

Développement d'une application de maintenance de logiciel embarqué

Dan Pham

Grenoble INP - Esisar

20 juin 2011

Plan

- 1 Contexte du stage
 - Présentation de l'entreprise
 - Problématique
- 2 Réalisation
 - Recherche de la solution
 - Solution retenue
 - Mise en place de la solution
- 3 Résultats
 - Évaluation de la solution
 - Démonstrations
- 4 Bilan

Sommaire

- 1 Contexte du stage
 - Présentation de l'entreprise
 - Problématique
- 2 Réalisation
 - Recherche de la solution
 - Solution retenue
 - Mise en place de la solution
- 3 Résultats
 - Évaluation de la solution
 - Démonstrations
- 4 Bilan

M.G.I.

Digital Printing and Finishing Technology

Expertise

- Constructeur de presses numériques multi-substrats
- Technologie jet d'encre
- Maîtrise quasi complète du produit

L'entreprise

- Siège social à Ivry-sur-Seine
- Orienté vers l'international
- Stratégie d'innovation



Produits concernés

JETcard



Produits concernés

JETvarnish



Objectifs

Contexte

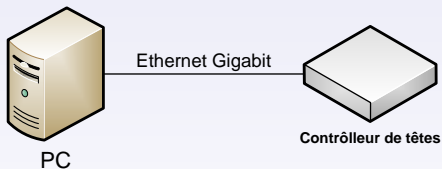
- Mises à jour requérant de la technicité
- Débits de l'USB 2.0 devenus insuffisants

Besoins

- Mise à jour via Ethernet Gigabit en point à point
- Vitesse utile de l'Ethernet : 700 Mbps
- Écriture sur mémoire flash du micro-logiciel sécurisée
- Gestion des mises à jour par interface graphique

Mise à jour à distance

Réalisé au cours du stage



Finalités

Retour sur investissement

- Économies sur les honoraires et déplacements des techniciens
- Éviter l'endommagement prématuré du matériel
- Maîtrise de l'Ethernet Gigabit



Finalités

Retour sur investissement

- Économies sur les honoraires et déplacements des techniciens
- Éviter l'endommagement prématuré du matériel
- Maîtrise de l'Ethernet Gigabit



Finalités

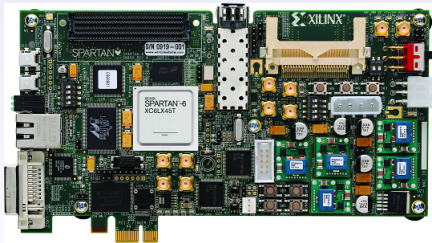
Retour sur investissement

- Économies sur les honoraires et déplacements des techniciens
- Éviter l'endommagement prématuré du matériel
- Maîtrise de l'Ethernet Gigabit



Moyens mis à disposition

Matériels



Xilinx SP605

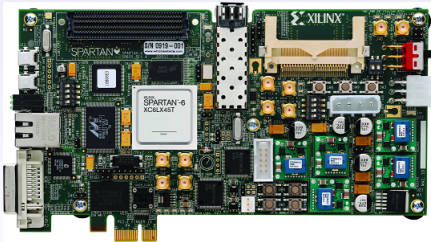
- FPGA Spartan-6 XC6SLX45T
- PHY Ethernet 10/100/1000Mbps
- Mémoire flash sérielle SPI 8Mo
- Prix attractif : 495\$

PC

- Processeur quad core

Moyens mis à disposition

Matériels



Xilinx SP605

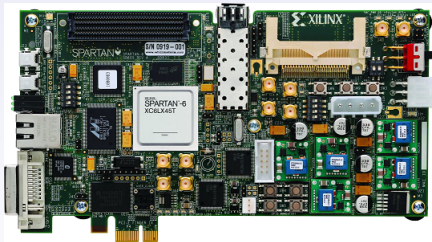
- FPGA Spartan-6 XC6SLX45T
- PHY Ethernet 10/100/1000Mbps
- Mémoire flash sérielle SPI 8Mo
- Prix attractif : 495\$

PC

- Processeur quad core

Moyens mis à disposition

Matériels



Xilinx SP605

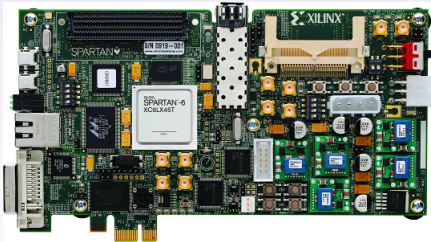
- FPGA Spartan-6 XC6SLX45T
- PHY Ethernet 10/100/1000Mbps
- Mémoire flash sérielle SPI 8Mo
- Prix attractif : 495\$

PC

- Processeur quad core

Moyens mis à disposition

Matériels



Xilinx SP605

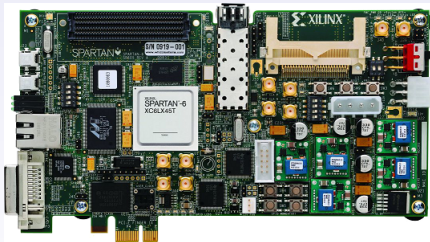
- FPGA Spartan-6 XC6SLX45T
- PHY Ethernet 10/100/1000Mbps
- Mémoire flash sérielle SPI 8Mo
- Prix attractif : 495\$

PC

- Processeur quad core

Moyens mis à disposition

Matériels



Xilinx SP605

- FPGA Spartan-6 XC6SLX45T
- PHY Ethernet 10/100/1000Mbps
- Mémoire flash sérielle SPI 8Mo
- Prix attractif : 495\$

PC

- Processeur quad core

Moyens mis à disposition

Logiciels

Langages et vérification

- Langage de description de matériel Verilog
- Vérification du code via Test Benches
- Analyseur logique ChipScope
- Protocoles et interface côté PC en C#
- Analyseur réseau Wireshark



Moyens mis à disposition

Logiciels

Langages et vérification

- Langage de description de matériel Verilog
- Vérification du code via Test Benches
- Analyseur logique ChipScope
- Protocoles et interface côté PC en C#
- Analyseur réseau Wireshark



Moyens mis à disposition

Logiciels

Langages et vérification

- Langage de description de matériel Verilog
- Vérification du code via Test Benches
- Analyseur logique ChipScope
- Protocoles et interface côté PC en C#
- Analyseur réseau Wireshark



Moyens mis à disposition

Logiciels

Langages et vérification

- Langage de description de matériel Verilog
- Vérification du code via Test Benches
- Analyseur logique ChipScope
- Protocoles et interface côté PC en C#
- Analyseur réseau Wireshark



Moyens mis à disposition

Logiciels

Langages et vérification

- Langage de description de matériel Verilog
- Vérification du code via Test Benches
- Analyseur logique ChipScope
- Protocoles et interface côté PC en C#
- Analyseur réseau Wireshark



Sommaire

- 1 Contexte du stage
 - Présentation de l'entreprise
 - Problématique
- 2 **Réalisation**
 - Recherche de la solution
 - Solution retenue
 - Mise en place de la solution
- 3 Résultats
 - Évaluation de la solution
 - Démonstrations
- 4 Bilan

Contraintes et critères de l'interface Gigabit

Contraintes fortes

- Accessibilité des protocoles réseau en C# sous Windows
- Performance minimale de 700 Mbps
- Code Verilog

Critères de choix

- Type de solution
- Type de licence
- Couches réseau
- Empreinte matérielle

Contraintes et critères de l'interface Gigabit

Contraintes fortes

- Accessibilité des protocoles réseau en C# sous Windows
- Performance minimale de 700 Mbps
- Code Verilog

Critères de choix

- Type de solution
- Type de licence
- Couches réseau
- Empreinte matérielle

Contraintes et critères de l'interface Gigabit

Contraintes fortes

- Accessibilité des protocoles réseau en C# sous Windows
- Performance minimale de 700 Mbps
- Code Verilog

Critères de choix

- Type de solution
- Type de licence
- Couches réseau
- Empreinte matérielle

Contraintes et critères de l'interface Gigabit

Contraintes fortes

- Accessibilité des protocoles réseau en C# sous Windows
- Performance minimale de 700 Mbps
- Code Verilog

Critères de choix

- Type de solution
- Type de licence
- Couches réseau
- Empreinte matérielle

Contraintes et critères de l'interface Gigabit

Contraintes fortes

- Accessibilité des protocoles réseau en C# sous Windows
- Performance minimale de 700 Mbps
- Code Verilog

Critères de choix

- Type de solution
- Type de licence
- Couches réseau
- Empreinte matérielle

Contraintes et critères de l'interface Gigabit

Contraintes fortes

- Accessibilité des protocoles réseau en C# sous Windows
- Performance minimale de 700 Mbps
- Code Verilog

Critères de choix

- Type de solution
- Type de licence
- Couches réseau
- Empreinte matérielle

Contraintes et critères de l'interface Gigabit

Contraintes fortes

- Accessibilité des protocoles réseau en C# sous Windows
- Performance minimale de 700 Mbps
- Code Verilog

Critères de choix

- Type de solution
- Type de licence
- Couches réseau
- Empreinte matérielle

Solution retenue

Solution retenue

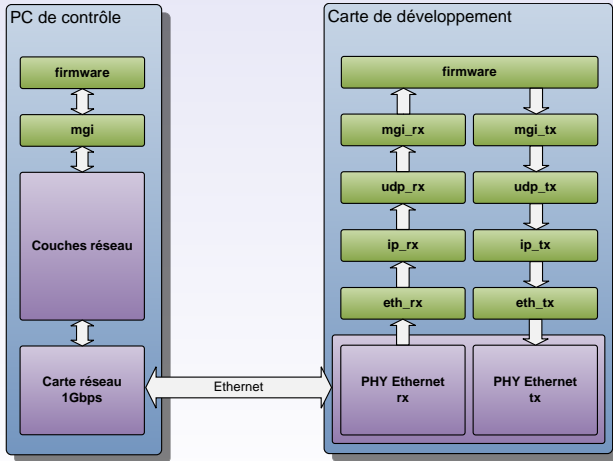
- Implémentation d'une pile Ethernet MAC/IP/UDP matérielle minimaliste
- Création d'un protocole assurant l'intégrité des données

Solution retenue

Solution retenue

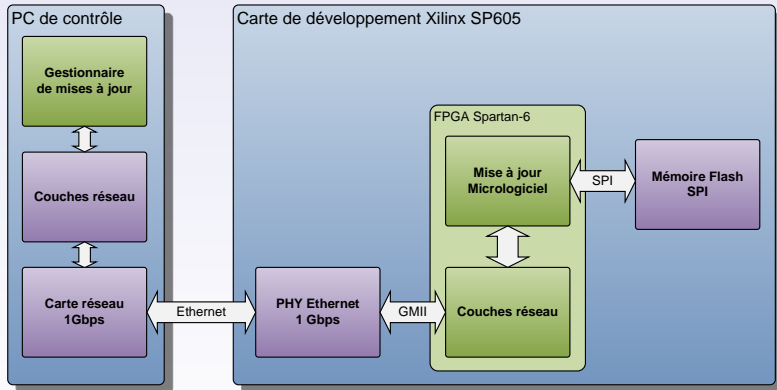
- Implémentation d'une pile Ethernet MAC/IP/UDP matérielle minimaliste
- Création d'un protocole assurant l'intégrité des données

Modèle en couches



Architecture matérielle et logicielle

Vue d'ensemble



Couche Physique

PHY Ethernet Gigabit

- Assure le codage, la gestion des collisions et la négociation du lien Ethernet
- Interface de communication parallèle 8 bits
- Fonctionnement à 125 MHz

Mise en place

- Phase de documentation difficile
- Logique supplémentaire pour échantillonner les signaux

Couche Physique

PHY Ethernet Gigabit

- Assure le codage, la gestion des collisions et la négociation du lien Ethernet
- Interface de communication parallèle 8 bits
- Fonctionnement à 125 MHz

Mise en place

- Phase de documentation difficile
- Logique supplémentaire pour échantillonner les signaux

Couche Physique

PHY Ethernet Gigabit

- Assure le codage, la gestion des collisions et la négociation du lien Ethernet
- Interface de communication parallèle 8 bits
- Fonctionnement à 125 MHz

Mise en place

- Phase de documentation difficile
- Logique supplémentaire pour échantillonner les signaux

Couche Physique

PHY Ethernet Gigabit

- Assure le codage, la gestion des collisions et la négociation du lien Ethernet
- Interface de communication parallèle 8 bits
- Fonctionnement à 125 MHz

Mise en place

- Phase de documentation difficile
- Logique supplémentaire pour échantillonner les signaux

Couche Physique

PHY Ethernet Gigabit

- Assure le codage, la gestion des collisions et la négociation du lien Ethernet
- Interface de communication parallèle 8 bits
- Fonctionnement à 125 MHz

Mise en place

- Phase de documentation difficile
- Logique supplémentaire pour échantillonner les signaux

Couche Ethernet MAC

Couche Ethernet MAC

- Permet de désigner l'interface physique cible de réception
- Protocole de résolution d'adresse (ARP) non nécessaire car réseau en point à point

Mise en place

- Debug difficile car accès impossible à la couche physique sous Windows
- Calcul de la somme de contrôle en émission
- Padding : Contrôle de la taille des trames émises

Couche Ethernet MAC

Couche Ethernet MAC

- Permet de désigner l'interface physique cible de réception
- Protocole de résolution d'adresse (ARP) non nécessaire car réseau en point à point

Mise en place

- Debug difficile car accès impossible à la couche physique sous Windows
- Calcul de la somme de contrôle en émission
- Padding : Contrôle de la taille des trames émises

Couche Ethernet MAC

Couche Ethernet MAC

- Permet de désigner l'interface physique cible de réception
- Protocole de résolution d'adresse (ARP) non nécessaire car réseau en point à point

Mise en place

- Debug difficile car accès impossible à la couche physique sous Windows
- Calcul de la somme de contrôle en émission
- Padding : Contrôle de la taille des trames émises

Couche Ethernet MAC

Couche Ethernet MAC

- Permet de désigner l'interface physique cible de réception
- Protocole de résolution d'adresse (ARP) non nécessaire car réseau en point à point

Mise en place

- Debug difficile car accès impossible à la couche physique sous Windows
- Calcul de la somme de contrôle en émission
- Padding : Contrôle de la taille des trames émises

Couche Ethernet MAC

Couche Ethernet MAC

- Permet de désigner l'interface physique cible de réception
- Protocole de résolution d'adresse (ARP) non nécessaire car réseau en point à point

Mise en place

- Debug difficile car accès impossible à la couche physique sous Windows
- Calcul de la somme de contrôle en émission
- Padding : Contrôle de la taille des trames émises

Couche Ethernet MAC

Couche Ethernet MAC

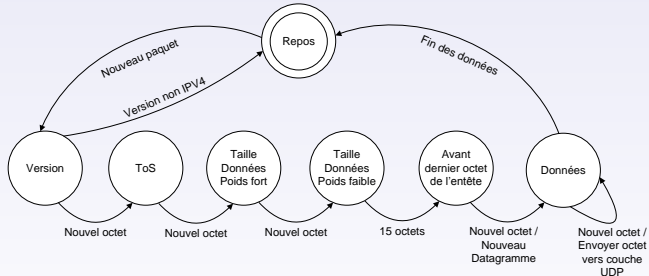
- Permet de désigner l'interface physique cible de réception
- Protocole de résolution d'adresse (ARP) non nécessaire car réseau en point à point

Mise en place

- Debug difficile car accès impossible à la couche physique sous Windows
- Calcul de la somme de contrôle en émission
- Padding : Contrôle de la taille des trames émises

Couche IP en réception

Machine d'états



Protocole d'intégrité

Protocole "MGI"

Couche MGI

- UDP : Acheminement des paquets non garanti
- Point à point : Les paquets arrivent dans l'ordre

Mise en place

- Diminuer au maximum la complexité dans la carte

Protocole d'intégrité

Protocole "MGI"

Couche MGI

- UDP : Acheminement des paquets non garanti
- Point à point : Les paquets arrivent dans l'ordre

Mise en place

- Diminuer au maximum la complexité dans la carte

Protocole d'intégrité

Protocole "MGI"

Couche MGI

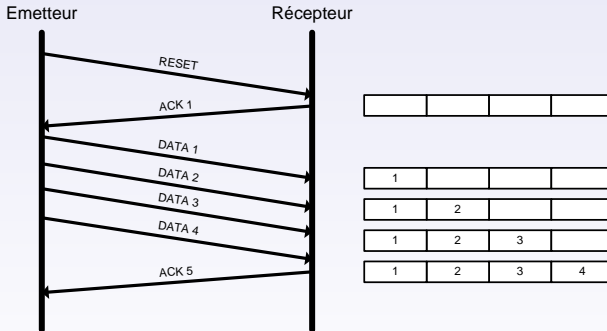
- UDP : Acheminement des paquets non garanti
- Point à point : Les paquets arrivent dans l'ordre

Mise en place

- Diminuer au maximum la complexité dans la carte

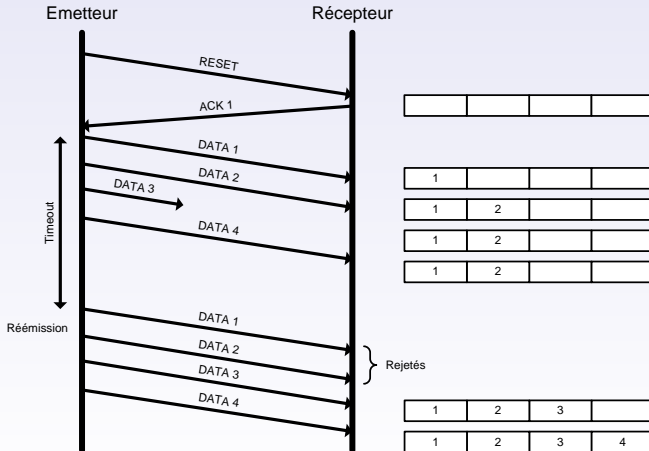
Protocole d'intégrité

Fonctionnement général



Protocole d'intégrité

Erreur

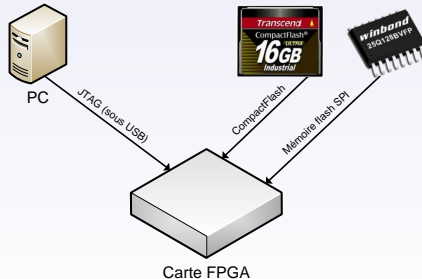


Couche Firmware

Modes de mise à jour du micrologiciel

Micrologiciel ou bitstream

- Tableau de bits contenant la configuration matérielle du FPGA utilisée au démarrage



Couche Firmware

Caractérisation du bitstream

Extraction du bitstream

- Spécification du bitstream non disponible
- Formé en deux parties : entête et données
- Champs de taille variable

000000	00 09 0f f0 0f f0 0f f0 0f f0 00 00 01 61 00 23	...ð.ð.ð.ð...a.#
000010	67 6f 6c 64 65 6e 5f 53 50 36 30 35 2e 6e 63 64	golden_SP605.ncd
000020	3b 55 73 65 72 49 44 3d 30 78 46 46 46 46 46 46	;UserID=0xFFFFFFFF
000030	46 46 00 62 00 0e 36 73 6c 78 34 35 74 66 67 67	FF.b..6s1x45tfgg
000040	34 38 34 00 63 00 0b 32 30 31 31 2f 30 34 2f 32	484.c..2011/04/2
000050	31 00 64 00 09 31 36 3a 31 32 3a 35 30 00 65 00	1.d..16:12:50.e.
000060	16 a6 74 ff ff ff ff ff ff ff ff ff ff ff ff	.!tyyyyyyyyyyyyyyy

Couche Firmware

Caractérisation du bitstream

Extraction du bitstream

- Spécification du bitstream non disponible
- Formé en deux parties : entête et données
- Champs de taille variable

000000	00 09 0f f0 0f f0 0f f0 0f f0 00 00 01 61 00 23	... ð.ð.ð.ð... a.#
000010	67 6f 6c 64 65 6e 5f 53 50 36 30 35 2e 6e 63 64	golden_SP605.ncd
000020	3b 55 73 65 72 49 44 3d 30 78 46 46 46 46 46 46	;UserID=0xFFFFFFFF
000030	46 46 00 62 00 0e 36 73 6c 78 34 35 74 66 67 67	FF.b..6s1x45tfgg
000040	34 38 34 00 63 00 0b 32 30 31 31 2f 30 34 2f 32	484.c..2011/04/2
000050	31 00 64 00 09 31 36 3a 31 32 3a 35 30 00 65 00	1.d..16:12:50.e.
000060	16 a6 74 ff ff ff ff ff ff ff ff ff ff ff ff	!tyyyyyyyyyyyyyyy

Couche Firmware

Caractérisation du bitstream

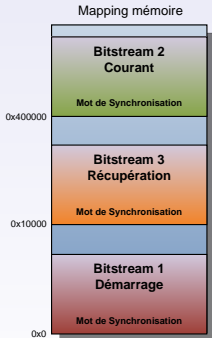
Extraction du bitstream

- Spécification du bitstream non disponible
- Formé en deux parties : entête et données
- Champs de taille variable

000000	00 09 0f f0 0f f0 0f f0 0f f0 00 00 01 61 00 23	...ð.ð.ð.ð...a.#
000010	67 6f 6c 64 65 6e 5f 53 50 36 30 35 2e 6e 63 64	golden_SP605.ncd
000020	3b 55 73 65 72 49 44 3d 30 78 46 46 46 46 46 46	;UserID=0xFFFFFFFF
000030	46 46 00 62 00 0e 36 73 6c 78 34 35 74 66 67 67	FF.b..6s1x45tfgg
000040	34 38 34 00 63 00 0b 32 30 31 31 2f 30 34 2f 32	484.c..2011/04/2
000050	31 00 64 00 09 31 36 3a 31 32 3a 35 30 00 65 00	1.d..16:12:50.e.
000060	16 a6 74 ff ff ff ff ff ff ff ff ff ff ff ff	!tyyyyyyyyyyyyyyy

Couche Firmware

Mode de récupération



Etapas de configuration

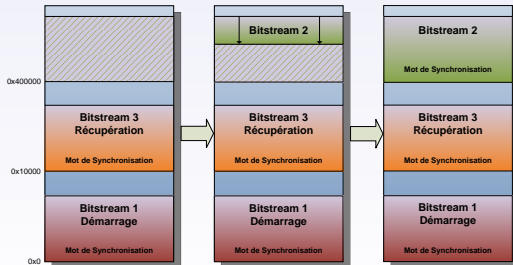
- Bitstream 1** Entête, synchronisation et adresse des deux bitstreams
- Bitstream 2** Architecture courante
- Bitstream 3** Récupération, réception via Ethernet et écriture sur flash SPI

Couche Firmware

Écriture du bitstream

Routines

- Implémentation des routines d'écriture, de lecture et d'effacement de la mémoire
- Remise à '1' de la mémoire avant réécriture

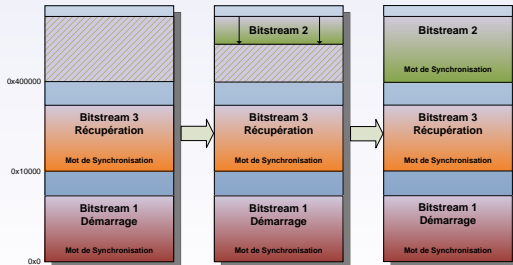


Couche Firmware

Écriture du bitstream

Routines

- Implémentation des routines d'écriture, de lecture et d'effacement de la mémoire
- Remise à '1' de la mémoire avant réécriture



Sommaire

- 1 Contexte du stage
 - Présentation de l'entreprise
 - Problématique
- 2 Réalisation
 - Recherche de la solution
 - Solution retenue
 - Mise en place de la solution
- 3 **Résultats**
 - **Évaluation de la solution**
 - **Démonstrations**
- 4 Bilan

Évaluation de l'empreinte matérielle

Empreinte matérielle

- Solution actuelle (sur Spartan-6 LX45T)

Slices	LUTs	FFs	BRAMs
487 /6822:7%	1308 /27288:5%	1487 /54576:3%	16 /116:13%

- Soft TEMAC Xilinx (sur Virtex-7)

Slices	LUTs	FFs	BRAMs
400	600	800	0

Évaluation de l'empreinte matérielle

Empreinte matérielle

- Solution actuelle (sur Spartan-6 LX45T)

Slices	LUTs	FFs	BRAMs
487 /6822:7%	1308 /27288:5%	1487 /54576:3%	16 /116:13%

- Soft TEMAC Xilinx (sur Virtex-7)

Slices	LUTs	FFs	BRAMs
400	600	800	0

Évaluation des performances

Méthodologie

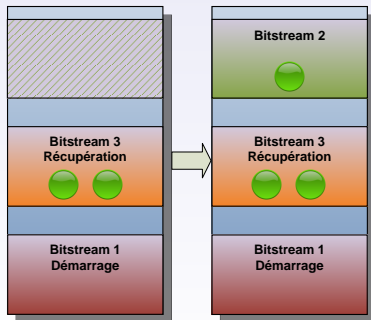
- Cartes Ethernet Gigabit PCI et PCI Express
- Utilisation de Jumbo frames 7Ko

	PCI (Mbps)	PCIe (Mbps)
Windows C# (UDP)	520	290
Windows C# DLL C++(UDP)	510	280
Windows C# (TCP)	330	990
Linux Sockets (UDP)	550	995

Démonstrations

Démonstration 1

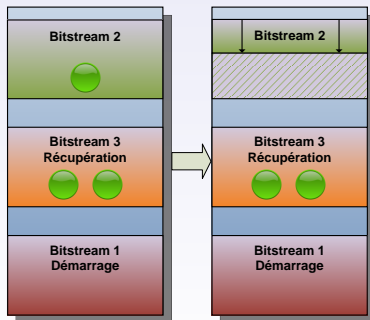
- Chargement d'un bitstream complet



Démonstrations

Démonstration 2

- Interruption lors du chargement d'un bitstream



Sommaire

- 1 Contexte du stage
 - Présentation de l'entreprise
 - Problématique
- 2 Réalisation
 - Recherche de la solution
 - Solution retenue
 - Mise en place de la solution
- 3 Résultats
 - Évaluation de la solution
 - Démonstrations
- 4 Bilan

Bilan

Difficultés rencontrées

- **Prise en main du Verilog, environnement C#**
- Synchronisation du bloc de réception
- Documentation du PHY et spécification du fichier bitstream
- Erreurs Xilinx incohérentes, erreur dans l'IP de FIFO
- Horloges multiples, machines d'états asynchrones

Bilan

Difficultés rencontrées

- Prise en main du Verilog, environnement C#
- Synchronisation du bloc de réception
- Documentation du PHY et spécification du fichier bitstream
- Erreurs Xilinx incohérentes, erreur dans l'IP de FIFO
- Horloges multiples, machines d'états asynchrones

Bilan

Difficultés rencontrées

- Prise en main du Verilog, environnement C#
- Synchronisation du bloc de réception
- Documentation du PHY et spécification du fichier bitstream
- Erreurs Xilinx incohérentes, erreur dans l'IP de FIFO
- Horloges multiples, machines d'états asynchrones

Bilan

Difficultés rencontrées

- Prise en main du Verilog, environnement C#
- Synchronisation du bloc de réception
- Documentation du PHY et spécification du fichier bitstream
- Erreurs Xilinx incohérentes, erreur dans l'IP de FIFO
- Horloges multiples, machines d'états asynchrones

Bilan

Difficultés rencontrées

- Prise en main du Verilog, environnement C#
- Synchronisation du bloc de réception
- Documentation du PHY et spécification du fichier bitstream
- Erreurs Xilinx incohérentes, erreur dans l'IP de FIFO
- Horloges multiples, machines d'états asynchrones

Bilan

Bilan

- Un projet enrichissant et conséquent
- Un développement typique de l'embarqué

Bilan

Bilan

- Un projet enrichissant et conséquent
- Un développement typique de l'embarqué

Merci de votre attention.

Étude des solutions existantes

Solutions pile protocolaire UDP/IP

- Description intégrale des protocoles en HDL
- Utiliser une hard-macro MAC avec description des autres protocoles en HDL
- Micro-contrôleur implémentant la MAC avec description des autres protocoles en HDL
- Processeur softcore exécutant le protocole en stand-alone
- Processeur softcore exécutant un OS intégrant la pile protocolaire

Étude des solutions existantes

Solutions pile protocolaire UDP/IP

- Description intégrale des protocoles en HDL
- Utiliser une hard-macro MAC avec description des autres protocoles en HDL
- Micro-contrôleur implémentant la MAC avec description des autres protocoles en HDL
- Processeur softcore exécutant le protocole en stand-alone
- Processeur softcore exécutant un OS intégrant la pile protocolaire

Étude des solutions existantes

Solutions pile protocolaire UDP/IP

- Description intégrale des protocoles en HDL
- Utiliser une hard-macro MAC avec description des autres protocoles en HDL
- Micro-contrôleur implémentant la MAC avec description des autres protocoles en HDL
- Processeur softcore exécutant le protocole en stand-alone
- Processeur softcore exécutant un OS intégrant la pile protocolaire

Étude des solutions existantes

Solutions pile protocolaire UDP/IP

- Description intégrale des protocoles en HDL
- Utiliser une hard-macro MAC avec description des autres protocoles en HDL
- Micro-contrôleur implémentant la MAC avec description des autres protocoles en HDL
- Processeur softcore exécutant le protocole en stand-alone
- Processeur softcore exécutant un OS intégrant la pile protocolaire

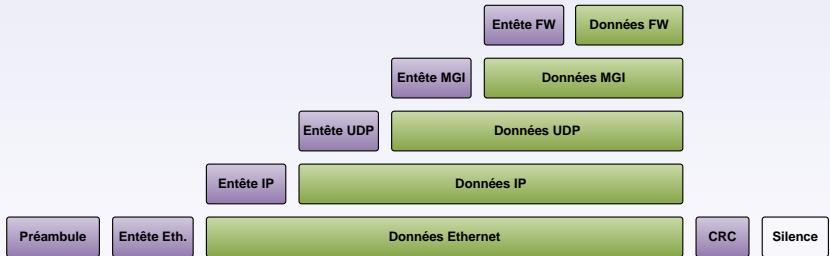
Étude des solutions existantes

Solutions pile protocolaire UDP/IP

- Description intégrale des protocoles en HDL
- Utiliser une hard-macro MAC avec description des autres protocoles en HDL
- Micro-contrôleur implémentant la MAC avec description des autres protocoles en HDL
- Processeur softcore exécutant le protocole en stand-alone
- Processeur softcore exécutant un OS intégrant la pile protocolaire

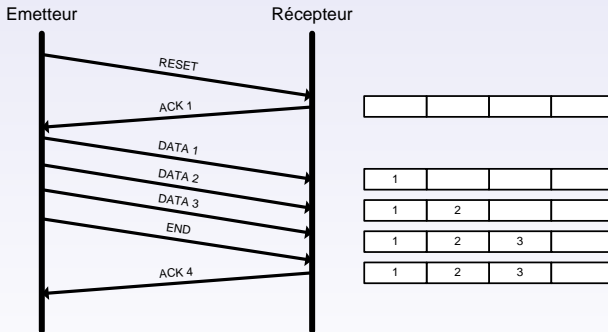
Modèle en couches

Encapsulation des protocoles



Protocole d'intégrité

Fin d'émission



Mise à jour à distance

But à terme

