d'entrées-sorties 0x60 et 0x64

Vidéo disponible à l'adresse : `http://homepages.laas.fr/nicomett/Videos`

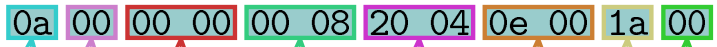


Accès à l'espace de configuration (1)

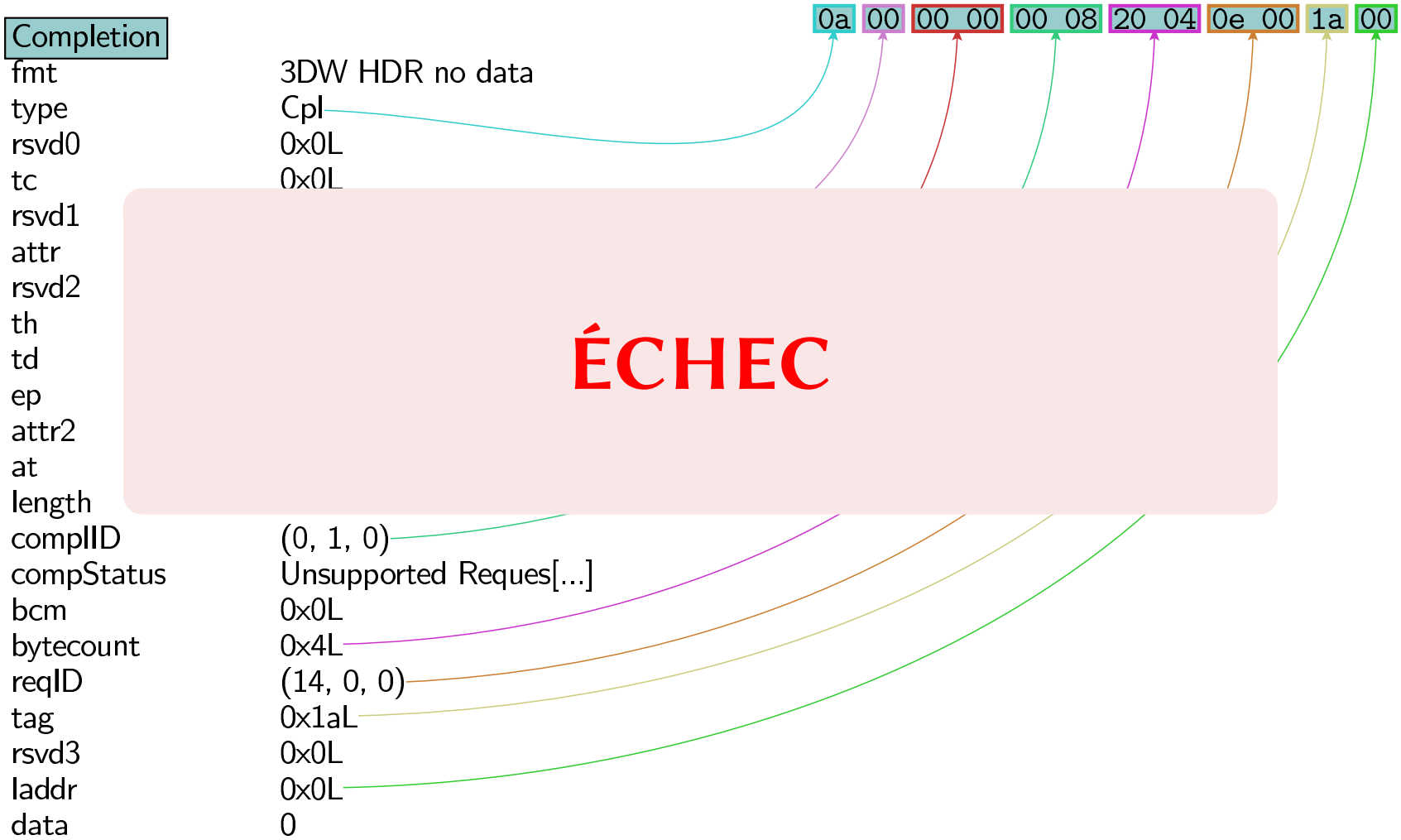


Accès à l'espace de configuration (2)

Completion	
fmt	3DW HDR no data
type	Cpl
rsvd0	0x0L
tc	0x0L
rsvd1	0x0L
attr	0L
rsvd2	0x0L
th	0L
td	0L
ep	0L
attr2	0L
at	0L
length	0x0L
complID	(0, 1, 0)
compStatus	Unsupported Reques[...]
bcm	0x0L
bytecount	0x4L
reqID	(14, 0, 0)
tag	0x1aL
rsvd3	0x0L
laddr	0x0L
data	0



Accès à l'espace de configuration (3)



Accès à l'espace de configuration (4)

Register: MISCCTRLSTS
Device: 0
Function: 0
Offset: 188h

Bit	Attr	Default	Description
			Enable System Error only for AER 1 = When this bit is set, the PCI Express errors do not trigger an MSI interrupt, regardless of the whether MSI is enabled or not. Whether or not PCI Express errors result in a system event like NMI/SMI/PMI/CPEI is dependent on whether the appropriate system error enable bits are set or not.
1	RWO	0h	Inbound Configuration Enable 0 = All inbound configuration transactions are sent a UR response by the receiving PCI Express port. 1 = Inbound configurations are allowed.
0	RW (Device 1–10), RO (Device 0)	0 (Device 1–10), 1 (Device 0)	Set Host Bridge Class Code 1 = Class code register indicates "Host Bridge".

Sommaire de la présentation

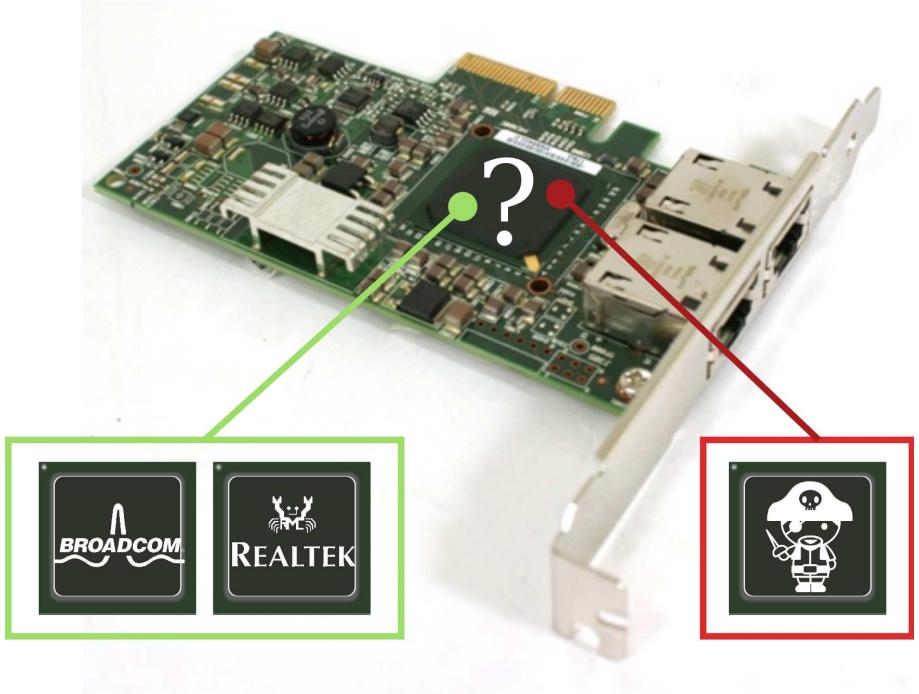
1. Élements d'architecture
2. IronHide : contrôleur d'entrées-sorties générique
3. Quelques résultats d'expérimentations
4. Conclusion & perspectives

Conclusion

	[Petroni *et al.* 04] [Carrier & Grand 04]	[Devine & Vissian 09]	IronHide
connectique	PCI	PCI	PCI-Express
requêtes	DMA	DMA	tous
performant	oui	oui	en cours
extensible	non	non	oui
programmable	non	oui	oui
adaptable	non	oui	oui

Travaux futurs (1)

Émulation d'un contrôleur d'entrées-sorties sur-étagère

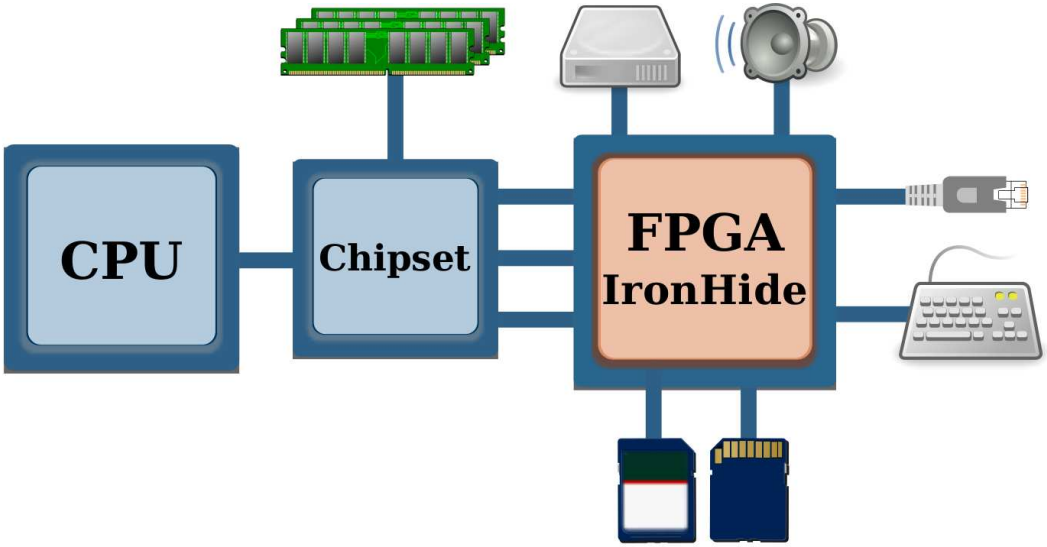


Quelques problématiques ouvertes :

- Quelles approches utiliser pour détecter un tel scénario d'attaque ?
  - → *device attestation protocol*, *device fingerprinting*, ...
- Comment peut-on s'en protéger efficacement ?

Travaux futurs (2)

IDS/NIDS dédié aux contrôleurs d'entrées-sorties



Quelques problématiques ouvertes :

- Peut-on établir un modèle de comportement pour les contrôleurs ?
  - → complétude, exactitude, mais surtout *stabilité* du modèle
  - → caractérisation d'un comportement déviant
- Quelles structures de données et quels algorithmes sont adaptés ?

IronHide
Plate-forme d'attaques par entrées-sorties

Fernand Lone Sang, Vincent Nicomette et Yves Deswarte

Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)
Groupe Tolérance aux fautes et Sûreté de Fonctionnement informatique (TSF)

Symposium sur la Sécurité des Technologies de l'Information et des Communications
SSTIC'2012 – Rennes

LAAS-CNRS

Références (1)

Brian Carrier & Joe Grand.
A Hardware-based Memory Acquisition Procedure for Digital Investigations.
Digital Investigation, vol. 1, no. 1, pages 50–60, February 2004.

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, 3-5 June 2009.

Nick L. Jr. Petroni, Timothy Fraser, Jesus Molina, & William A. Arbaugh.
Copilot - a Coprocessor-based Kernel Runtime Integrity Monitor.
In 13th USENIX Security Symposium, 9-13 August 2004.