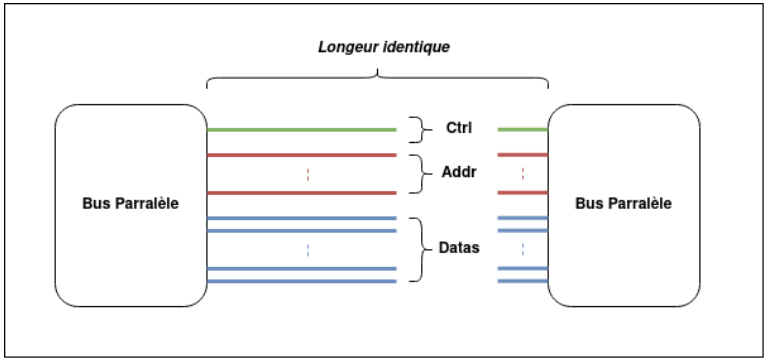
Introduction aux liens série à haut débit

Joachim Schmidt - joachim.schmidt@hesge.ch

*Communicating, Reconfigurable, Embedded Systems - CoRES - Hepia

Historique des bus de communication

- **Bus parallèle** transmet simultanément les bits d'un mot numérique sur des liens parallèles. Chaque mot est transmis en un cycle d'horloge. **Problèmes** : Limitation de la fréquence d'horloge (Génération d'erreur par retard). Nombre important de liens (taille des connecteurs). **Cas d'utilisation** : Bus parallèle (classique), Bus PCI, ...

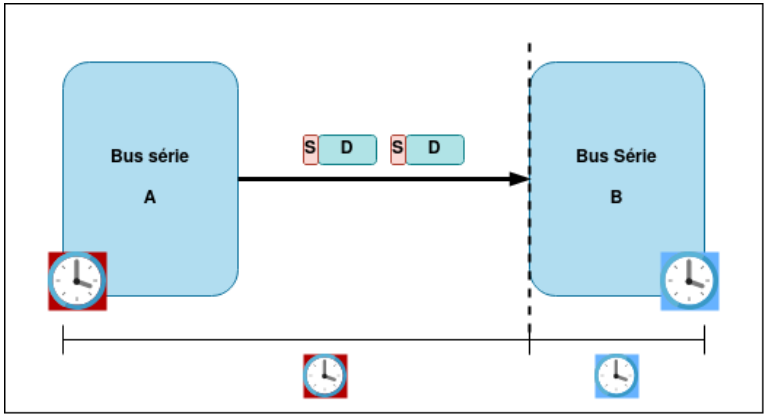


Historique des bus de communication

- **Bus série** transmet en série les bits d'un mot numérique sur un seul lien. Chaque mot de N bits est transmis en N cycle d'horloge.
  Modes de transmission série : Transmission série **asynchrone** (UART, Bus CAN, ...) et transmission série **synchrone** (HDMI, PCIe, USB, Ethernet, ...).

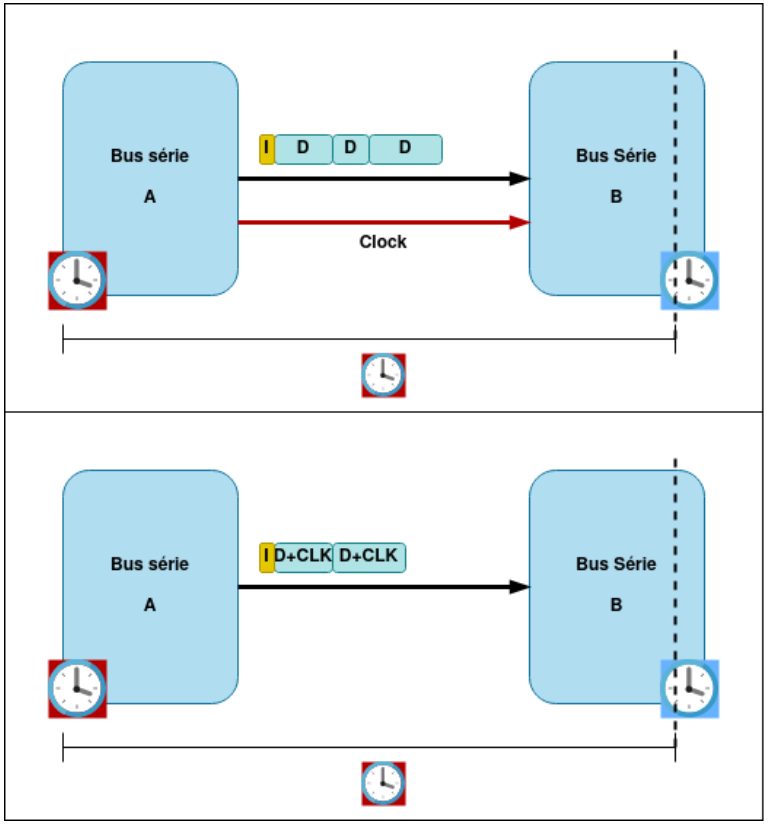
Transmission série asynchrone

- La **transmission série asynchrone** permet des débits limités. Chaque trame envoyée doit être précédée d'un signal de synchronisation (start). Chaque trame est en règle générale de taille constante. L'émetteur envoie les données avec son propre signal d'horloge. Le récepteur reçoit les données avec son propre signal d'horloge.



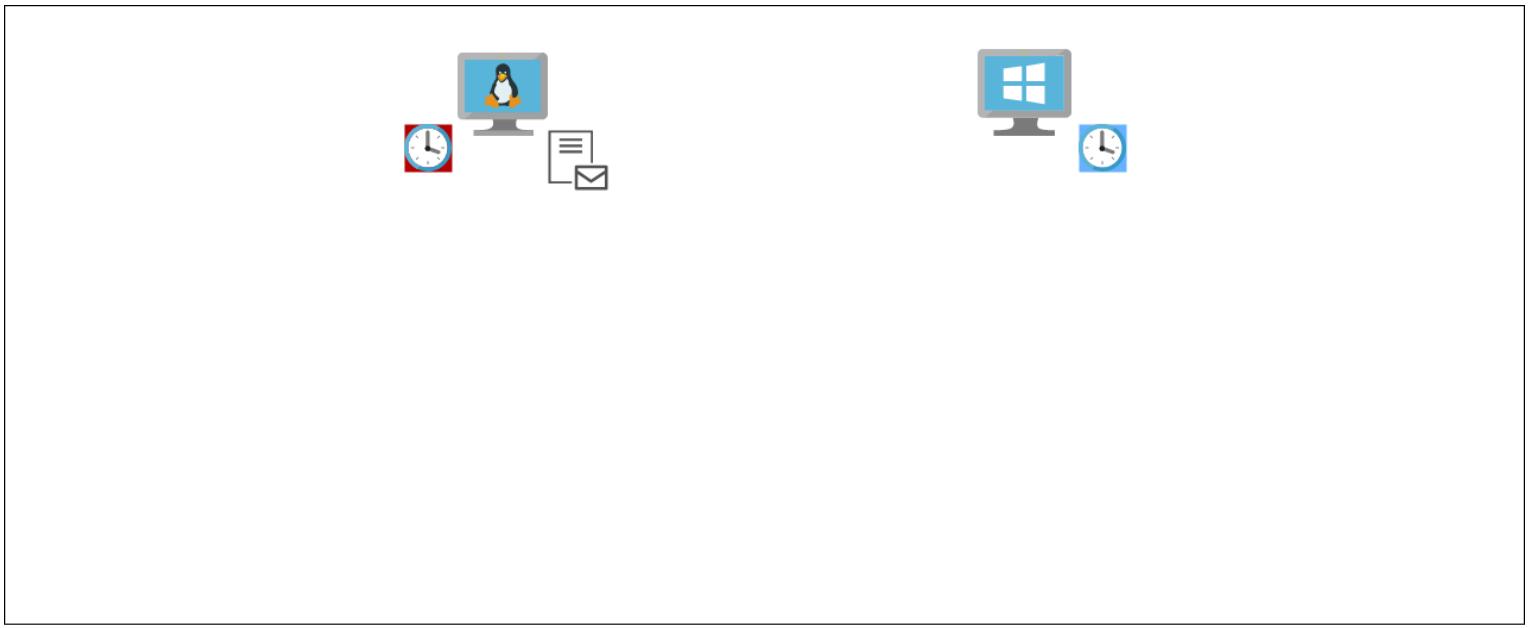
Transmission série synchrone

- La **transmission série synchrone** permet des débits très élevé (plusieurs dizaine de Gbps). On transmet systématiquement le signal d'horloge sur une ligne séparée ou sur la même ligne que les données. L'émetteur envoie les données, ainsi que son signal d'horloge. Le récepteur reçoit les données en utilisant le signal d'horloge de l'émetteur.

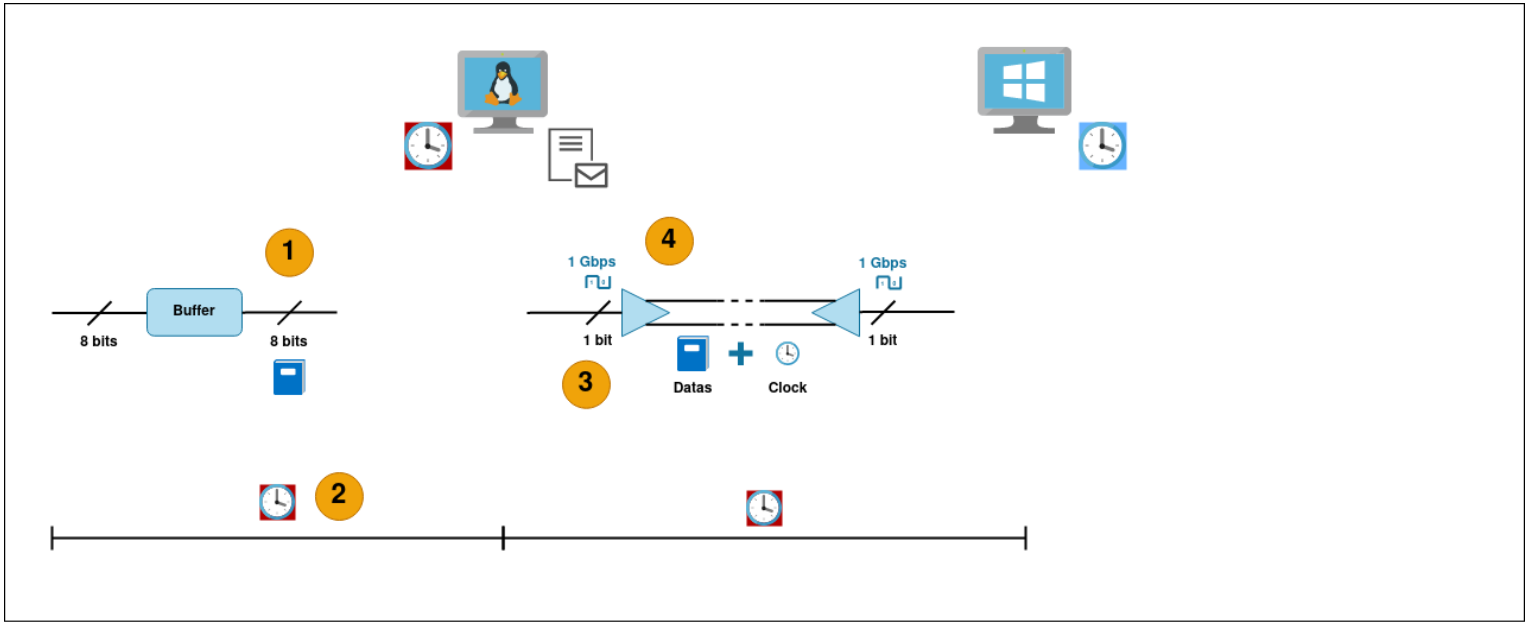


On va se concentrer sur les transmissions séries synchrones

- **Tux** veut envoyer un message à **Bill** et il ne veut perdre aucun mot de son message.
- Le système de Tux fonctionne avec une **horloge A de 100 MHz** (en réalité 100.001 MHz).
- Le système de Bill fonctionne avec une **horloge B de 100 MHz** (en réalité 99.999 MHz).
- Le débit de la transmission est de **1 Gbps**.

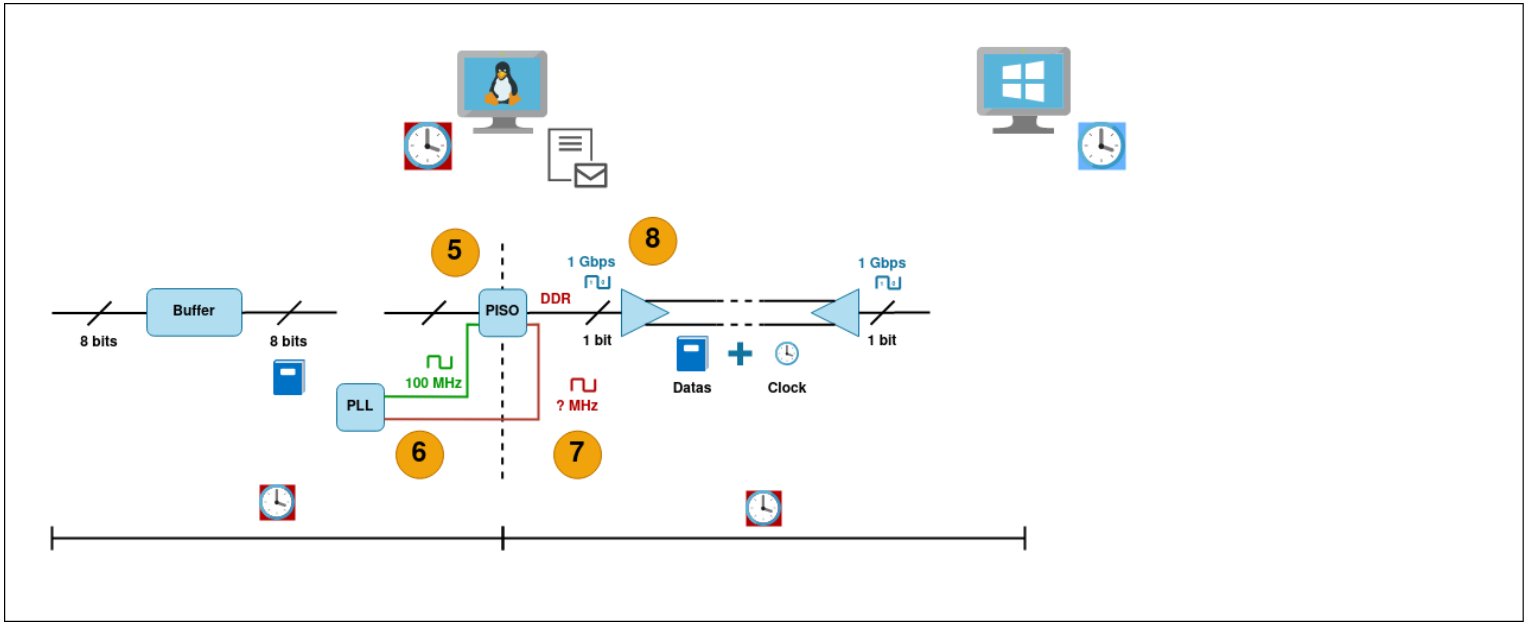


On va se concentrer sur les transmissions séries synchrones

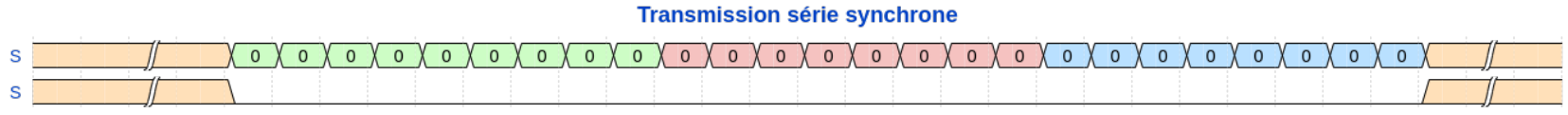


On va se concentrer sur les transmissions séries synchrones

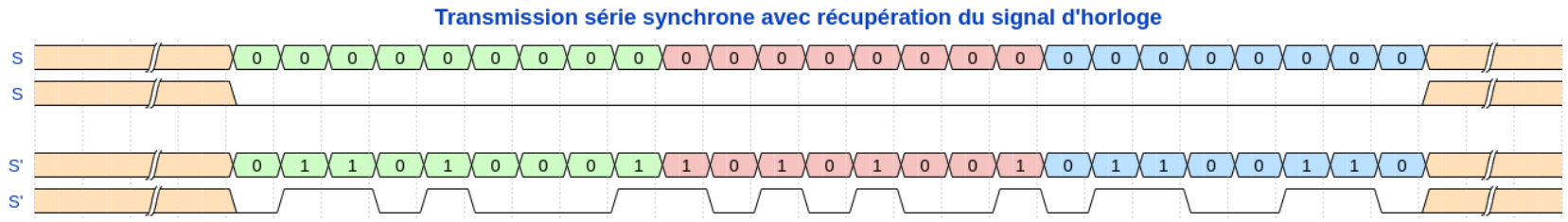
- **Q :** Quelle est la fréquence utilisée pour générer les données sur la ligne série ? Et est-ce que le débit binaire est juste ?



Comment encoder le signal d'horloge dans les données ?

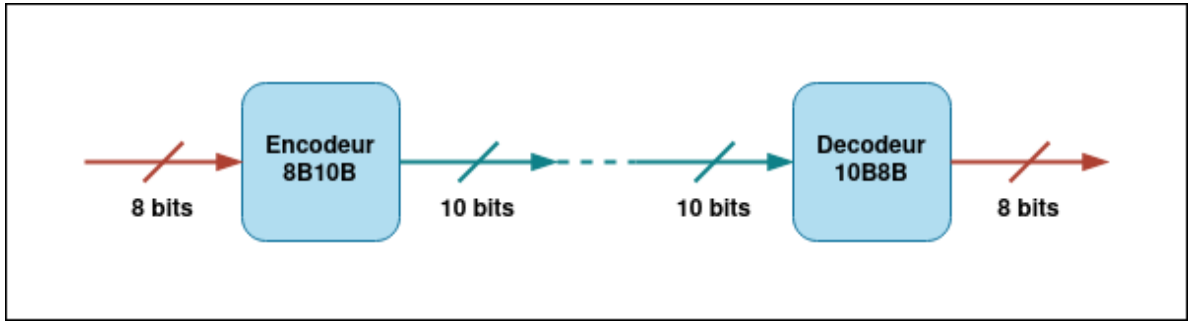


Comment encoder le signal d'horloge dans les données ?

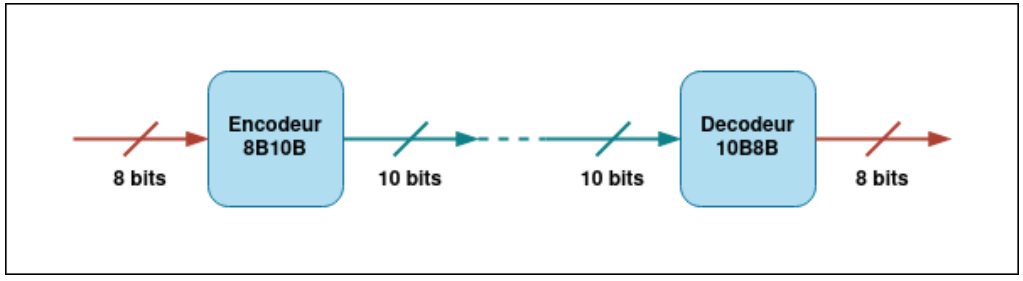


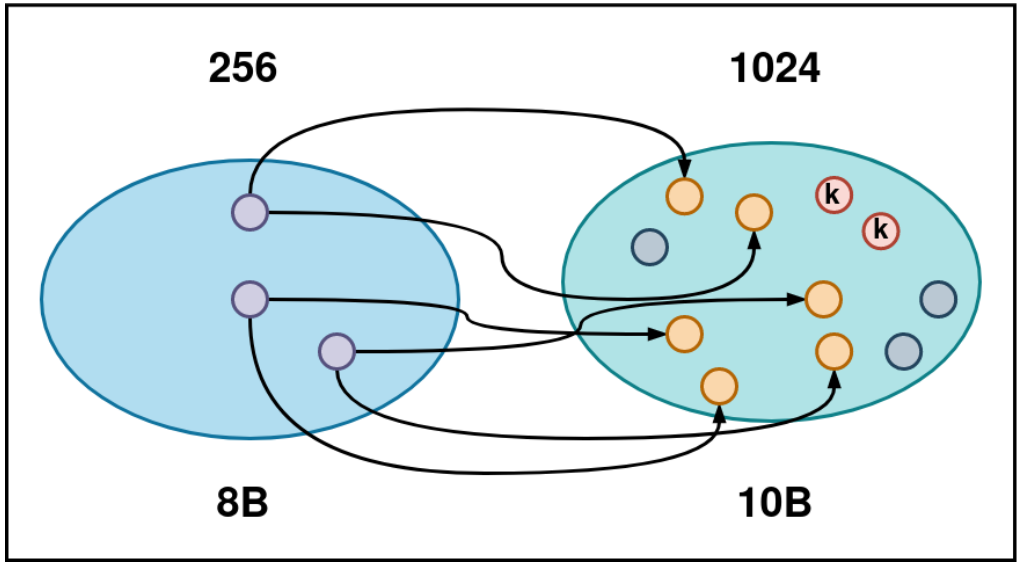
Pré-encodage 8B10B (variante 64B66B)

- On transforme nos mots de **8 bits** (256 mots possibles) en mot de **10 bits** (1024 mots possibles).
- Pour chaque mots de 8 bits, on garde deux mots de 10 bits symétriques. Et uniquement les mots ou on a suffisamment de transitions. On utilise pour les données 512 mots sur 1024.
- Les mots **0000000000** et **1111111111** ne sont jamais utilisés.
- L'objectif est d'avoir **en moyenne autant de 1 que de 0** sur la ligne.
- **L'amplitude de la composante continue** du signal est **en moyenne nulle**.
- Grâce aux transitions, on **transmet également le signal d'horloge qui a généré les données**.
- Parmi les 512 mots de 10 bits restants, on en garde **12** que l'on appel **caractères K** est que l'on utilise pour contrôler le flux.



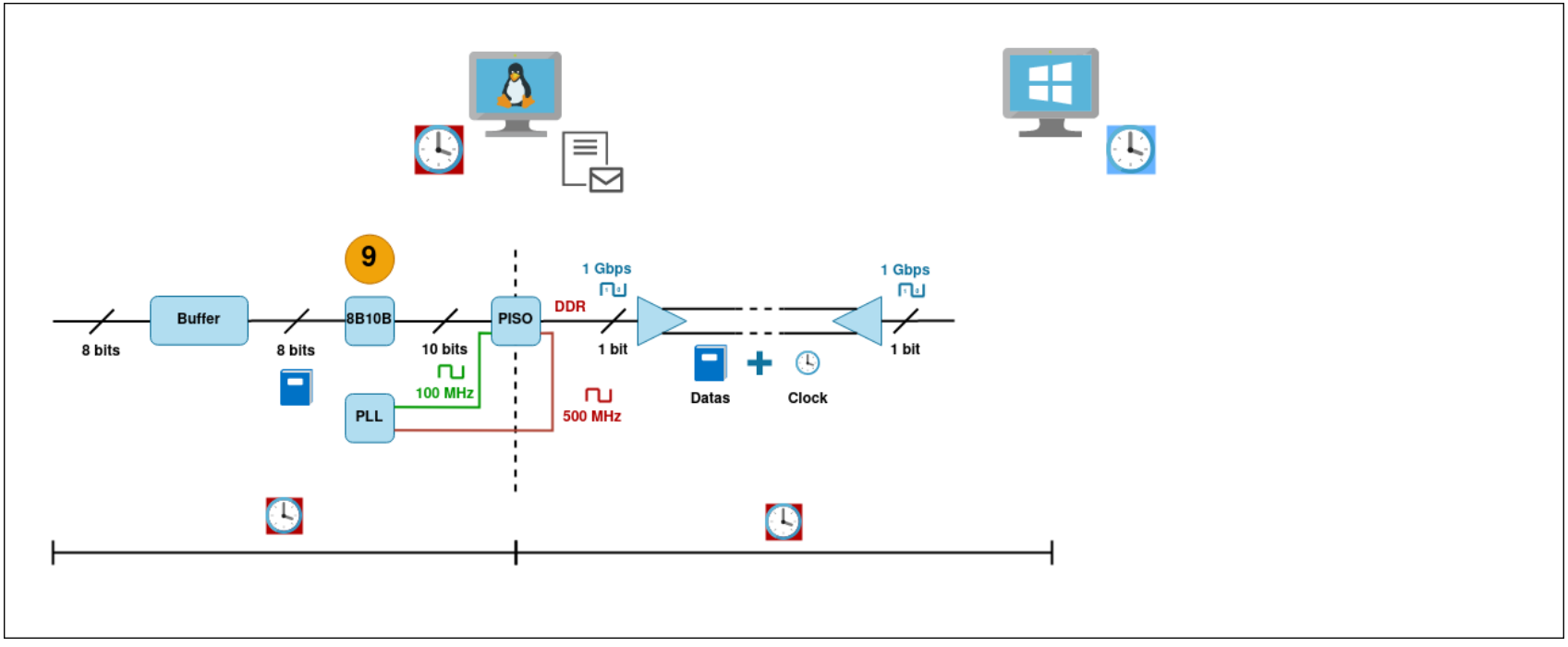
Pré-encodage 8B10B (variante 64B66B)





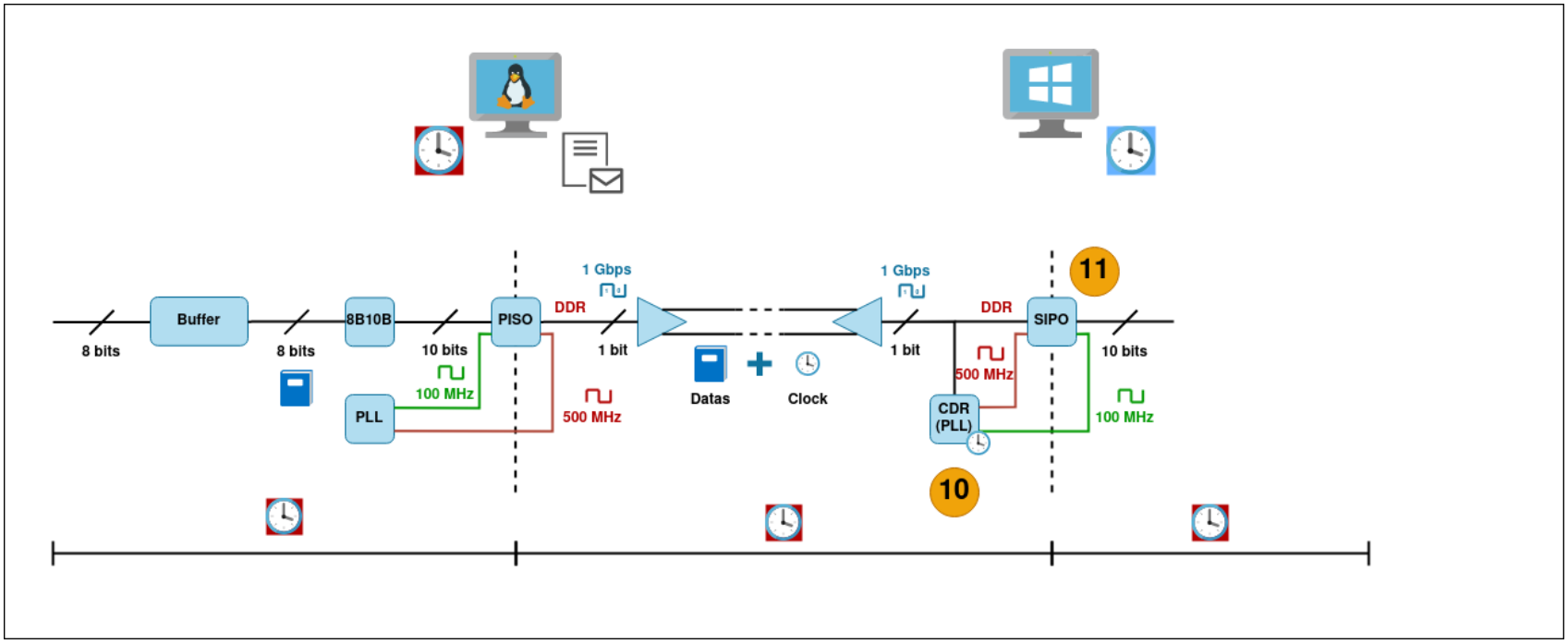
On transmet les données et le signal d'horloge à l'aide du pré-encodage 8B10B

- **Tux** a tout pour transmettre son message. Passons au cas de Bill...



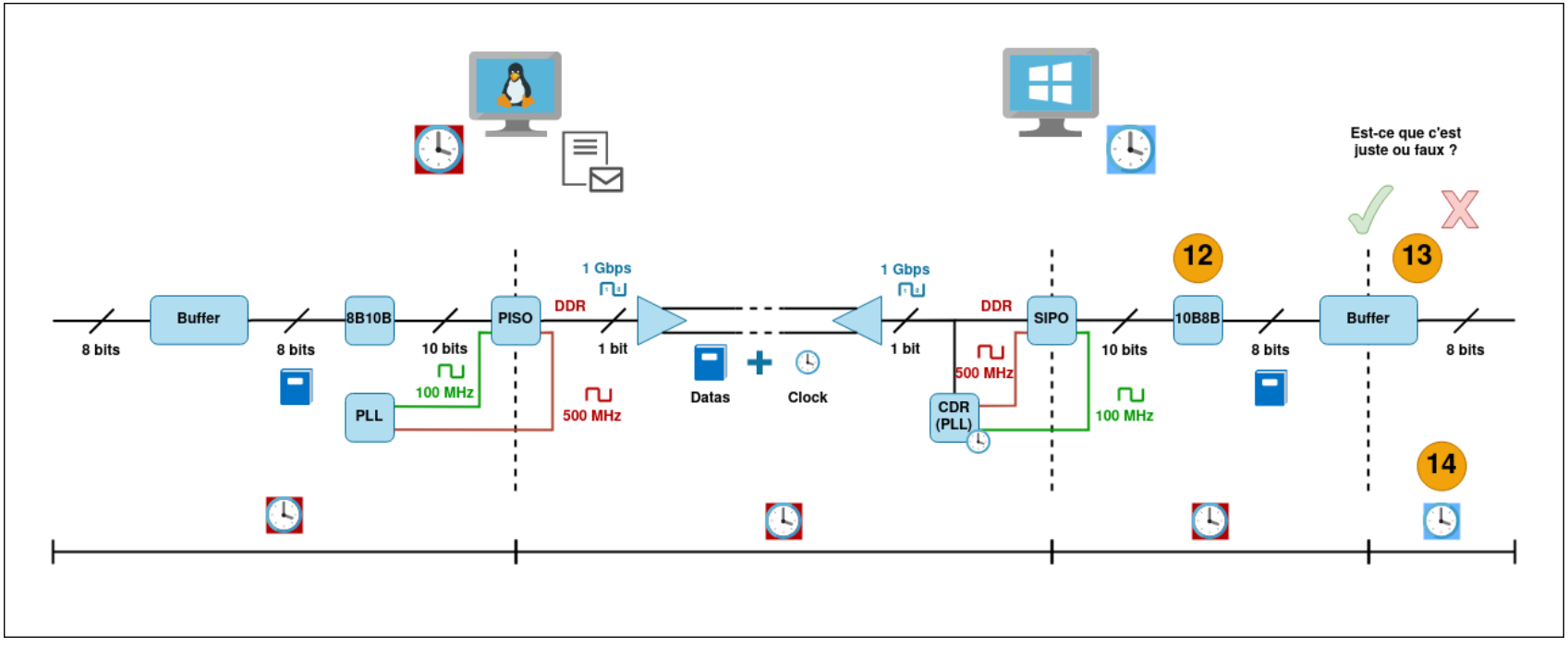
Récupération du signal d'horloge côté récepteur.

- **Bill** doit désérialiser les données en mots de 10 bits (on a encore le pré-encodage 8B10B). Pour cela, il utilise une PLL et génère les deux fréquences d'horloges. Une pour le côté série et une pour le côté parallèle.



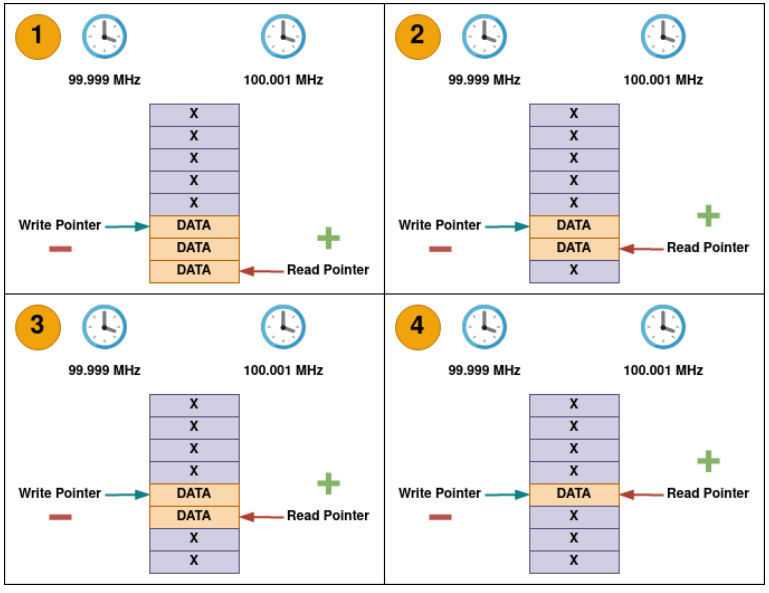
On décode les données en mots de 8 bits et on les récupèrent dans un buffer

- Est-ce que l'on peut sans autre lire les données dans le buffer avec l'horloge de **Bill** ?
- Si non, quel est le risque ?



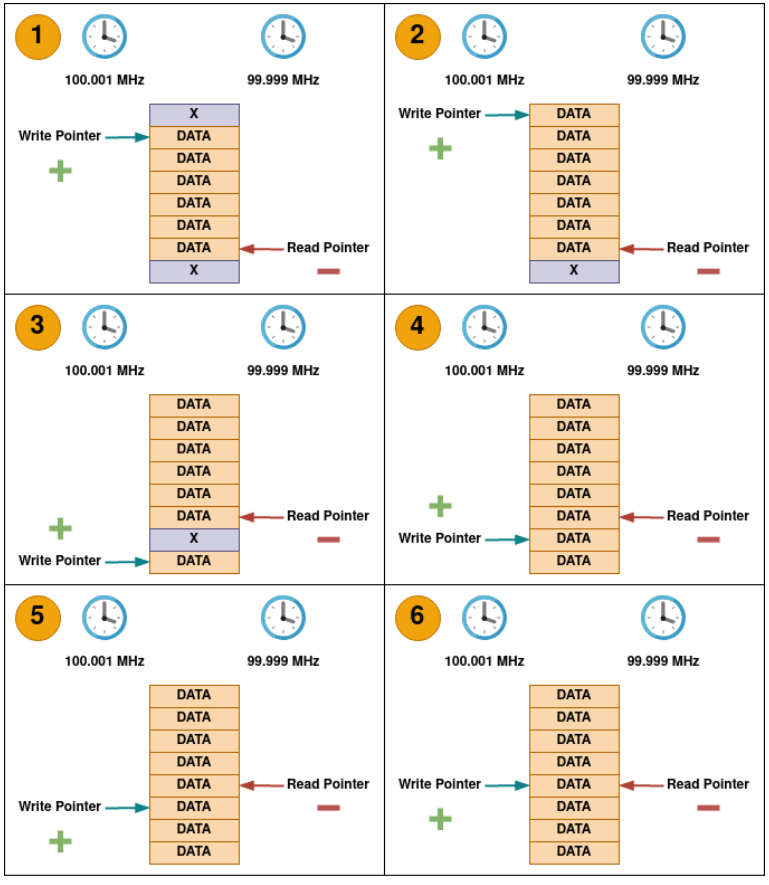
Premier cas dans lequel émetteur est plus lent que le récepteur.

- Dans ce premier cas, la lecture dans le buffer est plus rapide que l'écriture.
- Lorsque le pointeur de lecture rattrape le pointeur d'écriture, on relit des données que l'on avait déjà récupérées.



Deuxième cas dans lequel émetteur est plus rapide que le récepteur.

- Dans ce deuxième cas, l'écriture dans le buffer est plus rapide que la lecture.
- Lorsque le pointeur d'écriture rattrape le pointeur de lecture, on va écraser des données que l'on a pas encore récupérées.

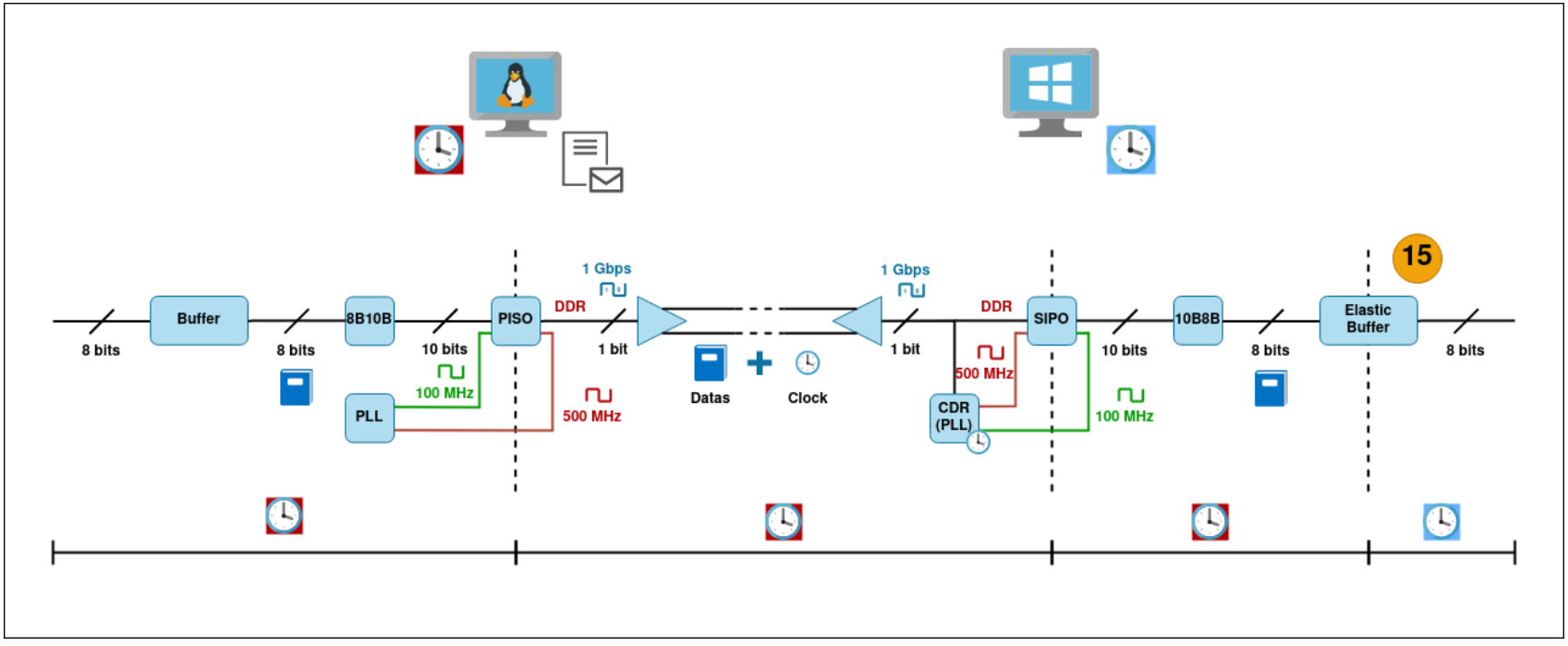


On décode les données en mots de 8 bits et on les récupèrent dans un buffer

- Est-ce que l'on peut sans autre lire les données dans le buffer avec l'horloge de **Bill** ?
- Si non, quel est le risque ?
- **Réponse** : Non, on va soit dupliquer des données, soit en perdre.

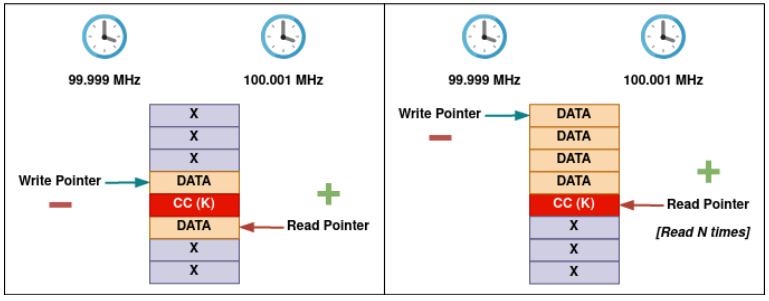
Le buffer élastique

- **Bill** va devoir utiliser un buffer spécifique, que l'on nomme **elastic buffer** et se coordonner avec **Tux** afin de l'utiliser correctement.



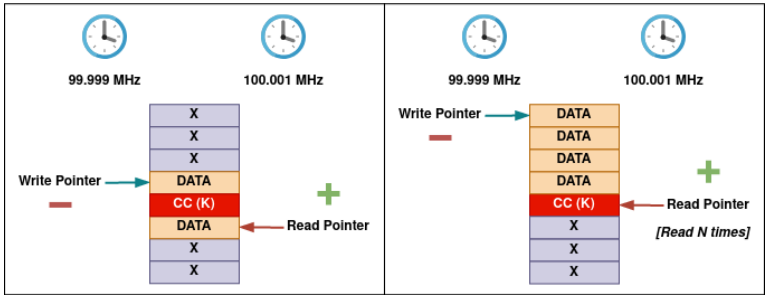
Premier cas dans lequel émetteur est plus lent que le récepteur avec le buffer élastique.

- Dans ce premier cas, la lecture dans le buffer est plus rapide que l'écriture.
- **Tux** a utilisé un certain nombre de **caractères K** du pré-encodage 8B10B appelé **CC** (Clock Compensation).
- Lorsque le pointeur de lecture tombe sur un symbole **CC**, ce même pointeur de lecture n'est plus incrémenté durant un certain nombre de cycles d'horloge.
- ...



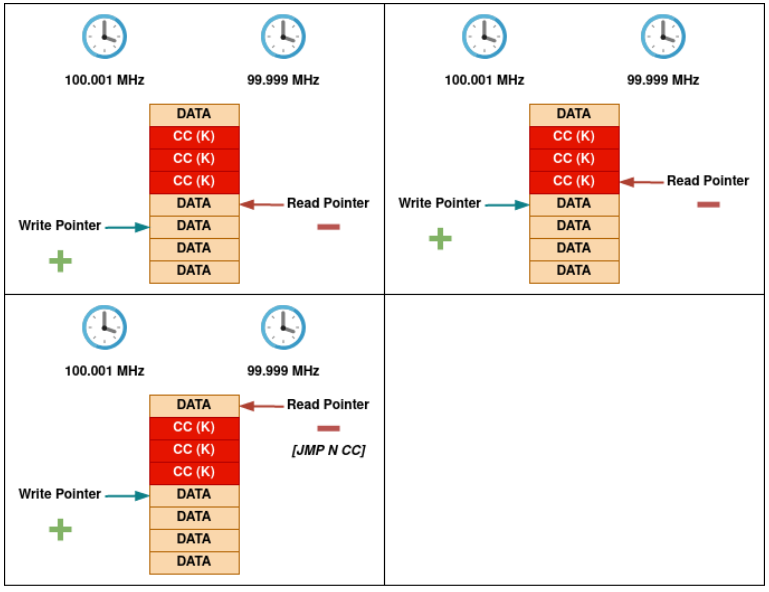
Premier cas dans lequel émetteur est plus lent que le récepteur avec le buffer élastique.

- ...
- Ce laps de temps, permet au pointeur d'écriture de s'éloigner à nouveau du pointeur de lecture.
- Le nombre de symbole **CC** et l'intervalle entre chaque groupe de symbole **CC** doivent être définis clairement entre l'émetteur et le récepteur.



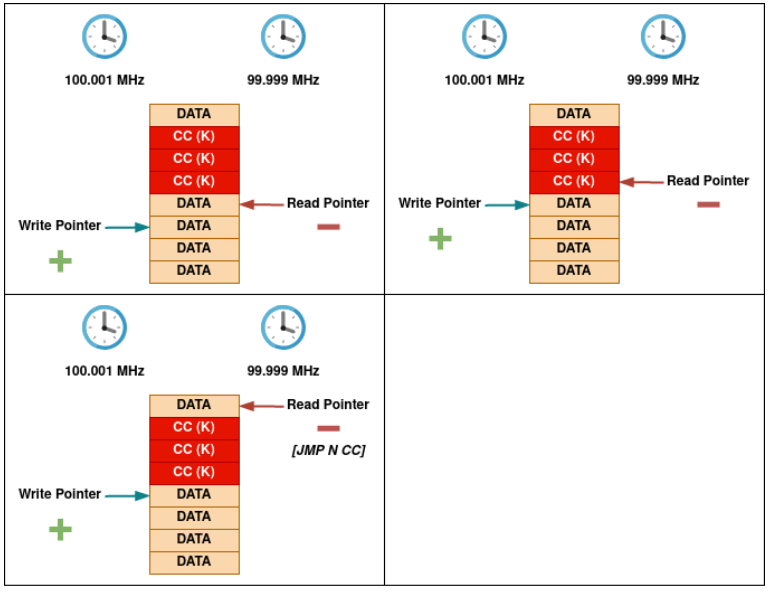
Deuxième cas dans lequel émetteur est plus rapide que le récepteur avec le buffer élastique.

- Dans ce deuxième cas, l'écriture dans le buffer est plus rapide que la lecture.
- **Tux** a utilisé un certain nombre de **caractères K CC** (Clock Compensation).
- Lorsque le pointeur de lecture tombe sur un symbole **CC**, ce même pointeur de lecture effectue un incrément du nombre de symbole **CC** introduit dans le flux.
- ...

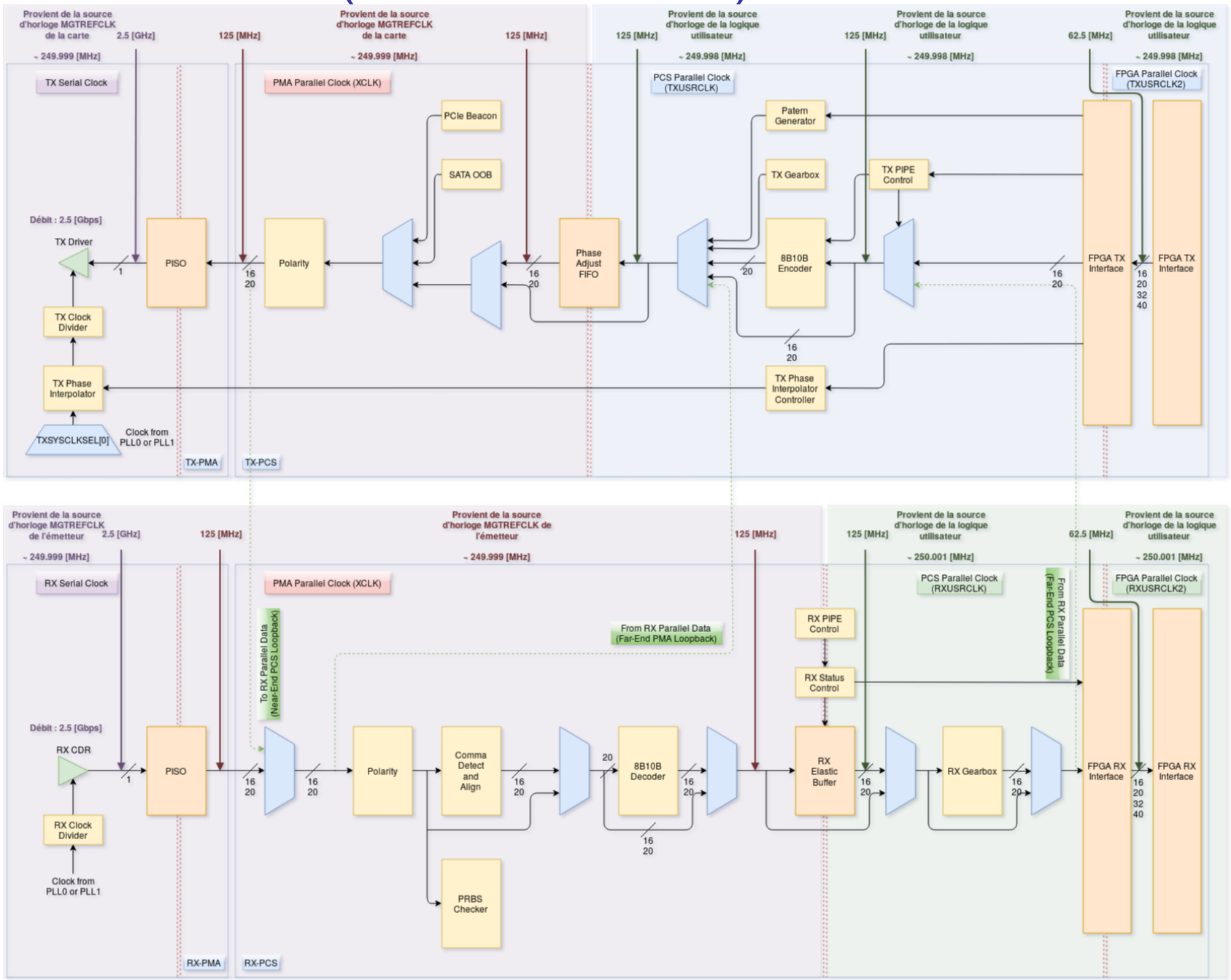


Deuxième cas dans lequel émetteur est plus rapide que le récepteur avec le buffer élastique.

- ...
- Ce saut permet au pointeur de lecture de s'éloigner à nouveau du pointeur d'écriture.

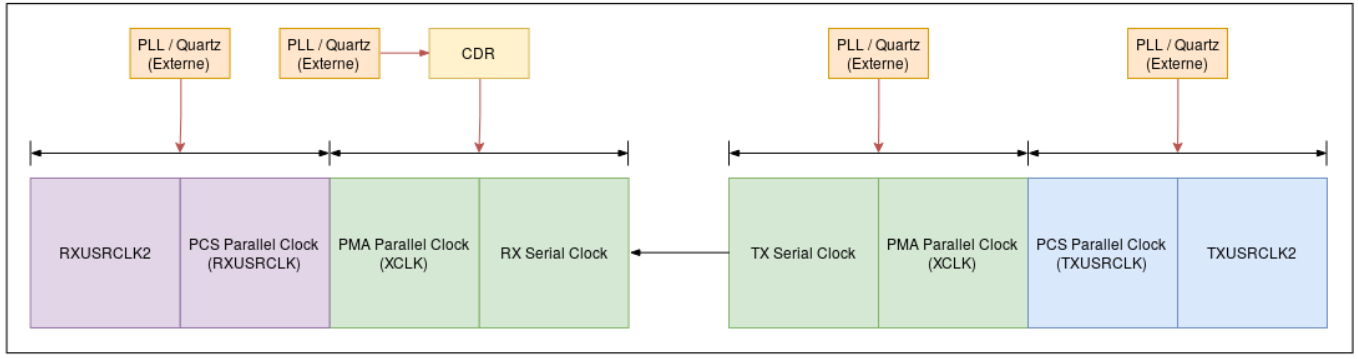


Transceiver GTP (Xilinx - 7 Series)

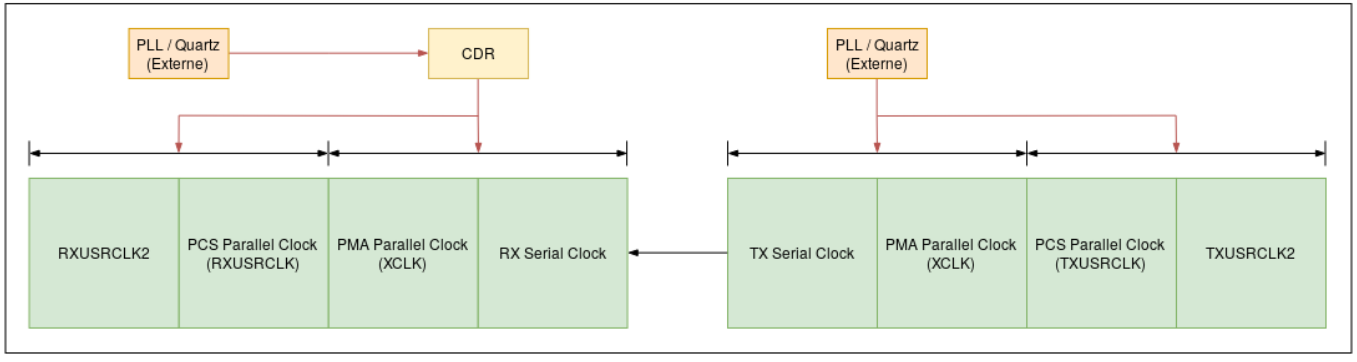


Transceiver GTP - Domaines d'horloges (Xilinx - 7 Series)

Configuration commune



Configuration faible latence



Configuration Aurora