h e p i a
Haute école du paysage, d'ingénierie
et d'architecture de Genève

LPSC

GTP et Aurora

Joachim Schmidt

Hepia - CoRES

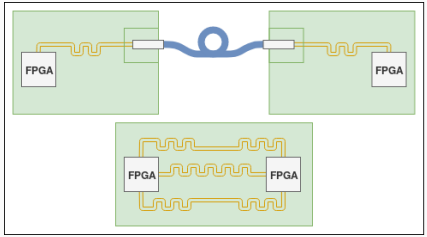
Table des matières

Introduction

Protocole Aurora

Principaux concepts

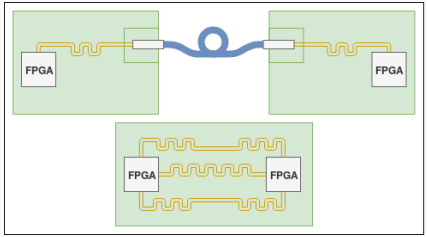
- Protocole développé par Xilinx.

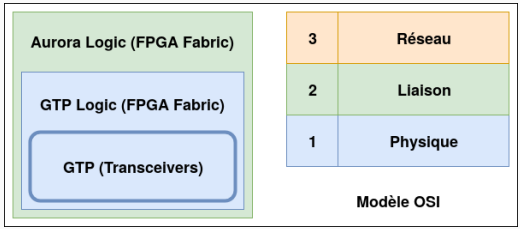


Protocole Aurora

Principaux concepts

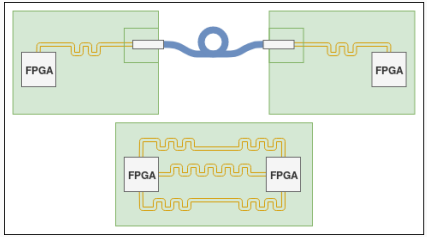
- Protocole développé par Xilinx.
- Protocole de communication de type *liaison* (couche 2 selon le modèle OSI).

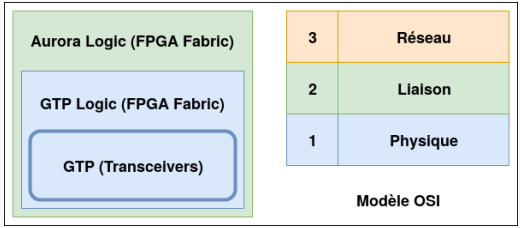




Principaux concepts

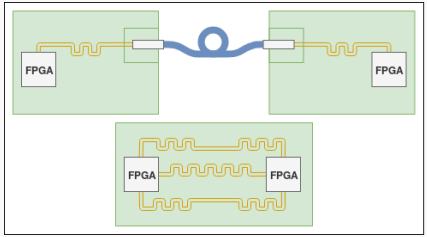
- Protocole développé par Xilinx.
- Protocole de communication de type *liaison* (couche 2 selon le modèle OSI).
- Permet des communications *point à point* (pas de notion d'adressage).

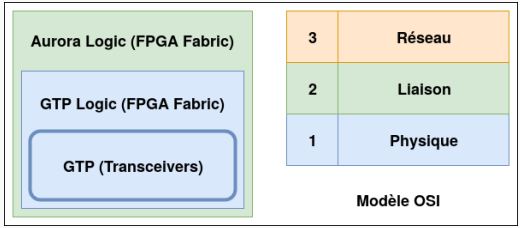




Principaux concepts

- Protocole développé par Xilinx.
- Protocole de communication de type *liaison* (couche 2 selon le modèle OSI).
- Permet des communications *point à point* (pas de notion d'adressage).
- Permet d'effectuer des communications *multi-gigabits*.





IP Core Aurora

Principales caractéristiques

- Débit compris entre 480 [Mbps] et 84.48 [Gbps].

IP Core Aurora

Principales caractéristiques

- Débit compris entre 480 [Mbps] et 84.48 [Gbps].
- Possibilité d'agréger jusqu'à 16 lignes pour les transceivers GTX/GTH (Serie 7 et Ultrascale).

Références [4]

IP Core Aurora

Principales caractéristiques

- Débit compris entre 480 [Mbps] et 84.48 [Gbps].
- Possibilité d'agréger jusqu'à 16 lignes pour les transceivers GTX/GTH (Serie 7 et Ultrascale).
- Possibilité d'agréger jusqu'à 4 lignes pour les transceivers GTP (Serie 7).

Références [4]

IP Core Aurora

Principales caractéristiques

- Débit compris entre 480 [Mbps] et 84.48 [Gbps].
- Possibilité d'agréger jusqu'à 16 lignes pour les transceivers GTX/GTH (Serie 7 et Ultrascale).
- Possibilité d'agréger jusqu'à 4 lignes pour les transceivers GTP (Serie 7).
- Pré-encodage 8B10B ou 64B66B.

Références [4]

IP Core Aurora

Principales caractéristiques

- Débit compris entre 480 [Mbps] et 84.48 [Gbps].
- Possibilité d'agréger jusqu'à 16 lignes pour les transceivers GTX/GTH (Serie 7 et Ultrascale).
- Possibilité d'agréger jusqu'à 4 lignes pour les transceivers GTP (Serie 7).
- Pré-encodage 8B10B ou 64B66B.
- Interface utilisateur AXI4 Streaming / Framing.

IP Core Aurora

Principales caractéristiques

- Débit compris entre 480 [Mbps] et 84.48 [Gbps].
- Possibilité d'agréger jusqu'à 16 lignes pour les transceivers GTX/GTH (Serie 7 et Ultrascale).
- Possibilité d'agréger jusqu'à 4 lignes pour les transceivers GTP (Serie 7).
- Pré-encodage 8B10B ou 64B66B.
- Interface utilisateur AXI4 Streaming / Framing.
- Communication Full-duplex ou Simplex.

Références [4]

IP Core Aurora

Principales caractéristiques

- Contrôle de flux utilisateur ou natif (*UFC* et *NFC*).

IP Core Aurora

Principales caractéristiques

- Contrôle de flux utilisateur ou natif (*UFC* et *NFC*).
- Logique Hot-plug.

IP Core Aurora

Principales caractéristiques

- Contrôle de flux utilisateur ou natif (*UFC* et *NFC*).
- Logique Hot-plug.
- Communication *asynchrone*.

Problème potentiel

La conséquence d'une communication *asynchrone* est que le temps de transmission est *non déterministe*. Dès lors, il est fortement conseiller de ne pas utiliser ce protocol pour des applications temps réel.

IP Core Aurora

- Transceivers *GTP* pour les FPGA Artix 7 et jusqu'aux Zynq 7020.

Références [4, 2]

IP Core Aurora

- Transceivers *GTP* pour les FPGA Artix 7 et jusqu'aux Zynq 7020.
- Débit maximum de *6.6 [Gbps]* par GTP.

Références [4, 2]

IP Core Aurora

- Transceivers *GTP* pour les FPGA Artix 7 et jusqu'aux Zynq 7020.
- Débit maximum de *6.6 [Gbps]* par GTP.
- Maximum de 4 *lignes* (GTP) par canal Aurora.

Références [4, 2]

IP Core Aurora

- Transceivers *GTP* pour les FPGA Artix 7 et jusqu'aux Zynq 7020.
- Débit maximum de *6.6 [Gbps]* par GTP.
- Maximum de 4 *lignes* (GTP) par canal Aurora.
- *Mots de 2 ou 4 [octets]* pour l'interface utilisateur (AXI4 Streaming / Framing).

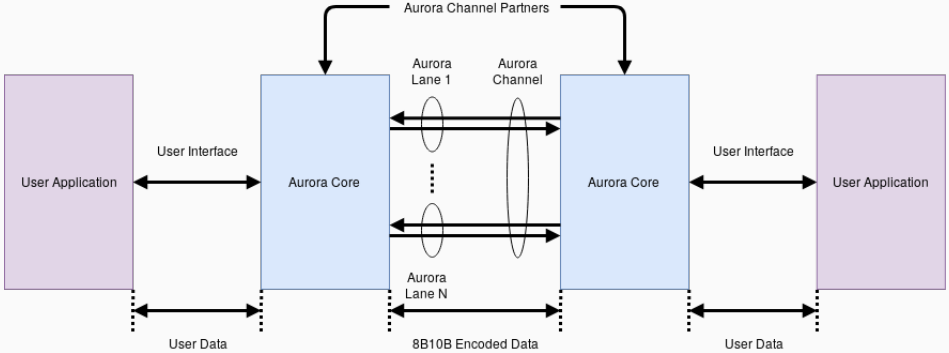
Références [4, 2]

IP Core Aurora

- Transceivers *GTP* pour les FPGA Artix 7 et jusqu'aux Zynq 7020.
- Débit maximum de *6.6 [Gbps]* par GTP.
- Maximum de 4 *lignes* (GTP) par canal Aurora.
- *Mots de 2 ou 4 [octets]* pour l'interface utilisateur (AXI4 Streaming / Framing).
- L'interface AXI4 peut être d'au plus *128 [bits]* (4 lignes * 4 [octets]).

Références [4, 2]

Schéma bloc



Agrégation de liens

Voir page 31

Ordre de transmission des octets

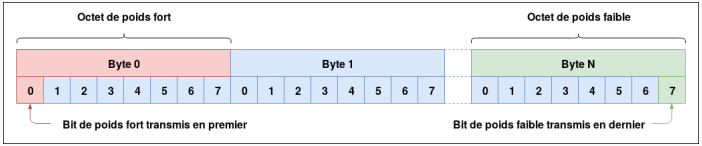
Dans le protocole Aurora les octets sont organisés en *big-endian*. Il s'agit de l'ordre de transmission réseau. Par conséquent, l'ordre des bits du vecteur *tdata* de l'interface AXI4 Streaming / Framing est *inversé*.

Big-endian

std_logic_vector(0 to (N - 1))

Little-endian

std_logic_vector((N - 1) downto 0)



Attention

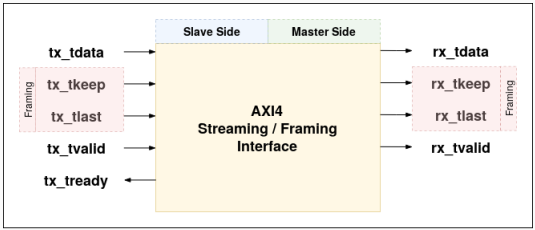
Les indices représentent l'ordre de transmission et non le poids des octets et des bits.

Interfaces utilisateurs

L'interface utilisateur utilise un bus AXI4 Stream décrit par la norme AMBA. Celui-ci permet de gérer des flux de données.

Deux modes sont disponibles

- Le mode *Streaming*, pour la gestion d'un simple flux de données.

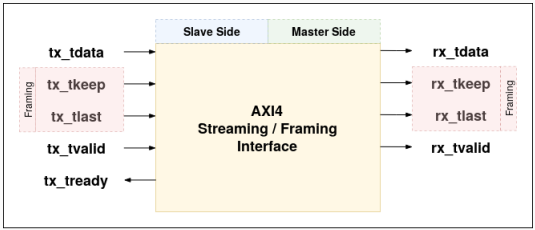


AXI4 Streaming / Framing

L'interface utilisateur utilise un bus AXI4 Stream décrit par la norme AMBA. Celui-ci permet de gérer des flux de données.

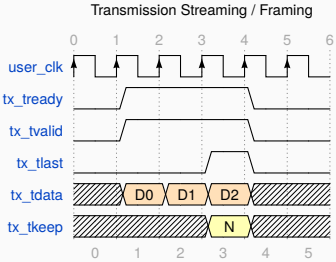
Deux modes sont disponibles

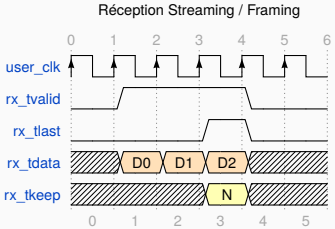
- Le mode *Streaming*, pour la gestion d'un simple flux de données.
- Le mode *Framing*, pour le gestion d'un flux de paquets de données (tkeep et tlast).



Références [4]

AXI4 Streaming / Framing





Le signal *tready* est désélectionné lorsque des caractères K sont transmis. Dans ce cas, on laisse *tvalid* actif, mais on ne modifie pas *tdata* au coup d'horloge suivant.

Contrôle de flux Aurora

Aurora propose deux interfaces de contrôle de flux depuis l'interface utilisateur.

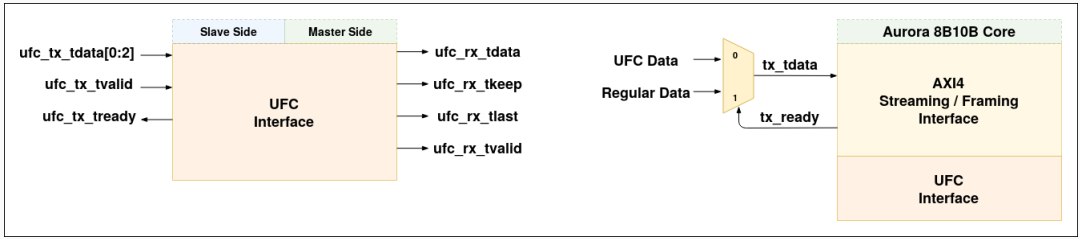
- L'interface *NFC* : Permet de contrôler le débit en insérant des messages IDLE. Le récepteur indique à l'interface Aurora partenaire combien de message IDLE elle doit insérer dans la transmission. Ceci est principalement utile pour éviter des débordement de FIFO.

Contrôle de flux Aurora

Aurora propose deux interfaces de contrôle de flux depuis l'interface utilisateur.

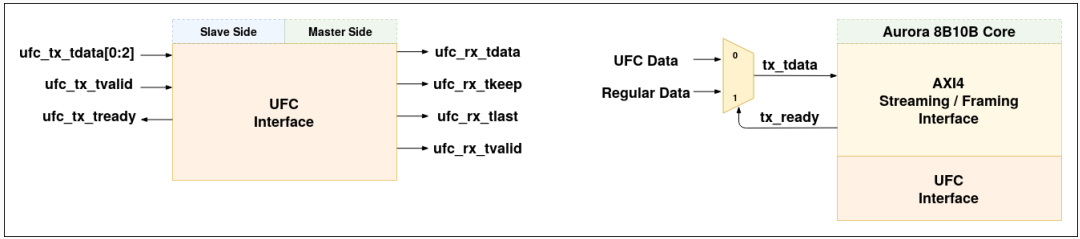
- L'interface *NFC* : Permet de contrôler le débit en insérant des messages IDLE. Le récepteur indique à l'interface Aurora partenaire combien de message IDLE elle doit insérer dans la transmission. Ceci est principalement utile pour éviter des débordement de FIFO.
- L'interface *UFC* : Permet d'insérer des messages prioritaires dans le flux de données standard.

User Flow Control



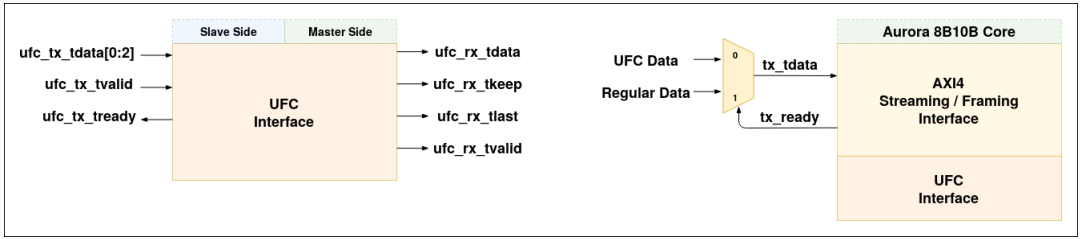
- Le signal *ufc_tx_tdata[0:2]* est utilisé pour spécifier la longeur des messages en mots.

User Flow Control



- Le signal *ufc_tx_tdata[0:2]* est utilisé pour spécifier la longeur des messages en mots.
- Le contenu des messages UFC est transmis par le même signal que les données conventionnelles, soit *tx_tdata*.

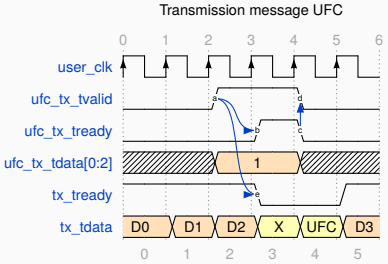
User Flow Control



- Le signal *ufc_tx_tdata[0:2]* est utilisé pour spécifier la longeur des messages en mots.
- Le contenu des messages UFC est transmis par le même signal que les données conventionnelles, soit *tx_tdata*.
- Le signal *tx_tready* est le sélecteur de la source des données.

Pour une interface AXI4 Framing sur 4 [octets]

Message UFC	*ufc_tx_tdata*
2 [octets]	0
4 [octets]	1
6 [octets]	2
8 [octets]	3



Le signal *tx_tready* est le sélecteur de la source de données.

Native Flow Control

- Le contrôle de flux NFC permet au récepteur de demander à l'émetteur de réguler le débit du flux de données.

Native Flow Control

- Le contrôle de flux NFC permet au récepteur de demander à l'émetteur de réguler le débit du flux de données.
- L'émetteur effectue le contrôle du flux de données en insérant des *caractères K IDLE*.

Références [4]

Native Flow Control

- Le contrôle de flux NFC permet au récepteur de demander à l'émetteur de réguler le débit du flux de données.
- L'émetteur effectue le contrôle du flux de données en insérant des *caractères K IDLE*.
- On peut également noter que l'envoie des *caractères K IDLE* permet notamment de conserver le signal d'horloge sur la ligne de transmission.

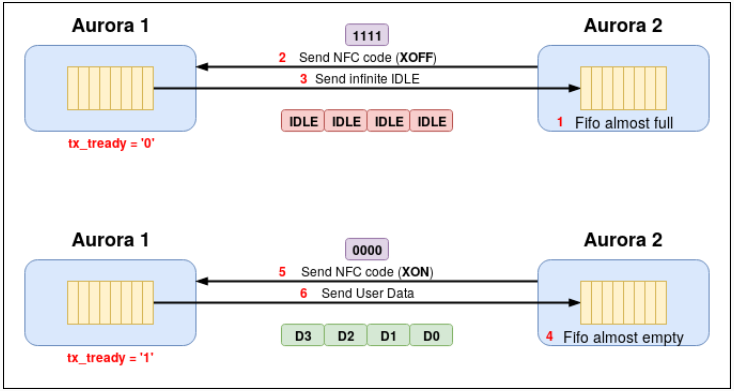
Native Flow Control

- Le contrôle de flux NFC permet au récepteur de demander à l'émetteur de réguler le débit du flux de données.
- L'émetteur effectue le contrôle du flux de données en insérant des *caractères K IDLE*.
- On peut également noter que l'envoie des *caractères K IDLE* permet notamment de conserver le signal d'horloge sur la ligne de transmission.
- Du côté du récepteur, la réception d'un *caractère K IDLE* se traduit par la mise à zéro du signal *tready*.

Native Flow Control

- Le contrôle de flux NFC permet au récepteur de demander à l'émetteur de réguler le débit du flux de données.
- L'émetteur effectue le contrôle du flux de données en insérant des *caractères K IDLE*.
- On peut également noter que l'envoie des *caractères K IDLE* permet notamment de conserver le signal d'horloge sur la ligne de transmission.
- Du côté du récepteur, la réception d'un *caractère K IDLE* se traduit par la mise à zéro du signal *tready*.
- Dans la pratique, on utilise l'interface NFC pour éviter les débordements de FIFO.

Native Flow Control



Mesure variation latence

Voir page 53

Références [4, 3]

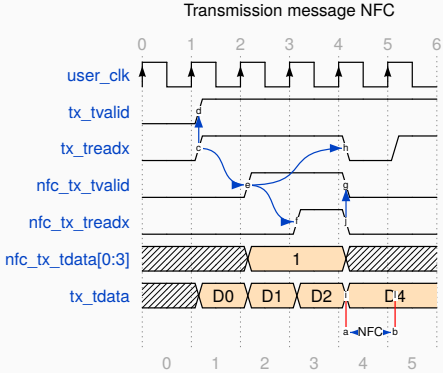
Codes NFC

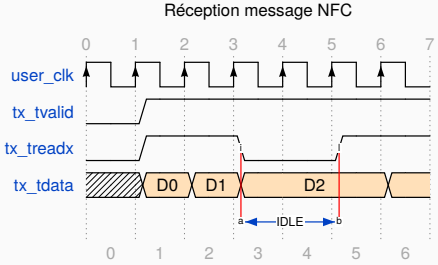
s_axi_nfc_tdata	*Cycles IDLE requis*
0000	0 (XON)
0001	2
0010	4
0011	8
0100	16
0101	32

s_axi_nfc_tdata	*Cycles IDLE requis*
0110	64
0111	128
1000	256
1001 to 1110	Reservé
1111	Infini (XOFF)

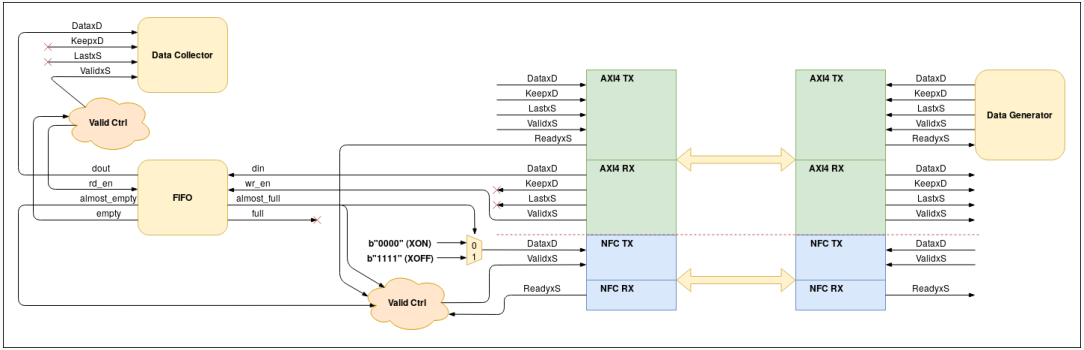
Références [4, 3]

Native Flow Control





Native Flow Control



Références [4]

Horloges et resets

- Vérifier que toutes les horloges (*init_clk* et *drp_clk*) externes au core Aurora sont bien connectées.

Horloges et resets

- Vérifier que toutes les horloges (*init_clk* et *drp_clk*) externes au core Aurora sont bien connectées.
- Attention l'horloge *drp_clk* est utilisée dans la machine d'état du reset côté RX.

Horloges et resets

- Vérifier que toutes les horloges (*init_clk* et *drp_clk*) externes au core Aurora sont bien connectées.
- Attention l'horloge *drp_clk* est utilisée dans la machine d'état du reset côté RX.
- L'horloge *user_clk* est disponible lorsque la partie GTP est initialisée.

Horloges et resets

- Vérifier que toutes les horloges (*init_clk* et *drp_clk*) externes au core Aurora sont bien connectées.
- Attention l'horloge *drp_clk* est utilisée dans la machine d'état du reset côté RX.
- L'horloge *user_clk* est disponible lorsque la partie GTP est initialisée.
- Les signaux resets (*gt_reset* et *reset*) sont asynchrones.

Horloges et resets

- Vérifier que toutes les horloges (*init_clk* et *drp_clk*) externes au core Aurora sont bien connectées.
- Attention l'horloge *drp_clk* est utilisée dans la machine d'état du reset côté RX.
- L'horloge *user_clk* est disponible lorsque la partie GTP est initialisée.
- Les signaux resets (*gt_reset* et *reset*) sont asynchrones.
- Le respect des timings pour les signaux resets (*gt_reset* et *reset*) sont très importants.

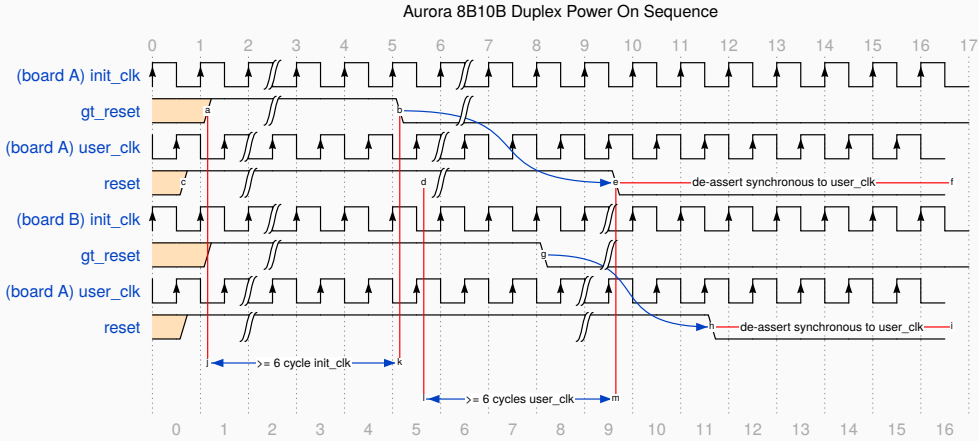
Horloges et resets

- Vérifier que toutes les horloges (*init_clk* et *drp_clk*) externes au core Aurora sont bien connectées.
- Attention l'horloge *drp_clk* est utilisée dans la machine d'état du reset côté RX.
- L'horloge *user_clk* est disponible lorsque la partie GTP est initialisée.
- Les signaux resets (*gt_reset* et *reset*) sont asynchrones.
- Le respect des timings pour les signaux resets (*gt_reset* et *reset*) sont très importants.
- Le signal *gt_reset* doit être géré avec l'horloge *init_clk*.

Horloges et resets

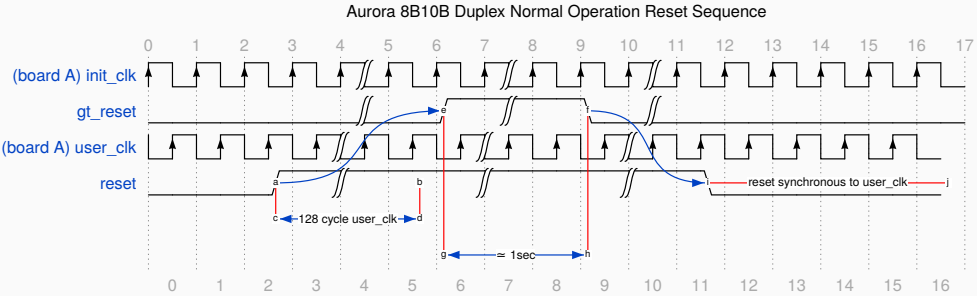
- Vérifier que toutes les horloges (*init_clk* et *drp_clk*) externes au core Aurora sont bien connectées.
- Attention l'horloge *drp_clk* est utilisée dans la machine d'état du reset côté RX.
- L'horloge *user_clk* est disponible lorsque la partie GTP est initialisée.
- Les signaux resets (*gt_reset* et *reset*) sont asynchrones.
- Le respect des timings pour les signaux resets (*gt_reset* et *reset*) sont très importants.
- Le signal *gt_reset* doit être géré avec l'horloge *init_clk*.
- Le signal *reset* doit être géré avec l'horloge *user_clk*.

Horloges et resets

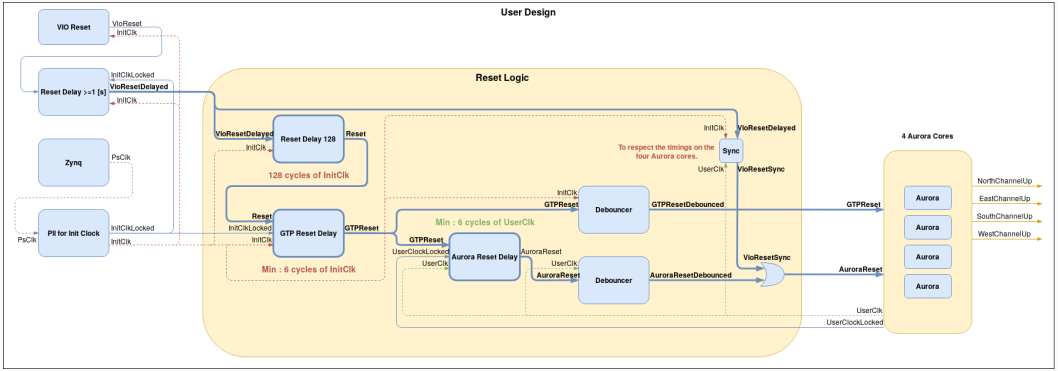


Références [4]

Horloges et resets



Horloges et resets



Références [4]

GTP et Aurora

Priorité de transmission des symboles

Data Type	*Priority*
Clock Compensation Sequences	Highest
Initialization Sequences	...
Native Flow Control PDUs	...
User Flow Control PDUs	...
Channel PDUs	...
Idle Sequences	Lowest

Inversion des priorités

Les priorités de *Channel PDUs* et *Idle Sequences* sont inversées durant le décompte d'un contrôle de flux. Durant cette période, les données utilisateurs (Channel PDUs) reçoivent la priorité la plus basse.

Symboles IDLE

- Les symboles IDLE sont utilisés pour effectuer un alignement des limites des mots et pour effectuer la liaison de canal lors de l'initialisation.

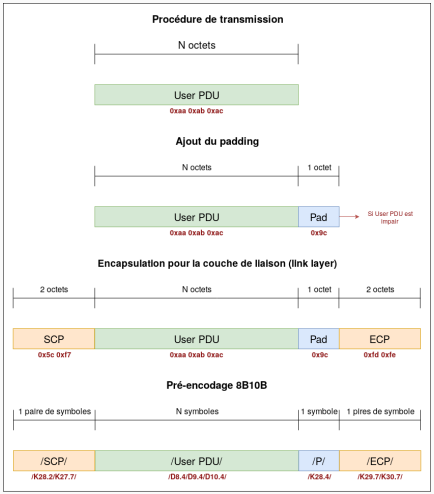
Symboles IDLE

- Les symboles IDLE sont utilisés pour effectuer un alignement des limites des mots et pour effectuer la liaison de canal lors de l'initialisation.
- Lors du fonctionnement, les symboles IDLE sont utilisés pour indiquer qu'il n'y a pas de données.

Symboles IDLE

- Les symboles IDLE sont utilisés pour effectuer un alignement des limites des mots et pour effectuer la liaison de canal lors de l'initialisation.
- Lors du fonctionnement, les symboles IDLE sont utilisés pour indiquer qu'il n'y a pas de données.
- L'insertion des symboles IDLE a lieu pendant les états d'attente et entre les PDUs de canal. Les groupes de symboles IDLE sont appliqués selon une séquence pseudo-aléatoire.

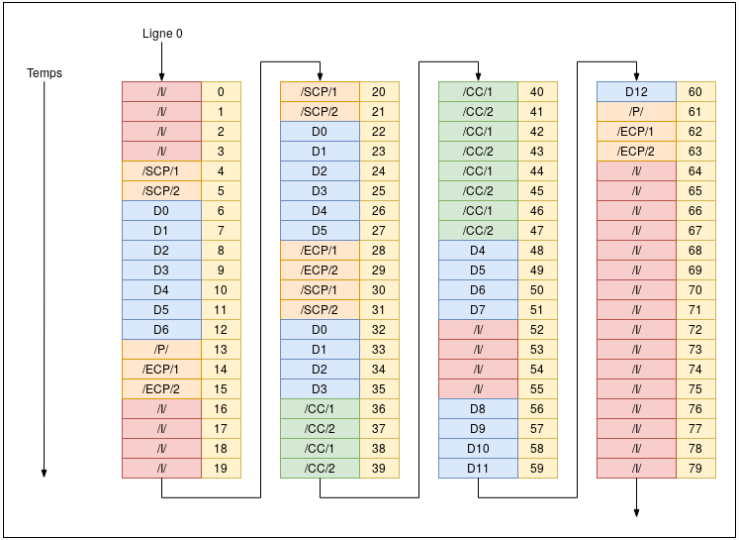
Procédure de transmission



Padding

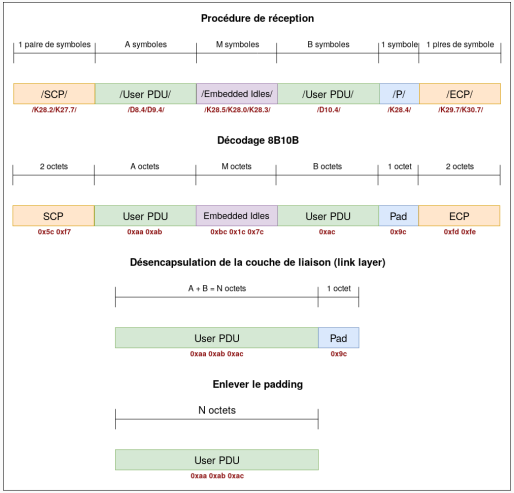
Le *Pad* est nécessaire uniquement si le nombre d'octet du User PDU est impair, car une transmission Aurora se fait toujours par paire de symboles. Ceci est dût au fait qu'un chemin de données dans un transceiver GTP (après désérialisation) est sur 20 bits (16 bits après décodage 8B10B), donc deux octets.

Procédure de transmission



Références [3]

Procédure de réception

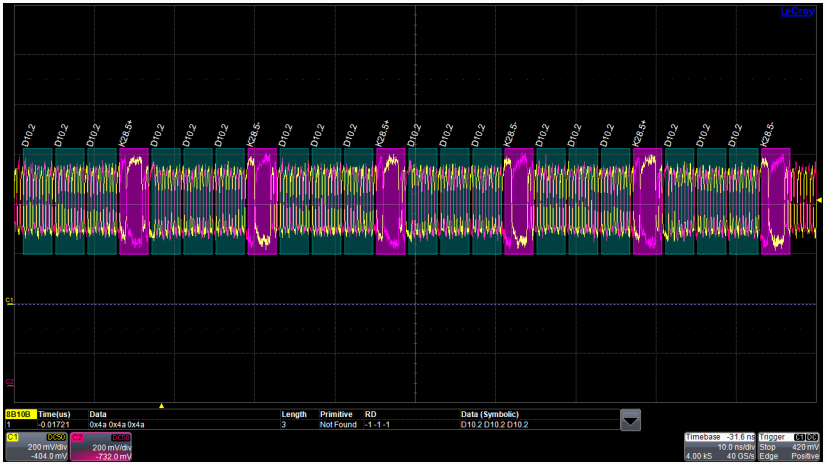


Caractères de contrôle utilisés dans Aurora

Ensemble	*Désignateur*	*Encodage*
Idle	/I/	/K/, /R/, /A/ sequence
Sync and Polarity	/SP/	/K28.5/D10.2/D10.2/D10.2/
Sync and Polarity Ack	/SPA/	/K28.5/D12.1/D12.1/D12.1/
Verification	/V/	/K28.5/D8.7/D8.7/D8.7/
Start of Cahnnel PDU	/SCP/	/K28.2/K27.7/
End of Channel PDU	/ECP/	/K29.7/K30.7/
Pad or Start of UFC PDU	/P/ ou /SUF/	/K28.4/
Comma	/K/	/K28.5/
Skip	/R/	/K28.0/
Channel Bonding	/A/	/K28.3/
Clock Compensation	/CC/	/K23.7/K23.7/
Start of NFC PDU	/SNF/	/K28.6/

Références [3]

Caractères de contrôle utilisés dans Aurora



Agrégation de liens

- L'agrégation de liens (channel bonding) permet d'obtenir un canal Aurora en utilisant plusieurs lignes GTP.

Schéma bloc

Voir page 7

Agrégation de liens

- L'agrégation de liens (channel bonding) permet d'obtenir un canal Aurora en utilisant plusieurs lignes GTP.
- Le principale intérêt est l'augmentation du débit binaire du canal.

Schéma bloc

Voir page 7

Agrégation de liens

- L'agrégation de liens (channel bonding) permet d'obtenir un canal Aurora en utilisant plusieurs lignes GTP.
- Le principale intérêt est l'augmentation du débit binaire du canal.
- Le principale inconvénient est la nécécité d'avoir plus de câbles coaxiaux ou fibres optiques.

Schéma bloc

Voir page 7

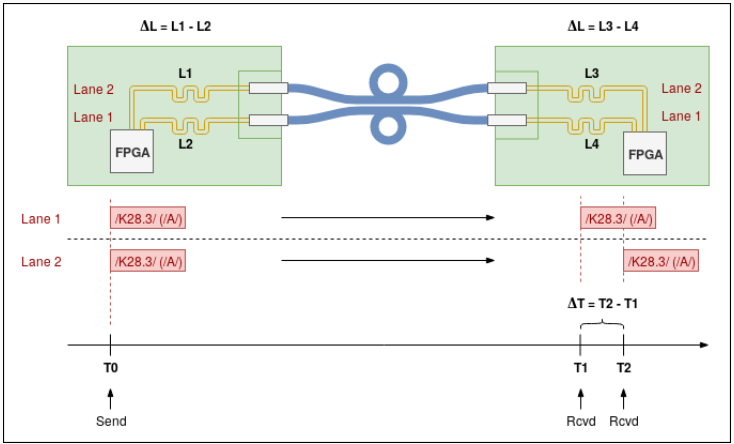
Agrégation de liens

- L'agrégation de liens (channel bonding) permet d'obtenir un canal Aurora en utilisant plusieurs lignes GTP.
- Le principale intérêt est l'augmentation du débit binaire du canal.
- Le principale inconvénient est la nécécité d'avoir plus de câbles coaxiaux ou fibres optiques.
- Il est également nécessaire d'effectuer une procédure d'initialisation spécifique pour compenser les différences de latence entre chaques liens du canal.

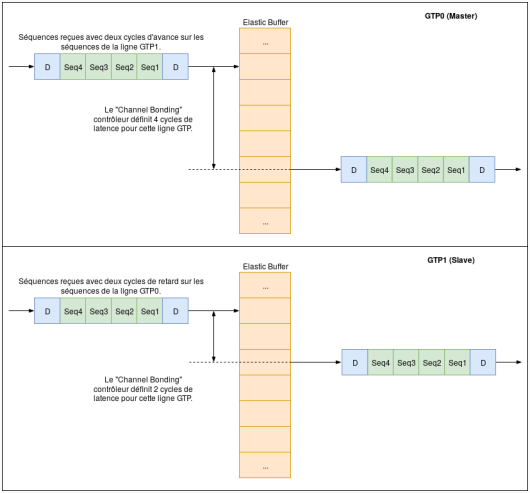
Schéma bloc

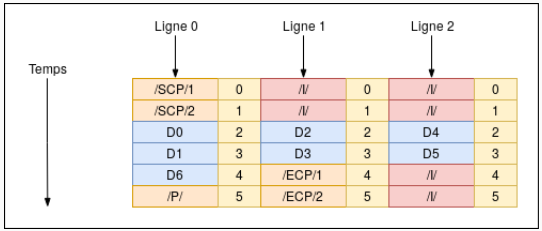
Voir page 7

Agrégation de liens - Différence de latence



Agrégation de liens - Correction de la latence



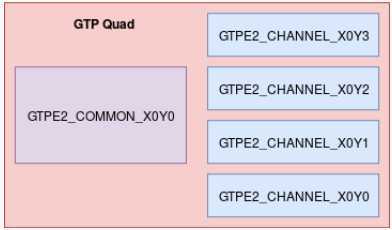


Mode d'envoi d'une transmission multi-lignes

Les données et les caractères de contrôle sont toujours envoyés par paire et ordonnés sur les lignes de manière incrémentale. Les caractères de contrôle peuvent être envoyés sur n'importe quelle ligne.

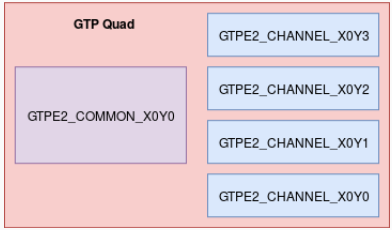
Organisation des transceivers GTP

- Les transceivers GTP sont organisés en *GTP Quad*.



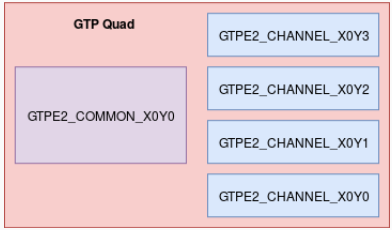
Organisation des transceivers GTP

- Les transceivers GTP sont organisés en *GTP Quad*.
- Un à quatre *GTP Quad* pour la gamme Artix 7.



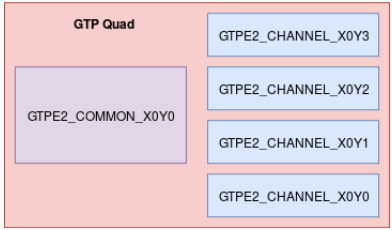
Organisation des transceivers GTP

- Les transceivers GTP sont organisés en *GTP Quad*.
- Un à quatre *GTP Quad* pour la gamme Artix 7.
- Quatres transceivers et deux PLLs (communes aux quatres transceivers) par *GTP Quad*.

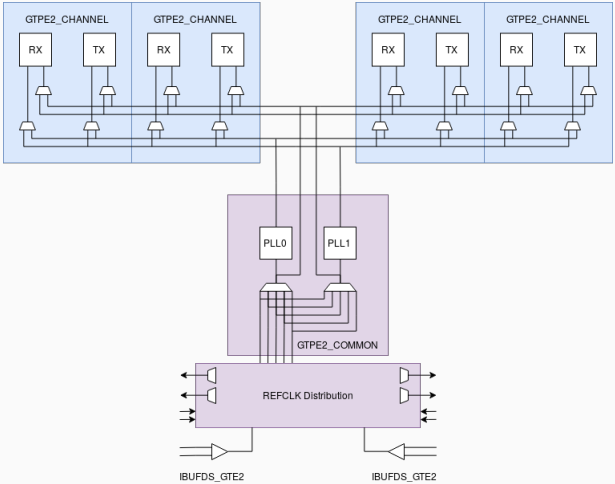


Organisation des transceivers GTP

- Les transceivers GTP sont organisés en *GTP Quad*.
- Un à quatre *GTP Quad* pour la gamme Artix 7.
- Quatres transceivers et deux PLLs (communes aux quatres transceivers) par *GTP Quad*.
- Deux sources d'horloge externes et d'autres sources internes.



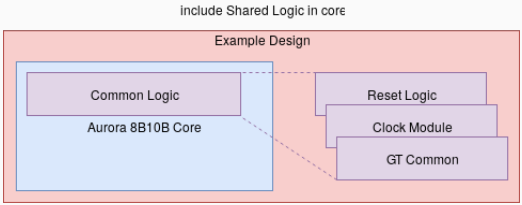
Organisation des transceivers GTP

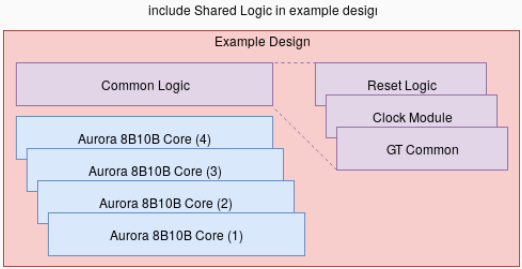


Références [2]

Il y a deux modes d'organisation pour Aurora

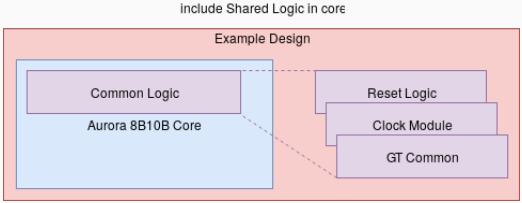
- La *partie commune* est instanciée à *l'intérieur du core Aurora.*

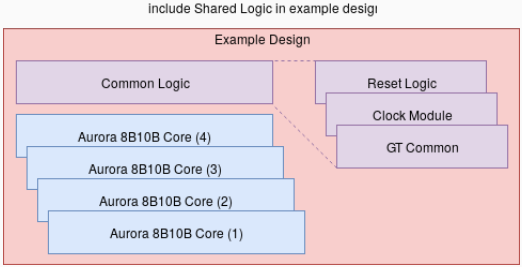




Il y a deux modes d'organisation pour Aurora

- La *partie commune* est instanciée à *l'intérieur du core Aurora.*
- La *partie commune* est instanciée à *l'extérieur du core Aurora.*





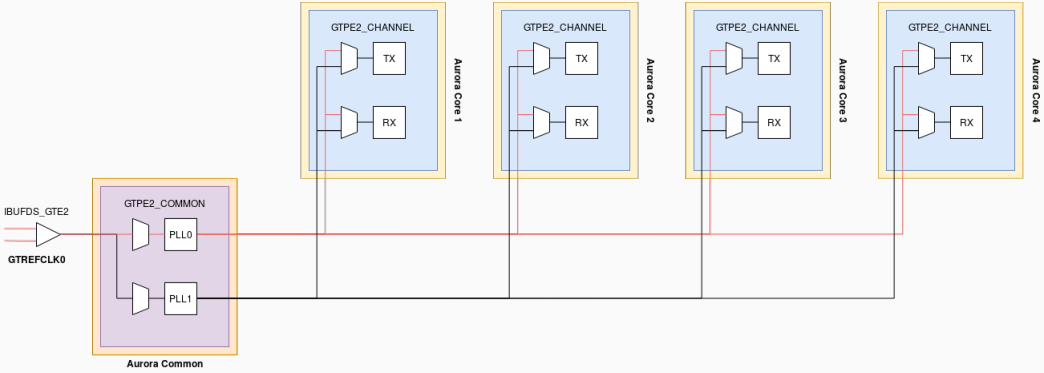
Organisation Aurora

- Dans le cas d'une application où l'on souhaite avoir 4 cores Aurora (1 ligne par core) indépendants, il faut impérativement instancier la partie commune (partagée par les cores) à l'extérieur.

Organisation Aurora

- Dans le cas d'une application où l'on souhaite avoir 4 cores Aurora (1 ligne par core) indépendants, il faut impérativement instancier la partie commune (partagée par les cores) à l'extérieur.
- Dans le cas où l'on instancie la partie commune à l'intérieur, un seul core peut être utilisé (éventuellement avec plusieurs lignes).

Sur la figure suivante, il y a quatre cores instanciés (1 transceivers par core) indépendants.



Dérive entre les horloges

Il est important de noter qu'un core Aurora utilise une horloge utilisateur issue d'un oscillateur interne à la carte. Quant aux données reçues, celles-ci utilisent l'horloge récupérée à partir de la transmission série (CDR). Par conséquence, on se retrouve avec deux sources d'horloge différentes et donc une dérive d'horloge à compenser.

Les conséquence sont les suivantes

- L'utilisation d'un buffer élastique est nécessaire.

Dérive entre les horloges

Il est important de noter qu'un core Aurora utilise une horloge utilisateur issue d'un oscillateur interne à la carte. Quant aux données reçues, celles-ci utilisent l'horloge récupérée à partir de la transmission série (CDR). Par conséquence, on se retrouve avec deux sources d'horloge différentes et donc une dérive d'horloge à compenser.

Les conséquence sont les suivantes

- L'utilisation d'un buffer élastique est nécessaire.
- Le lien série est non-déterministe.

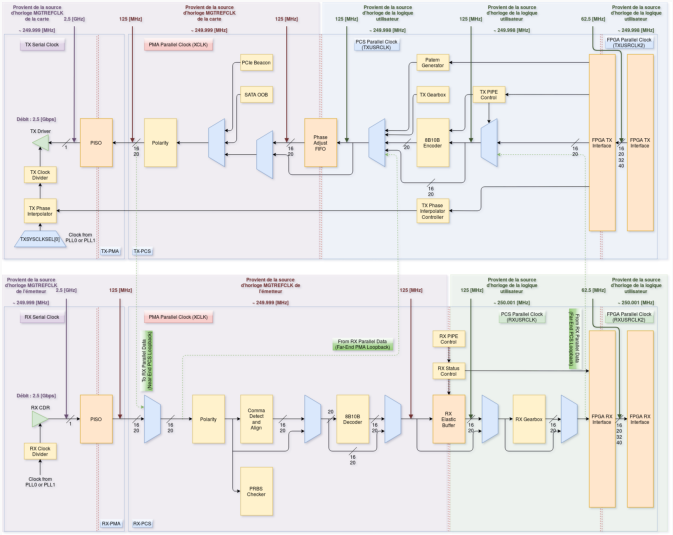
Dérive entre les horloges

Il est important de noter qu'un core Aurora utilise une horloge utilisateur issue d'un oscillateur interne à la carte. Quant aux données reçues, celles-ci utilisent l'horloge récupérée à partir de la transmission série (CDR). Par conséquence, on se retrouve avec deux sources d'horloge différentes et donc une dérive d'horloge à compenser.

Les conséquence sont les suivantes

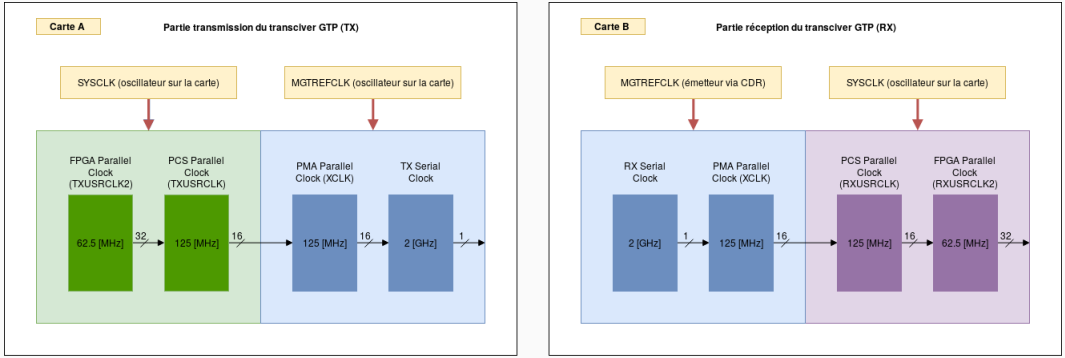
- L'utilisation d'un buffer élastique est nécessaire.
- Le lien série est non-déterministe.
- Ce type de communication fonctionne très bien pour la plus part des applications standards, mais ne doit pas être utilisé pour des applications qui requièrent des transmissions en temps réel (déterministe).

Latence variable dans Aurora - Buffer élastique



Références [2]

Latence variable dans Aurora - Domaines d'horloges

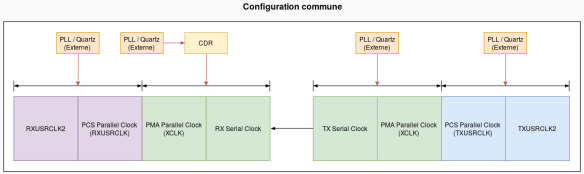


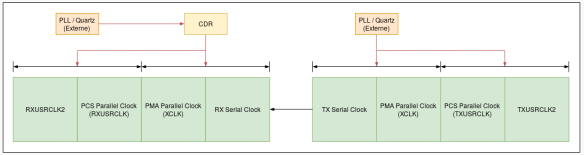
Les horloges MGTREFCLK utilisées du côté TX pour la carte A et du côté RX pour la carte B proviennent du même oscillateur MGTREFCLK de la carte A. Par conséquent, les zones violettes représentent un seul et même domaine d'horloge. De plus, les domaines d'horloge TX Serial Clock, PMA Parallel Clock (XCLK carte A), RX Serial Clock et PMA Parallel Clock (XCLK carte B) proviennent également de cette même source d'horloge.

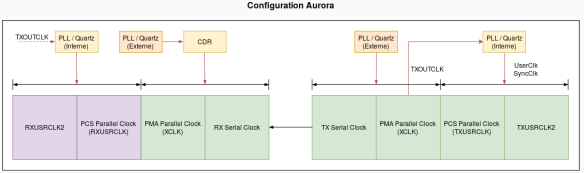
Les horloges SYSCLK de la carte A et SYSCLK de la carte B sont deux oscillateurs différents qui se trouvent chacun sur une carte. Par conséquent, les zones vertes représentent deux domaines d'horloge différents, dont l'un se trouve sur la carte A et l'autre sur la carte B. Les domaines d'horloges PCS Parallel Clock (TXUSRCLK) et FPGA Parallel Clock (TXUSRCLK2) proviennent de l'oscillateur SYSCLK de la carte A. Quant aux domaines d'horloge PCS Parallel Clock (RXUSRCLK) et FPGA Parallel Clock (RXUSRCLK2), ils proviennent de l'oscillateur SYSCLK de la carte B.

Références [2]

Latence variable dans Aurora - Domaines d'horloges







Types de latence

Voir page 47

Références [2]

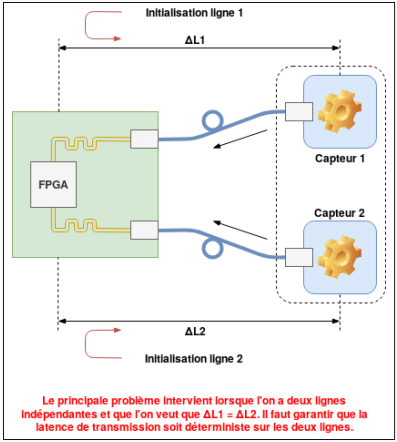
- Le protocole Aurora fourni un mécanisme capable de compenser une dérive d'horloge pouvant atteindre ± 100 [ppm].

- Le protocole Aurora fourni un mécanisme capable de compenser une dérive d'horloge pouvant atteindre ± 100 [ppm].
- Le symbole CC sont insérés périodiquement pendant les états d'attente et entre les PDUs de canal.

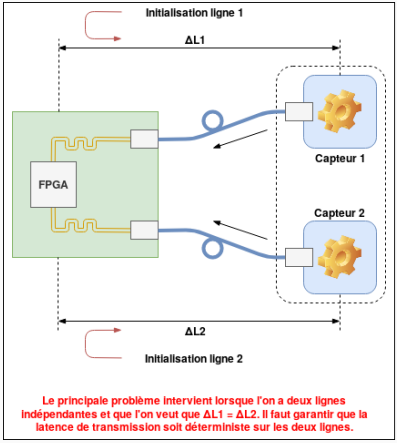
- Le protocole Aurora fourni un mécanisme capable de compenser une dérive d'horloge pouvant atteindre ± 100 [ppm].
- Le symbole CC sont insérés périodiquement pendant les états d'attente et entre les PDUs de canal.
- La séquence de compensation d'horloge consiste en six paires de symboles CC.

- Le protocole Aurora fourni un mécanisme capable de compenser une dérive d'horloge pouvant atteindre ± 100 [ppm].
- Le symbole CC sont insérés périodiquement pendant les états d'attente et entre les PDUs de canal.
- La séquence de compensation d'horloge consiste en six paires de symboles CC.
- La séquence de compensation d'horloge est transmise tous les 10000 [bytes] par ligne (5000 cycles pour une ligne sur 2 [bytes] et 2500 cycles pour une ligne sur 4 [bytes]).

Dans quels cas une latence fixe est requise

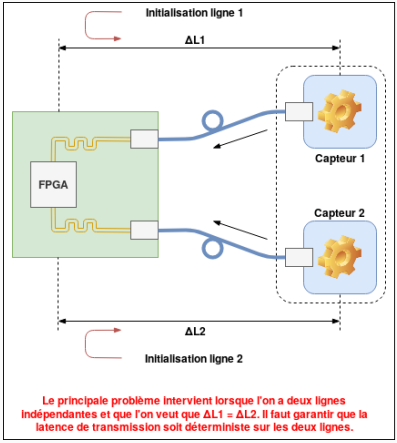


Dans quels cas une latence fixe est requise



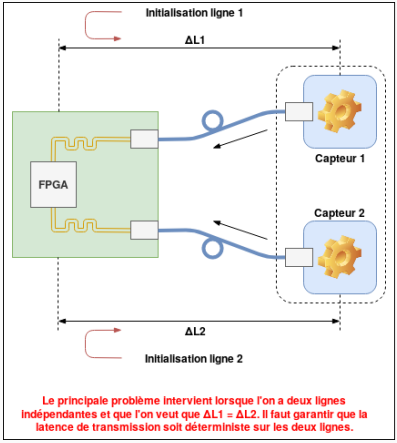
- On peut mesurer un timestamp au niveau de l'émetteur.

Dans quels cas une latence fixe est requise



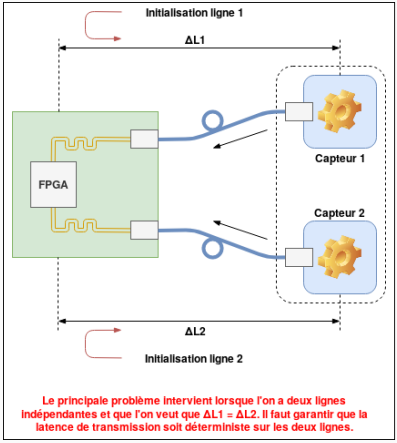
- On peut mesurer un timestamp au niveau de l'émetteur.
- Il faut garantir que les capteurs soit synchrones.

Dans quels cas une latence fixe est requise



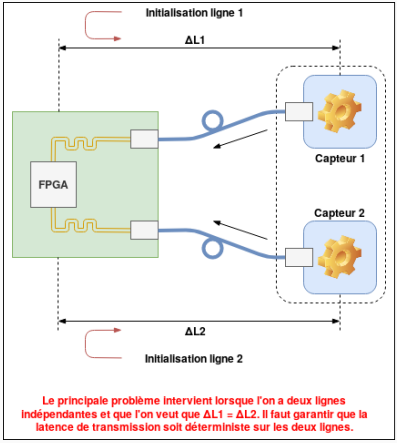
- On peut mesurer un timestamp au niveau de l'émetteur.
- Il faut garantir que les capteurs soit synchrones.
- Si on mesure un timestamp au niveau du récepteur.

Dans quels cas une latence fixe est requise



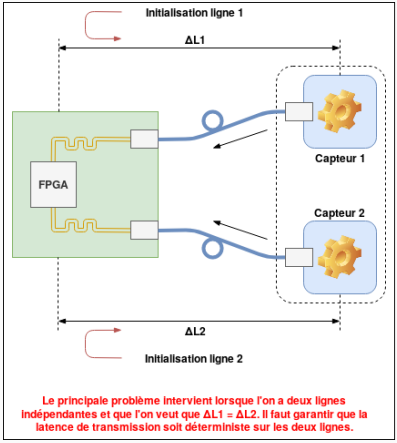
- On peut mesurer un timestamp au niveau de l'émetteur.
- Il faut garantir que les capteurs soit synchrones.
- Si on mesure un timestamp au niveau du récepteur.
- Il faut garantir que la latence des transceivers soit identique, ou la compenser.

Dans quels cas une latence fixe est requise



- On peut mesurer un timestamp au niveau de l'émetteur.
- Il faut garantir que les capteurs soit synchrones.
- Si on mesure un timestamp au niveau du récepteur.
- Il faut garantir que la latence des transceivers soit identique, ou la compenser.
- Il faut connaitre la latence des capteurs aux transceivers.

Dans quels cas une latence fixe est requise



- On peut mesurer un timestamp au niveau de l'émetteur.
- Il faut garantir que les capteurs soit synchrones.
- Si on mesure un timestamp au niveau du récepteur.
- Il faut garantir que la latence des transceivers soit identique, ou la compenser.
- Il faut connaitre la latence des capteurs aux transceivers.
- Si il y a une différence, compenser celle-ci en ajoutant un décalage aux timestamps.

Initialisation 1

Latence variable	*Latence fixe (1)*	*Latence fixe (2)*
25	25	25
26	25	25
23	25	25

Initialisation 2

Latence variable	*Latence fixe (1)*	*Latence fixe (2)*
35	35	25
33	35	25
37	35	25

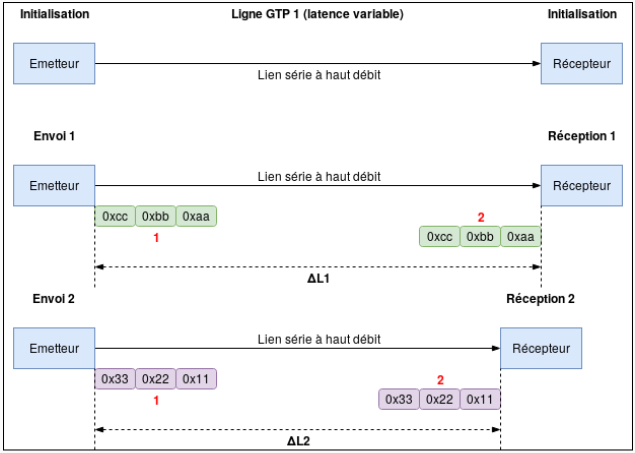
Conditions

Latence variable	*Latence fixe (1)*	*Latence fixe (2)*
-	Temps traitement fixe	Temps de traitement fixe
-	Fréquence d'horloges identique (source)	Fréquence d'horloges identique (source)
-	-	Phases d'horloges identique

Latence fixe

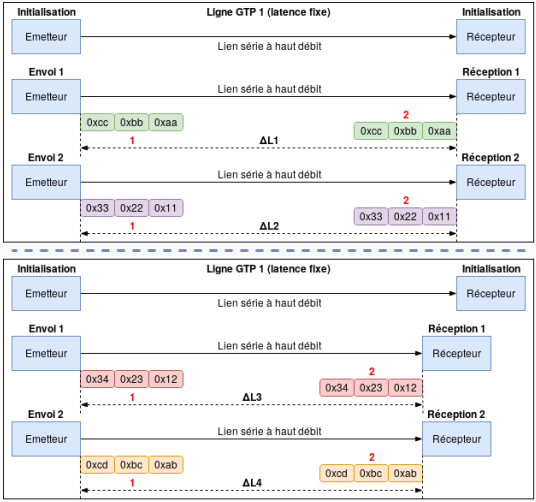
Voir page 43

Types de latence - Latence variable



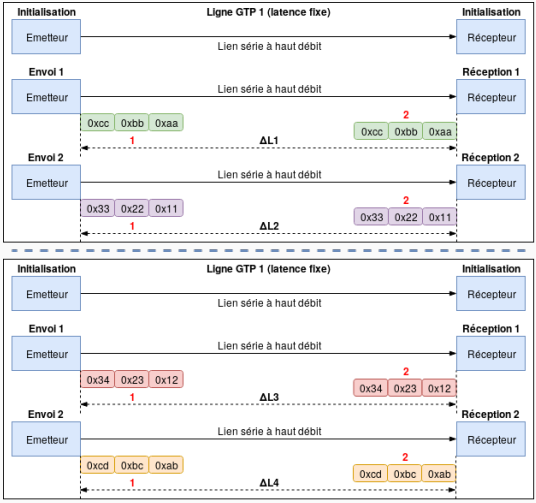
ΔL1 ≠ ΔL2 : Latence variable au cours du temps

Types de latence - Latence fixe (1)



ΔL1 = ΔL2 et ΔL3 = ΔL4 : Latence fixe au cours du temps mais variable d'une initialisation à l'autre

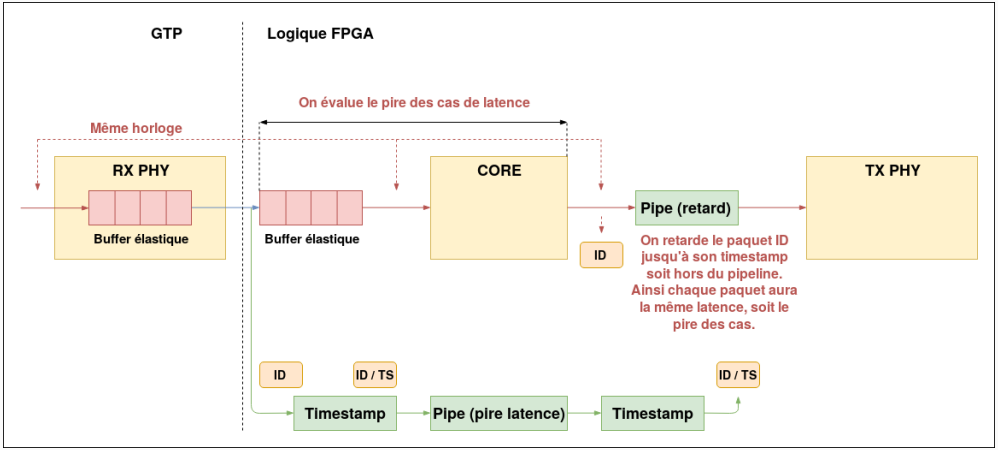
Types de latence - Latence fixe (2)



ΔL1 = ΔL2 et ΔL3 = ΔL4 : Latence fixe au cours du temps et invariable d'une initialisation à l'autre

Références [2]

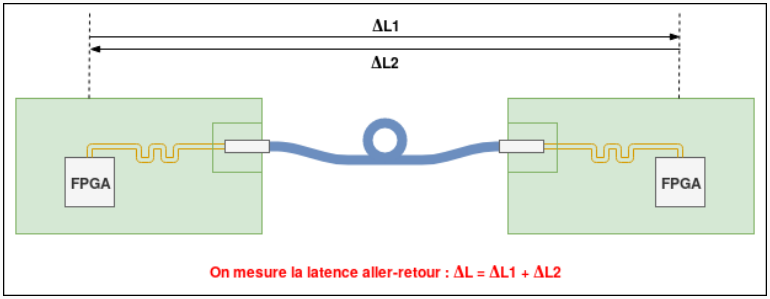
Solution apportée pour un mandataire - Correction latence variable



Optimisation des ressources matérielles

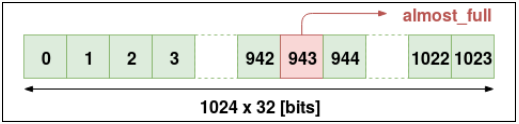
Lors d'un développement et pour optimiser l'utilisation des ressources matérielles, il est souvent utile de dimensionner correctement les FIFOs. Par conséquent, on doit être en mesure de déterminer le pire des cas de la variation de latence.

Le montage suivant est une mise en oeuvre pour effectuer ce type de mesure



Références [4, 2]

- Je mesure comme pire latence 71 cycles d'horloge.



La fin de ma FIFO me permet de réceptionner les mots envoyés par l'émetteur, le temps que ce dernier aie reçu le message NFC *(XOFF)* et qu'il puisse bloquer la transmission. Cette marge correspond à un aller-retour en terme de cycles d'horloge.

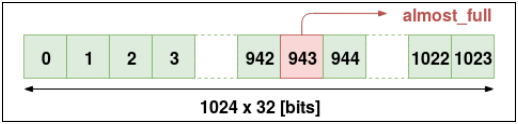
Native Flow Control

Voir page 15

Références [4, 2]

Mesure de la variation de latence

- Je mesure comme pire latence 71 cycles d'horloge.
- Je peux arrondir à 80 cycles d'horloge, pour avoir une petite marge.



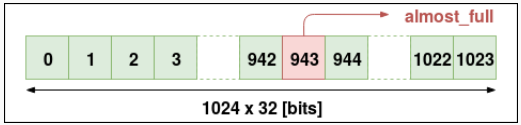
La fin de ma FIFO me permet de réceptionner les mots envoyés par l'émetteur, le temps que ce dernier aie reçu le message NFC *(XOFF)* et qu'il puisse bloquer la transmission. Cette marge correspond à un aller-retour en terme de cycles d'horloge.

Native Flow Control

Voir page 15

Mesure de la variation de latence

- Je mesure comme pire latence 71 cycles d'horloge.
- Je peux arrondir à 80 cycles d'horloge, pour avoir une petite marge.
- Je place le seuil *"almost full"* de ma FIFO à $1023 - 80 = 943$



La fin de ma FIFO me permet de réceptionner les mots envoyés par l'émetteur, le temps que ce dernier aie reçu le message NFC *(XOFF)* et qu'il puisse bloquer la transmission. Cette marge correspond à un aller-retour en terme de cycles d'horloge.

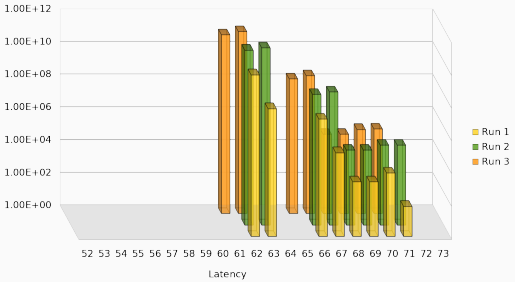
Native Flow Control

Voir page 15

Références [4, 2]

Mesure de la variation de latence

L'histogramme (échelle logarithmique) suivant présente trois exécutions différentes.

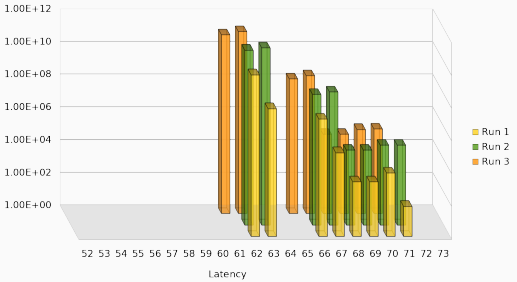


- La latence moyenne dépend des conditions d'initialisation (reset asynchrone, phase d'init. de la PLL, état init. du buffer élastique), mais reste identique au cours du temps.

Références [4, 2]

Mesure de la variation de latence

L'histogramme (échelle logarithmique) suivant présente trois exécutions différentes.

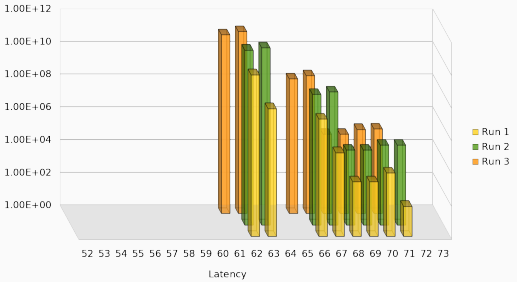


- La latence moyenne dépend des conditions d'initialisation (reset asynchrone, phase d'init. de la PLL, état init. du buffer élastique), mais reste identique au cours du temps.
- l'étalement est dût aux différents caractères (K), soit caractères de compensation d'horloge (CC) et caractères IDLE envoyés selon une séquence pseudo-aléatoire.

Références [4, 2]

Mesure de la variation de latence

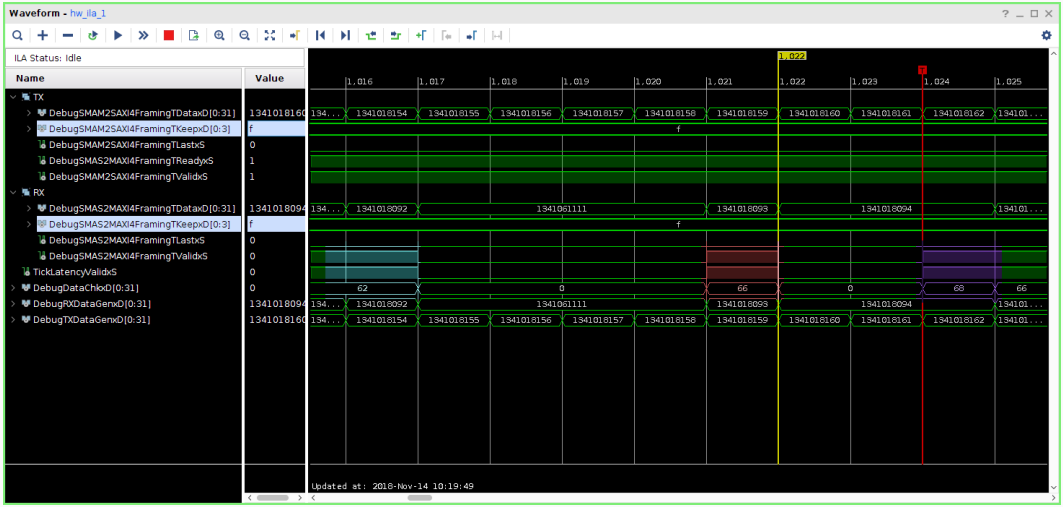
L'histogramme (échelle logarithmique) suivant présente trois exécutions différentes.



- La latence moyenne dépend des conditions d'initialisation (reset asynchrone, phase d'init. de la PLL, état init. du buffer élastique), mais reste identique au cours du temps.
- l'étalement est dût aux différents caractères (K), soit caractères de compensation d'horloge (CC) et caractères IDLE envoyés selon une séquence pseudo-aléatoire.
- La latence est non-déterministe.

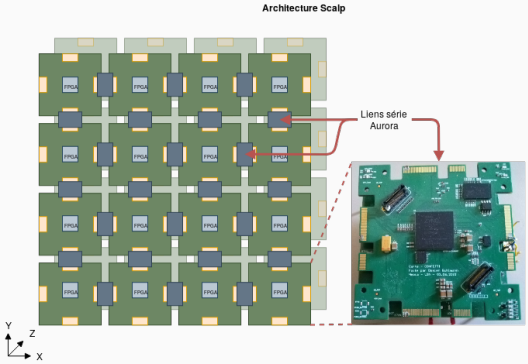
Références [4, 2]

Mesure de la variation de latence



Références [4, 2]

Aurora dans le projet SOMA / Scalp



Scalp NoC	
Aurora	A définir
GTP	LVDS

Protocoles

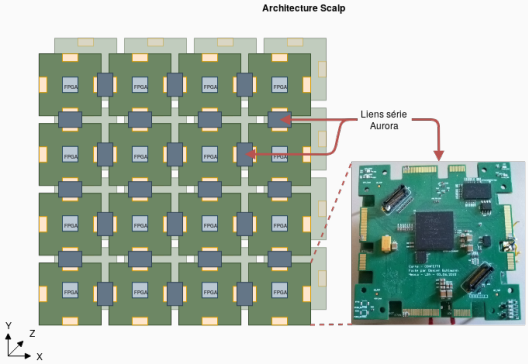
3	Réseau
2	Liaison
1	Physique

Modèle OSI

- SOMA est un projet qui vise à développer une architecture cellulaire auto-organisée et distribuée.

Références [1]

Aurora dans le projet SOMA / Scalp



Scalp NoC	
Aurora	A définir
GTP	LVDS

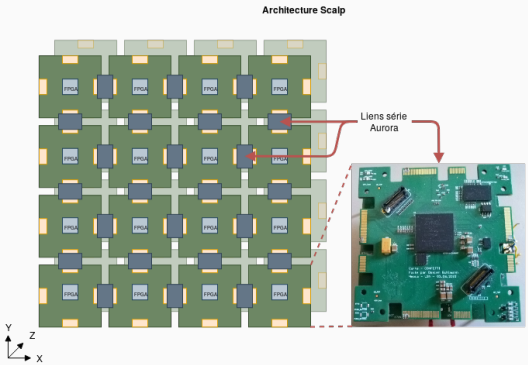
Protocoles

3	Réseau
2	Liaison
1	Physique

Modèle OSI

- SOMA est un projet qui vise à développer une architecture cellulaire auto-organisée et distribuée.
- Scalp est la plateforme matérielle composée d'un grand nombre de cartes équipées d'un FPGA Zynq 7015.

Références [1]



Scalp NoC	
Aurora	A définir
GTP	LVDS

Protocoles

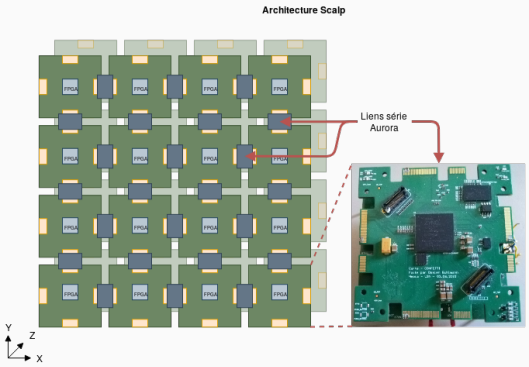
3	Réseau
2	Liaison
1	Physique

Modèle OSI

- SOMA est un projet qui vise à développer une architecture cellulaire auto-organisée et distribuée.
- Scalp est la plateforme matérielle composée d'un grand nombre de cartes équipées d'un FPGA Zynq 7015.
- La communication entre les cartes s'effectue au travers d'un NoC (Network on Chip).

Références [1]

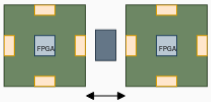
Aurora dans le projet SOMA / Scalp



- SOMA est un projet qui vise à développer une architecture cellulaire auto-organisée et distribuée.
- Scalp est la plateforme matérielle composée d'un grand nombre de cartes équipées d'un FPGA Zynq 7015.
- La communication entre les cartes s'effectue au travers d'un NoC (Network on Chip).
- Le NoC (couche de routage) repose sur des liaisons point à point qui utilisent dans un premier temps le protocole Aurora.

Références [1]

Aurora dans le projet SOMA / Scalp



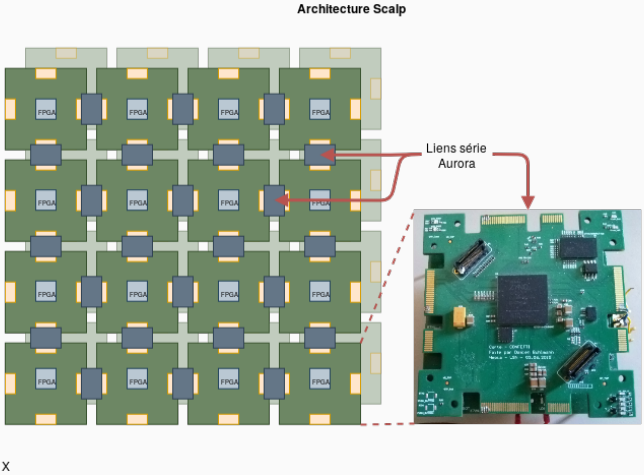
On connecte les cartes FPGA pour former un réseau suivant une topologie en forme de grille.

Scalp NoC	
Aurora	A définir
GTP	LVDS

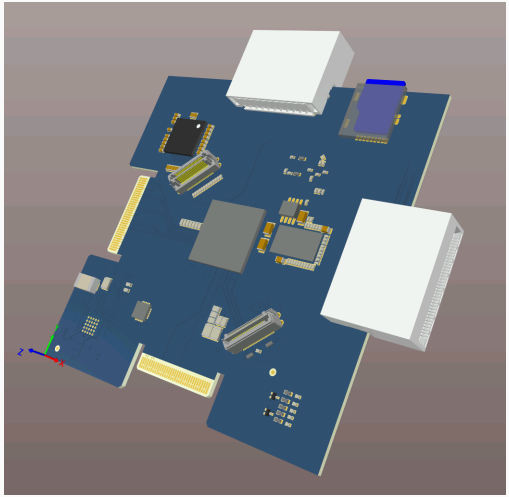
Protocoles

3	Réseau
2	Liaison
1	Physique

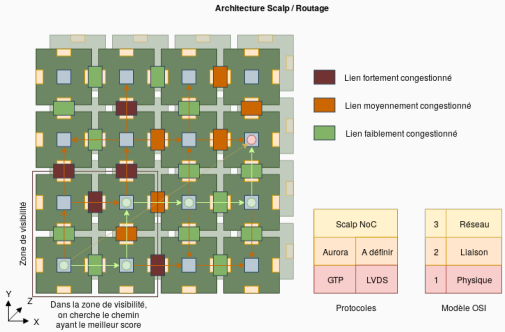
Modèle OSI



Aurora dans le projet SOMA / Scalp

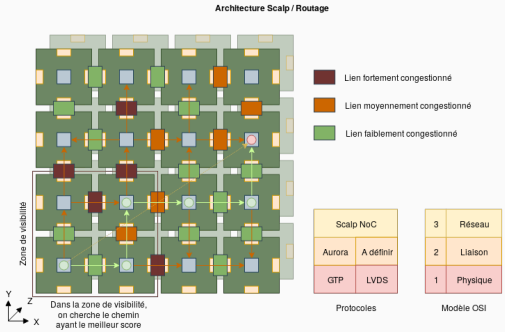


Aurora dans le projet SOMA / Scalp



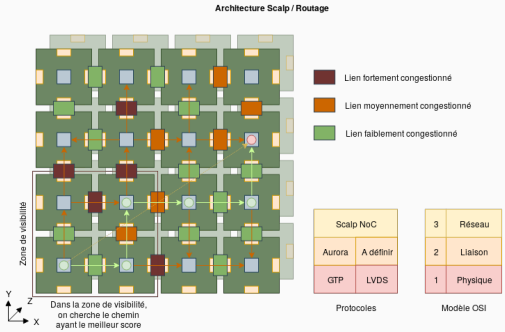
- La couche réseau Scalp NoC utilise Aurora comme protocole pour les liaisons 2D (liens GTP).

Aurora dans le projet SOMA / Scalp



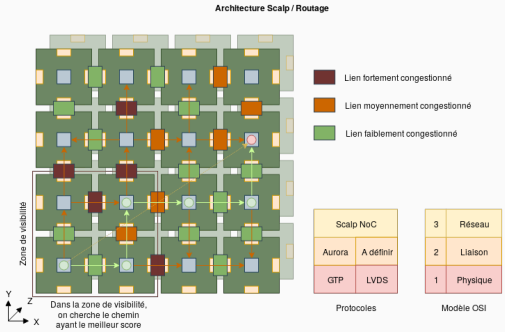
- La couche réseau Scalp NoC utilise Aurora comme protocole pour les liaisons 2D (liens GTP).
- La couche réseau Scalp NoC utilisera un autre protocole pour les liaisons 3D (liens LVDS).

Aurora dans le projet SOMA / Scalp



- La couche réseau Scalp NoC utilise Aurora comme protocole pour les liaisons 2D (liens GTP).
- La couche réseau Scalp NoC utilisera un autre protocole pour les liaisons 3D (liens LVDS).
- Le contrôle de flux NFC est utilisé pour bloquer le flux en cascade et éviter des dépassements de FIFO (pertes de données).

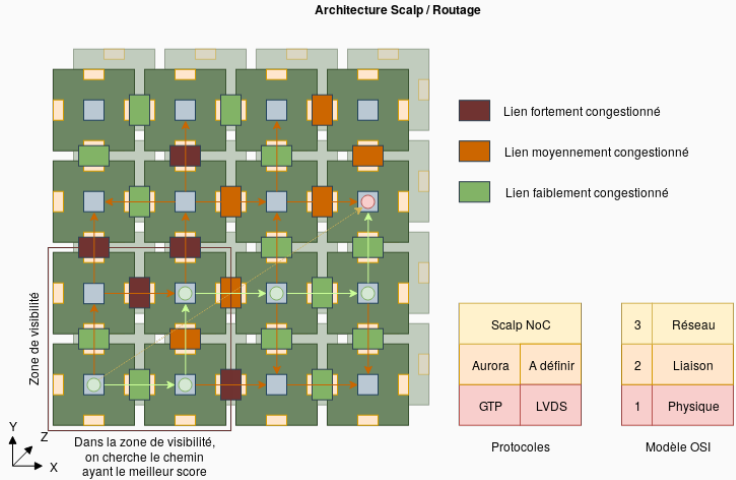
Aurora dans le projet SOMA / Scalp



- La couche réseau Scalp NoC utilise Aurora comme protocole pour les liaisons 2D (liens GTP).
- La couche réseau Scalp NoC utilisera un autre protocole pour les liaisons 3D (liens LVDS).
- Le contrôle de flux NFC est utilisé pour bloquer le flux en cascade et éviter des dépassements de FIFO (pertes de données).
- Le contrôle de flux UFC est utilisé pour des communications prioritaires entre cartes voisines.

Références [1]

Aurora dans le projet SOMA / Scalp



Customiser une IP Aurora

Pourquoi customiser une IP Aurora ?

- Avec l'IP Xilinx générée par le wizard, une IP pour chaque GTP.

Customiser une IP Aurora

Pourquoi customiser une IP Aurora ?

- Avec l'IP Xilinx générée par le wizard, une IP pour chaque GTP.
- Avoir une IP unique et configurer génériquement quel GTP doit être utilisé.

Customiser une IP Aurora

Pourquoi customiser une IP Aurora ?

- Avec l'IP Xilinx générée par le wizard, une IP pour chaque GTP.
- Avoir une IP unique et configurer génériquement quel GTP doit être utilisé.
- Lorsque l'on a une carte de développement et une carte de production qui on des horloges de références différentes. Configuration générique sans avoir besoin de regénérer les IP.

Customiser une IP Aurora

Pourquoi customiser une IP Aurora ?

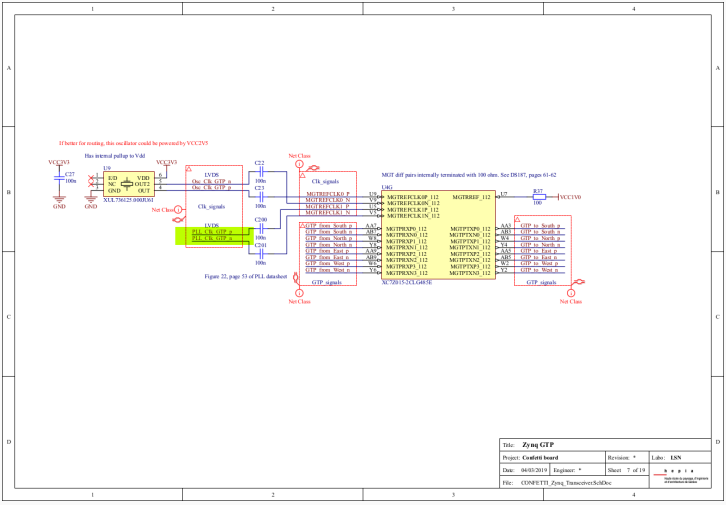
- Avec l'IP Xilinx générée par le wizard, une IP pour chaque GTP.
- Avoir une IP unique et configurer génériquement quel GTP doit être utilisé.
- Lorsque l'on a une carte de développement et une carte de production qui on des horloges de références différentes. Configuration générique sans avoir besoin de regénérer les IP.
- Lorsque l'on veut utiliser des composants spécifiques des GTP et qui ne sont pas proposés par Xilinx. (Par ex. : PRBS).

Customiser une IP Aurora

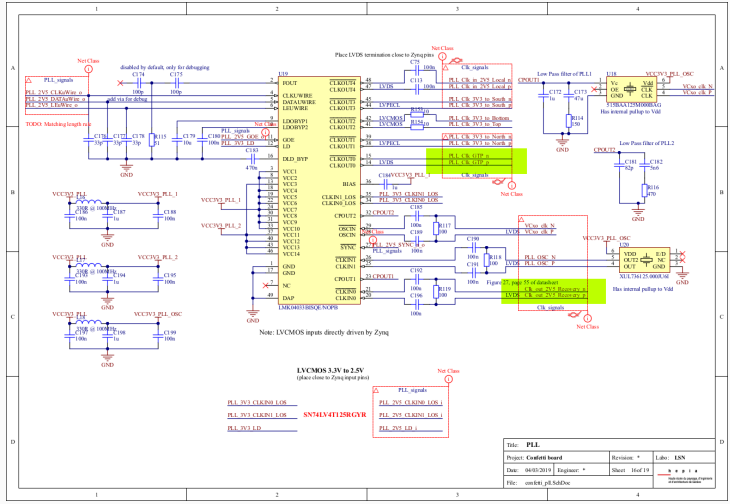
Pourquoi customiser une IP Aurora ?

- Avec l'IP Xilinx générée par le wizard, une IP pour chaque GTP.
- Avoir une IP unique et configurer génériquement quel GTP doit être utilisé.
- Lorsque l'on a une carte de développement et une carte de production qui on des horloges de références différentes. Configuration générique sans avoir besoin de regénérer les IP.
- Lorsque l'on veut utiliser des composants spécifiques des GTP et qui ne sont pas proposés par Xilinx. (Par ex. : PRBS).
- Lorsque l'on veut mettre en oeuvre des communications synchrones avec Aurora (recovery clock pour la partie utilisateur et bypass du buffer élastique). Par défaut, les communications Aurora sont asynchrones.

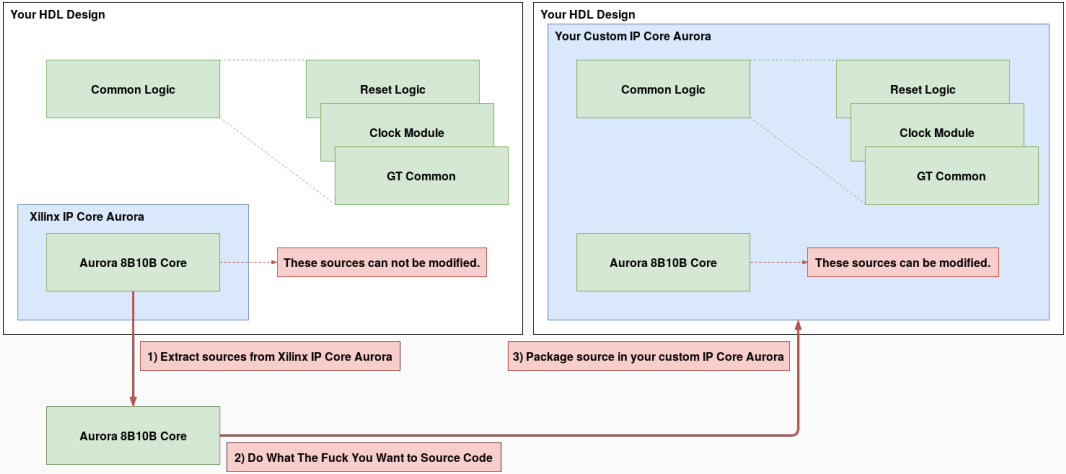
Customiser une IP Aurora



Customiser une IP Aurora



Customiser une IP Aurora



Références [4]

Questions?

Mise en oeuvre d’Aurora

Configuration de l'IP Core Aurora

Configuration de la couche physique

- Choisir la largeur d'une ligne (16 ou 32 [bits]).

Configuration de l'IP Core Aurora

Configuration de la couche physique

- Choisir la largeur d'une ligne (16 ou 32 [bits]).
- Choisir le débit d'une ligne (par ex. 2.5 [Gbps]).

Configuration de l'IP Core Aurora

Configuration de la couche physique

- Choisir la largeur d'une ligne (16 ou 32 [bits]).
- Choisir le débit d'une ligne (par ex. 2.5 [Gbps]).
- Choisir la fréquence de l'horloge GT Refclk (voir la documentation de la carte).

Références [4]

Configuration de l'IP Core Aurora

Configuration de la couche physique

- Choisir la largeur d'une ligne (16 ou 32 [bits]).
- Choisir le débit d'une ligne (par ex. 2.5 [Gbps]).
- Choisir la fréquence de l'horloge GT Refclk (voir la documentation de la carte).
- Choisir la fréquence de l'horloge d'initialisation. Celle-ci est fournie par l'utilisateur et devrait être un multiple l'horloge GT Refclk pour éviter les problèmes de timing lors de changement de domaine d'horloge.

Références [4]

Configuration de l'IP Core Aurora

Configuration de la couche physique

- Choisir la largeur d'une ligne (16 ou 32 [bits]).
- Choisir le débit d'une ligne (par ex. 2.5 [Gbps]).
- Choisir la fréquence de l'horloge GT Refclk (voir la documentation de la carte).
- Choisir la fréquence de l'horloge d'initialisation. Celle-ci est fournie par l'utilisateur et devrait être un multiple l'horloge GT Refclk pour éviter les problèmes de timing lors de changement de domaine d'horloge.
- Choisir la fréquence de l'horloge DRP Clk (reconfiguration dynamique). Laisser par défaut si cette option n'est pas utilisée ou choisir la même fréquence que l'horloge d'initialisation.

Références [4]

Configuration de l'IP Core Aurora

Configuration de la couche liaison

- Choisir le mode de transmission (simplex ou full-duplex).

Références [4]

Configuration de l'IP Core Aurora

Configuration de la couche liaison

- Choisir le mode de transmission (simplex ou full-duplex).
- Choisir le type d'interface utilisateur (streaming ou framing).

Références [4]

Configuration de l'IP Core Aurora

Configuration de la couche liaison

- Choisir le mode de transmission (simplex ou full-duplex).
- Choisir le type d'interface utilisateur (streaming ou framing).
- Choisir le type d'interface de controle de flux (aucun, UFC, NFC, UFC + NFC).

Références [4]

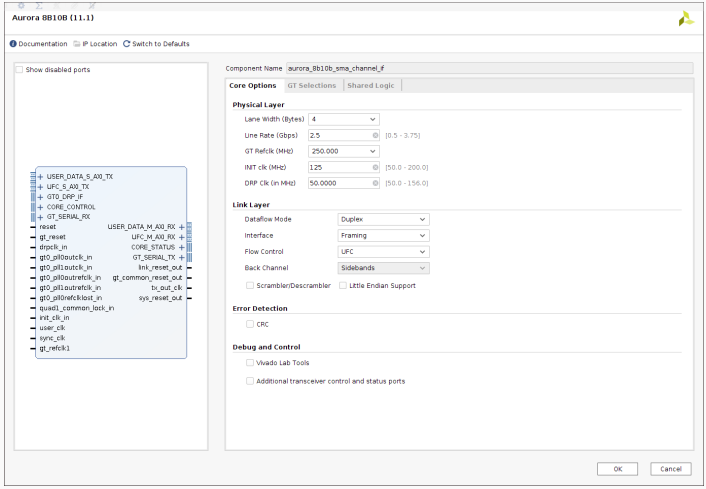
Configuration de l'IP Core Aurora

Configuration de la couche liaison

- Choisir le mode de transmission (simplex ou full-duplex).
- Choisir le type d'interface utilisateur (streaming ou framing).
- Choisir le type d'interface de controle de flux (aucun, UFC, NFC, UFC + NFC).
- Laisser les autres options par défaut si elles ne sont pas utilisées.

Références [4]

Configuration de l'IP Core Aurora



Références [4]

Configuration de l'IP Core Aurora

Configuration des GTP

- Choisir le nombre de lignes pour le canal.

Configuration de l'IP Core Aurora

Configuration des GTP

- Choisir le nombre de lignes pour le canal.
- Assigner les lignes aux GTP.

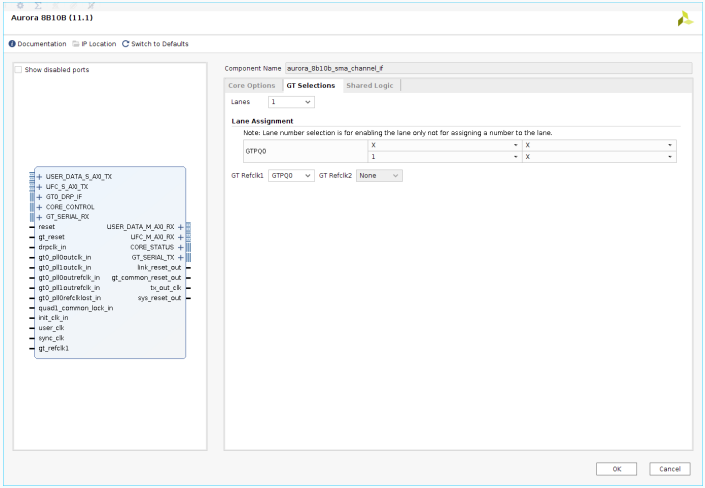
Configuration de l'IP Core Aurora

Configuration des GTP

- Choisir le nombre de lignes pour le canal.
- Assigner les lignes aux GTP.
- Assigner la source d'horloge au GTP Quad.

Références [4]

Configuration de l'IP Core Aurora



Références [4]

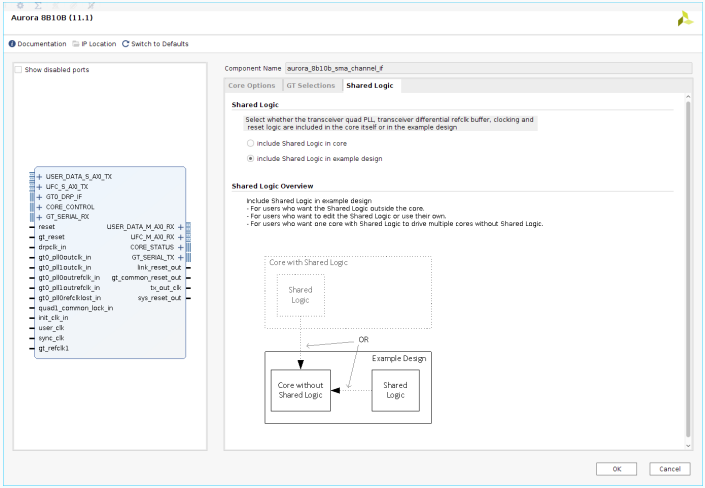
Configuration de l'IP Core Aurora

Logique partagée

- Choisir où se situe la logique partagée. A l'intérieur du core ou à l'extérieur du core.

Références [4]

Configuration de l'IP Core Aurora



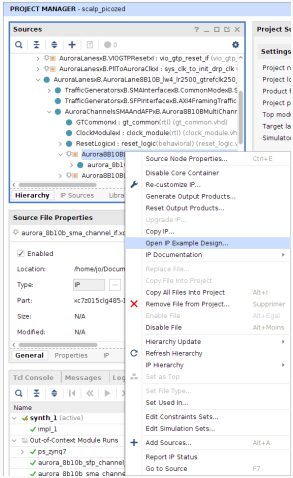
Références [4]

Configuration de l'IP Core Aurora

Sélection du projet d'exemple Aurora

- Sélectionner la nouvelle IP Aurora et sélectionner "Open IP Example Design...".

Configuration de l'IP Core Aurora



Références [4]

Configuration de l'IP Core Aurora

Projet d'exemple Aurora

- Les sources du core Aurora sont celles surlignées en rouge. Celles-ci sont déja présentes dans votre projet.

Configuration de l'IP Core Aurora

Projet d'exemple Aurora

- Les sources du core Aurora sont celles surlignées en rouge. Celles-ci sont déja présentes dans votre projet.
- Copier dans votre projet les sources du "top level" et de la partie communes. Celles-ci sont surlignées en vert.

Références [4]

Configuration de l'IP Core Aurora

Projet d'exemple Aurora

- Les sources du core Aurora sont celles surlignées en rouge. Celles-ci sont déja présentes dans votre projet.
- Copier dans votre projet les sources du "top level" et de la partie communes. Celles-ci sont surlignées en vert.
- Pour finir, retrouner à votre projet Vivado incluant votre IP Aurora et adaptez les sources que vous venez de copier selon vos besoins.

Références [4]

Projet d'exemple Aurora

- Le code peut (doit) éventuellement être nettoyé. Notamment en enlevant les instances VIO (Virtual Input Output) et les instances ILA Core. Ceci permet de garder uniquement le minimum de code nécessaire et ainsi améliorer la lisibilité.

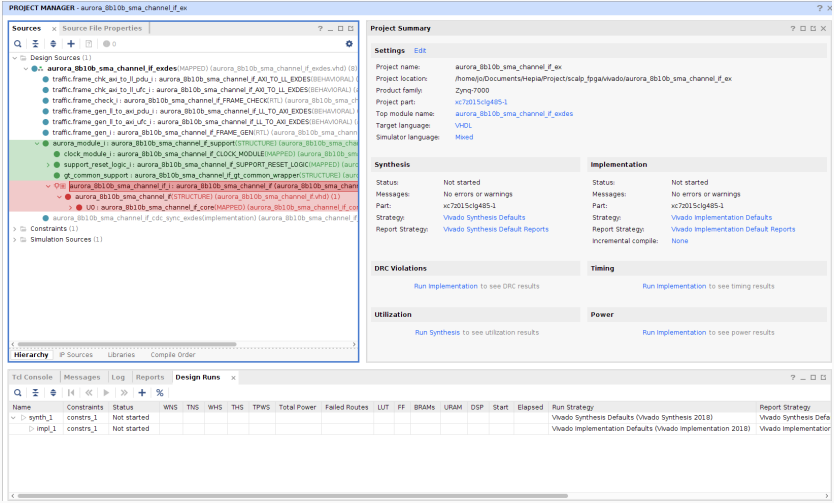
Configuration de l'IP Core Aurora

Projet d'exemple Aurora

- Le code peut (doit) éventuellement être nettoyé. Notamment en enlevant les instances VIO (Virtual Input Output) et les instances ILA Core. Ceci permet de garder uniquement le minimum de code nécessaire et ainsi améliorer la lisibilité.
- Eventuellement pour plus de lisibilité, réécrire le code des sources importées from scratch.

Références [4]

Configuration de l'IP Core Aurora



Références [4]

Références

Références i

J. S. C. A. D. B. F. V., D. B. and A. U.
Self-configurable 3-d cellular adaptive platform.
2018.

Xilinx.
7 Series FPGAs GTP Transceivers (UG482).
2014.

Xilinx.
Aurora 8B/10B Protocol Specification (SP002).
2014.

Xilinx.
Aurora 8B/10B v11.0 LogiCORE IP Product Guide (PG046).
2016.