THESE DE DOCTORAT
DE L'UNIVERSITE Pierre et Marie CURIE

Spécialité : **INFORMATIQUE**

Présentée par : **Anne PIERRE DUPLESSIX**

Pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITE PARIS VI

Sujet de la thèse :

Etude et réalisation d'une liaison série à 1 GBaud indépendante du codage des données

Soutenue le : **4 avril 1997**

Devant le jury composé de :

Alain Greiner	*Directeur de thèse*
Andreas Kaiser	*Rapporteur*
Patrick Garda	*Rapporteur*
Pascal Sainrat	*Examinateur*
Roland Marbot	*Examinateur*
Francis Devos	*Examinateur*

Remerciements

Je souhaite remercier ici toutes les personnes qui m'ont accompagnée pendant la réalisation de cette thèse.

En premier lieu, je souhaite exprimer toute ma gratitude à Roland Marbot et son équipe: Pascal Couteaux, Jean-Claude Le Bihan et Reza Nezamzadeh qui m'ont accueillie à BULL et m'ont aidée à mener à bien mon projet tout au long de ces trois années.
Je remercie également l'équipe 'technologie' de BULL pour son soutien technique.

Je remercie chaleureusement Alain Greiner, directeur du MASI, qui à travers l'enseignement du DEA m'a permis de découvrir et d'apprécier le monde de la micro-électronique et qui a pris la peine de corriger ce mémoire avec patience et rigueur.

Je remercie les membres du jury Francis Devos et Pascal Sainrat et plus particulièrement les rapporteurs Andreas Kaiser et Patrick Garda pour le soin qu'ils ont apporté à l'examen de ce document.

Pour finir, je souhaite remercier ici mes parents: leurs encouragements m'ont accompagnée durant ma longue scolarité et aujourd'hui qu'elle se termine enfin, je leur dédie ce travail en témoignage de ma gratitude.

Table des matières

Introduction

Face à la montée en puissance de traitement des microprocesseurs, développer de nouveaux modes de transmission est devenu crucial. La capacité de traitement des microprocesseurs n'est plus limitée par la puissance intrinsèque du circuit mais par la quantité d'information que le circuit peut échanger avec ses périphériques, ses mémoires ou d'autres circuits.
Ainsi les applications multimédia nécessitent la transmission et l'affichage à l'écran d'images animées. Pour que l'accès à ces images soit rapide et disponible à tout moment, des débits de l'ordre du Gbits/sec sont nécessaires.

Tous les éléments qui participent à la transmission des données sont concernés par cette évolution. Ceci comprend d'une part les éléments physiques c'est-à-dire les ports d'entrée/sortie (au niveau du circuit et de la carte) et le moyen de transmission (ruban, câble coaxial, fibre optique, etc.), d'autre part les protocoles de transmission, de contrôle d'erreur, de routage, etc.
Nous avons réalisé l'étude d'un port d'entrée/sortie adapté à ce nouvel environnement. Les qualités visées sont une faible surface occupée, une consommation minimum, un débit de l'ordre du Gbits/sec et une grande souplesse d'adaptation aux protocoles de transmission.

Le sujet de cette thèse est la conception et la réalisation d'une liaison série haut débit adaptée à n'importe quel type de codage des données et à deux types de support physiques câble coaxial ou fibre optique.

Le document présent s'articule de la manière suivante :

Dans le premier chapitre, nous exposons le contexte technique dans lequel prend place notre étude. Il s'agit d'adapter les principes mis au point pour la liaison série HS-Link chez BULL dans le cadre du projet ESPRIT OMI-HIC. La principale innovation consiste pour notre liaison série à s'affranchir de toute contrainte de codage sur les données. Cette absence de contrainte sur les données se traduit par une restriction sur les signaux d'horloge utilisés par l'émetteur et le récepteur : ils doivent avoir des fréquences égales à 10^{-4} près.

Dans le deuxième chapitre, nous présentons deux macrocellules liaison série réalisées en CMOS : HS-Link et Hot Pads.
HS-Link offre un exemple très robuste du point de vue de la sérialisation des données. Au niveau du désérialiseur, le principe utilisé s'appuie sur le codage 8B/12B défini dans la norme IEEE 1355 associée à HS-Link. HS-Link ne satisfait pas aux spécifications que nous visons car elle ne peut pas recevoir n'importe quel type de données.
Hot Pads vise les mêmes objectifs que notre étude. Bien que correctes d'un point de vue fonctionnel, les solutions proposées pour réaliser la macrocellule liaison série nous paraissent difficile à mettre en oeuvre et peu fiables en terme d'immunité au bruit.
Notre étude reprend les principes de sérialisation utilisés dans HS-Link et innove du point de vue de la désérialisation.

Dans le troisième chapitre, nous présentons les principes fonctionnels originaux mis au point pour le désérialiseur. L'échantillonnage des bits du signal série s'appuie sur des séries de signaux d'horloge décalés entre eux d'une période-bit. Ces signaux sont maintenus en phase avec le signal série par l'intermédiaire d'un retard variable qui compense le décalage progressif entre le signal série et l'horloge de référence du désérialiseur. Les bits échantillonnés sont ensuite resynchronisés sur un signal d'horloge récupéré CKR et regroupés en mots par référence à un mot spécifique que l'on cherche à identifier en sortie du sérialiseur.

Le quatrième chapitre présente de manière détaillée divers éléments originaux de la macrocellule liaison série. Il s'agit du retard variable, du comparateur de phase, de la boucle à verrouillage de retards, de la bascule d'échantillonnage du signal série, de l'arbre de OU-exclusifs et du circuit d'adaptation d'impédance. Ces éléments ont été conçus spécifiquement dans le cadre de la réalisation de liaisons série en technologie CMOS. Nous présentons ici les schémas et les principes de fonctionnement de ces éléments.

Le cinquième chapitre présente le circuit FIBULE qui nous permet de tester les principes de désérialisation détaillés au chapitre III. A titre d'exemple de transmission d'un codage quelconque des données, FIBULE implémente la couche physique du protocole Fibre Channel. Ce circuit intègre une macrocellule liaison série, un bloc auto-test et un bloc d'adaptation d'impédance.
Ce chapitre explicite la mise en pratique des principes exposés dans le troisième chapitre.

Le sixième chapitre présente la méthodologie de test adoptée pour la version 0.5µ de FIBULE. Les résultats obtenus sur carte de test sont détaillés et commentés.
De plus, nous présentons dans ce chapitre, une étude analogique sur le jitter associé au signal série. Il s'agit de quantifier la part de jitter introduite par chaque élément de la liaison série. Pour cela on compare une évaluation du jitter obtenu par simulation SPICE avec les résultats mesurés lors des tests sur carte.

Au terme de ce document, nous analysons les objectifs atteints et nous proposons une étude critique du travail réalisé.

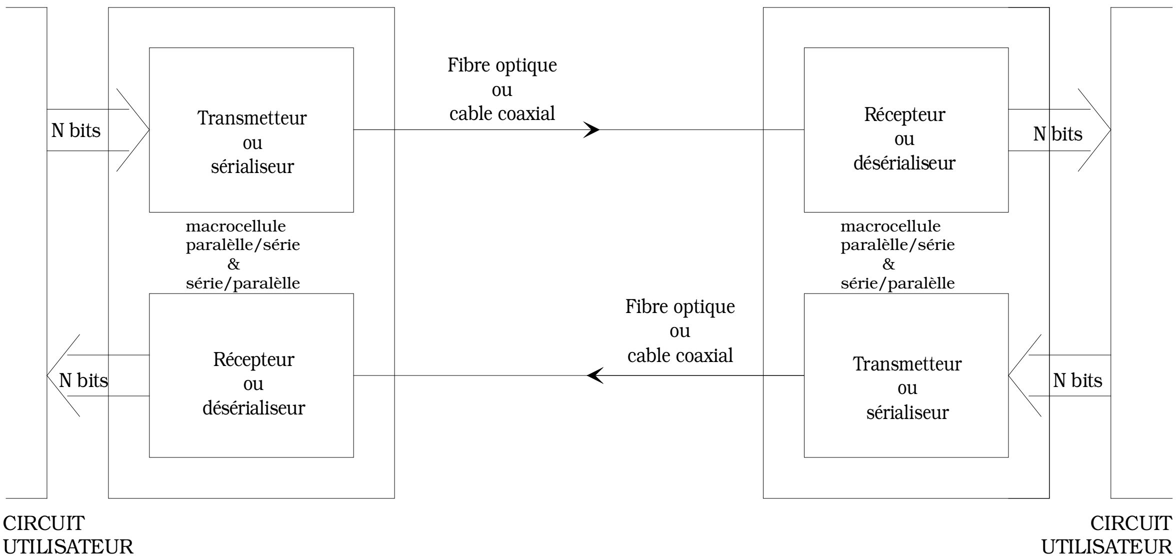
Chapitre I : Présentation du sujet

La transmission de données numériques peut se faire de deux façons:

- En parallèle : tous les bits d’un mot de largeur arbitraire sont transmis simultanément sur plusieurs fils. Cette méthode convient pour des distances courtes (quelques mètres) et permet des débits importants en gardant sur chaque fil un débit faible.

- En série : les bits sont transmis un par un sur un fil unique. Pour égaler les débits obtenus en parallèle, la fréquence de transmission doit être beaucoup plus élevée. Par contre cette méthode convient aux transmissions sur longue distance. C’est d’ailleurs le système utilisé dans l’industrie des télécommunications.

La transmission des données en série résout un problème inhérent à la transmission en parallèle : on élimine les problèmes de déphasage entre les bits à l’arrivée (dû à des parcours de longueur inégale).
La réalisation d’une liaison série a donc pour objectif la transmission de données numériques sur longue distance. Toutefois on peut également envisager son utilisation en remplacement des bus classiques si le débit délivré est suffisant.



Les liaisons série permettent de réaliser des connexions point à point.
Une liaison série bidirectionnelle se décompose en deux parties:

- Le sérialiseur réalise la transformation parallèle/série et envoie les données sérialisées sur un câble ou une fibre optique.

- Le désérialiseur reçoit les données et réalise la transformation série/parallèle.

Dans notre cas, le sérialiseur et le désérialiseur sont regroupés dans une macrocellule unique et plusieurs macrocellules peuvent être implantées pour un même circuit utilisateur.
La transmission sur câble coaxial se fait directement. Pour transmettre sur une fibre optique, il faut utiliser un convertisseur signal électrique/signal optique.

A/ Contexte du travail réalisé

Dans le cadre du projet Esprit OMI-HIC, la société Bull a développé une technique d'interconnexion à la fois modulaire, de coût modéré et à faible délai d'attente baptisée HS Link (High Speed Link). Ce projet a abouti à la définition de la norme IEEE 1355 qui définit un protocole de communication série point à point entre processeurs, entre cartes, voire entre systèmes (s'ils ne sont pas distants de plus de quelques mètres). Ce protocole définit trois gammes de débit: 100 à 200 Mbits/sec, 1 Gbits/sec et 3 Gbits/sec.

Pour la première gamme, la norme IEEE 1355 repose sur la technologie de communication série DS-Link mis au point par la société Inmos pour le Transputer T9000.

Pour les deux gammes supérieures, elle repose sur la technologie HS Link de Bull. Sa principale particularité est de se présenter sous la forme d'une macrocellule intégrable à la

périphérie du circuit utilisateur qui peut être un processeur, une mémoire, un commutateur ou un routeur.
La version CMOS doit couvrir la gamme autour du Gbits/sec, la version BiCMOS celle comprise entre 2.5 et 3 Gbits/sec.

Un circuit de démonstration BULLIT a été réalisé en technologie CMOS 0.8µ pour prouver la validité de la technologie HS Link. Il intègre 2 liaisons série bidirectionnelles (sous forme de macrocellules) et un protocole simple pour gérer le démarrage des transmissions et la mesure du taux d'erreur. Ce circuit a permis de démontrer la possibilité de faire des liaisons série haut débit en technologie CMOS.

Les caractéristiques de la macrocellule HS Link sont les suivantes :

- Débit de 1 Gbits/sec
- Faible puissance consommée : ~ 0.3 W
- Surface occupée : 1 mm^2
- Le récepteur n'a pas de signal d'horloge local, il utilise l'horloge récupérée dans le signal série. Le principe de transmission des données est totalement asynchrone.
- Utilisation d'un codage des données 8B/12B spécifique à la technique de transmission.

Les applications visées sont la réalisation de réseaux d'interconnexion hautes performances pour systèmes multiprocesseurs hétérogènes.

B/ Objectifs

La mise en oeuvre de la technologie HS Link sous la forme du circuit BULLIT a permis de montrer que des liaisons série CMOS peuvent égaler les performances en débit des liaisons série AsGa ou bipolaire en offrant des avantages importants en terme de consommation et de surface occupée.
Cependant la technologie HS Link ne peut pas remplacer n'importe quelle liaison série car elle ne fonctionne qu'avec son propre codage des données.

Le but de notre étude est donc de généraliser le concept de liaison série mis en place pour le projet OMI-HIC à tous les types de codage de données c'est-à-dire de concevoir et mettre au point une technologie liaison série qui soit:

- adaptée à n'importe quel type de codage des données ce qui implique de réaliser des transmissions en mode plésiochrone

- capable de délivrer un débit de 1 Gbits/sec en CMOS
- intégrable dans un circuit utilisateur sous la forme d'une macrocellule
- adaptée aux transmissions en mode plésiochrone : l'émetteur et le récepteur disposent chacun d'une horloge de référence générée par un quartz oscillant. Aux dispersions près, les deux quartz oscillent à la même fréquence.
- capable de s'interfacer avec un câble coaxial ou une fibre optique
- de consommation faible pour ne pas perturber le fonctionnement du circuit principal
- de superficie aussi réduite que possible pour qu'un nombre important de liaisons série puissent être implantées sur un même circuit intégré.

Ces objectifs permettent de répondre aux besoins d'une transmission à courte distance (quelques mètres) mais aussi de viser le marché beaucoup plus vaste des applications de type télécommunication (en utilisant par exemple des fibres optiques pour propager le signal série).

Dans les chapitres III et IV, nous présentons les principes et les schémas d'une macrocellule de sérialisation/désérialisation correspondant aux objectifs ci-dessus. Pour démontrer la faisabilité du schéma proposé, nous avons conçu et réalisé une macro-cellule pour des mots de 10 bits, sans contrainte de codage. Ce choix nous permet d'être compatible avec le codage 8B/10B d'IBM utilisé dans le protocole Fibre Channel et de comparer les performances de la macrocellule étudiée avec celles d'autres liaisons série ayant les mêmes caractéristiques en débit. Pour ce faire, nous avons choisi de réaliser un circuit-test (FIBULE) qui respecte le protocole Fibre Channel et sa norme associée ANSI X3T11.

Chapitre II : Etat de l'art

Dans ce chapitre nous allons présenter quelques liaisons série existantes et leurs caractéristiques.
A travers cette étude, nous voulons mettre en évidence les difficultés qu'il faut résoudre lors de la conception d'une liaison série.

Nous avons rassemblé dans le tableau ci-dessous différentes technologies de liaisons série dont les performances en débit se rapprochent de celles que nous visons, c'est-à-dire 1Gbits/sec.

Produit	Société	Technologie de fabrication	Débit en Gbits/sec	Puissance consommée en W	Nombre de CI nécessaires	Type de codage utilisé
Hot Rod	Triquint	AsGa	1	3	2	4b/5b
G-Taxi	Vitesse	AsGa	1.25	3.5	2	FCS
G-Link	HP	Bipolaire	1.4	5	2	SCI-serial
Hot Pads	Sun/ LSI Logic	CMOS	0.65 - 1	0.5	/	/
HS Link	Bull/ ST	CMOS	0.7 - 1	0.3	/	IEEE 1355

En observant leurs différentes caractéristiques (le débit, la puissance consommée, le nombre de circuits intégrés qui forment la liaison série, la technologie de fabrication), on constate que les liaisons série présentées dans ce tableau se divisent essentiellement en deux groupes :

- Le premier privilégie la performance. Il s'agit de liaisons série réalisées en technologie AsGa ou bipolaire et pour lesquelles l'émetteur et le récepteur sont présentés sous la forme de deux circuits séparés. La puissance consommée par ces circuits varie de 3 à 5 W.

- Le second privilégie la limitation du coût en terme de surface et de puissance consommée. Ce sont des liaisons série réalisées en technologie CMOS, disponibles sous forme de macrocellule. La puissance consommée par une macrocellule varie de 0.3 à 0.5 W.

Parmi les liaisons série présentées dans le tableau, nous en présentons deux plus en détail : ce sont les liaisons série HS-Link et Hot Pads dont les objectifs en terme d'intégration sont les plus proches des nôtres.

A/ Présentation de HS-Link

La liaison HS-Link développée par BULL respecte le standard IEEE 1355. Ce standard décrit le protocole logique qui gère la transmission du signal série ainsi que les propriétés physiques et électriques des supports de transmission et des connecteurs utilisés.
BULLIT est le circuit mis en oeuvre pour démontrer les capacités de la macro-cellule liaison série HS-Link. Il est réalisé en technologie CMOS.

Les caractéristiques associées aux liaisons série dans le standard IEEE 1355 sont les suivantes :

- La transmission des données est asynchrone.

Le fonctionnement du récepteur est basé uniquement sur la réception du signal série. Aucun signal d'horloge local n'est nécessaire. Le récepteur extrait le signal d'horloge inclus dans le signal série et fonctionne avec ce signal.
Aucune spécification n'est imposée à l'émetteur concernant la fréquence d'émission du signal série (entre 0.7 et 1 GHz).

- IEEE 1355 définit un codage 8B/12B des données.

Ce codage est un aspect fondamental du standard car c'est lui qui permet la transmission asynchrone des données. Chaque début de mot est marqué par une transition montante du signal série. Le récepteur isole cette transition et l'utilise comme signal de synchronisation.
D'autre part, ce codage est conçu de manière à éliminer la composante continue du signal série et il comporte un procédé de détection d'erreurs isolées.
Soit d[7:0] où d[7] est le bit de poids fort, le caractère de 12 bits correspondant à cette donnée est :

bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8	bit 9	bit 10	bit 11
Start	AP	e[0]	e[1]	e[2]	e[3]	e[4]	e[5]	e[6]	e[7]	I	Stop

(‘1’)											(‘0’)

où e[0:7] est égal soit à d[0:7] (I=0) soit à son inverse (I=1),
AP=1 si le nombre de ‘1’ dans le byte d[0:7] est pair
AP=0 si le nombre de ‘1’ dans le byte d[0:7] est impair
I permet de savoir si la donnée est inversée

Ce codage offre 256 caractères de données et 126 caractères de contrôle.

- IEEE 1355 définit un protocole de communication

Ce protocole est utilisé pour l’initialisation et l’arrêt d’une transmission ainsi que pour gérer, pendant la transmission, le contrôle de flux.

Le contrôle du flux de données nécessite l’échange de caractères de contrôle spéciaux représentant des crédits d’émission.

1/ Présentation de la macrocellule

La macrocellule occupe une surface de 2 mm^2 en technologie CMOS 0.5µ. Alimentée sous +3.3V, elle consomme une puissance de 0.3W pour un débit variant entre 0.7 et 1 Gbits/sec.
Le sérialiseur prend en entrée des caractères de 12 bits et un signal d'horloge CK.
Le désérialiseur reçoit le signal série et en extrait la transition 01 (Stop/Start) qui marque le début d'un caractère pour l'utiliser comme signal de synchronisation. Il fonctionne donc sans signal d'horloge externe et fournit en sortie, avec les données parallèles, ce signal d'horloge récupéré.

2/ Fonctionnement du sérialiseur

L'idée de base du sérialiseur est d'utiliser plusieurs signaux d'horloge basse fréquence décalés en phase au lieu d'un unique signal d'horloge haute fréquence.
Ces multiples signaux d'horloge sont réalisés en utilisant une boucle à verrouillage de retards (DLL). Le signal d'horloge local CK est successivement retardé par 12 éléments de retards variables identiques contrôlés par une commande numérique. Pour déterminer la durée des retards, CK est comparé en phase avec la sortie du 12ème retard CK(12) et le résultat ainsi obtenu est utilisé pour contrôler un compteur/décompteur. La valeur de ce compteur détermine la durée du retard telle que CK et CK(12) soient en phase. Ainsi la durée d'un élément de retard correspond à la durée d'un bit dans le signal série.

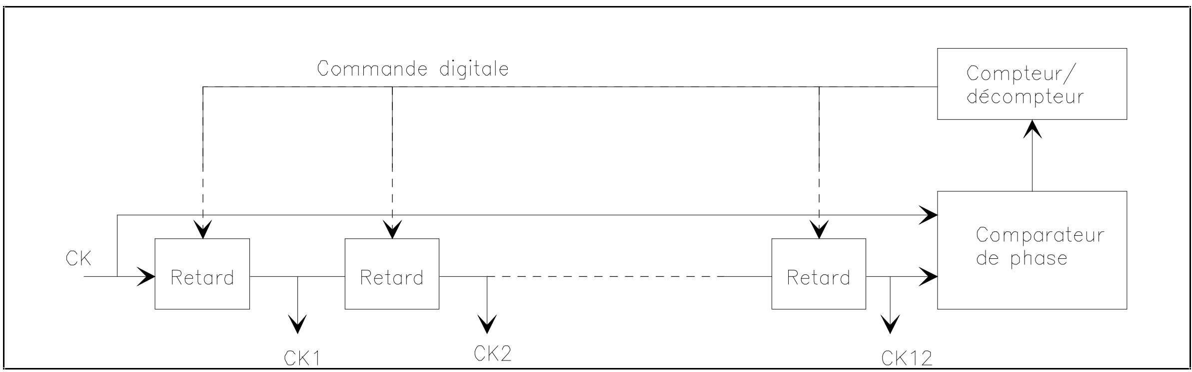
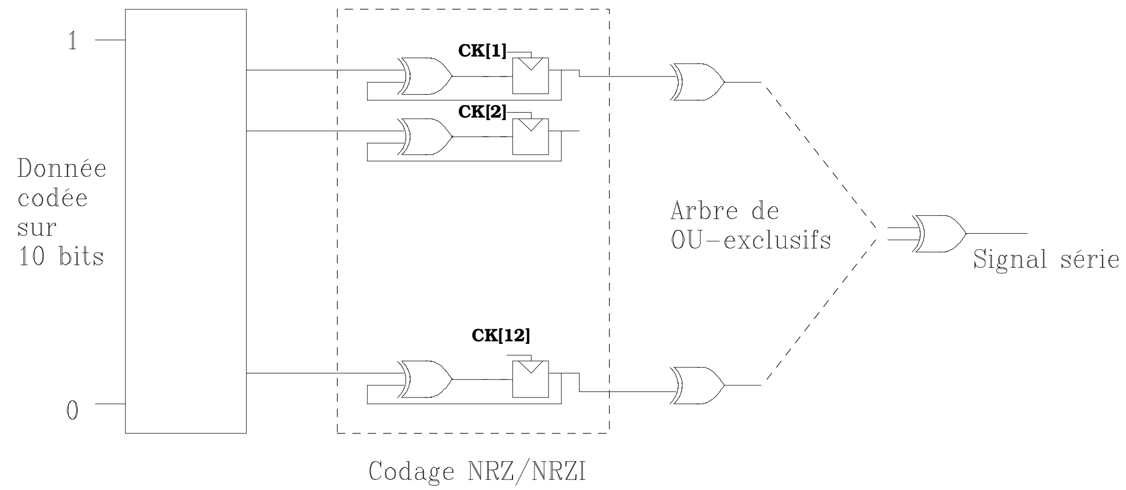


Schéma de principe de la boucle à verrouillage de retards

La sérialisation utilise un arbre de OU-exclusifs. Ce schéma a pour particularité que sa sortie réalise une transition chaque fois qu'une des ses entrées bascule, les autres entrées étant stables.
Ceci implique l'utilisation d'un codage NRZI des données présentées en entrée de l'arbre de OU-exclusifs.
Chaque bit est synchronisé sur un des signaux d'horloge CK(i) et présenté en entrée de l'arbre de OU-exclusifs



Réalisation du signal série

Ce procédé garantit un jitter minimum sur le signal de sortie. Il a fait l'objet d'un brevet déposé au nom de la société BULL par R.Marbot, JC.Lebihan, A.Cofler et R.Nezamzadeh sous la référence EP3350 (juillet 1993).

3/ Fonctionnement du désérialiseur

Les données et le signal d'horloge qui leur est associé forment un signal unique pour éviter les décalages de phase dus à des parcours de longueurs différentes.
Dans HS-Link, le début de chaque mot est marqué grâce aux bits Stop/Start par un front montant. Au niveau du désérialiseur, ce front montant est extrait du signal série entrant pour servir d'horloge de base au circuit à verrouillage de retards (DLL) qui génère les 12 horloges d'échantillonnage : l'émetteur transmet son horloge au récepteur qui la reconstitue.

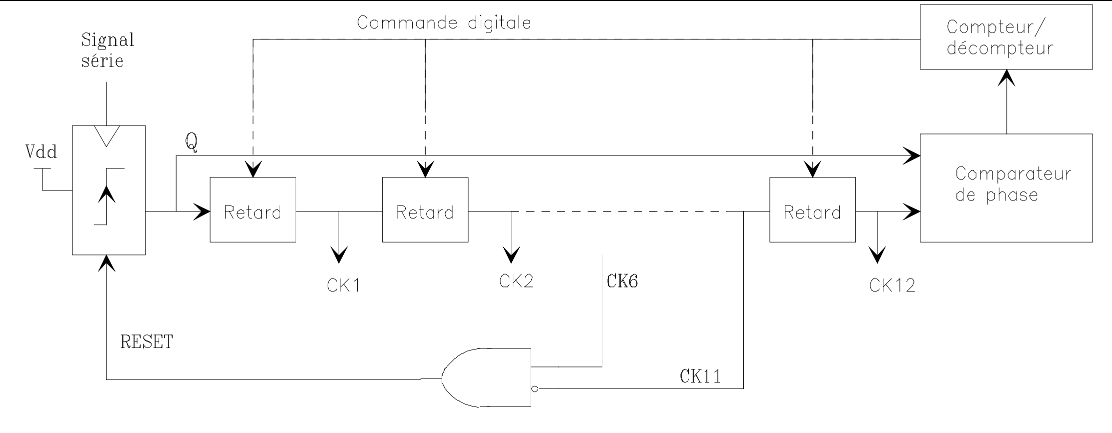


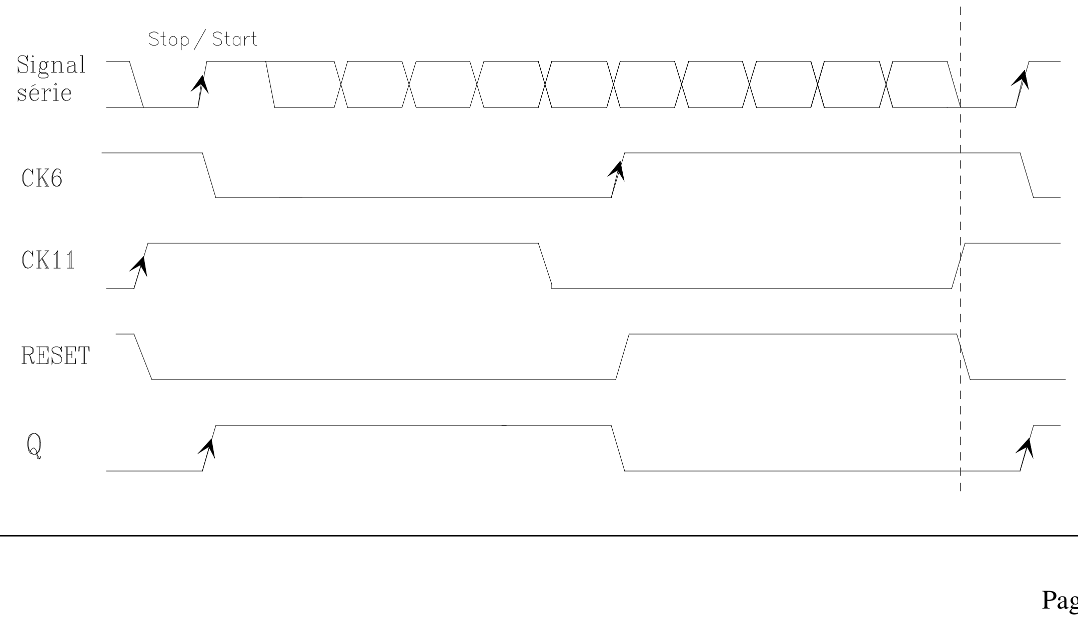
Schéma de principe du bloc de récupération de l'horloge transmise

En début de transmission, on transmet un caractère spécifique IDLE de la forme "111111000000". Ceci revient à transmettre uniquement la fréquence des caractères pour permettre à la boucle DLL de se stabiliser.

Quand Q et CK12 sont en phase, on peut commencer à transmettre des données. le système reste calé sur le front montant de début de caractère grâce au signal RESET.

- RESET est actif entre les fronts montants de CK6 et CK11, il force alors la valeur de Q à 0.
- RESET est relâché après le front montant de CK11. Le front montant suivant sur le signal série (c'est-à-dire la transition montante qui marque le début du caractère suivant) va mémoriser le 1 présenté en entrée de la bascule et générer un front montant sur Q.

Le signal série est échantillonné par la série de signaux d'horloge CK[1:12] et les valeurs obtenues sont synchronisées sur le signal d'horloge récupéré avant d'être proposées en sortie du désérialiseur.



Chronogramme associé au schéma de récupération de l'horloge transmise

4/ Analyse

La macrocellule liaison série HS-Link réalisée en technologie CMOS est intégrable dans un circuit utilisateur et sa consommation est faible. La seule restriction à l'utilisation de HS-Link est le respect du codage 8B/12B.
Le principe de fonctionnement du désérialiseur ne permet pas d'adapter cette liaison série à n'importe quel type de codage des données à cause du mécanisme de récupération d'horloge qui fait explicitement l'hypothèse d'un front montant au début de chaque caractère.

B/ Présentation de Hot Pads

Il s'agit d'une liaison série réalisée en technologie CMOS. Elle se présente sous la forme d'une macrocellule (incluant sérialiseur et désérialiseur) intégrable à la périphérie du circuit utilisateur. Son rôle est de sérialiser et de désérialiser des mots de 10 bits. Alimentée sous +3.3V, elle consomme une puissance de 0.5W pour un débit variant entre 0.65 et 1 Gbits/sec.
Cette macro-cellule n'impose pas un codage particulier pour les caractères transmis. En revanche, l'émetteur et le récepteur fonctionnent en mode plésiochrone : l'émetteur et le récepteur sont contrôlés par des horloges de fréquences très voisines.

1/ Présentation de la macro-cellule

Les macro-cellules sont regroupées par quatre et partagent deux blocs analogiques : un adaptateur d'impédance et une boucle à verrouillage de phase (PLL) utilisée comme générateur de signaux d'horloge multiphases.
La boucle PLL génère deux séries de 40 signaux d'horloge de même fréquence que l'horloge système. La première série M_CLK[0:39] est utilisée pour sérialiser les données en émission, la seconde E_CLK[0:39] pour sur-échantillonner le signal série à l'entrée du récepteur.
Le sérialiseur et le désérialiseur fonctionnent chacun avec leur propre horloge locale générée par un quartz.

2/ Fonctionnement du sérialiseur

La PLL comprend un oscillateur contrôlé en tension (VCO) composé de 20 retards variables, 2 diviseurs de fréquence, un comparateur de phase, un amplificateur et un filtre passe-bas. Le principe de fonctionnement de la PLL est d'asservir la phase d'un signal généré par l'oscillateur sur la phase d'un signal de référence.

L'oscillateur contrôlé en tension est un convertisseur tension/fréquence.
Pour obtenir une série d'horloges multiphases, on utilise comme oscillateur une série de retards variables à structure différentielle et mis en boucle. Les retards sont constitués d'une résistance contrôlée en tension Rctrl. Leur temps de traversée est ajusté pour mettre en phase l'horloge système et une des horloges générées par l'oscillateur.

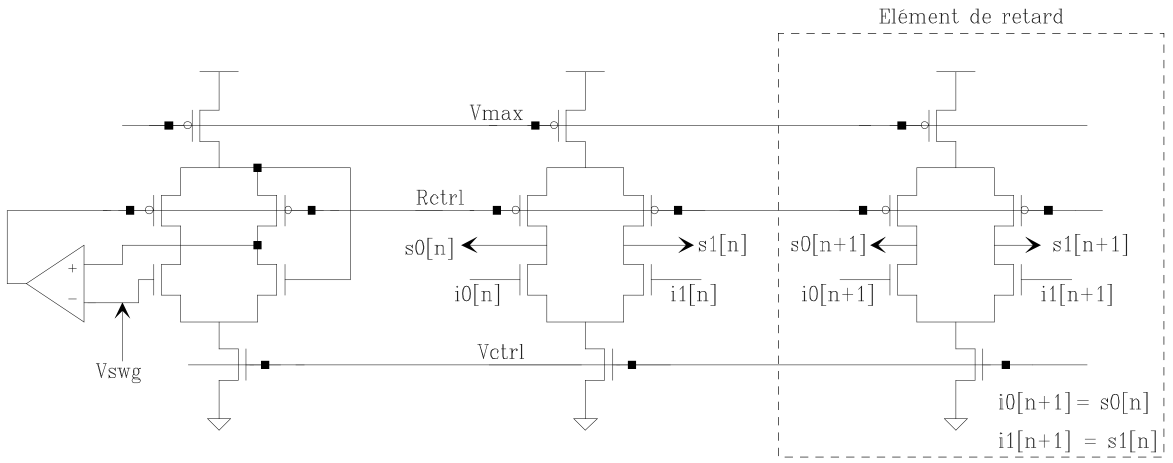
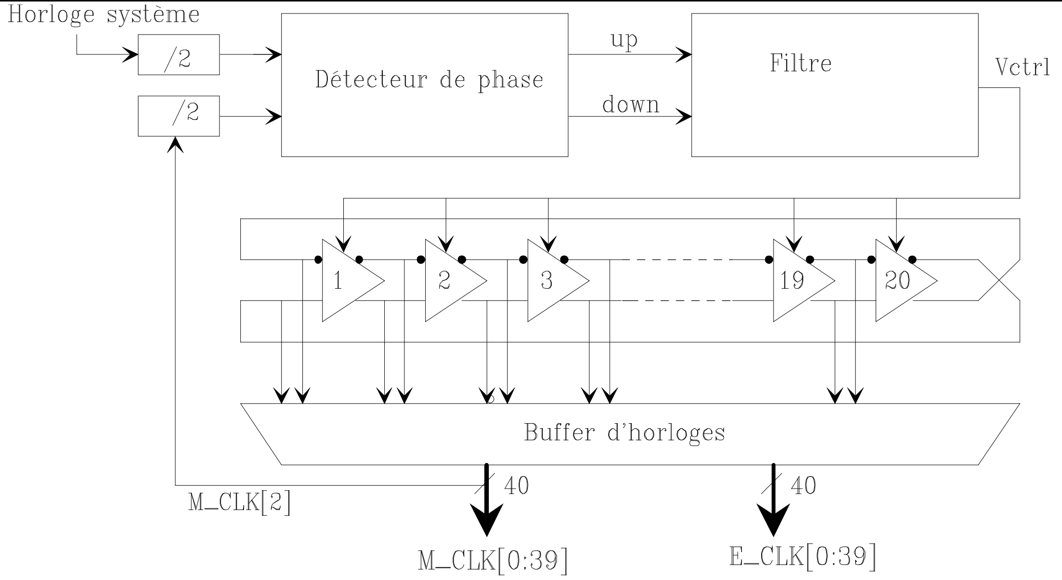


Schéma de l'élément de retard au niveau transistor

Le schéma de l'élément de retard est décrit ci-dessus.
Vctrl est la sortie du comparateur de phase; il permet de contrôler la durée des retards.
Vswg représente la valeur minimale pour les signaux internes s0 et s1. Vmax contrôle la valeur maximale pour ces mêmes signaux. L'ampleur du swing est définit pour un élément de retard fictif sur lequel on ajuste la valeur Rctrl. Cette valeur est ensuite propagée à tous les retards.
En ajustant Vswg on réalise un compromis entre la précision de la valeur du retard et la fréquence maximale que l'on peut atteindre.

Le fonctionnement de la PLL comprend une phase d'acquisition et une phase d'accrochage.
Pendant la phase d'acquisition, la PLL recherche dans le domaine de fréquence couvert par l'oscillateur celle qui correspond au signal de référence.
Pendant le phase d'accrochage, les signaux de référence et de l'oscillateur sont de même fréquence. Quand l'erreur de phase présente en sortie du comparateur de phase est constante, l'oscillateur est calé.



Boucle à verrouillage de phase (PLL)

Les données parallèles TX_DATA[0:9] sont synchronisées sur deux signaux d'horloge en opposition de phase M_CLK[2] (qui est en phase avec l'horloge système) et M_CLK[22]. Ainsi les valeurs Q[0:9] sont stables lorsqu'elles sont présentées en entrée du sérialiseur.

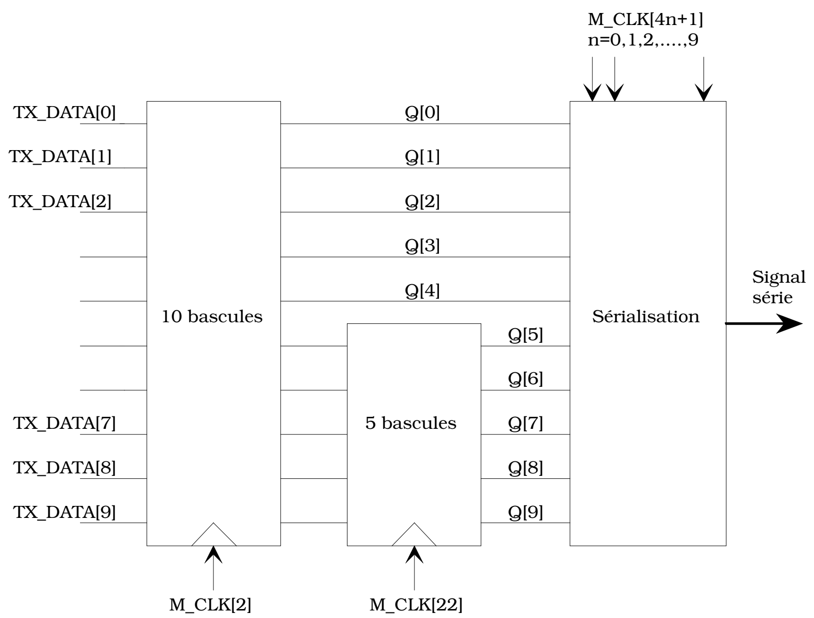
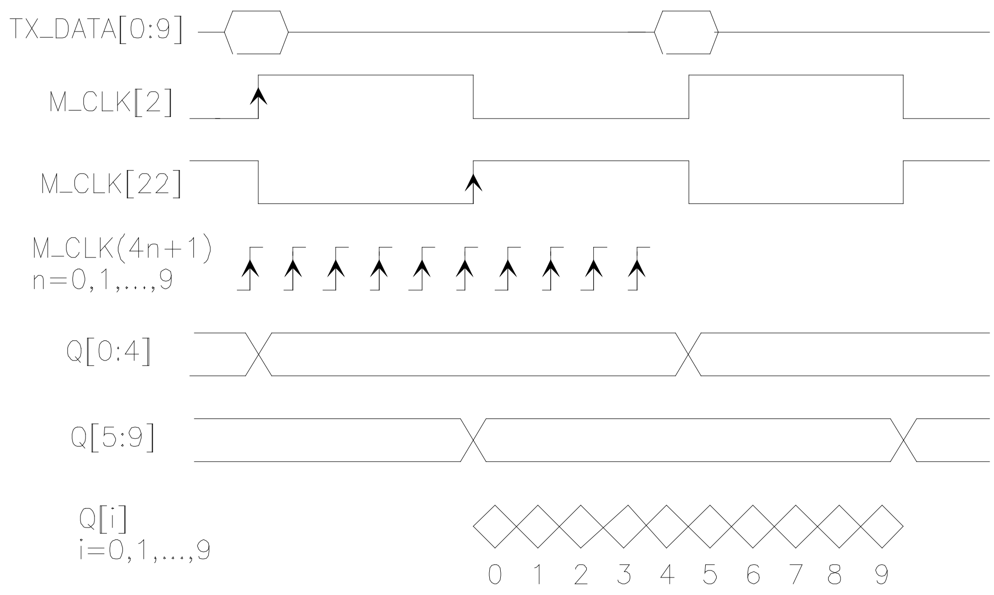
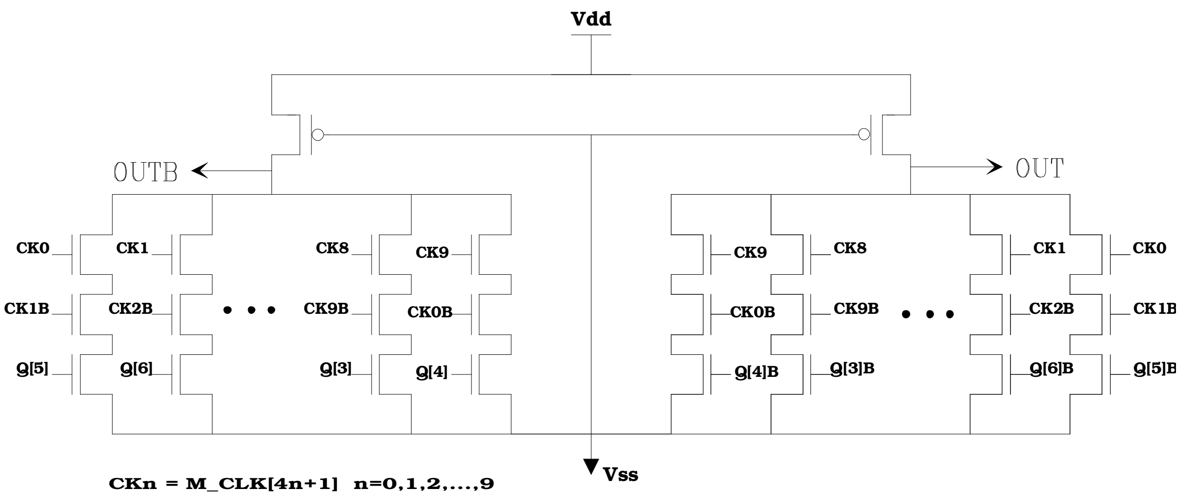


Schéma du sérialiseur

La sérialisation des données Q[0:9] s'effectue dans un premier temps en synchronisant chacune des données sur un des signaux d'horloge M_CLK[4n+1] où n=0,1,...,9. Ces signaux sont décalés en phase à la fréquence-bit que l'on souhaite obtenir en sortie du sérialiseur.



Chronogramme du sérialiseur



Réalisation du signal série

La sérialisation des données est réalisée par l'intermédiaire du schéma présenté ci-dessus. Le passage d'une donnée sur le signal série est synchronisé par deux signaux d'horloge consécutifs de telle façon que chaque donnée est transmise sur le signal série pendant un temps T égal à un dixième de la période de l'horloge système.

3/ Fonctionnement du désérialiseur

Contrairement à HS-Link, Hot Pads travaille en mode plésiochrone. L'émetteur et le récepteur fonctionnent chacun avec leur propre signal d'horloge et les fréquences de ces deux signaux doivent être égales à 10^{-4} près.
Le signal série représentant dix bits consécutifs est échantillonné par la série de 40 signaux d'horloge E_CLK[0:39] générés par la boucle PLL. Chaque bit est échantillonné quatre fois. Une logique de majorité est utilisée pour reconstituer les mots transmis à partir des valeurs échantillonnées.

Nous ne disposons pas de détails sur le procédé de désérialisation. En effet la description fonctionnelle de la liaison série Hot Pads présentée ici correspond à l'article proposé dans le IEEE Journal of Solid State Circuit (Vol. 30 n°4, Avril 1995) par K.Lee, S.Kim, G.Ahn et D-K.Jeong. Dans cet article, le principe de désérialisation décrit n'est pas conforme aux spécifications : il impose des contraintes sur le codage des données transmises.
Depuis, des améliorations ont été apportées à ce circuit pour le rendre totalement indépendant du codage. Ces améliorations n'ont pas fait l'objet d'une publication à ce jour.

4/ Analyse

Telle qu'elle est disponible aujourd'hui, la liaison série Hot Pads répond à toutes les spécifications que nous avons défini dans la problématique.
Hot Pads n'impose aucune restriction sur le codage des données. Cette absence de contrainte sur le codage est remplacée par une contrainte sur les fréquences respectives de l'émetteur et du récepteur qui doivent être égales à 10^{-4} près.

C/ Conclusion

HS-Link et Hot Pads représentent deux visions possibles d'une liaison série CMOS fonctionnant à 1 Gbits/sec.
HS-Link réalise une transmission asynchrone. Une contrainte sur le codage des données permet l'utilisation d'une large plage de fréquences de transmission.
Hot Pads réalise une transmission plésiochrone. L'émetteur et le récepteur fonctionnent avec des horloges de même fréquence à 10-4 près. Ce procédé permet de s'affranchir de toute contrainte sur le codage des données.

1/ Principes de sérialisation

Le procédé de sérialisation est basé sur la génération de séries de signaux d'horloge multiphases de basse fréquence. Pour générer ces signaux, HS-Link utilise une boucle DLL et Hot Pads une boucle PLL.
Le principe de la PLL est d'usage courant : la plupart des liaisons série l'emploient. Cependant, dans le contexte de génération d'horloges multiphases, le principe de la DLL utilisée dans HS-Link nous parait plus fiable et plus résistant.

- Pour une DLL, le déphasage entre les signaux CK1, CK2,...,CKn et le signal d'horloge de référence CK est réévalué à chaque période de CK. Ainsi l'erreur commise sur la valeur du retard ne s'accumule pas dans le temps comme avec une PLL où la différence de fréquence entre l'horloge de référence et l'horloge générée a un effet cumulatif sur le déphasage.

- Dans une DLL, la commande digitale du retard est mémorisée dans un registre. La boucle reste calibrée même si la transmission est temporairement interrompue.

- La DLL ayant une boucle de contrôle numérique est peu sensible au bruit et son comportement est régi par une simple équation linéaire. Au contraire, la boucle de contrôle de la PLL est analogique. Elle est sensible au bruit et nécessite un signal de référence de bonne qualité pour pouvoir accrocher.

Les techniques mises en oeuvre dans HS-Link et dans Hot Pads pour sérialiser les données sont très similaires et nous permettent d'atteindre nos objectifs. Ce n'est donc pas sur la sérialisation que vont porter nos efforts.

2/ Principes de désérialisation

C'est au niveau du désérialiseur que se situe la principale différence entre HS-Link et Hot Pads :

- Pour le premier, on extrait le signal d'horloge du signal série et on l'utilise en entrée de la boucle DLL pour générer des séries de signaux d'horloge basse fréquence. Chaque série de 12 signaux d'horloge échantillonne 12 bits consécutifs du signal série tels que ces bits forment un mot complet.

- Pour le second, on ne peut pas utiliser le signal d'horloge transmis pour échantillonner le signal série. On génère des séries de signaux d'échantillonnage basse fréquence à partir du signal d'horloge local. On obtient une série de valeurs que l'on tente d'associer entre elles à l'aide d'un algorithme pour retrouver la valeur des bits transmis.
  La précision des quartz est de l'ordre de 10^{-4} ce qui signifie que le décalage de phase entre les deux horloges va évoluer dans le temps et qu'il faudra compenser en temps réel cette évolution. C'est le mode plésiochrone.

3/ Conclusion

La liaison série HS-Link ne satisfait pas nos spécifications puisqu'elle impose un codage spécifique des données. Toutefois elle nous apporte une technique de sérialisation très intéressante fondée sur la boucle à verrouillage de retard (DLL).

La liaison série Hot Pads satisfait nos spécifications : elle est disponible sous la forme d’une macro-cellule CMOS de faible consommation et n’impose aucune contrainte sur le codage des données.
Malheureusement, le mécanisme de désérialisation n’est pas publié. De plus, plusieurs choix techniques nous paraissent discutables :

- L'utilisation d'une boucle à verrouillage de phase (PLL) pour générer la série de signaux d'horloge décalés en phase est une source d'imprécision qui risque d'entraîner de nombreuses erreurs lors de la transmission : le fonctionnement d’une boucle PLL est très sensible au bruit et à l’environnement.

- De plus, la réalisation de la PLL est d’une grande complexité technologique. Elle nécessite la mise au point d’un retard variable dont la valeur atteint 250 ps avec une précision de l’ordre de quelques ps.

- Le principe de récupération du signal série par sur-échantillonnage semble également être une source d'erreurs potentielles puisque chaque bit n'est échantillonné que quatre fois.

Notre étude vise l’obtention des mêmes fonctionnalités que Hot Pads mais utilise pour cela des techniques différentes. Nous espérons obtenir une plus grande fiabilité pour la macro-cellule.

Chapitre III : Principe de la solution

Ce chapitre met en évidence les principes du mécanisme de désérialisation et de récupération d’horloge que nous proposons pour une liaison série sans contrainte de codage.

La partie sérialisation n'est pas détaillée ci-dessous car elle ne comporte pas d'innovation par rapport au principe mis en oeuvre dans la liaison HS Link.

Par contre, la partie désérialisation fait l'objet d'une étude approfondie.
Le problème posé est le suivant : on cherche à réaliser un récepteur qui puisse échantillonner n’importe quel signal série et en extraire les mots émis sans imposer aucune contrainte sur le codage. On accepte en revanche la contrainte d’une communication plésiochrone : les fréquences des horloges de transmission et de réception ne doivent pas différer de plus de 10^{-4}.

Dans le processus de désérialisation du signal série, on distinguera deux difficultés :

A/ la récupération des bits par un échantillonnage approprié du signal série reçu,

B/ la reconstruction des mots par assemblage des bits dans leur disposition originale.

A/ Récupération des bits

La récupération des bits transmis en série passe essentiellement par un échantillonnage efficace du signal série. Nous avons choisi d'échantillonner chaque bit une fois seulement quand sa valeur logique est stable. Le problème de l'échantillonnage se ramène donc à la réalisation (à partir de l'horloge locale et du signal série) de un ou plusieurs signaux d'horloge alignés sur le signal série en fréquence et en phase.

1/ Méthode d'échantillonnage du signal série

Soit CKref1 le signal d'horloge de fréquence F1 local de l'émetteur.
Soit CKref2 le signal d'horloge de fréquence F2 local du récepteur tel que

$F2 = F1$ à 10^{-4} près

Soit N le nombre de bits compris dans un mot.

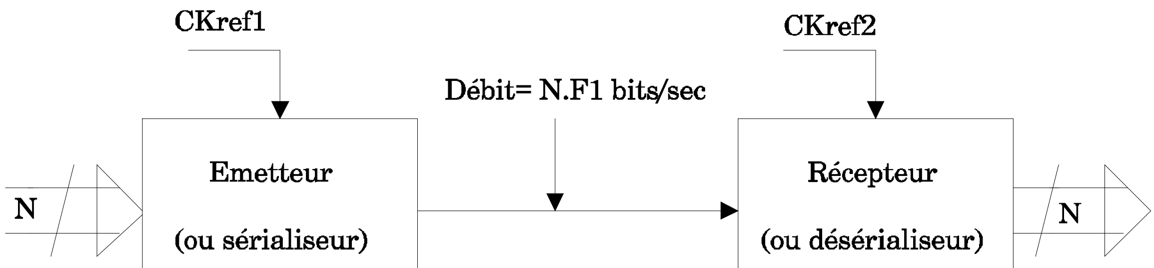


Schéma d'un lien série

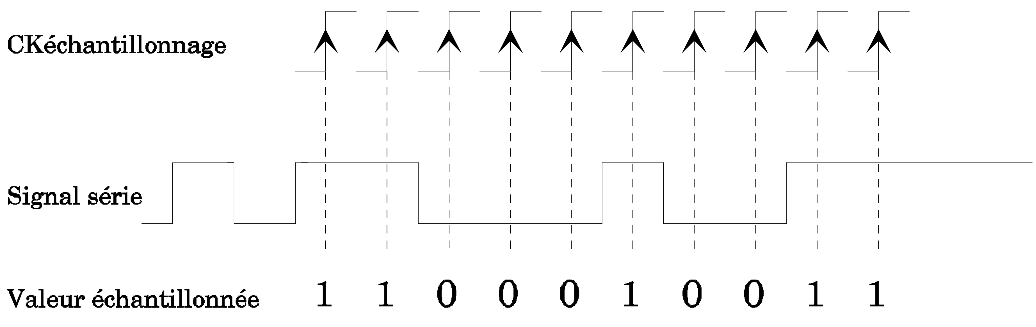
Le signal reçu par le désérialiseur consiste en une suite de niveaux logiques 0 ou 1 de durée L.T où

L est un entier non nul
T est la durée moyenne d'un bit $T = 1/N.F1$.

Il faut donc disposer d'un signal d'horloge d'échantillonnage de fréquence égale au débit N.F1 du signal série.
Une fois calé en phase, ce signal pourra échantillonner tous les bits successivement.

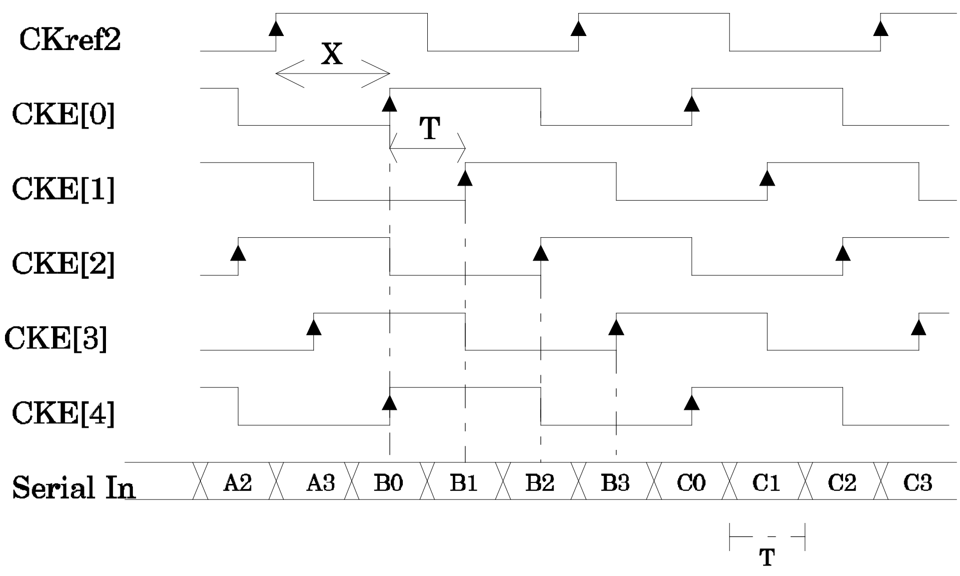
Les débits visés par cette technologie sont de l'ordre du Gbits/sec. Un signal d'horloge de fréquence ~1 GHz est difficile à asservir en fréquence et en phase.

A la place d'un signal d'horloge unique à très haute fréquence, on a choisi d'utiliser N signaux d'horloge de fréquence F1 déphasés de T qui échantillonnent chacun un bit tous les N bits.



Exemple d'échantillonnage du signal série

Nous réalisons tout d'abord l'ajustement en phase des signaux d'échantillonnage : le signal CKref2 est décalé d'une valeur X de manière à obtenir un signal d'horloge CKE[0] placé au milieu d'un bit quelconque du signal série.
Nous réalisons ensuite l'ajustement en fréquence des signaux d'échantillonnage : le signal d'horloge CKE[0] est à son tour décalé N fois d'une valeur T pour générer les signaux CKE[1:N].
La détermination des retards variables X et T permet donc de réaliser un échantillonnage satisfaisant du signal série.



Génération des signaux d'échantillonnage CKE[0:N] pour N=4

2/ Détermination de la valeur du retard T

Par principe CKE[N] et CKE[0] sont en phase et échantillonnent les mêmes bits. On va utiliser cette propriété pour déterminer la valeur du retard T entre deux signaux d'horloge.

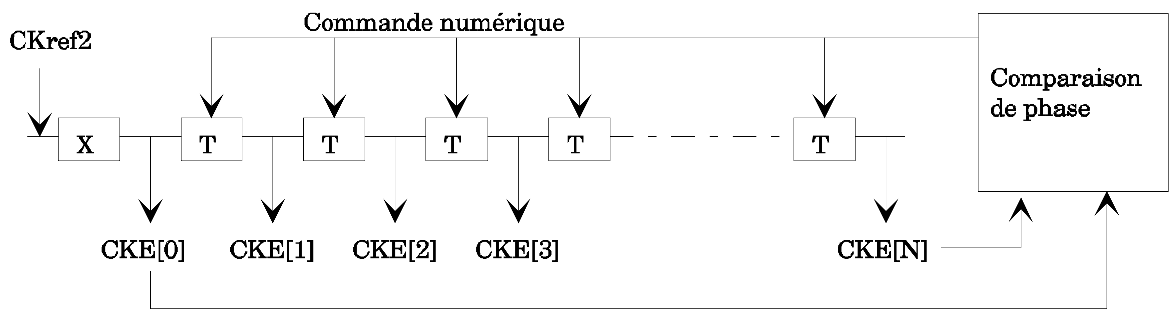


Schéma de principe de la boucle à verrouillage de retards

En décalant N fois CKE[0] d'un temps T, on génère N signaux d'horloge CKE[1:N] de même fréquence F2 qui viendront échantillonner N bits successifs. La valeur de T est déterminée en comparant les phases de CKE[0] et de CKE[N].
On utilise ici le principe de Delay Locked Loop (DLL).

La valeur de T est déterminée à partir de CKref2. Elle correspond à la durée d'un bit avec une erreur de 10^{-4}. Donc, si CKE[0] est parfaitement centré sur le bit B0 et pour N=10, CKE[10] sera décalé par rapport au centre du bit B10 de $10 \times 10^{-4} = 10^{-3}$ période-bit. Cette erreur est totalement négligeable par rapport aux jitters inhérents au système et étudiés plus loin.

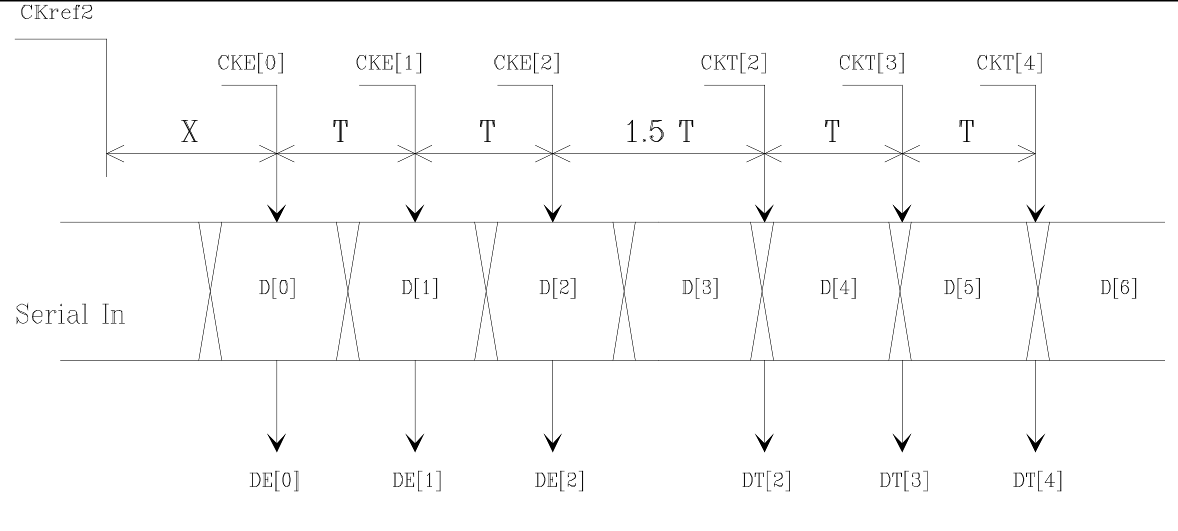
3/ Détermination de la valeur du retard X

3.1/ Détermination du déphasage permanent

Supposons dans un premier temps que l'émetteur et le récepteur fonctionnent avec des horloges CKref1 et CKref2 de même fréquence (F1=F2).

Dans ce cas la valeur de X compense le décalage de phase introduit par le temps de propagation du signal série sur le lien physique. Cette valeur est constante.
La valeur de X est déterminée par la recherche des transitions sur le signal série.

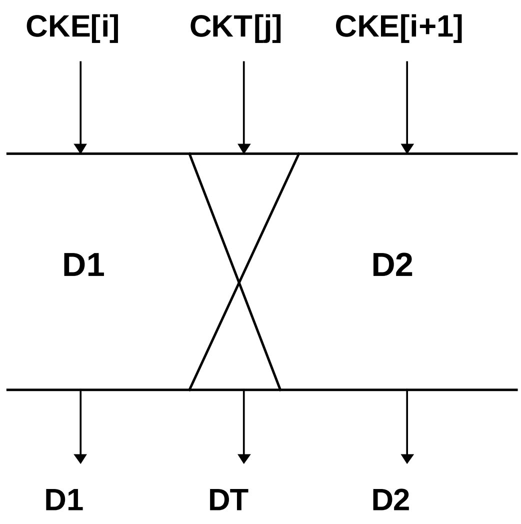
Pour repérer les transitions on génère une deuxième série de signaux d'horloge CKT[0:N] décalée d'une durée de 1.5 T par rapport à la première. Cette série de signaux CKT[0:N] est supposée échantillonner le signal série au moment des transitions entre les bits si la valeur du retard X est correcte.



Echantillonnage du signal série

Echantillonner un signal instable est une source de métastabilité potentielle. Pour éviter de propager cette métastabilité, nous avons mis au point une bascule spécifique qui garantit la stabilisation de la valeur échantillonnée en deux cycles d’horloge au maximum.

Pour déterminer la valeur de X, on repère chaque transition du signal série et l'on compare les trois valeurs échantillonnées autour de cette transition.



Une transition est détectée quand D1 est différent de D2.

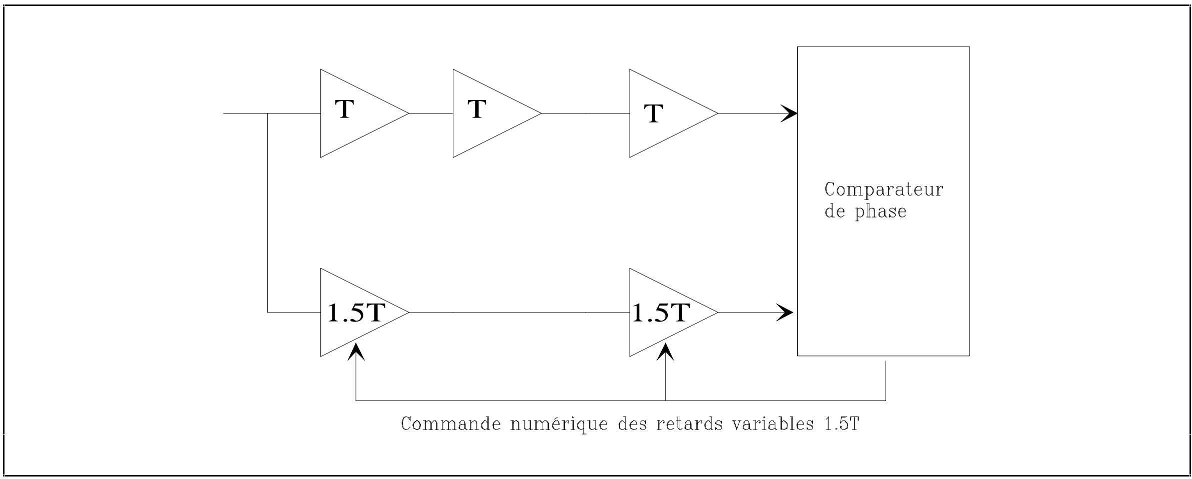
Pour chaque transition, on analyse la valeur de DT.

Si DT=D1,
la valeur du retard X doit augmenter

Si DT=D2,
la valeur du retard X doit diminuer

Par ajustements successifs, la valeur du retard X telle que les bits soient échantillonnés en leur milieu est atteinte.

Pour réaliser la série de signaux d’horloge CKT[0:N] destinée à échantillonner le signal série au moment des transitions, il faut disposer d’un retard 1.5T. On définit ce retard en remarquant que 2 x 1.5T = 3T.



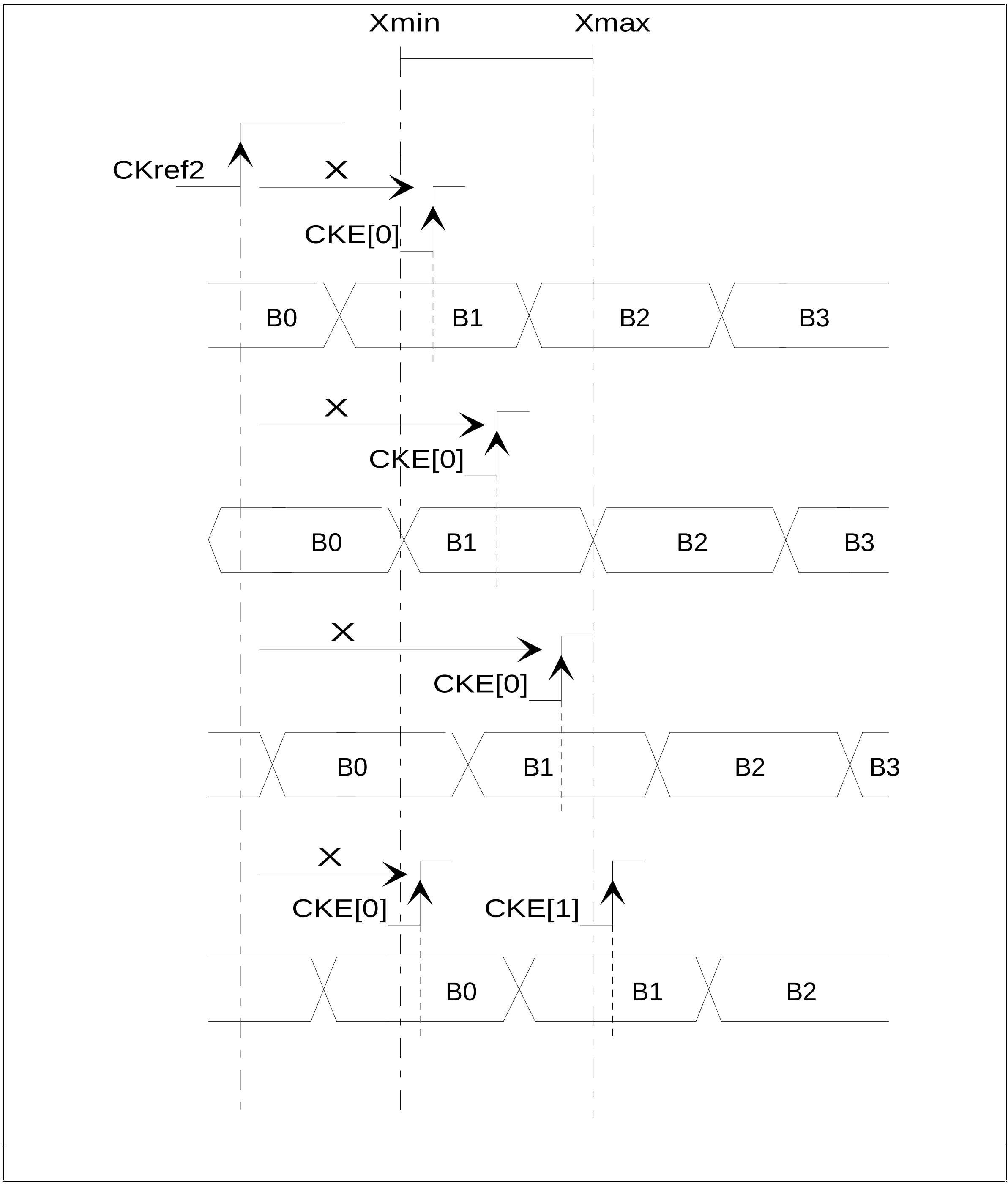
3.2/ Détermination du déphasage en mode plésiochrone

Revenons maintenant sur notre hypothèse F1=F2.
F1 et F2 sont les fréquences de CKref1 et CKref2 respectivement signaux d'horloge de l'émetteur et du récepteur. Ces deux signaux sont générés par deux quartz indépendants. Or la précision des quartz est généralement de l'ordre de 10^{-4}. On dit alors que l'émetteur et le récepteur fonctionnent en mode plésiochrone : leurs signaux d'horloge sont de même fréquence à la précision des quartz près.
Ces deux fréquences sont suffisamment proches pour ne pas perturber le principe de fonctionnement décrit ci-dessus.

L’imprécision sur la valeur de la fréquence implique que le déphasage entre CKref2 et le signal série n'est pas constant donc la valeur du retard X n'est pas constante. C'est une fonction soit croissante soit décroissante du temps. En principe la valeur de X doit donc croître ou décroître à l'infini.
Cette difficulté est résolue en définissant la valeur de X modulo T.

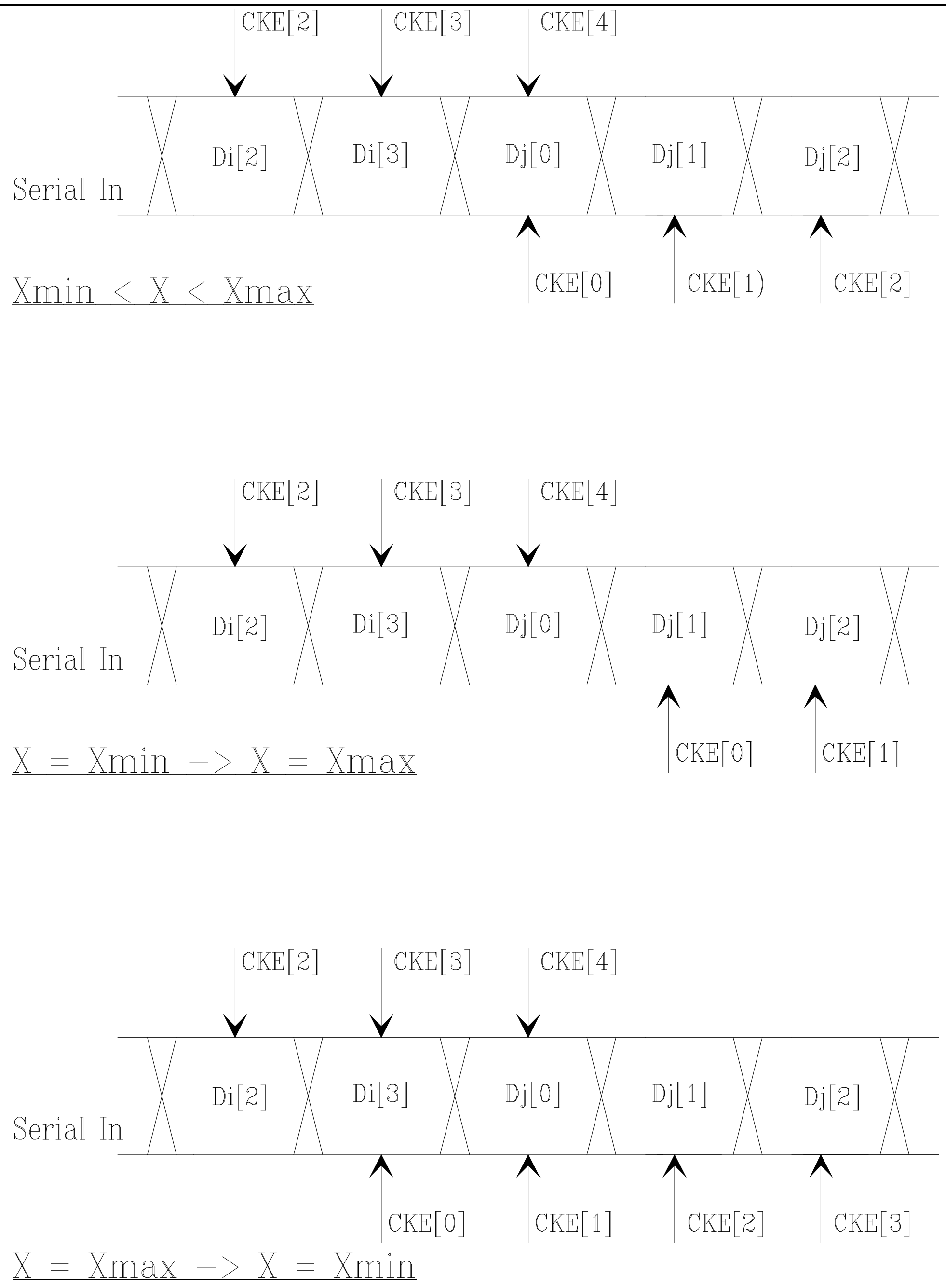
On pose $X\min \leq X \leq X\max$ tels que Xmax = Xmin + T

Quand la valeur de X atteint l'extrémité Xmin (respectivement Xmax) et que la recherche des transitions montre que X doit décroître (respectivement croître), on affecte à X la valeur Xmax (respectivement Xmin) et tous les signaux qui s'en déduisent CKE[0:N] et CKT[0:N] sont décalés de T.



Le chronogramme ci-dessus illustre le décalage progressif du signal série par rapport à CKref2. La valeur du retard X augmente pour compenser ce décalage jusqu'à atteindre sa valeur maximale Xmax.

Quand la valeur du retard X passe de Xmax à Xmin, les signaux d'échantillonnage CKE[0:N] sont décalés d'un bit. Ce phénomène est explicité à travers le schéma suivant.



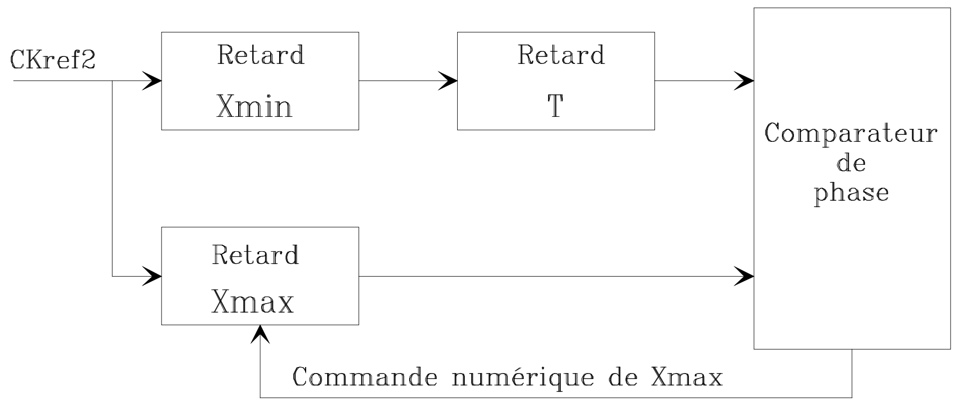
Effets des sauts de la valeur de X pour N=4

Quand la valeur de X est comprise strictement entre Xmin et Xmax, l’échantillonnage est redondant. Le front de CKE[N] et le front du CKE[0] suivant échantillonnent le même bit.

Quand la valeur du retard passe de Xmin à Xmax, les bits échantillonnés par le front de CKE[N] et le front de CKE[0] suivant ne sont plus les mêmes. On constate ici la nécessité de disposer de N+1 signaux d’horloge pour échantillonner le signal série.
Quand la valeur du retard passe de Xmax à Xmin, l’échantillonnage est deux fois redondant. Le bit échantillonné par le front de CKE[N-1] l’est aussi par le front du CKE[0] suivant. Le bit échantillonné par le front de CKE[N] l’est aussi par le front du CKE[1] suivant.

3.3/ Calcul de Xmin et Xmax

On définit la valeur de Xmin a priori. Xmin correspond à la valeur minimale du retard variable. On applique au second retard la commande correspondant à un retard de durée T. Le comparateur de phase permet alors de déterminer la commande numérique correspondant à un retard de durée Xmax.



<u>Boucle de commande de la valeur de Xmax</u>

<u>4/ Conclusion</u>

La méthode proposée garantit les propriétés suivantes:

- Aucune contrainte n'est imposée sur le codage des données.
- Chaque bit est échantillonné quand sa valeur logique est la plus stable.
- Chaque bit est échantillonné au moins une fois.

Cette méthode nécessite une phase de mise en route. Il faut un certain temps pour que les retards T, 1.5T et X prennent leurs valeurs respectives. Pendant cette période dite phase de calibration des retards, le signal série n'est pas échantillonné correctement.
Lors de l'utilisation de la liaison série, il faut donc prévoir un protocole d'initialisation pour envoyer un signal série pendant la phase de calibration des retards sans tenir compte des données obtenues en sortie du récepteur.

Le schéma suivant présente les deux boucles DLL utilisées pour génerer les horloges d'échantillonnage du signal série ainsi que le contrôle du retard 1.5T.

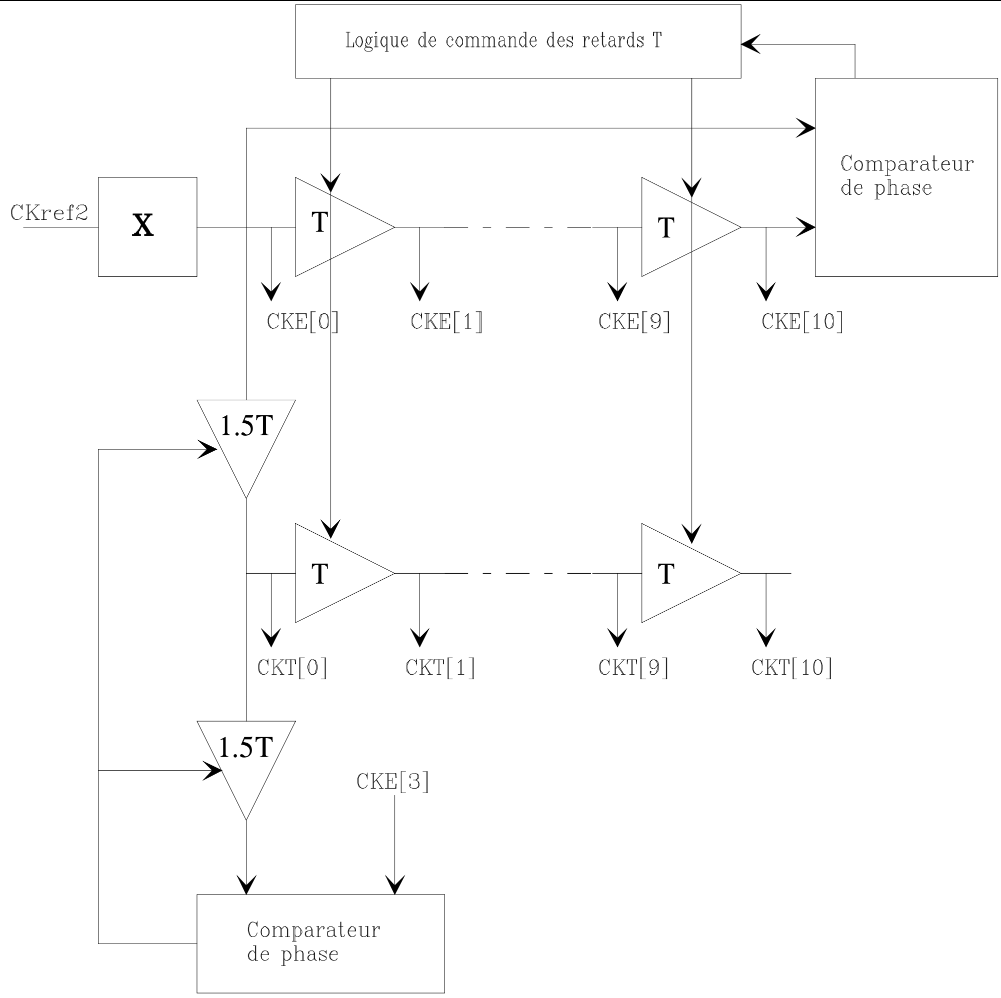


Schéma récapitulatif du générateur d'horloges multiphases pour N=10

avec les boucles de commandes des retards variables T et 1.5T

B/ Reconstruction des mots

1/ Méthode d'identification des limites des mots

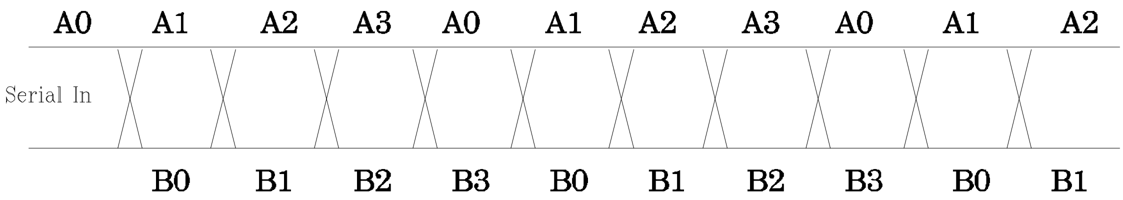
On dispose à présent de N+1 valeurs de bits qu'il faut placer correctement dans le flot de données quand la valeur du retard X passe de Xmax à Xmin ou inversement.
A partir de cette série de valeurs logiques on cherche à retrouver les mots présentés en entrée de l'émetteur.

Cette partie est délicate puisque l’on ne veut pas imposer de contrainte sur le codage des données. Il faut pourtant disposer d’un moyen pour déterminer les limites entre les mots. Nous avons choisi de définir un mot spécifique de référence MOT_REF codé sur N bits. Ce mot est choisi par l'utilisateur en fonction de ses besoins et pour s'adapter au protocole utilisé. Toutefois le choix de MOT_REF par rapport aux codes licites doit être fait de telle façon que ce mot ne peut se retrouver par chevauchement sur deux codes licites.

Au début de chaque transmission la logique du circuit regroupe arbitrairement les bits extraits en séquences de N bits consécutifs. Puis on compare chaque séquence de N bits au mot spécifique MOT_REF.

- Si la séquence de N bits choisie arbitrairement ne correspond pas à MOT_REF, il faut décaler l’assemblage des bits de 1 bit et reprendre les comparaisons ceci jusqu’à l’identification de MOT_REF dans le signal série reçu. Ce processus doit aboutir après N essais au plus.

- Lorsque plusieurs séquences consécutives de N bits correspondent à MOT_REF, on cesse ce processus de recherche d’alignement : les bits sont alors correctement regroupés.



Exemple de décalage de l’assemblage des bits en mots pour des mots de 4 bits
On passe de l’assemblage de type A à celui de type B

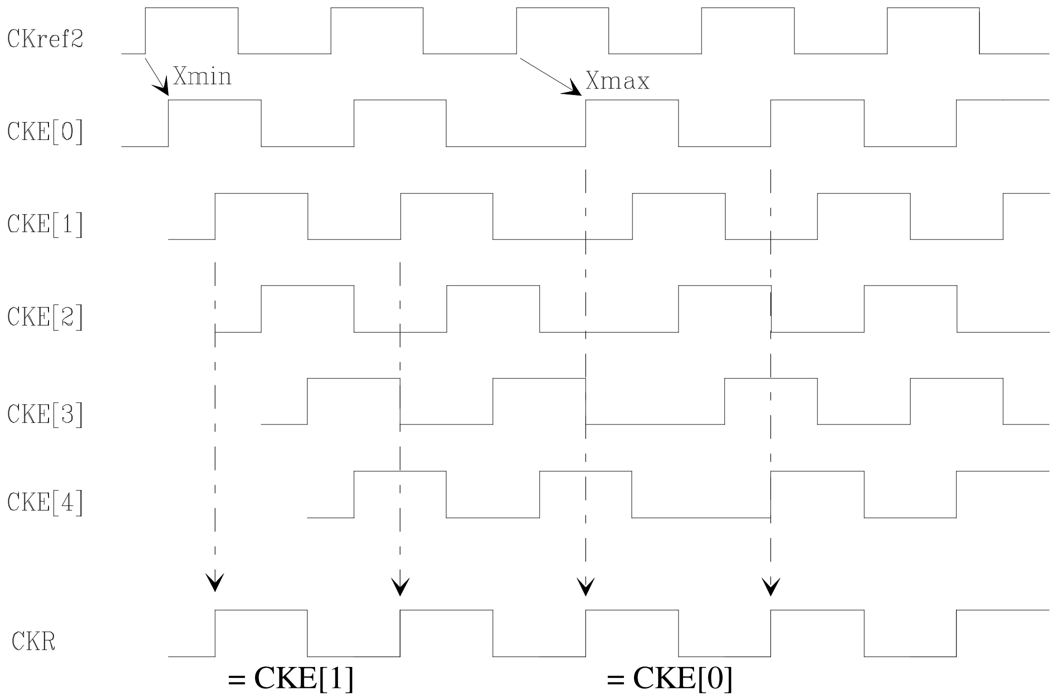
Une fois que MOT_REF a été identifié, on considère que la liaison série est prête à transmettre les données véritables. Les données sont restituées en maintenant le même regroupement des bits qui a permis d’identifier MOT_REF. Ainsi aucun codage spécifique n’est imposé aux données.

2/ Récupération de l’horloge transmise

Pour permettre l’échantillonnage des mots ainsi reconstruits, on cherche à fabriquer un signal d’horloge de fréquence F1 c’est-à-dire à récupérer le signal d’horloge transmis avec le signal série.

Si l’on utilise comme horloge de sortie du désérialiseur le signal d’horloge local CKref2, la différence en fréquence entre F2 et F1 conduira soit à une accumulation de mots non lus (F1>F2), soit à une déficience de mots à lire (F1<F2).
Utiliser le signal d’horloge récupéré permet de fournir en sortie du désérialiseur les mots à la fréquence F1. Les débits d’entrée et de sortie du désérialiseur sont les mêmes.

Dans ce but, on utilise les signaux d’horloge CKE[0:N] qui servent déjà à échantillonner le signal série. Soit un de ces signaux CKE[i], il est toujours en phase avec le signal série. Cependant il n’a pas une période régulière car de temps en temps le retard X va atteindre une de ces extrémités et sauter à l’autre. Dans cette situation, le signal d’horloge CKE[i] a pour un cycle une période de (N-1) x T ou (N+1) x T au lieu de N x T. Ce signal n’a pas une fréquence stable; il ne peut pas être utilisé comme signal d’horloge récupéré.

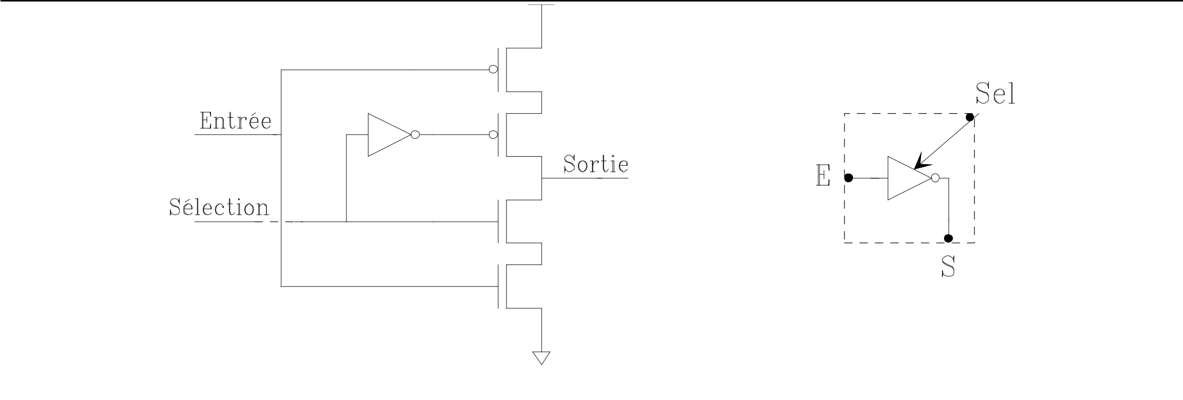


Exemple de détermination de CKR pour N=4 et un passage de Xmin à Xmax

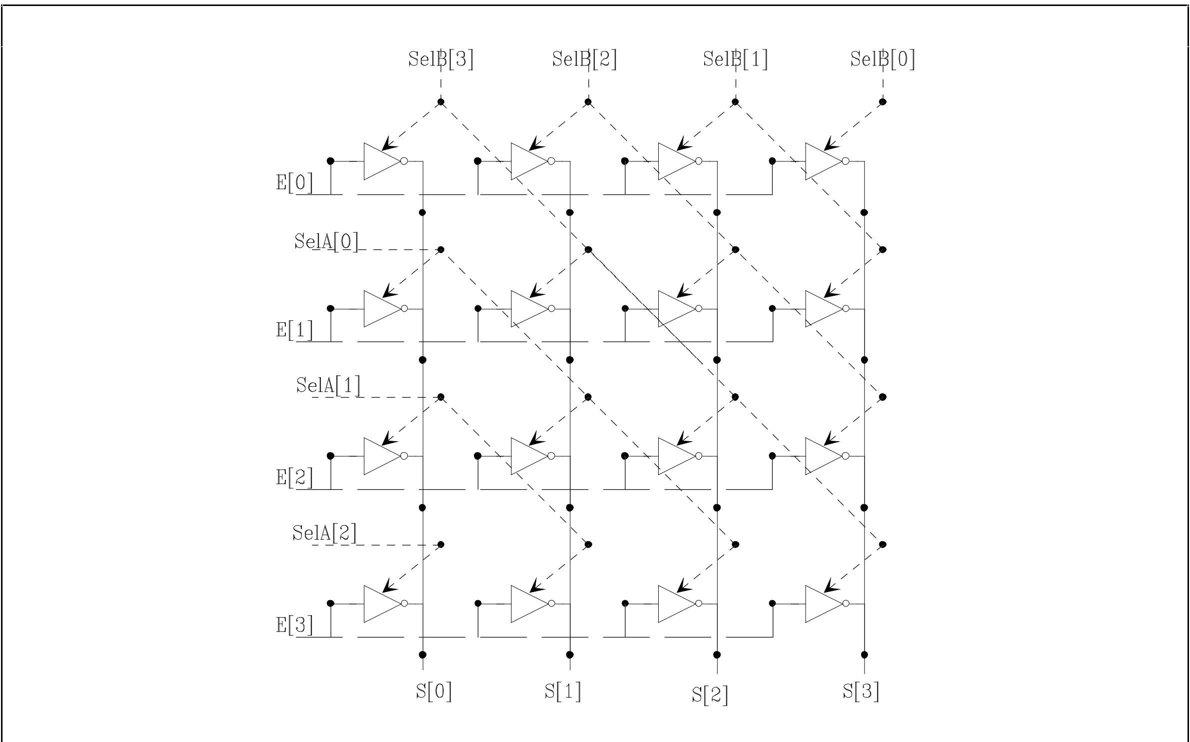
Par contre, si au moment du changement de valeur du retard X de Xmin à Xmax (ou de Xmax à Xmin) on choisit comme signal d’horloge récupéré CKE[i-1] (respectivement CKE[i+1]) au lieu de CKE[i], on obtient un signal CKR de fréquence F1.

3/ Assemblage des bits en mots

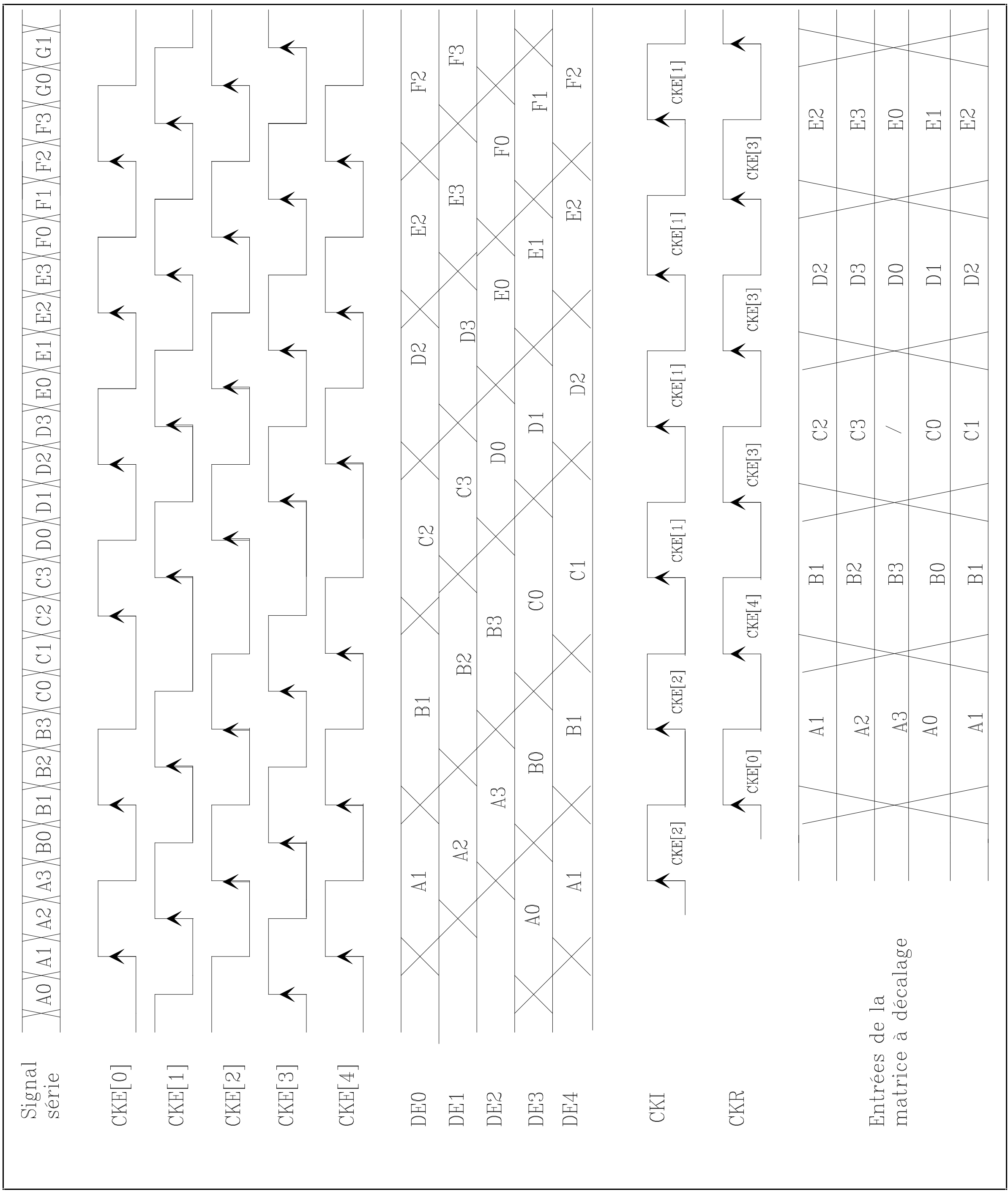
La matrice à décalage (‘barrel-shifter’) permet de réorganiser les bits dans le bon ordre et d’éliminer les bits redondants. L’ordre des bits présentés en entrée de la matrice évolue en fonction du choix de CKR parmi les CKE[0:10].
L’élément de base de la matrice à décalage est un inverseur commandé du type :



Ces éléments assemblés forment une matrice à A entrées et B sorties.
Ci-dessous, un exemple de matrice à décalage pour A=B=4



En entrée de la matrice on présente un bus de 4 bits synchrones E[0:3]. Le mot que l'on veut obtenir en sortie est une combinaison de ces quatre bits par décalage. A chaque combinaison correspond une commande pour la matrice à décalage de manière à orienter la traversée des bits dans la matrice.



Dans le cas où N=4,

<u>le chronogramme ci-dessus illustre les conséquences d'un saut de la valeur du retard X de Xmin à Xmax sur le choix de CKR et sur l'utilisation de la matrice à décalage</u>

Le choix de CKR découpe la série de bits en groupes formant un seul mot. La matrice à décalage réorganise les bits dans le bon ordre en fonction de la valeur de CKR.
La matrice à décalage prend en entrée les N+1 bits échantillonnés par CKE[0:10] puis synchronisés sur CKR.

Dans l'exemple du chronogramme précédent, le signal série est constitué de mots de 4 bits A[0:3], B[0:3], C[0:3], D[0:3], etc.
Le signal série est échantillonné par 5 signaux CKE[0:4].

Au niveau de la troisième série de signaux CKE, la valeur du retard X passe de Xmin à Xmax. Les signaux CKE sont décalés de la durée d'un bit vers la droite.
Soit DE[i] la valeur du signal série échantillonnée par CKE[i].

On choisit CKI tel que CKI échantillonne les bits 0 et 1 de chaque mot.

CKI est le signal inverse de CKR. CKI sert à pré-échantillonner les N/2 premiers bits de chaque mot.

Lors du saut de la valeur du retard X, le choix de CKI parmi les CKE[0:4] passe de CKE[2] à CKE[1].

On choisit CKR tel que CKR échantillonne les bits 2 et 3 de chaque mot ainsi que les bits pré-échantillonnés par CKI.

Le choix de CKI et CKR se fait en cherchant à identifier MOT_REF en sortie de la matrice à décalage. Tant que ce mot n'est pas reconnu, CKR et CKI parcourent toutes les valeurs possibles de CKE.

Lors du saut de la valeur du retard X, le choix de CKR parmi les CKE[0:4] passe de CKE[0] à CKE[3]. On voit apparaître dans ce cas un cycle de transition pendant lequel CKR = CKE[4]. Il s'agit du cycle pendant lequel CKE[4] et le CKE[0] suivant ne sont pas en phase à cause du saut de X. Cette transition permet de préserver la régularité de CKR.

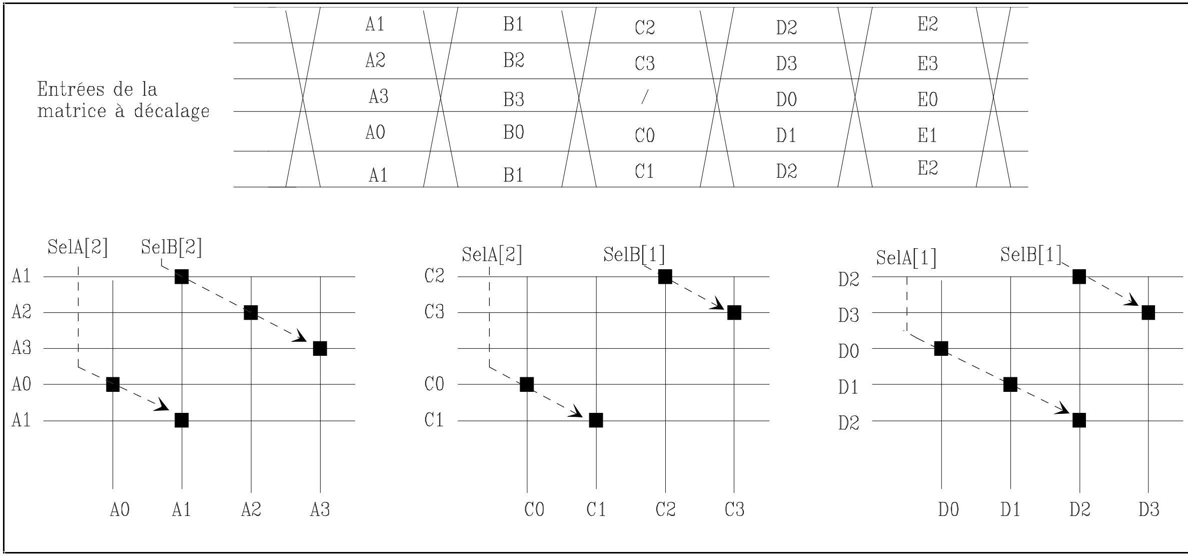
CKR échantillonne les N/2 derniers bits de chaque mot ainsi que les bits pré-échantillonnés par CKI.

On obtient en entrée de la matrice à décalage les 4 bits de chaque mot synchronisés sur le front montant de CKR.
Le choix de CKR détermine l'ordre dans lequel sont présentés les bits en entrée de la matrice à décalage. Quand le retard passe de Xmax à Xmin ou inversement le choix de CKR est décalé en conséquence ainsi que les commandes de la matrice à décalage.

CKR = CKE[0]	S[0:3] = {E[3], E[0], E[1], E[2]}	SelA[2] = 1 et SelB[2] = 1
CKR = CKE[1]	S[0:3] = {E[0], E[1], E[2], E[3]}	SelB[3] = 1
CKR = CKE[2]	S[0:3] = {E[1], E[2], E[3], E[0]}	SelA[0] = 1 et SelB[0] = 1
CKR = CKE[3]	S[0:3] = {E[2], E[3], E[0], E[1]}	SelA[1] = 1 et SelB[1] = 1

Les commandes de la matrice à décalage sont fixées pour un choix donné de CKR et on vérifie SelA[i] = SelB[i]. Cependant, lors d'un saut de la valeur du retard X, les valeurs de SelA et SelB ne changent pas systématiquement au même cycle de CKR ce qui engendre la nécessité de les différencier.

Reprenons l'exemple d'un saut de la valeur du retard X de Xmin à Xmax décrit précédemment pour un mot de 4 bits.



Pour le mot C, SelA[2] = 1 et SelB[1] = 1. La matrice à décalage sert également au niveau des sauts de la valeur du retard X à gérer les états transitoires.

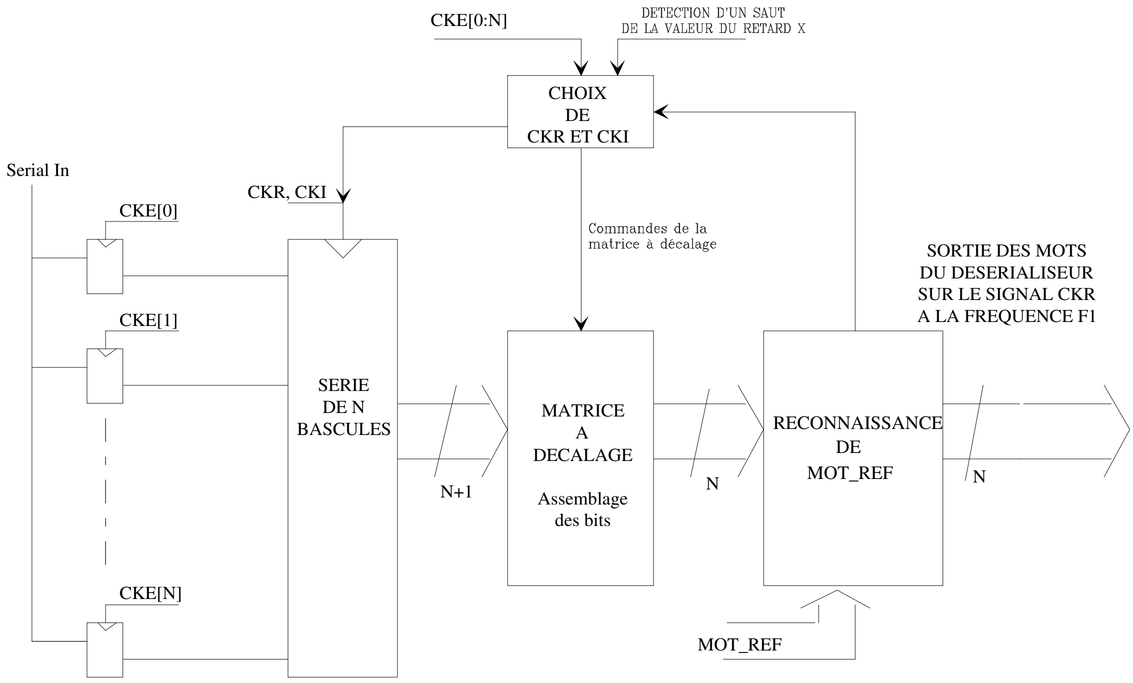


Schéma général du traitement des données dans le désérialiseur

Identification de MOT_REF

A la sortie de la matrice, les mots sont comparés à MOT_REF.

- Si la reconnaissance se fait sur dix mots consécutifs, un signal déclare à l'utilisateur que le récepteur est calibré sur le signal série.

- Si MOT_REF n'est pas détecté parmi les mots reçus, on force le décalage du signal d'horloge récupéré CKR de 1. Ainsi on parcourt tous les regroupements possibles de N bits.

C/ Conclusion

Le principe de désérialisation présenté dans ce chapitre répond au principal critère imposé de notre étude : aucune restriction n'est imposée au codage des données pour assurer une bonne réception du signal.

Par contre, on constate la nécessité d'un protocole d'initialisation de la liaison série. Cette mise en place se fait en deux temps :

- Phase de calibration des retards T, 1.5T et X : le signal série reçu doit comporter un grand nombre de transitions pour faciliter l'ajustement de la valeur du retard X.

- Phase d'identification des mots : le signal reçu doit inclure de façon régulière le mot de référence MOT_REF pour permettre au récepteur d'identifier les limites des mots.

Notons que la sécurité de la transmission est gérée par l'utilisateur : c'est à l'utilisateur de surveiller le signal de calibration et de relancer un processus de calibration si nécessaire.

Chapitre IV : Circuits critiques

Nous présentons dans ce chapitre divers éléments originaux de la liaison série qui ont donné lieu à une étude approfondie. Il s'agit :

- du retard variable
- du comparateur de phase
- de la boucle à verrouillage de retard
- de l'arbre de OU-exclusifs
- de la bascule d'échantillonnage du signal série
- du circuit d'adaptation d'impédance

Ces éléments ont fait l'objet d'analyses et de simulations temporelles avec les outils VTIsim de COMPASS et SPICE. Ces cellules ont été dessinées et optimisées pour une technologie CMOS 0.5µ.

A/ Le retard variable

Le retard variable à commande numérique est l'élément de base de la boucle à verrouillage de retards (DLL).

Il apparaît dans la macrocellule liaison série sous deux formes. La première correspond au retard inclus dans la DLL et permettant d'asservir le temps T. La deuxième correspond aux autres retard qui apparaissent dans la macrocellule. Il s'agit du retard 1.5T qui réalise les signaux d'horloge destinés à échantillonner les transitions du signal série. Il s'agit également du retard X qui permet l'ajustement en phase des signaux d'échantillonnage des données par rapport au signal série.

Le retard variable est conçu pour avoir une variation de délai strictement monotone et aussi linéaire que possible en fonction de la commande numérique qu'on lui applique.
Cet élément a été mis au point par A.Cofler, M.Combes, JC.Lebihan, R.Marbot et R.Nezamzadeh. Il fait l'objet du brevet 'circuit à retard variable' sous la référence 92.03527, mars 1992.

1/ Interpolation de deux signaux

Soit un retard de valeur fixe constitué par deux inverseurs en série, on réalise l'interpolation de courant entre les signaux A et B par l'intermédiaire d'une commande numérique.

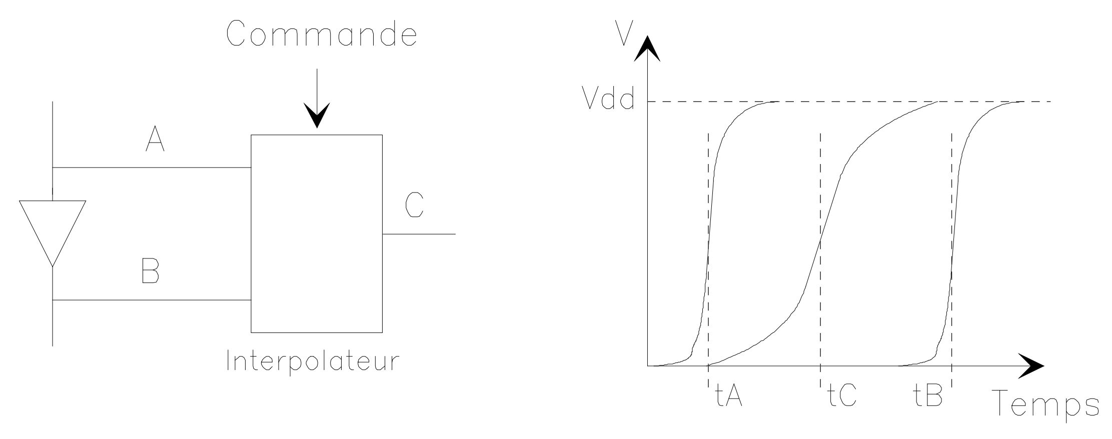


Schéma de principe du retard variable

La différence t_B - t_A correspond au retard fixe. C'est une caractéristique de la technologie employée.
Le signal C possède une pente plus faible que celle des signaux A et B et le temps t_C prend un certain nombre de valeurs comprises entre t_A et t_B, suivant la valeur de la commande numérique.

L'interpolateur reçoit une commande numérique sur 3 bits. En fonction de cette commande, les courants respectifs des branches A et B sont contrôlées de manière complémentaire.

Selon le poids respectif associé aux entrées A et B, le signal issu de l’interpolation verra sa phase passer linéairement de la phase en sortie du signal A (a[0:2]=[1, 1, 1]) à celle du signal B (a[0:2]=[0, 0, 0]). Les largeurs des transistors dans les branches contrôlées par a0, a1, a2 doivent être dans les rapports 1/2/4.

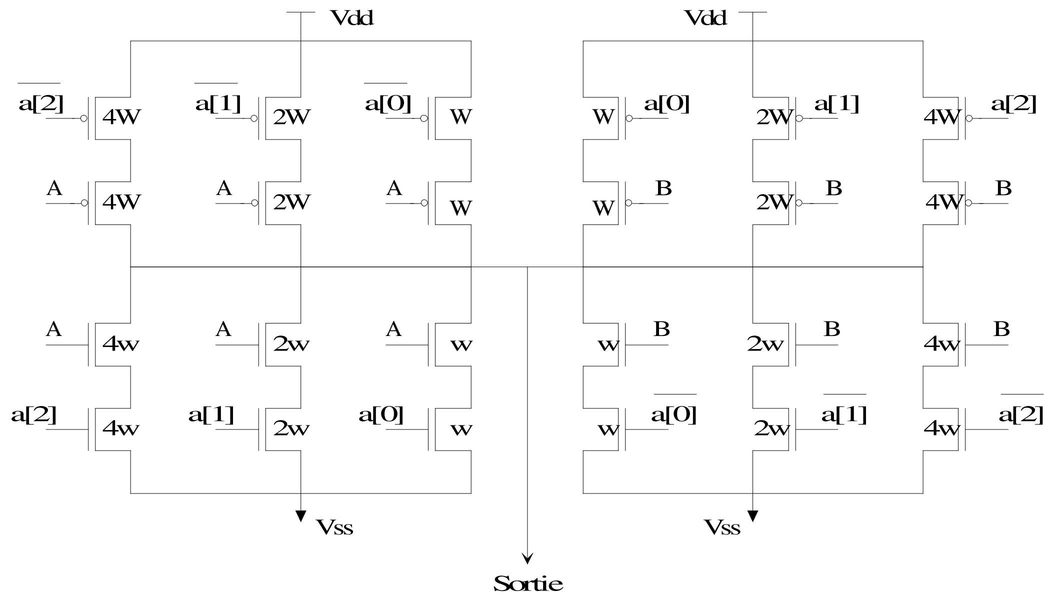


Schéma de l’interpolateur au niveau transistor

2/ Réalisation d’un retard variable

Pour étendre la plage de valeurs que peut prendre le retard, il suffit d'ajouter des retards à valeurs fixes en série et de réaliser l'interpolation sur deux valeurs de retards contiguës.

La commande d’un retard variable se décompose en deux parties :

- la première définit les entrées de l’interpolateur (gam[2:0]),
- la deuxième définit le poids respectif de chacune des deux entrées (a[0:2]) au sein de l’interpolateur.

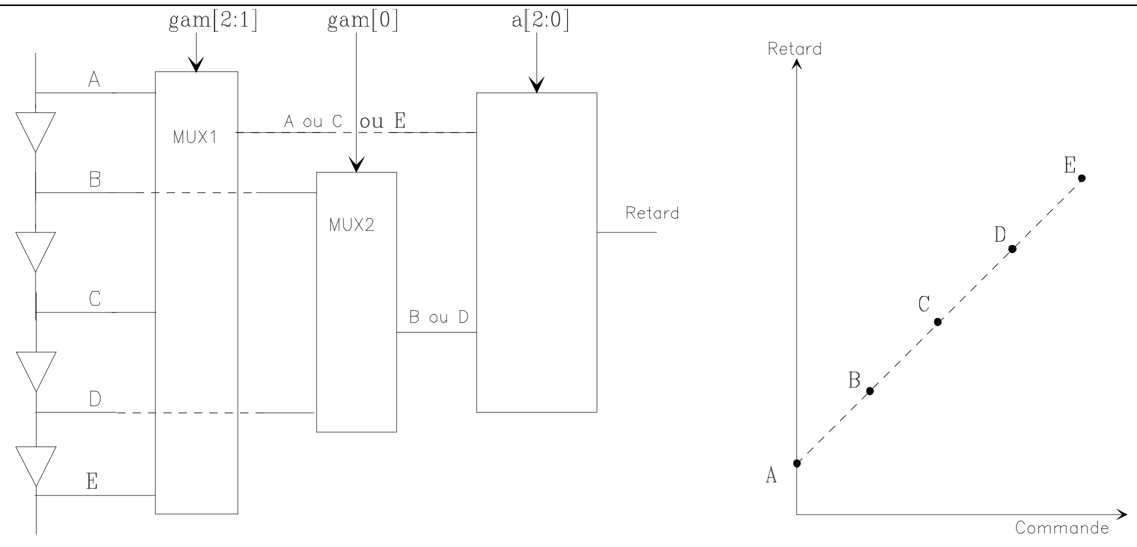
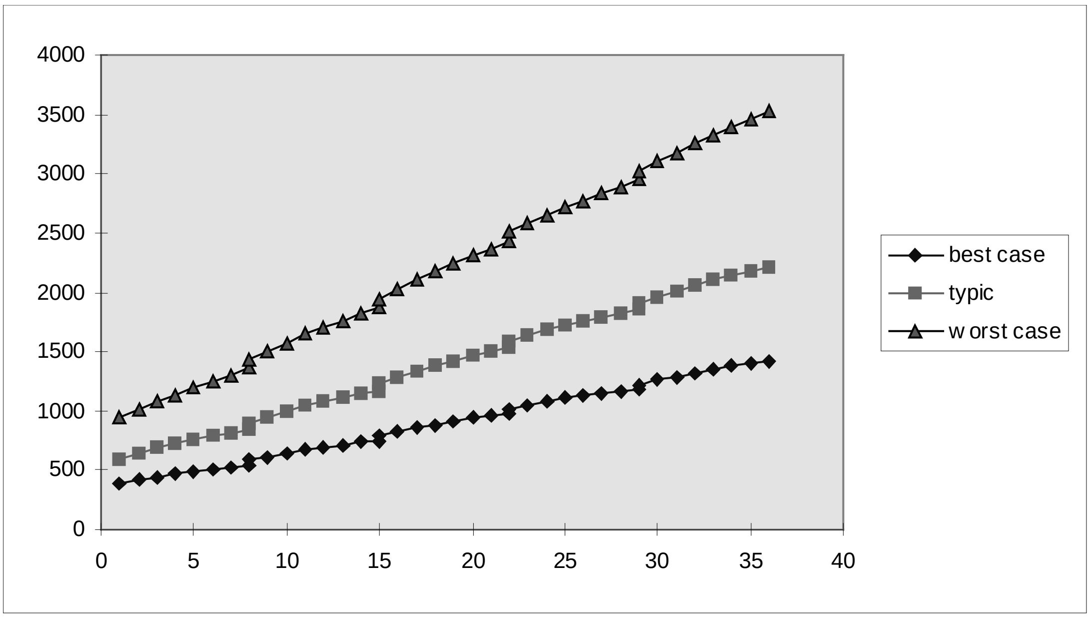


Schéma du retard variable

3/ Résultats de simulation et de test

La figure ci-dessous est le résultat d'une simulation SPICE du retard variable en CMOS 0.5µ.



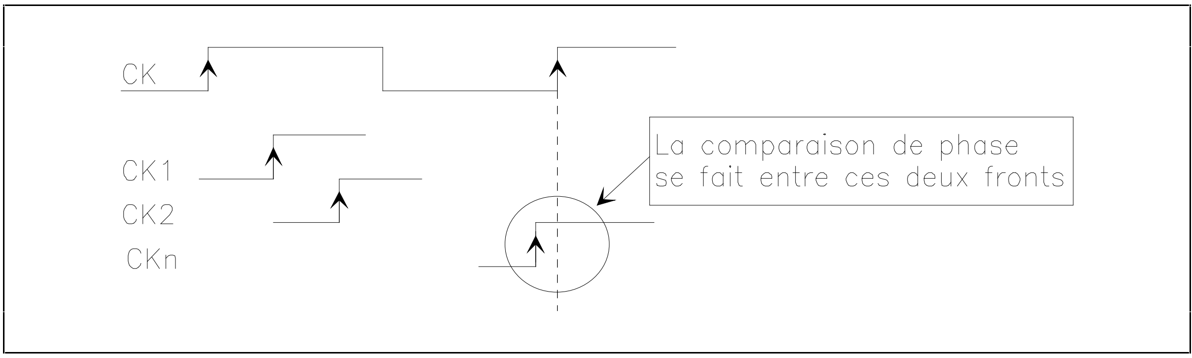
Dans les conditions les plus rapides (process 'best case', température 0°C, Vdd=3.6V), le retard peut varier entre 390 et 1410 ps.
Dans les conditions typiques (process 'typic', température 25°C, Vdd=3.3V), le retard peut varier entre 600 et 2210 ps.
Dans les conditions les plus lentes (process 'worst case', température 70°C, Vdd=3.0V), le retard peut varier entre 940 et 3500 ps.

Ceci nous donne une évaluation de la plage de fonctionnement en fréquence du DLL qui sera disponible quelque soit le process et les conditions d'utilisation : entre 940 et 1410 ps, soit entre 70 et 106 MHz.

Le pas de variation du retard est environ de 40 ps pour une technologie CMOS 0,5µ en conditions d'utilisation typiques.

B/ Le comparateur de phase

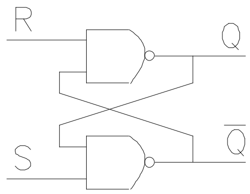
Le but du comparateur de phase est de mesurer le signe du déphasage entre deux fronts montants CK et CKn où CKn est le signal obtenu à partir de CK après avoir traversé n retards variables.
Le résultat du comparateur doit apparaître sous la forme soit d'une demande d'incrémentation de la valeur de la commande numérique des retards variables (CKn précède CK comme ci-dessous), soit d'une demande de décrémentation de cette valeur (CK précède CKn).



Les caractéristiques requises pour le comparateur de phase sont les suivantes :

- résolution inférieure au pas de réglage du retard variable (< 40 ps)
- zone d'indécision aussi réduite que possible
- symétrie complète dans le traitement des deux entrées CK et CKn.

La solution choisie est une bascule R/S réalisée en portes NAND.



R	S	Qn	$\overline{Qn}$
0	0	1	1
1	0	0	1
0	1	1	0
1	1	Qn−1	$\overline{Qn-1}$

L'état R=S=1 est un état mémorisant.

Dans notre cas, R et S sont CK et CKn. Les deux entrées étant à l'état bas, la première qui commute crée un front descendant sur la sortie du NAND correspondant. Quand la deuxième entrée passe à 1, l'état précédent est conservé.

Qn = 1 indique que le front montant est arrivé sur S en premier.

Qn = 0 indique que le front montant est arrivé sur R en premier.

Lorsque les fronts sont parfaitement simultanés, la bascule est dans un état métastable et la moindre perturbation la fera basculer d'un coté ou de l'autre.
Pour cette raison, on n’applique pas directement aux commandes des retards la demande d’incrémentation ou de décrémentation engendrée par une comparaison de phase. On préfère accumuler plusieurs demandes consécutives dans le même sens pour prendre une décision valide. Arbitrairement, on temporise les demandes d’incrémentation ou de décrémentation sur quatre cycles d’horloge. Il faut obtenir quatre demandes dans le même sens en sortie du comparateur de phase pour modifier la commande des retards.

C/ La boucle à verrouillage de retard (DLL)

La boucle à verrouillage de retard (Delay Locked Loop) est utilisée dans la liaison série pour réaliser plusieurs séries de signaux d'horloge multiphases de même fréquence.

Dans le sérialiseur, elle fournit les signaux d'échantillonnage des données en parallèle et réalise ainsi le décalage dans le temps des bits présentés en entrée de l'arbre de OU-exclusifs.
Dans le désérialiseur, elle fournit les signaux d'échantillonnage du signal série pour récupérer la valeur des bits et le signal d'horloge associé.

L’ensemble des éléments qui constituent la DLL ont été mis au point par l’équipe de R.Marbot chez BULL dans le cadre du développement de HS Link.

1/ Principe

Comme nous l’avons vu au chapitre III, la boucle à verrouillage de retard est basée sur la propagation d'un front le long d'une chaîne de retards.
Le dispositif est contrôlé par une boucle d'asservissement de manière à obtenir la simultanéité de deux événements extraits de la chaîne de retards. Ici on réalise une

comparaison de phase entre CK et CKn. Le résultat obtenu commande un compteur/décompteur qui définit la commande numérique qu'il faut appliquer aux retards.

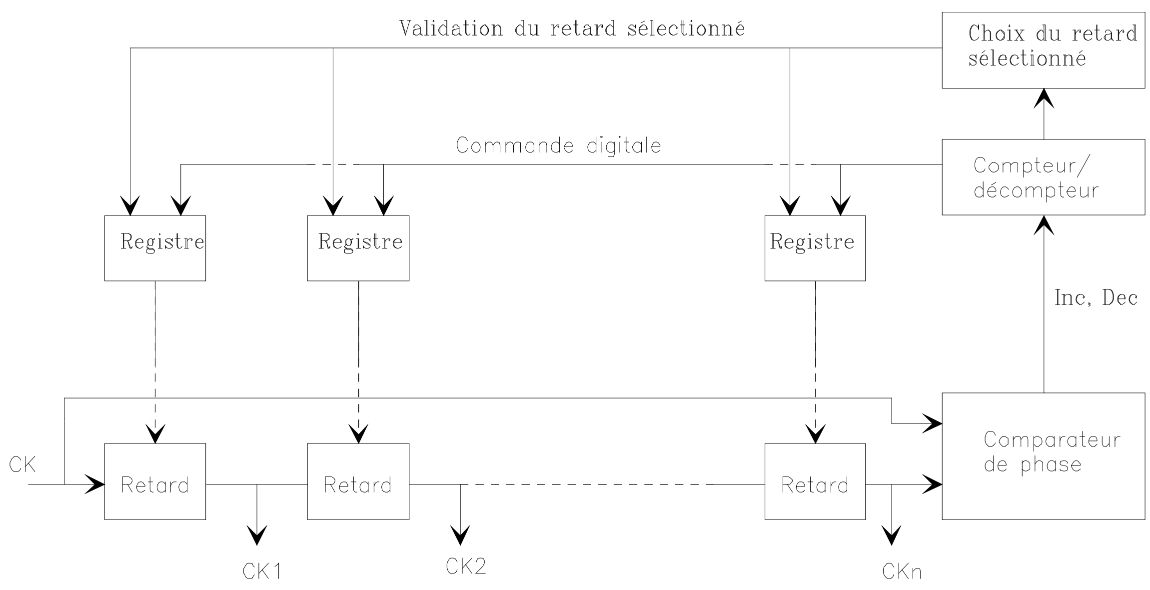


Schéma de la DLL

Par principe de construction, le fonctionnement de la DLL est très stable et garantit une bonne précision sur le décalage en phase des signaux d'horloge CKi générés. Pour cela, il s'appuie :

- sur la variation linéaire du retard et sur la précision de son pas de variation
- sur la précision du comparateur de phase
- sur la rapidité de réaction de la boucle d'asservissement. En mode fonctionnel, la valeur du retard est ajustée tous les quatre cycles d'horloge ce qui garantit une parfaite adaptation aux variation de température ou de tension d'alimentation.

2/ Mécanisme d'incrémentation ou de décrémentation des retards

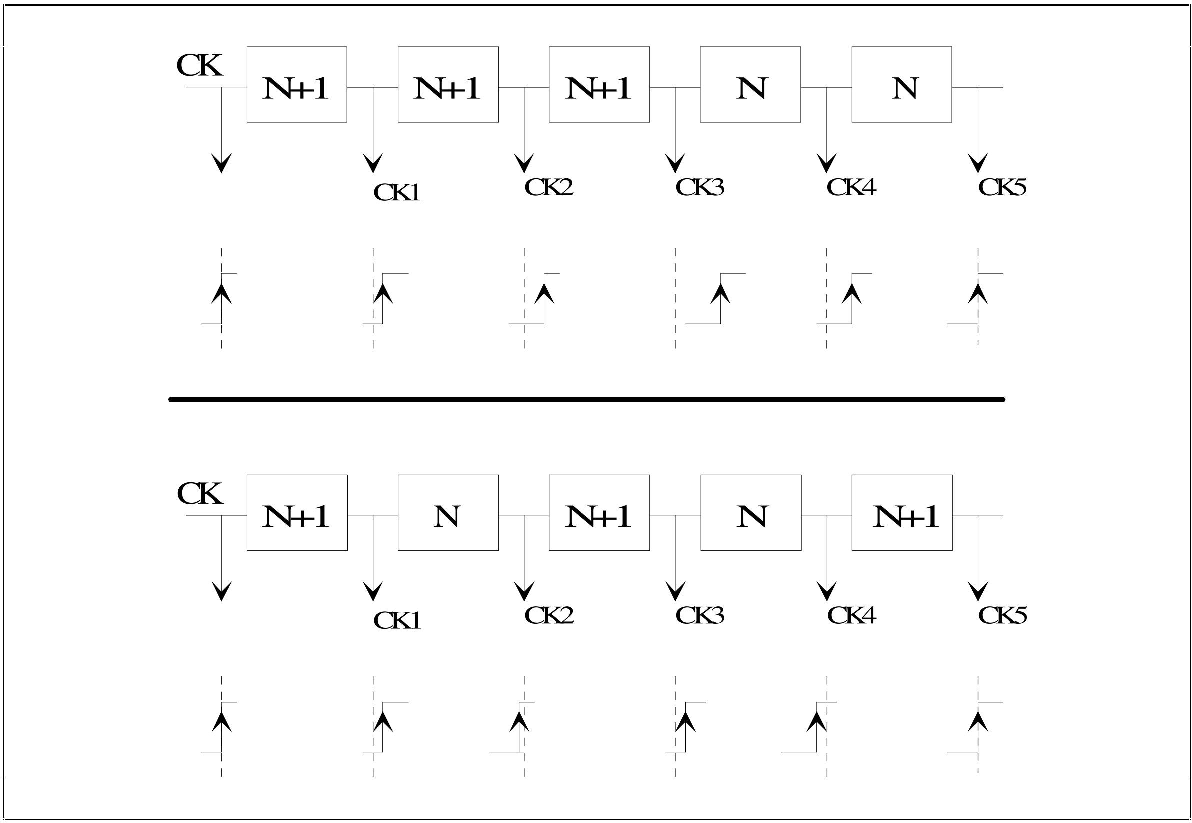
Lorsque le comparateur de phase détecte un décalage entre CK et CKn, il transmet une demande d'incrémentation ou de décrémentation des retards. Pour garder une bonne précision il est préférable de faire varier un seul retard à la fois. Ceci nous donne un pas d'ajustement d'environ 40 ps pour la boucle.

Chaque retard reçoit donc deux signaux sur son registre de commande :

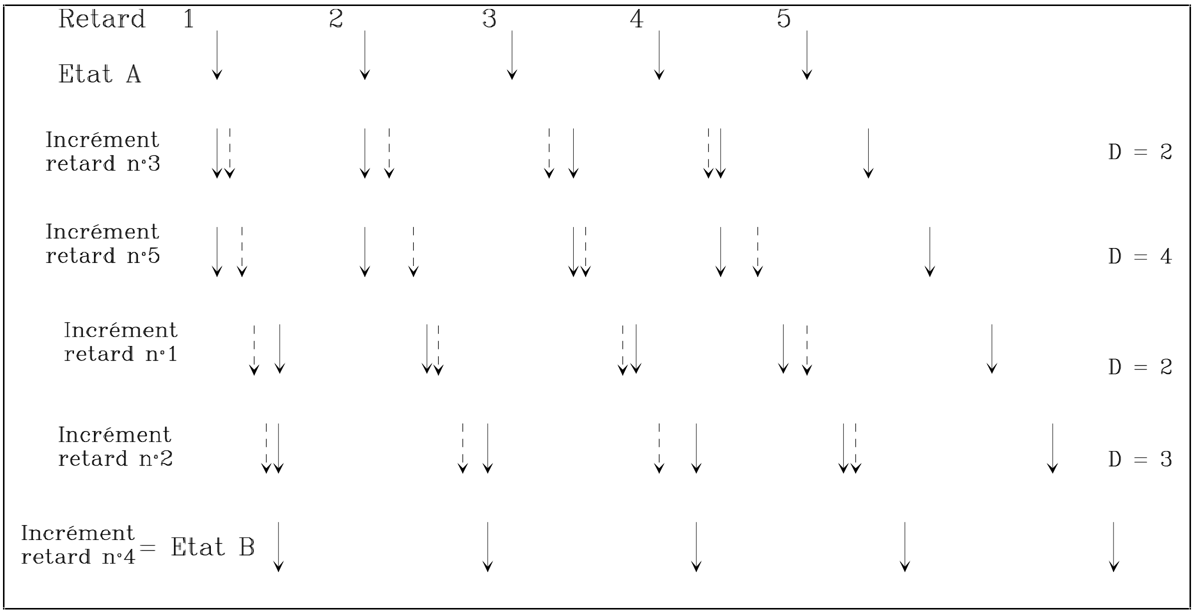
- Une valeur du retard qui est mémorisée
- Un signal de validation qui autorise l’écriture dans un seul registre à la fois.

L'ordre dans lequel les retards sont modifiés est un élément important pour garantir un minimum de jitter sur la série d'horloges générées par la boucle.

Si les retards sont modifiés dans l'ordre du schéma, le décalage en phase entre les signaux CK et CKi où i=1,2,...,5 s'éloigne notablement de la valeur optimale de retard définie par $(3*(T_{N+1}) + 2*(T_N)) / 5$ et indiquée ci-dessous par la ligne en pointillé.



Si les retards sont sélectionnés dans un ordre judicieux, on peut obtenir un décalage minimum entre la position du front montant de chaque CKi et sa position optimale.
Le choix de l’ordre de sélection des retards est effectué en parcourant tous les arrangements possibles des retards qui composent la DLL.
La figure ci-dessous présente le passage pour cinq retards de l’état A à l’état B. Les retards sont incrémentés dans l’ordre 3, 5, 1, 2 et 4. Dans chaque cas, on compare la position du front d’horloge en sortie du retard (flèche en trait continue) avec sa position idéale (flèche en trait pointillé).
D est l’écart maximum à la position idéale en valeur absolue pour les cinq fronts d’horloge après l’incrément d’un retard.

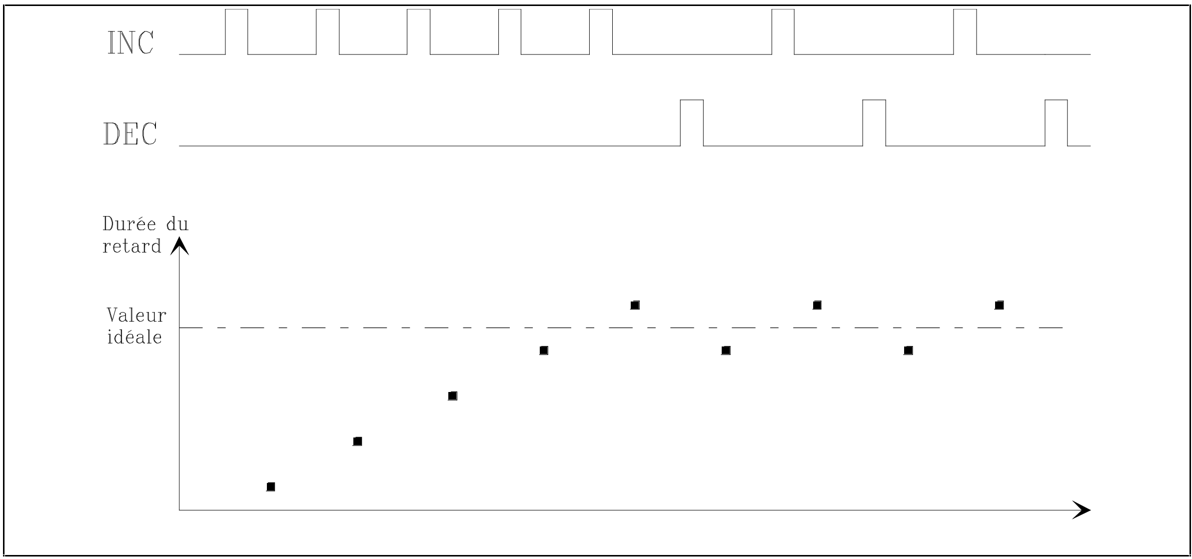


Dmax est le plus grand des D pour le passage d'un état A à un état B.
Pour chaque arrangement des retards, on calcule Dmax. L'arrangement qui offre le Dmax le plus faible correspond à celui qui produit en moyenne les fronts d'horloge les plus proches de leur position optimale.

<u>3/ Calibration de la DLL</u>

L'emploi de la DLL nécessite la mise en place d'une phase de calibration des retards au moment du démarrage de la liaison série.

Au moment de la mise sous tension, la commande numérique imposée aux retards est la valeur qui correspond à un retard minimum.
Si l'horloge CK a sa fréquence comprise dans la plage de fonctionnement de la DLL, le mécanisme d'asservissement fait croître la valeur des retards jusqu'à obtenir CK et CKn en phase.
Quand CK et CKn sont en phase, la commande du retard oscille entre deux valeurs autour de la position idéale. Le comparateur de phase génère alternativement une demande d'incrément et une demande de décrément.
On considère que le retard est calibré lorsque huit décréments ont été générés.



Il faut de même calibrer la valeur des différents retards variables utilisés dans la liaison série et ceci en respectant leur ordre d’interdépendance.
L’utilisation d’une procédure d’initialisation permet de gérer les phases de calibration. Cette procédure enchaîne les événements suivants :

- reset
- calibration des retards T
- calibration des retards 1.5T

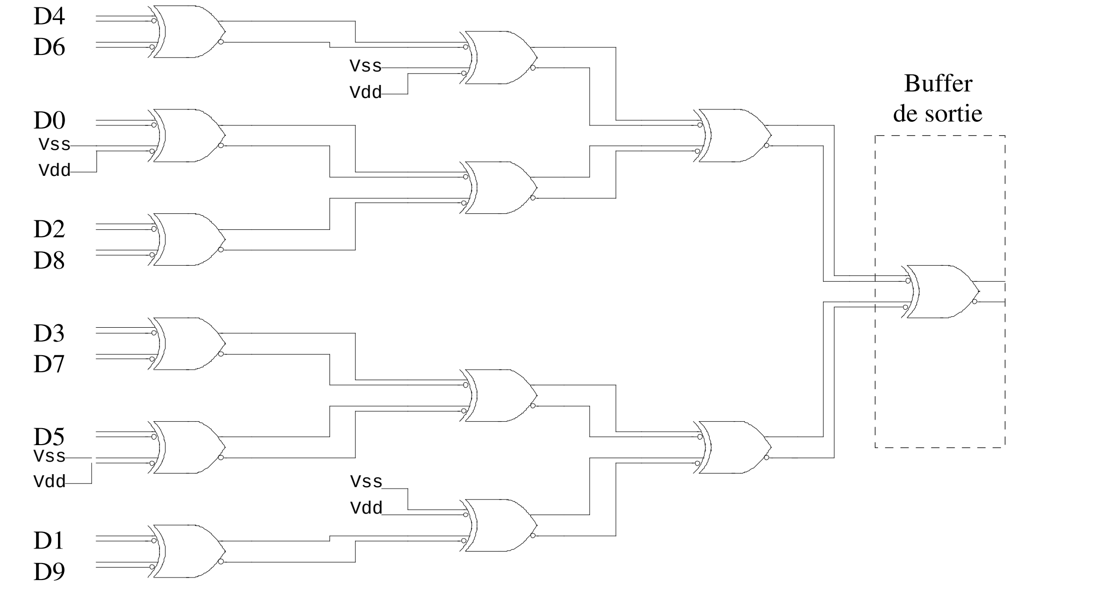
On effectue ensuite en parallèle la calibration de la valeur maximale Xmax du retard X, la calibration du retard X et la recherche du mot de référence COMMA.

D/ L’arbre de OU-exclusifs

Ce schéma constitue le principe de base du sérialiseur. Il est composé de OU-exclusifs différentiels qui garantissent une parfaite symétrie dans la transmission des signaux sur le lien série. De plus son implantation a été soigneusement réalisée de manière à égaliser les longueurs des chemins que peuvent emprunter les signaux de données.

Pour éviter de propager des signaux à très hautes fréquences, le dernier OU-exclusif différentiel fonctionne à la fréquence maximale. Il est placé dans le buffer de sortie. Ainsi les signaux internes au circuit ne dépassent pas la fréquence moitié du signal série.

Pour les mêmes raisons, les dix entrées de l'arbre sont associées deux à deux de manière à ce que les deux entrées d'un OU-exclusif commutent avec un délai entre elles d'au moins deux fois la durée d'un bit. C'est pourquoi nous avons dissociés les entrées paires et impaires.



Exemple d’implantation de l’arbre de OU-exclusifs pour 10 entrées

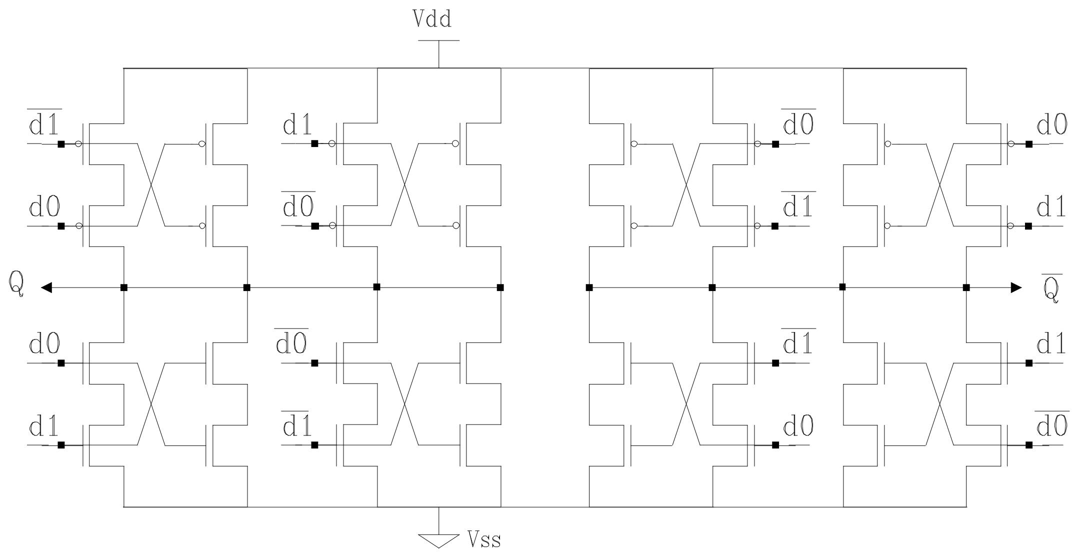


Schéma du OU-exclusif différentiel au niveau transistor

L’utilisation d’une structure différentielle répond à une exigence de parfaite symétrie dans le traitement des entrées de l’arbre de OU-exclusifs.
Pour générer un signal série de bonne qualité, la propagation d’une transition sur une des entrées de l’arbre jusqu’à la sortie doit avoir la même durée quelque soit l’entrée concernée et quelque soit la nature de la transition (montante ou descendante). Le schéma présenté ci-dessus respecte ces exigences.

E/ La bascule d'échantillonnage du signal série

C'est la bascule utilisée dans le désérialiseur pour échantillonner le signal série. Elle a été spécifiquement étudiée pour limiter les problèmes de métastabilité pendant la phase d’ajustement de la valeur du retard X.
Cette bascule est le résultat d’une étude menée chez BULL par J-M.Boudry

Le problème posé est le suivant : on échantillonne le signal série de fréquence Fd avec un signal d'horloge généré par le désérialiseur. Ces deux signaux sont asynchrones. On cherche à évaluer la probabilité d'obtention d'un état métastable dans le cas où l’horloge d’échantillonnage et le signal série échantillonné sont indépendants l’un de l’autre (retard X non ajusté):

On définit la période d'obtention d'un état métastable comme : $P = \dfrac{e^{\frac{Tu}{To}}}{2\ F_c\ F_d\ T_p}$

où :

- F_c est la fréquence de l'horloge d'échantillonnage ~100MHz
- F_d est la fréquence du signal échantillonné ~100MHz
- T_p est la fenêtre temporelle pendant laquelle le risque de métastabilité est présent (majoré à 1ns)
- Tu est le temps utile qu'on laisse au latch pour perdre sa métastabilité
- To est la constante de temps avec laquelle évolue l'exponentielle pour quitter l'équilibre métastable

Le dénominateur donne la fréquence moyenne de coïncidence entre les fronts de l'horloge d'échantillonnage et les fronts du signal échantillonné dans la fenêtre temporelle T_p=1ns.
La période moyenne de ces coïncidences est alors
$P_{mc} = \dfrac{1}{2\times10^8\times10^8\times10^{-9}} = \dfrac{1}{2\times10^7} = 5\times10^{-8}$

Quand la bascule est laissée dans un état voisin de l'équilibre métastable, il s'éloigne de cet état par une exponentielle. To est la constante de temps de cette exponentielle.
To a été estimé par un jeu de simulations SPICE en pire cas à 0,11ns.
Tu représente le temps dont dispose la bascule pour quitter la métastabilité. La période d'horloge étant environ de 10ns, on soustrait de cette valeur 1ns pour le temps de traversée de la bascule et 2ns pour se donner une marge de sécurité. On prend donc Tu=7ns.

On arrive alors à $P = e^{\frac{7\times10^{-9}}{1.1\times10^{-10}}}\times5\times10^{-8} = 2.7\times10^{13}$ ans

Cette valeur nous permet d'utiliser la bascule ci-dessous pour échantillonner le signal série au moment présumé des transitions puisque la valeur obtenue sera toujours stable pour les étages suivants.

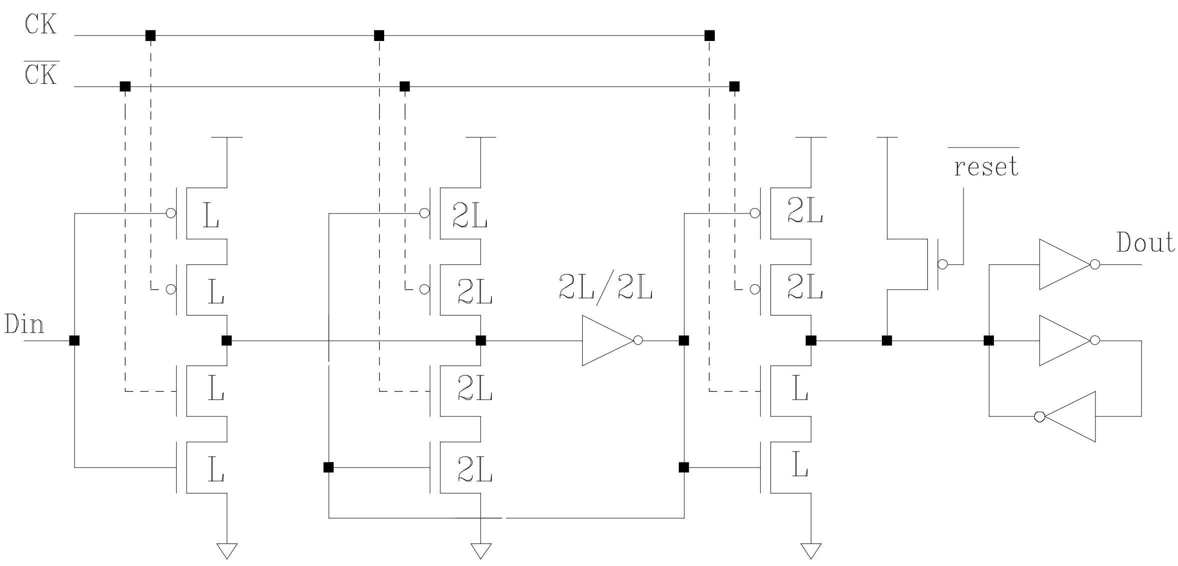


Schéma de la bascule d'échantillonnage du signal série

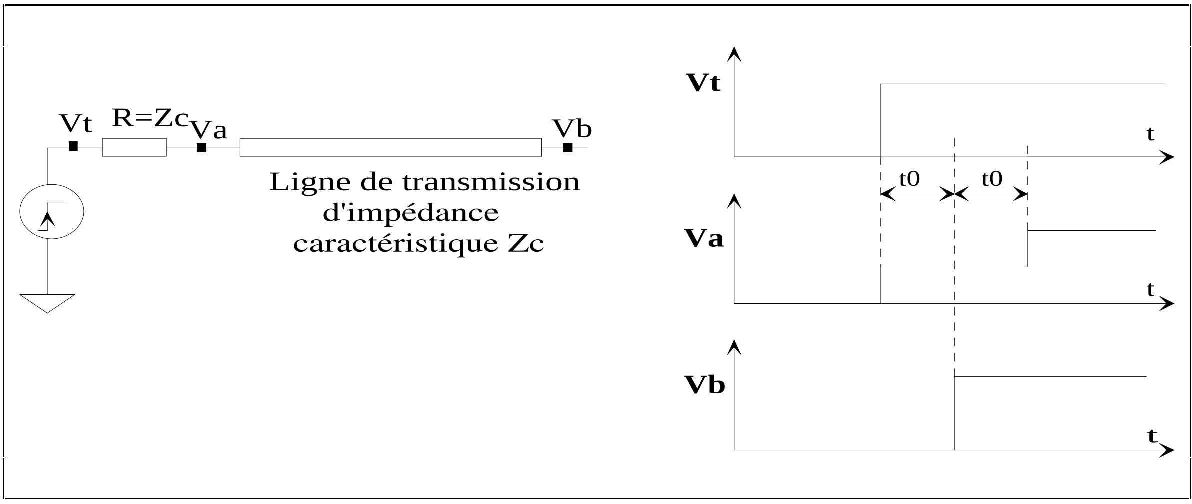
La taille des transistors de la boucle de réaction est plus importante que pour une bascule classique. Cela permet d’augmenter le produit gain * bande passante et de réduire l’effet des éléments parasites que constituent le circuit d’entrée et l’inverseur de sortie.

Une fois ajustée la valeur du retard X, le signal série est échantillonné d’une part au centre de chaque bit et d’autre part au niveau de chaque transition.
L’échantillonnage des bits ne peut plus donner lieu à un état métastable.
Par contre l’échantillonnage des transitions engendre nécessairement des états métastables. Ceux-ci sont traités en retardant de plusieurs cycles l’utilisation des valeurs échantillonnées.

F/ Circuit d’adaptation d’impédance

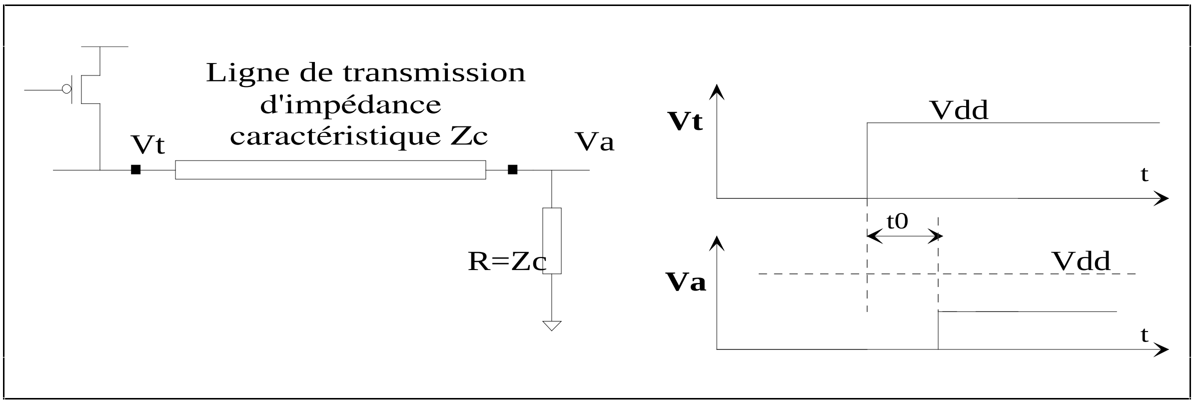
La transmission du signal série d’un circuit à un autre à travers une ligne de transmission ne doit pas introduire de distorsions sur le signal transmis. Pour avoir les meilleures conditions possibles de transmission du signal, il faut adapter la ligne de transmission à son impédance caractéristique.
L’adaptation d’impédance peut se faire de différentes façons :

- Adaptation série : l’adaptation est réalisée au niveau de l’émetteur.



La ligne est terminée par une impédance infinie (en pratique on utilise une impédance très grande par rapport à Zc). Une transition sur Vt est transmise en Vb au bout du temps t0 de traversée de la ligne.
Cette technique présente l'avantage de ne pas consommer en continu et de transmettre des signaux sans modifier le swing. Elle est, par contre, très sensible aux bruits de mode commun (la ligne non terminée se comporte comme une antenne). De plus, cette technique est difficilement réalisable en CMOS car l'impédance de sortie d'un driver CMOS varie en fonction des commutations de son entrée.

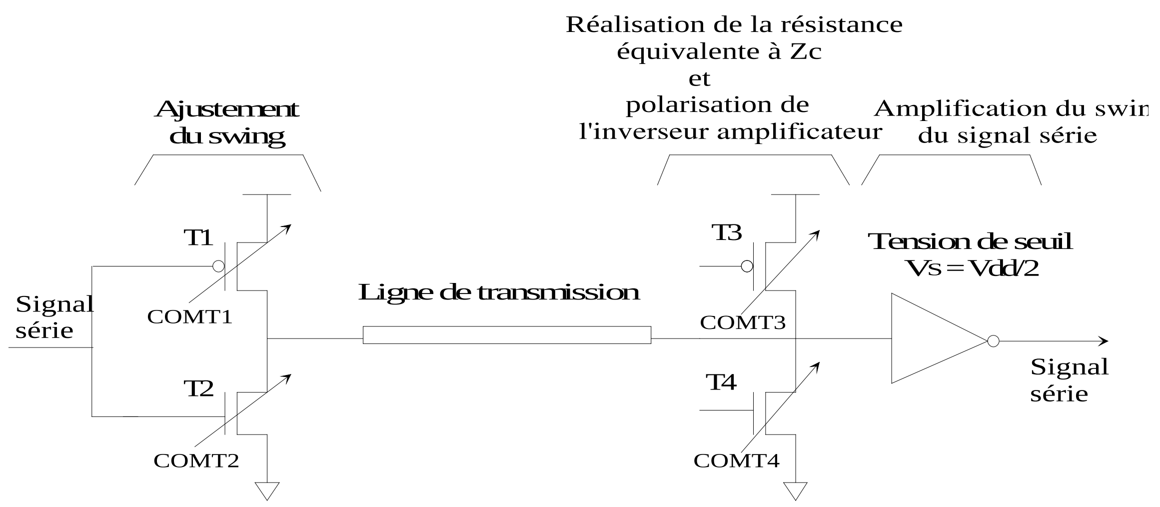
- Adaptation parallèle : l'adaptation est réalisée en fin de ligne.



Une transition sur Vt est transmise en Va au bout du temps t0 de traversée de la ligne avec un swing réduit dans le rapport Zc/Zc + Zs. Cette technique consomme en continu et transmet des signaux à excursion réduite, elle nécessite donc des amplificateurs en réception afin de restituer les niveaux CMOS. Par contre, cette technique présente l'avantage d'être très robuste aux perturbations sur la ligne.
Pour cette raison, nous avons adopté le principe de l'adaptation parallèle.

Le schéma suivant est l'application de ce principe à la macrocellule liaison série.
A l'émission, la valeur du swing sur la ligne est contrôlée en ajustant la taille des transistors T1 et T2 (on suppose la ligne adaptée).
A la réception, le pont T3/T4 réalise la résistance équivalente à Zc.
D'autre part, cette technique d'adaptation d'impédance implique une réduction du swing pour le signal transmis. Le signal doit donc être amplifié après réception. L'amplification

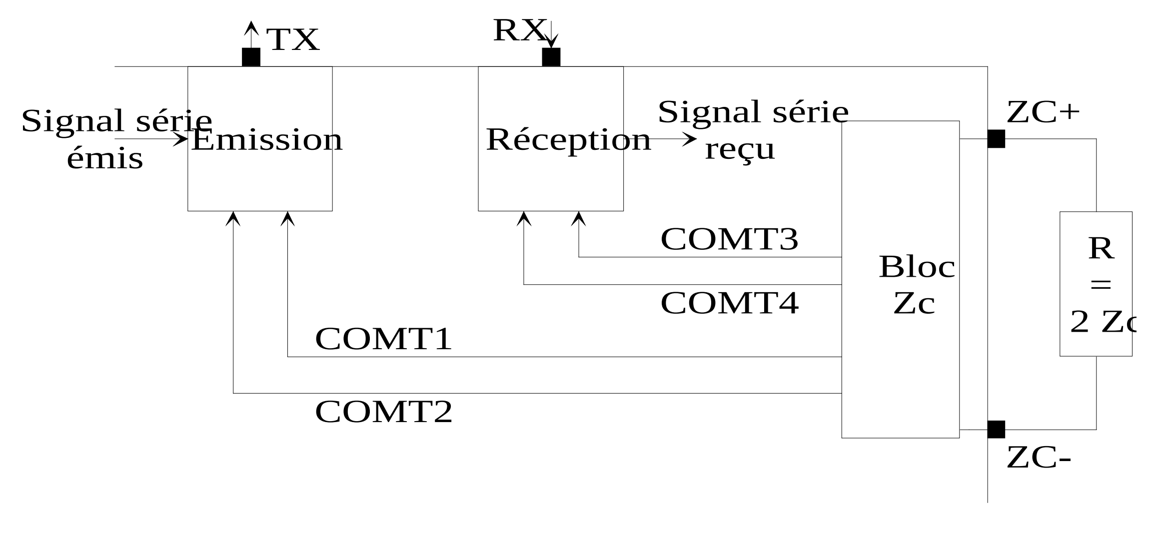
d'un signal induit une limitation de bande passante et une distorsion qui doivent rester compatibles avec les contraintes de la liaison. Le meilleur inverseur en CMOS (single ended) est un inverseur polarisé dans sa zone de commutation. Ajuster le rapport de taille entre les transistors T3 et T4 permet de polariser l'inverseur amplificateur.



<u>Adaptation d'impédance et de swing sur le signal série</u>

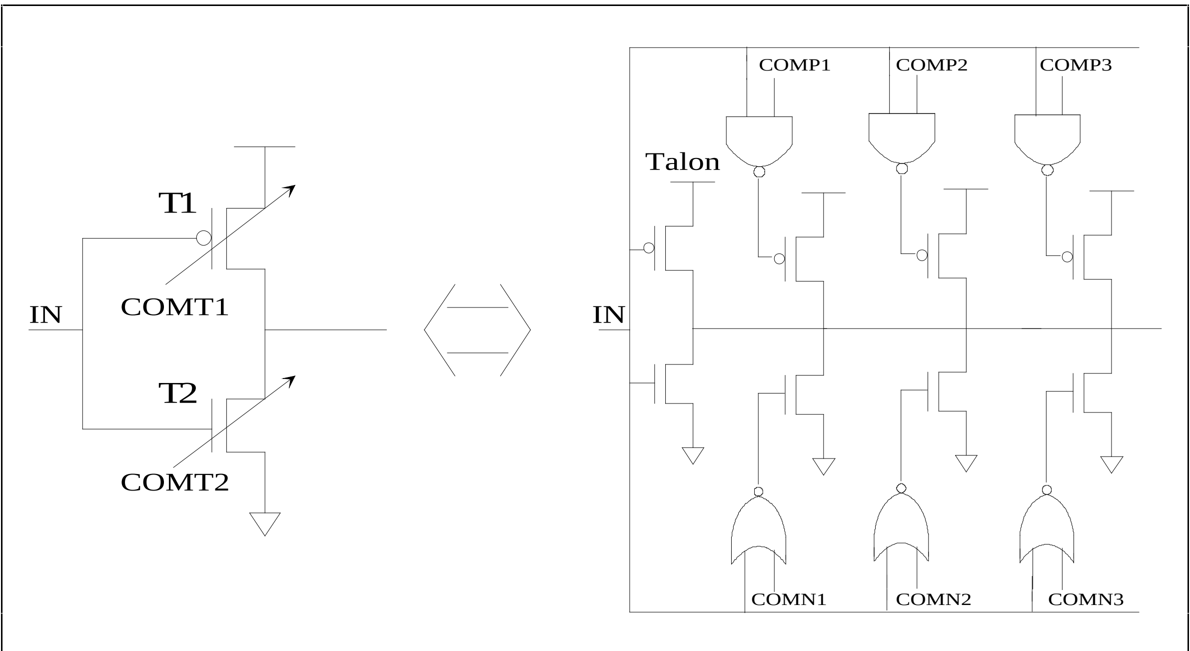
Ce schéma découple les parties rapides de traitement du signal série (schéma ci-dessus) des parties de contrôle qui réalisent l'adaptation de la ligne de transmission. En effet le contrôle des parties analogiques (c'est-à-dire le contrôle de la taille des transistors T1, T2, T3 et T4) est réalisé en parallèle par une boucle d'asservissement statique purement numérique. C'est la fonction du bloc Zc.

Le bloc Zc détermine la taille des transistors T1 à T4 sur des schémas équivalents au chemin de données sur la ligne de transmission et transmet les valeurs obtenues aux buffers d'émission et de réception.



- Réalisation d'un transistor à taille variable

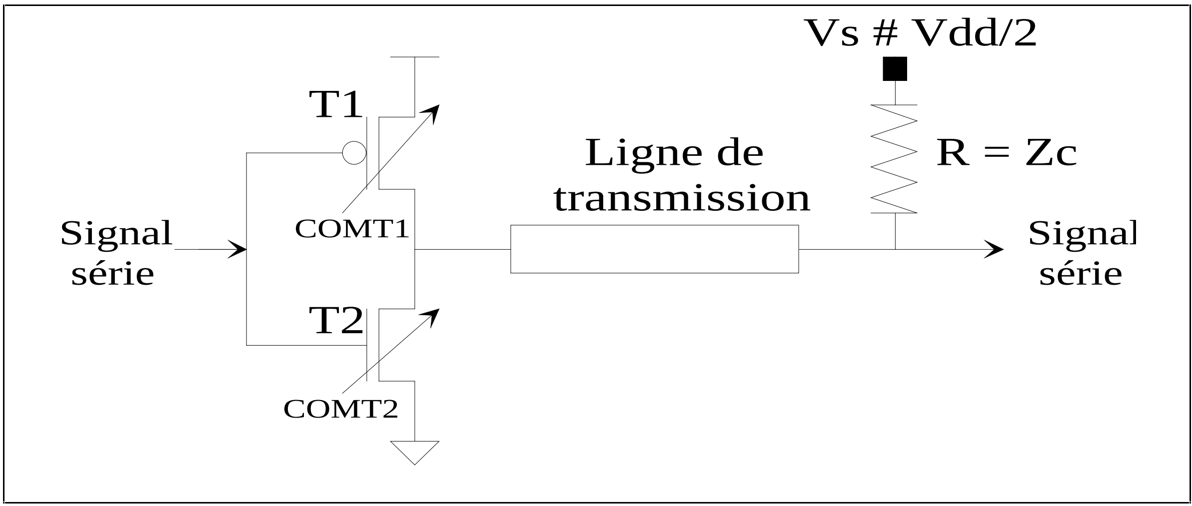
Pour compenser les dispersions de process et de conditions de fonctionnement du circuit, il est nécessaire de pouvoir ajuster en temps réel la taille des transistors T1 à T4. Nous utilisons pour cela des transistors à taille variable dont la taille est définie par une commande numérique.



Un transistor à taille variable est obtenu en plaçant en parallèle des transistors et en contrôlant numériquement le nombre de transistors actifs. Pour obtenir une meilleure précision dans l'ajustement de la taille du transistor, on définit une taille minimale avec le talon. Le nombre de transistors implémentés dans ce réseau et leurs tailles respectives dépendent de la gamme de courants de sortie qu'on veut garantir (quelque soient les conditions de travail: dispersions de process, gamme de température et de tension d'alimentation), et de la résolution qu'on veut atteindre sur le contrôle de ces courants.

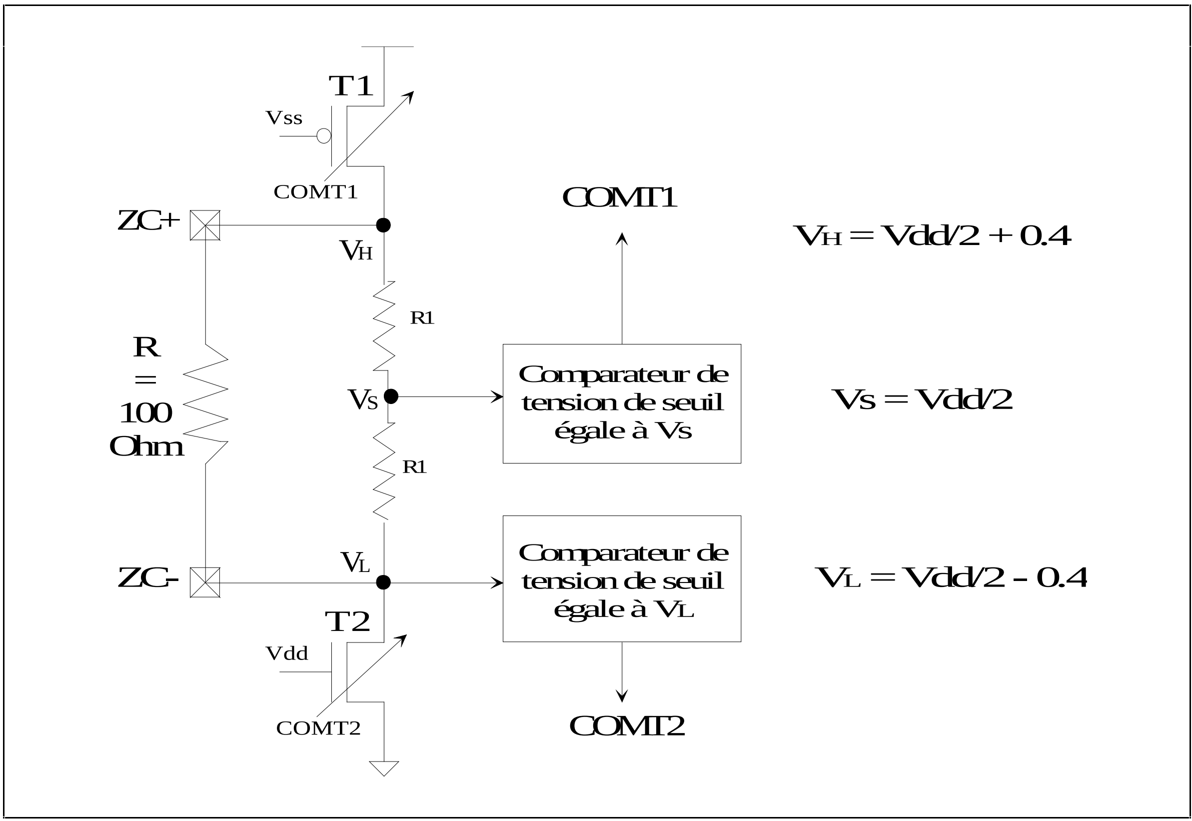
1/ Emission

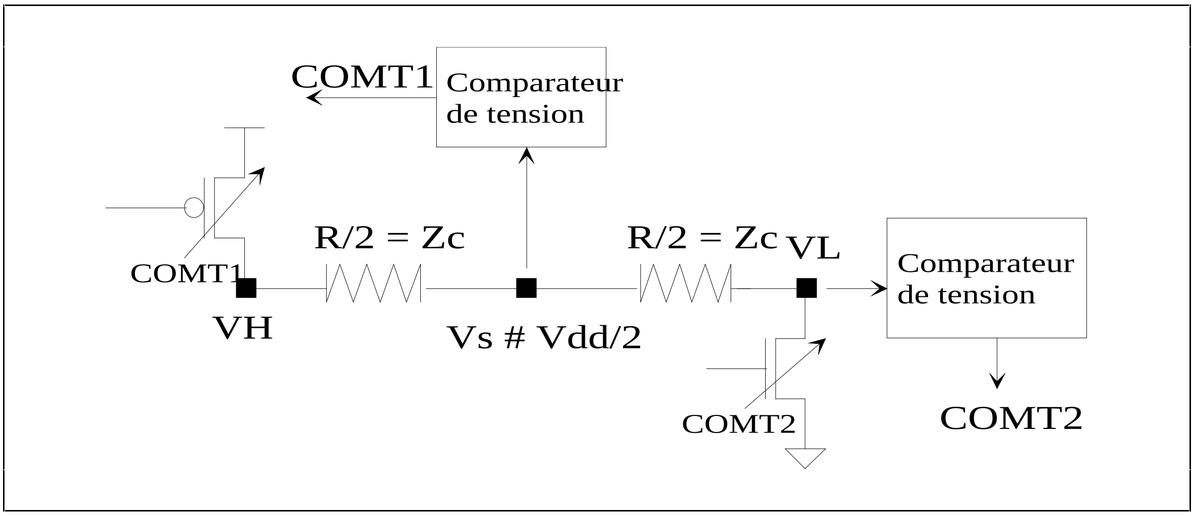
En émission, il faut contrôler le swing du signal série: le swing passe de 3.3V (niveaux CMOS) à 0.8V centré autour de Vdd/2 (swing compatible avec les niveaux ECL). Cette adaptation est réalisée par l'intermédiaire d'un inverseur CMOS dont on ajuste la taille des transistors P (T1) et N (T2).
Les tailles respectives de T1 et T2 sont déterminées en supposant que la ligne de transmission est correctement adaptée comme ci-dessous.



Pour déterminer les tailles de T1 et T2, on réalise dans le bloc Zc un schéma qui reproduit les conditions d'émission sur la ligne. COMT1 et COMT2 commandent la taille des transistors T1 et T2 respectivement.

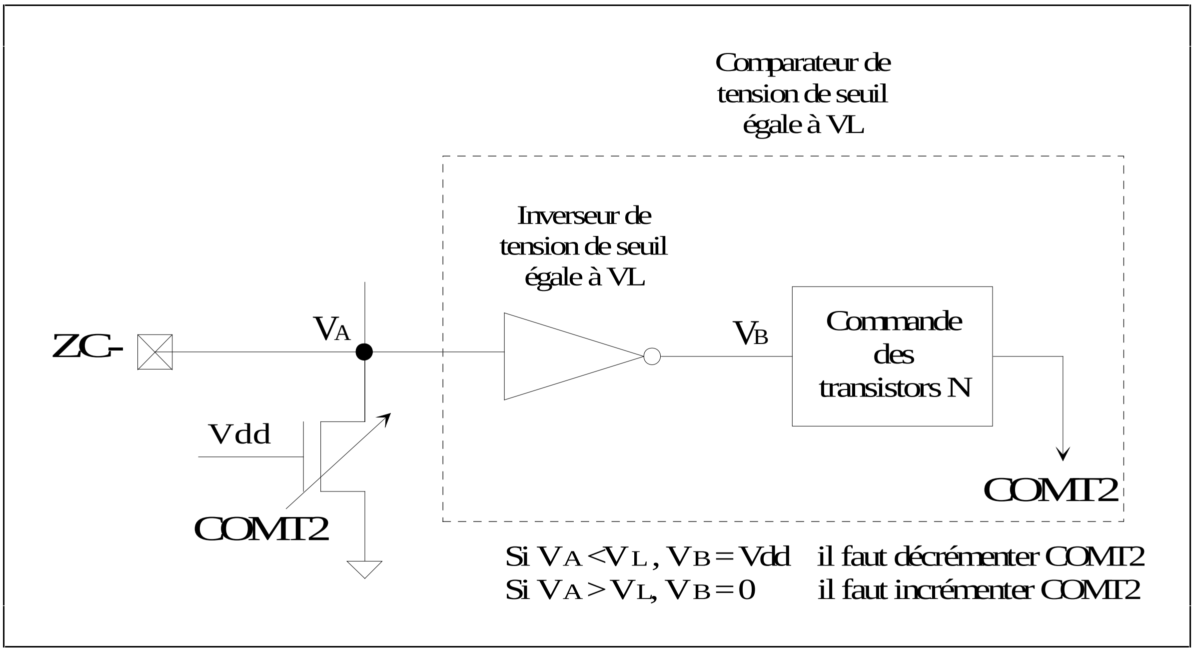
On place aux bornes de ZC+ et ZC- une résistance externe R = 100 Ohm équivalente à deux fois l'impédance caractéristique de la ligne de transmission. En interne, R1 est grand par rapport à R. On se ramène donc au schéma suivant où l'intensité du courant qui traverse la résistance R est contrôlée par la taille des transistors P et N commandée respectivement par COMT1 et COMT2.





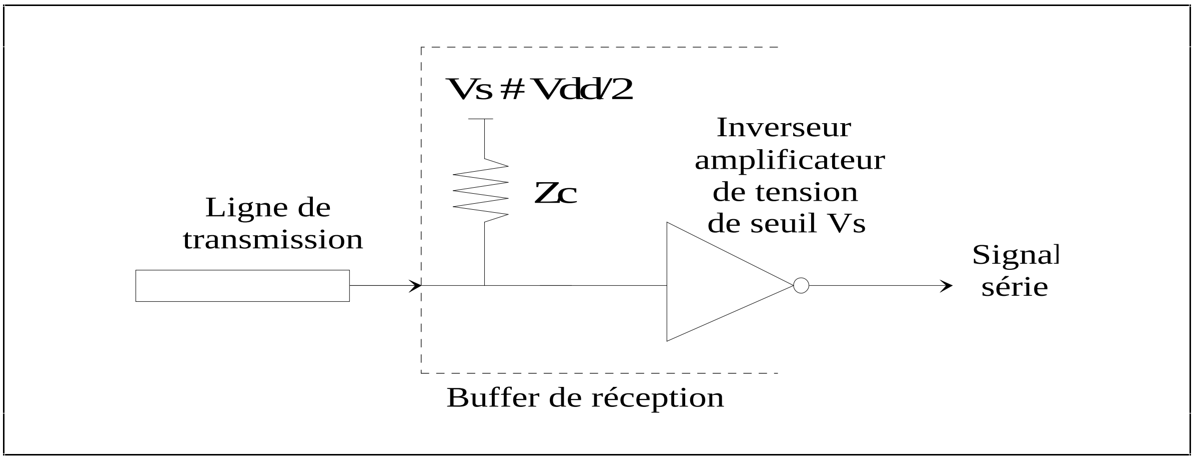
Les boucles de commandes de COMT1 et COMT2 sont purement numériques.
On ajuste COMT1 et COMT2 pour obtenir : V_S = Vdd/2 et V_L = Vdd/2 - 0.4 Volt.
La tension du signal série en émission varie de + ou - 0.4 Volt autour de Vdd/2.

La comparaison de tension à V_S et à V_L est réalisée par un inverseur taillé spécifiquement dans son rapport Wp/Wn pour que sa tension de seuil soit V_S ou V_L. L'ajustement de ce rapport est obtenu par simulation SPICE et doit être redéfini pour chaque process. La logique de génération des commandes des transistors N et P est illustrée dans le schéma ci-après pour la commande COMT2.

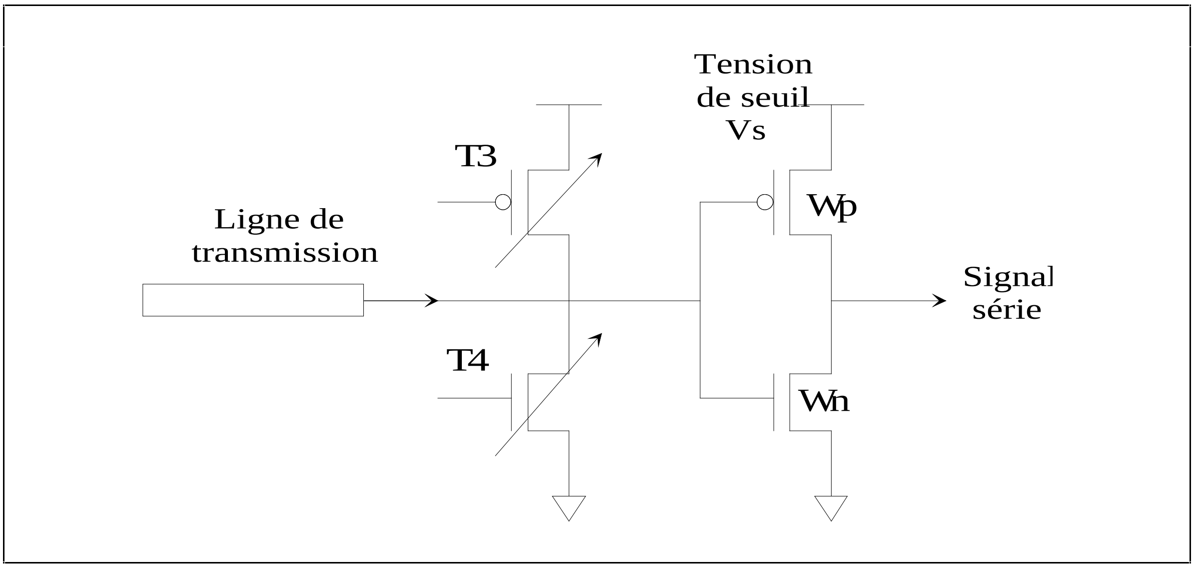


2/ Réception

Le signal série arrive sur le récepteur avec un swing de 0.8V. Il convient de l'amplifier pour restaurer des niveaux CMOS et il faut également adapter la ligne de transmission sur son impédance caractéristique.



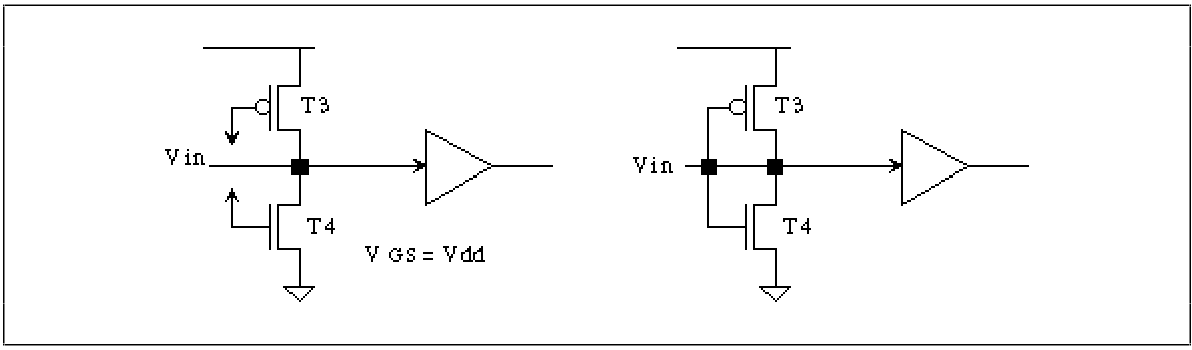
L'amplificateur de réception est un inverseur CMOS centré (tension de seuil Vs # Vdd/2). La ligne de transmission est adaptée grâce à la résistance équivalente réalisée par T3 et T4. Le buffer de réception adopte donc le schéma suivant:



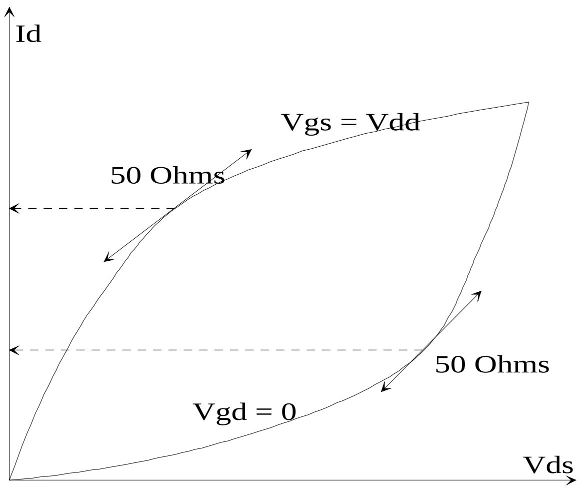
Le pont T3/T4 a une double fonction :

- polariser le signal d'entrée à Vs (pour que l'inverseur travaille dans sa zone d'amplification).
- terminer la ligne sur son impédance caractéristique (50 Ohm).

Le rôle de l’inverseur de réception est d’amplifier le signal série et de restaurer le swing pour l’adapter aux transistors CMOS. Le rapport Wp/Wn de l’inverseur est choisi pour équilibrer son comportement (fronts montants / fronts descendants).
Deux schémas sont possibles pour faire le pont T3/T4:

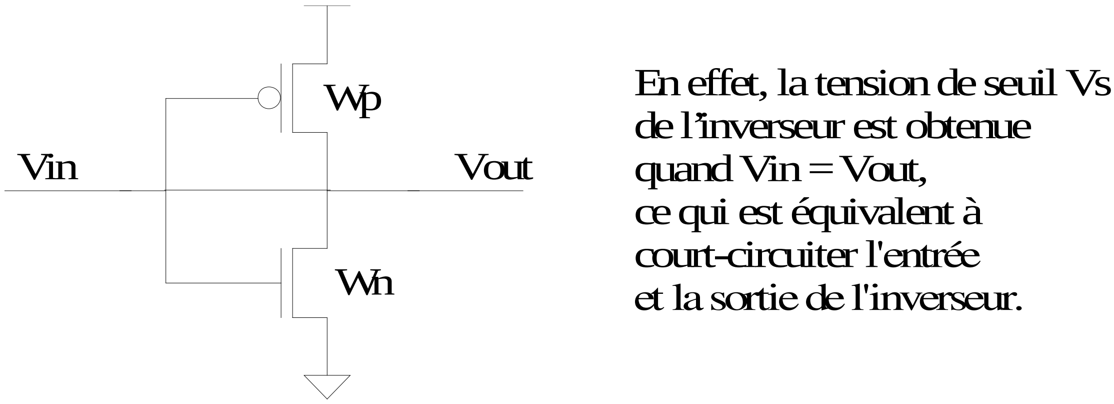


- Pour le premier, Vgs = Vdd. Ce montage consomme plus que les autres (voir diagramme ci-dessous).



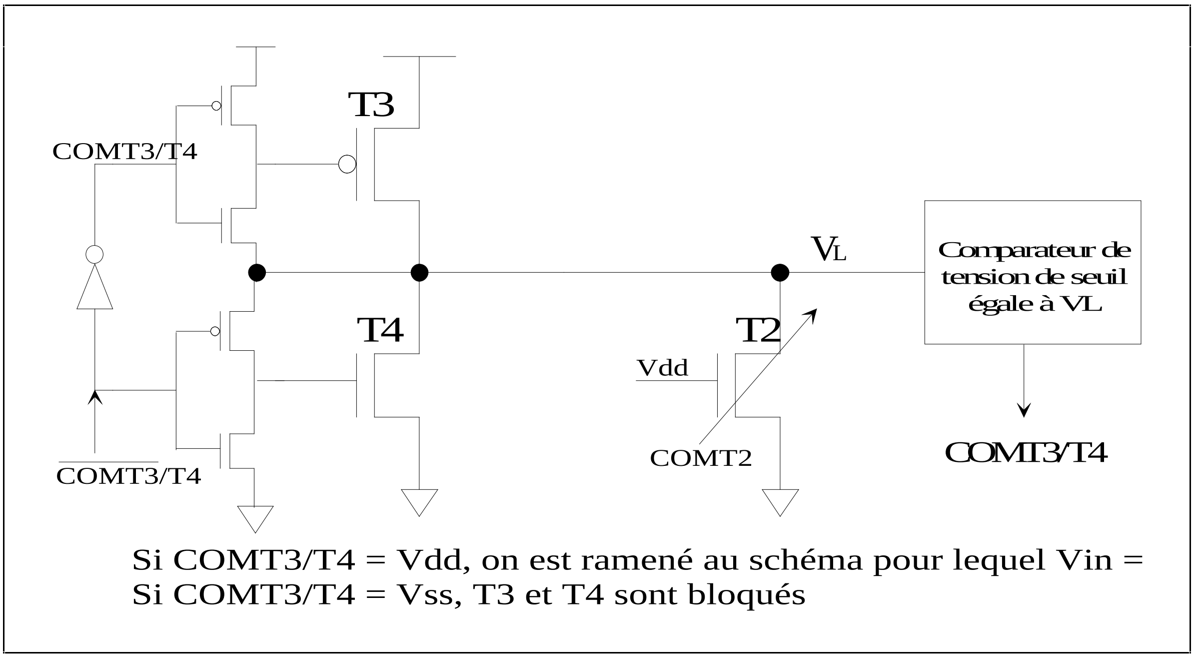
De plus il nécessite un réglage indépendant de T3 et T4 qui fonctionnent différemment par rapport à l’inverseur, donc ne fournissent pas naturellement Vs.

- Le second schéma utilisé offre deux avantages décisifs: il limite la consommation et permet de fixer le seuil à celui de l'inverseur. De plus ce montqge compense naturellement les dispersions dues au process.



Les transistors T3 et T4 sont actifs quand leur grille est connectée à leur drain. Quand les transistors correspondants à T3 et T4 sont actifs, ils génèrent la tension de polarisation de l’entrée de l’amplificateur. Par définition, celle-ci est égale à Vs.

Le rapport des transistors T3/T4 est choisi identique à celui de l’inverseur. Par définition T3 et T4 fournissent Vs. Il reste à régler leur taille pour obtenir l’impédance d’adaptation. C’est le rôle de Zc.
Pour déterminer les tailles de T3 et T4 (c’est-à-dire la valeur de la commande numérique COMT3/T4), on réalise le montage suivant dans le bloc Zc:



Ce montage ajuste COMT3/T4 en fonction de la tension mesurée aux bornes du transistor N commandé par COMT2.
Dans la boucle d’émission, le transistor T2 génère, dans une résistance de 50 Ohm reliée à Vs, un courant qui décale le potentiel du drain jusqu’à VL. Un transistor N de même taille

décalant le potentiel du drain à VL quand il injecte un courant dans le pont T3/T4 implique T3//T4 = 50 Ohm. Par ailleurs le pont T3/T4 donne au repos une tension Vs.

La commande COMT3/T4 est appliquée aux transistors T3 et T4 du buffer de réception du signal série. Une seule commande ajuste la taille de T3 et T4 car les transistors N et P sont associés par couple tels que le rapport de taille entre le P et le N soit égal au rapport Wp/Wn de l'inverseur d'amplification (ceci polarise le signal d'entrée à Vs # Vdd/2).
Les transistors T3 et T4 sont polarisés tels que $V_{GS} = V_{DS}$.
Dans cette configuration, la résistance équivalente à chaque transistor a un comportement quasi linéaire en fonction de V_{DS}, dans la plage utile.

Ce procédé d'adaptation d'impédance a fait l'objet d'un brevet déposé par la société BULL au nom de R.Marbot, J-C.Lebihan, A.Cofler et R.Nezamzadeh-Moosavi (réf. FR 3448).

Chapitre V : Description du circuit de démonstration FIBULE

Le circuit FIBULE est une application des principes exposés dans les deux chapitres précédents. Nous avons réalisé un circuit susceptible de fonctionner dans un environnement standard : le protocole Fibre Channel. Ceci nous permet de comparer directement nos résultats avec ceux d’autres liaisons série existantes.

Le circuit FIBULE est composé d’une macrocellule liaison-série bidirectionnelle, d’un bloc d’auto-test et d’un bloc d’adaptation d’impédance.
Nous commencerons par une description générale du circuit FIBULE en détaillant les implications de la norme Fibre Channel.
Nous présenterons ensuite le sérialiseur et le désérialiseur puis les éléments annexes du circuit comme le module d’auto-test et le bloc d’adaptation d’impédance.

A/ Présentation de FIBULE

FIBULE peut être utilisé pour tous types de données codées sur 10 bits.
FIBULE intègre les fonctions de sérialisation et de désérialisation et possède pour faciliter le test un chemin de rebouclage qui relie en interne la sortie-série du sérialiseur à l'entrée-série du désérialiseur. FIBULE comprend aussi un adaptateur d'impédance qui assure une réception optimale du signal transmis.

FIBULE a pour but de valider les principes originaux présentés au chapitre III. A ce titre, FIBULE comporte de nombreuses possibilités de test destinées à identifier les limites de fonctionnement des différents éléments qui composent la liaison série :

- il existe un chemin de ‘scan’ qui permet de lire le contenu de certains registres de la liaison série et de vérifier ainsi l'état de la macrocellule à différents instants de la transmission.

- il existe un bloc auto-test. Des données aléatoires sont transmises à travers le chemin de rebouclage interne et sont comparées à elles-mêmes après réception. Ceci permet de tester les fonctionnalités dynamiques de la liaison série au niveau du wafer (sans attendre la mise en boîtier).

FIBULE ne réalise aucun protocole de communication et aucun mécanisme de détection d’erreur. Ces fonctions seront prises en charge par les couches supérieures de protocole. En particulier FIBULE est une liaison série compatible avec la norme ANSI X3T11 Fibre Channel.
Fibre Channel est un protocole de communication haut débit permettant la mise en oeuvre de protocoles haut niveau tels que SCSI, IP et HIPPI. Fibre Channel offre la possibilité à ces différents protocoles de coexister sur un même réseau reliant stations de travail, banques de données et super-ordinateurs.

FIBULE implémente la couche physique de transmission de Fibre Channel qui se fait à un débit de 1.0625 Gbits/sec. Afin d’assurer la compatibilité avec la norme Fibre Channel, la liaison physique est réalisée par un signal différentiel respectant les niveaux ECL (2 fils dans chaque sens). FIBULE peut transmettre directement sur des câbles coaxiaux ou s'interfacer avec des transformateurs optiques.

L'interface du circuit FIBULE est compatible avec le document "10-Bit Interface Specification" (révision 0.30a, Déc1994) qui définit une interface de signaux commune et standard entre la couche physique de Fibre Channel et la couche du protocole.

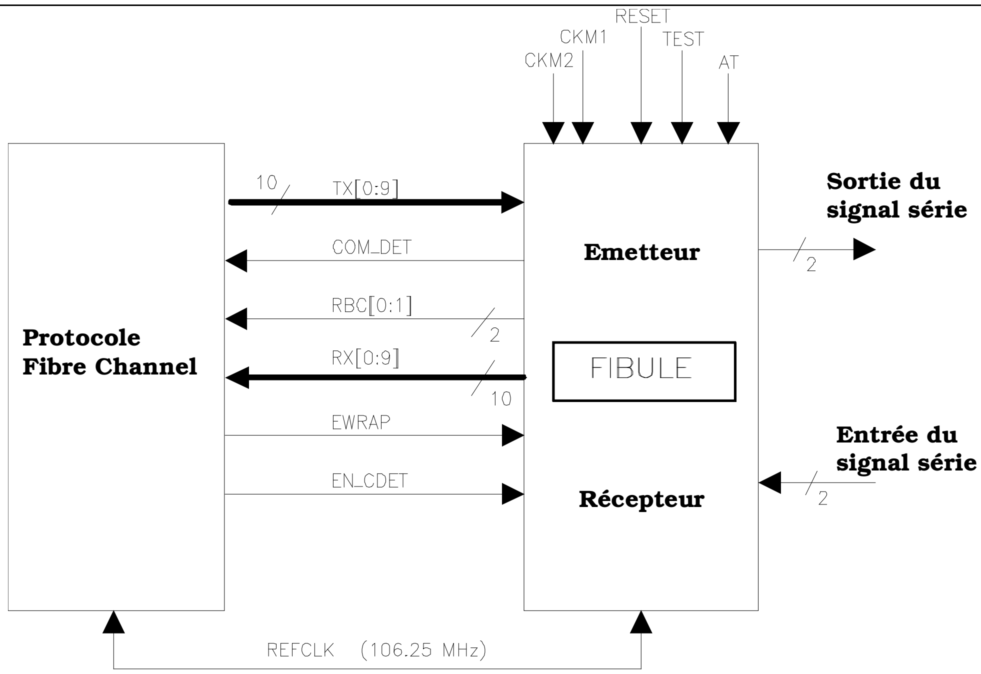


Schéma de l'interface de FIBULE

L'interface de FIBULE comporte les signaux suivants :

- COM_DET est lié à la détection du mot de synchronisation COMMA.
- EN_CDET permet de valider la recherche de détection du mot de synchronisation COMMA.
- EWRAP valide le rebouclage en interne de la sortie du sérialiseur sur l'entrée du désérialiseur.
- RBC[0:1] sont les signaux d'horloge récupérés par le désérialiseur dans le signal série. Ces deux signaux sont à la même fréquence et en opposition de phase. Quand la liaison est calibrée, leur fréquence est égale à la fréquence moitié de la fréquence-byte du signal série.
- REFCLK est l'horloge système qui définit le débit de transmission sur la liaison série (106.25 MHz pour Fibre Channel).
- RX+, RX- sont les entrées différentielles du signal série pour le désérialiseur.
- RX[0:9] est la donnée parallèle sur 10 bits présentées en sortie du désérialiseur.
- TX+, TX- sont les sorties différentielles du signal série pour le sérialiseur.

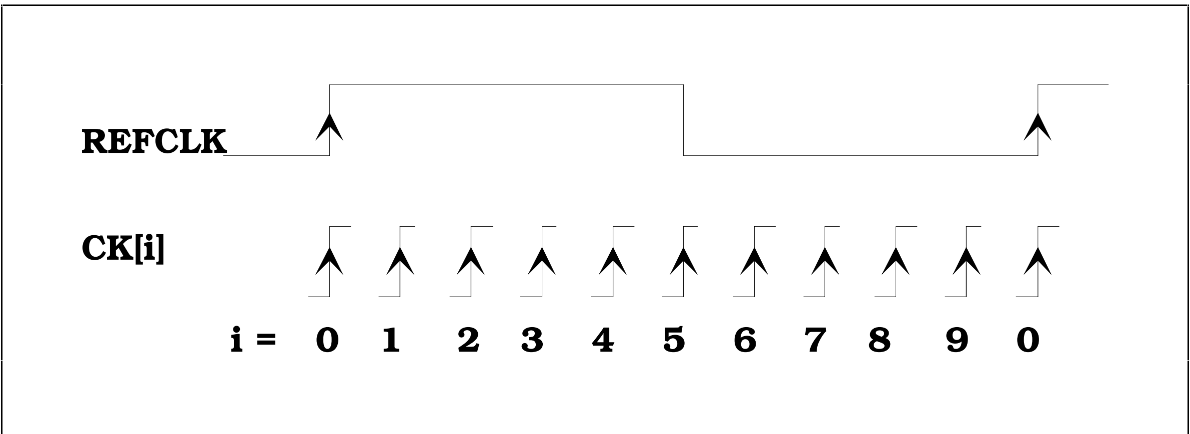
- TX[0:9] est la donnée parallèle sur 10 bits présentée en entrée du sérialiseur. L'ordre d’émission des bits est de TX[0] à TX[9].

B/ Le sérialiseur

Le sérialiseur reçoit en entrée des mots de 10 bits TX[0:9] et un signal d’horloge local REFCLK généré par un quartz. Les mots sont pris en entrée du sérialiseur sur le front montant de REFCLK. Les 10 bits TX[0:9] sont envoyés un par un en sortie de l’arbre de OU-exclusifs en commençant par TX[0] et en terminant par TX[9].

1/ Principe de fonctionnement

Les mots sont traités de la manière suivante :
Chaque bit TX[i] pour i = 0,1,...,9 est échantillonné dans une bascule sur le front montant de REFCLK.
On utilise une boucle à verrouillage de délai (DLL) pour générer une série de signaux d’horloge CK[0:9] de même fréquence que REFCLK. Ces signaux sont décalés en phase les uns par rapport aux autres de T/10.



Pour pouvoir échantillonner chaque bit sans risque de métastabilité, on les sépare en deux groupes : le premier TX[0:4] est synchronisé sur CK[6], le second TX[5:9] sur CK[1]. On obtient alors le bus Q[0:9]

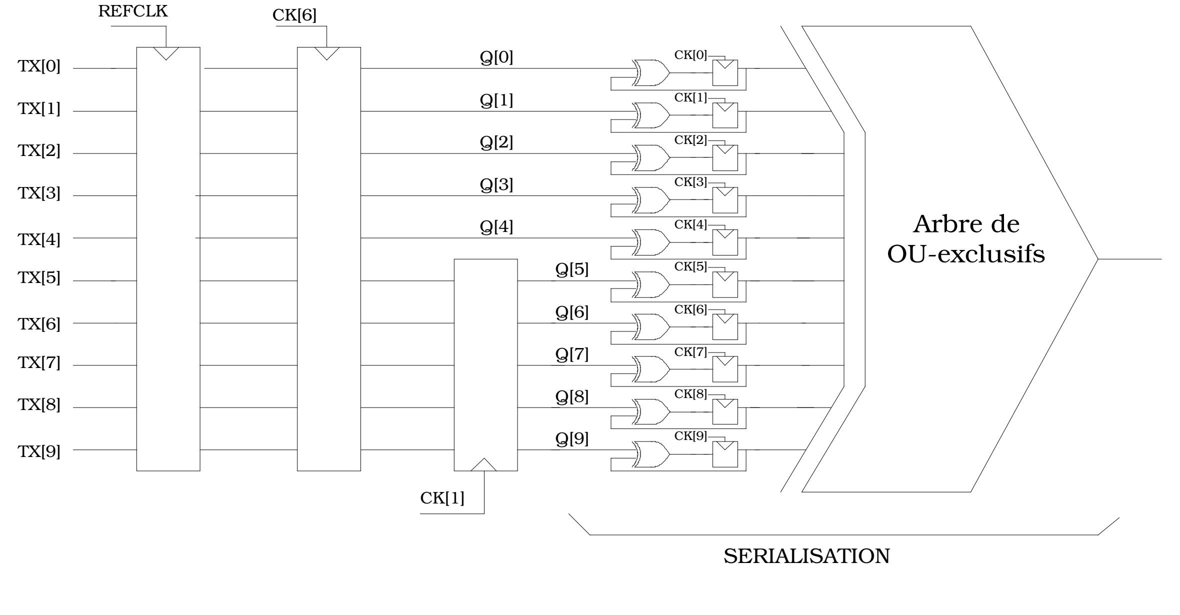
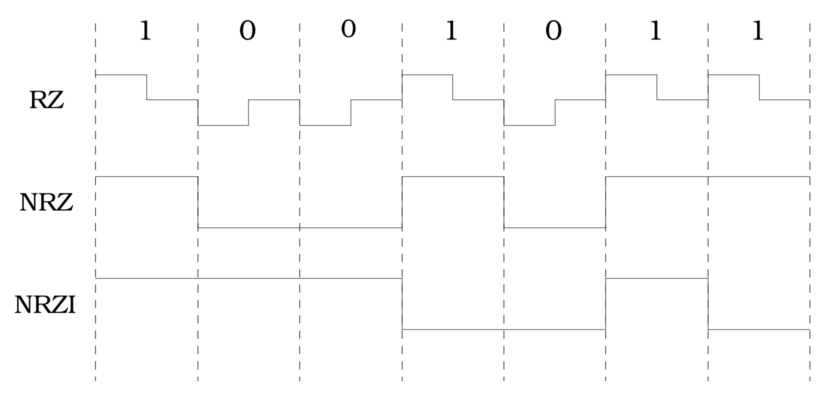


Schéma général du sérialiseur

La sérialisation des bits Q[0:9] est réalisée en décalant les bits dans le temps par échantillonnage avec le signal CK[i] correspondant et en traversant l'arbre de OU-exclusifs décrit au chapitre IV.

2/ Codages NRZ et NRZI

L'opération de sérialisation réalise sur le mot une transformation NRZ/NRZI.
Pour mieux comprendre ce phénomène et ses conséquences, nous rappelons ici les principes des codages NRZ et NRZI et le passage de l'un à l'autre.



RZ : Return to Zero

Le signal série passe à Vdd pour une valeur 1 puis revient à la masse. Il passe à -Vdd pour une valeur 0 puis revient à la masse.

Ce signal présente l'avantage de comporter beaucoup de transitions ce qui facilite la récupération de l'horloge transmise.
Par contre, le fait de coder chaque bit avec deux transitions limite le débit du signal série.

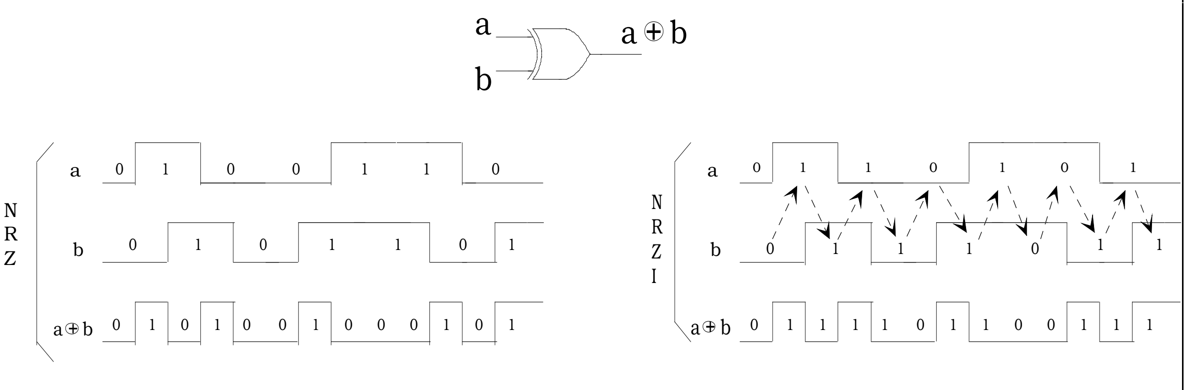
NRZ : Non Return to Zero

Le signal série passe à Vdd pour une valeur 1 et à Vss pour une valeur 0.
C'est le codage le plus simple et le plus utilisé. Pour garantir une bonne récupération du signal d'horloge, il faut éviter de transmettre de longues séries de 0 ou de 1.
Pour une même fréquence de transmission, le débit sera deux fois plus important avec un codage NRZ qu'avec un codage RZ.

NRZI : Non Return to Zero-Invert on 1

Ce codage est réalisé à partir du codage NRZ. A partir d'une valeur initiale Vdd ou Vss du signal série, chaque 1 déclenche une transition et chaque 0 maintient le signal à la même valeur.
Ce codage est utilisé en remplacement du codage NRZ dans le cas suivant : les données ne comportent jamais plus de N valeurs 0 consécutives. En choisissant le codage NRZ-Invert on 1, on peut garantir au moins une transition tous les N bits sur le signal série. Ceci permet de récupérer l'horloge transmise et de maintenir la synchronisation entre l'émetteur et le récepteur.
On peut aussi choisir que 0 déclenche une transition et 1 maintient la valeur précédente.

Au cours de la sérialisation des données, l'arbre de OU-exclusifs réalise de fait un codage NRZI. Ceci est facile à vérifier sur un OU-exclusif dont les entrées sont déphasées de 180°.

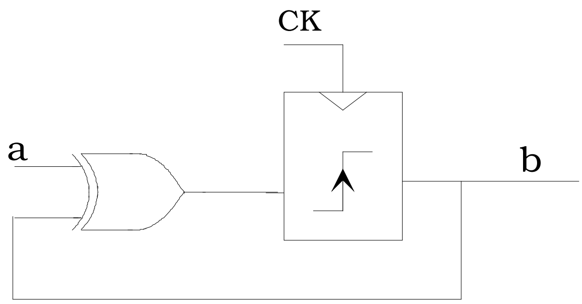


Transformation par un OU-exclusif de signaux codés en NRZ ou en NRZI

Si les signaux d'entrée a et b du OU-exclusif correspondent à un code NRZ, la séquence de bits en sortie du OU-exclusif ne correspond pas aux entrées.
Par contre si les signaux d'entrée a et b correspondent à un code NRZI, la séquence de bits en sortie représente le code NRZI du signal d'entrée sérialisé tel que le montre le parcours des flèches.

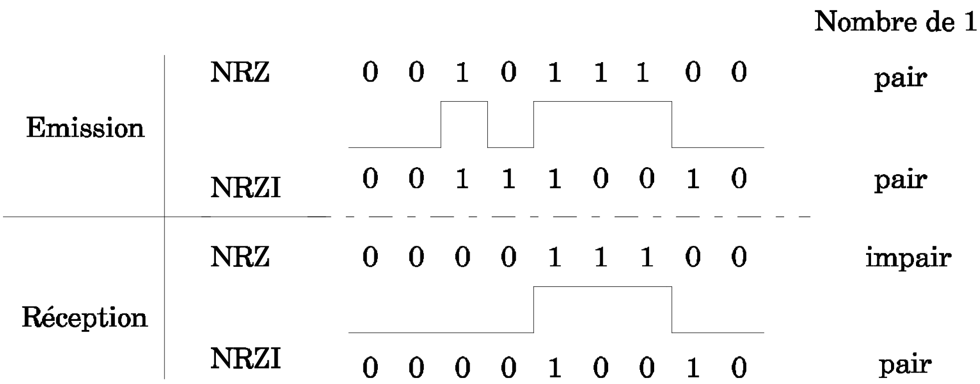
Ainsi en disposant en entrée de l’arbre de OU-exclusifs les bits en parallèle tels que chaque bit soit codé en NRZI et décalé dans le temps par rapport aux autres, on obtient en sortie le signal série codé en NRZI.

TX[0:9] est codé en NRZ. On réalise la transformation NRZ/NRZI sur chaque bit avec le schéma suivant :

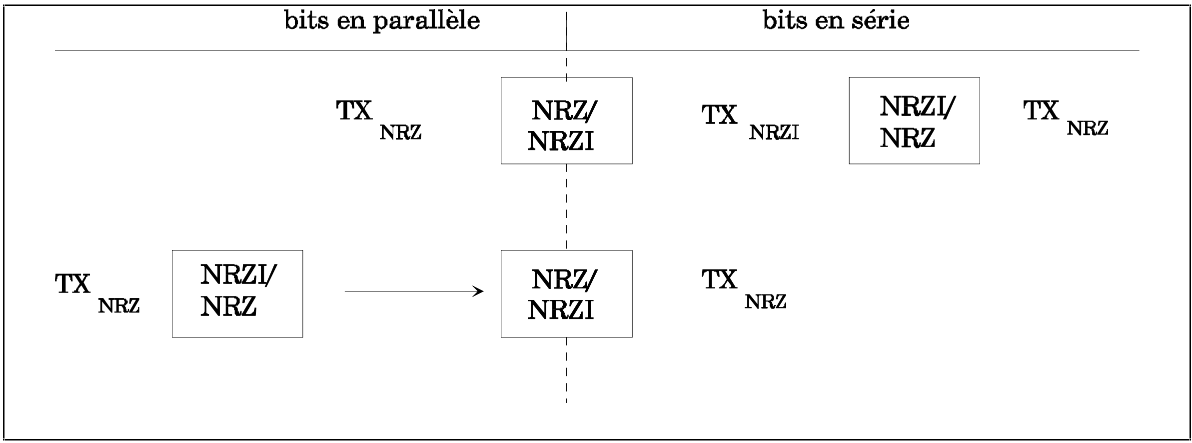


a_{n+1}	b_n	b_{n+1}
0	0	0
0	1	1
1	1	0
1	0	1

La sérialisation transforme les bits en parallèle Q[0:9] codé en NRZ en signal série codé en NRZI. Il reste alors à réaliser la transformation inverse NRZI/NRZ.
Il est préférable de transmettre sur la ligne un signal série codé en NRZ. Son immunité au bruit est meilleure, car on peut détecter une erreur en utilisant un calcul de parité.

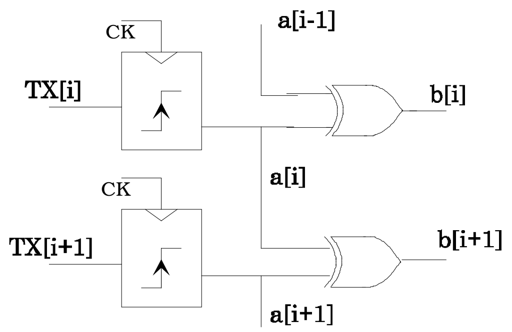


Pour transmettre le signal série en codage NRZ, il faudrait donc en principe réaliser la transformation NRZI/NRZ à la sortie de l’arbre de OU-exclusif, c’est-à-dire sur un signal à 1 Gbit/sec.
Pour pallier à cette difficulté, on choisit de placer la transformation NRZI/NRZ en amont de son inverse en utilisant la propriété de commutativité de cette transformation.



A la sortie du sérialiseur, on dispose directement du signal série codé en NRZ.

La transformation NRZI/NRZ est réalisée par le schéma suivant :

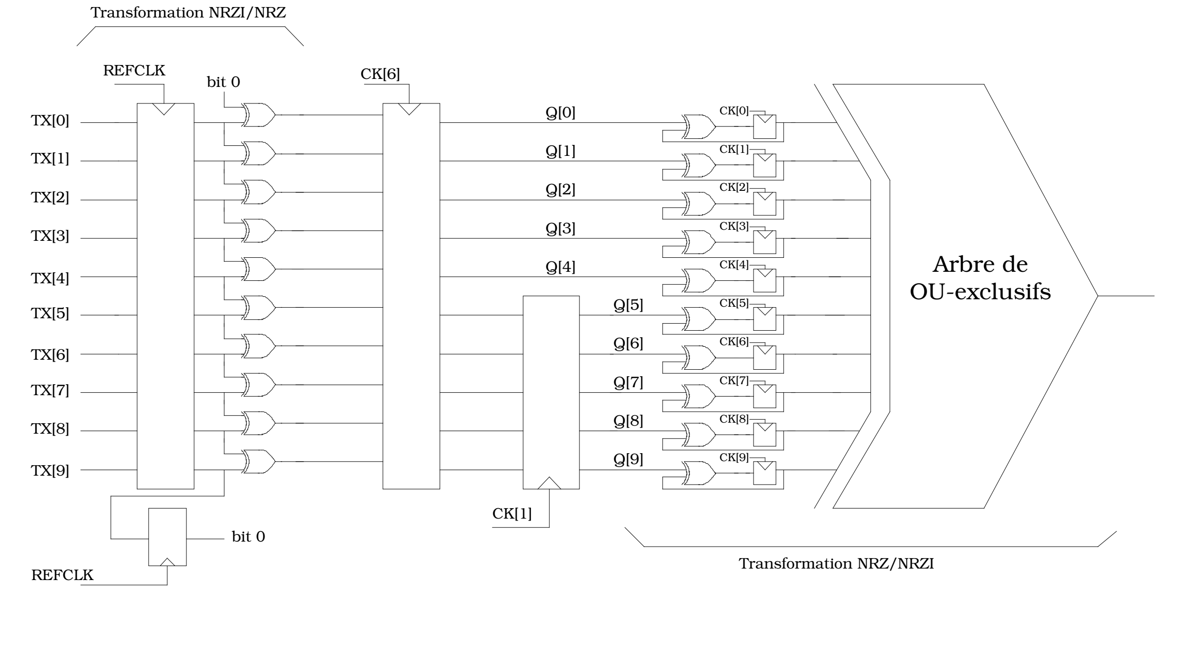


Transformation NRZI/NRZ

a[i]	a[i+1]	b[i+1]
0	1	1
1	1	0
1	0	1
0	0	0

Deux bits consécutifs différents correspondent à une transition donc à un 1. Deux bits consécutifs de même valeur correspondent à un 0.

Le schéma complet du chemin de données du sérialiseur est finalement celui présenté ci-dessous.
Ce schéma est intégralement réalisé en cellules ‘custom’.



C/ Le désérialiseur

Le désérialiseur reçoit en entrée le signal série différentiel RX^+, RX^- et le signal d'horloge de référence REFCLK généré par le quartz d'horloge du sérialiseur. Lorsque la transmission se fait entre deux chips séquencés par des quartz différents, mais spécifiés pour la même fréquence de fonctionnement, la dispersion des quartz induit une différence relative entre l'horloge de l'émetteur et celle du récepteur inférieure à 10^{-4}. Le désérialiseur présente en sortie les mots en parallèle RX[0:9] synchronisés sur un des signaux d'horloge récupérés RBC[0] ou RBC[1].
La seule condition sur le codage des données est que le signal série reçu comporte au moins une transition par byte.

Le désérialiseur propose en sortie deux signaux d'horloges récupérés RBC[0] et RBC[1]. Dans le cadre du protocole Fibre Channel, les mots de 10 bits sont regroupés par 4 et l'alignement est tel que les mots 1 et 3 sont disponibles sur le front de RBC[0] et les mots 0 et 2 sur le front de RBC[1].
Si aucune donnée n'est présente sur l'entrée série ou si le signal présent ne correspond pas aux normes du désérialiseur définies ci-dessus, la fréquence des signaux d'horloge RBC[0:1] sera la fréquence moitié de REFCLK.

Le désérialiseur convertit le signal série en mots de 10 bits. Si l'entrée EN_CDET est placée à 1, la procédure de recherche des limites des mots est lancée. Le regroupement correct des bits en mots est réalisé en cherchant à reconnaître dans le flot de bits un caractère spécial nommé COMMA. Le caractère COMMA est programmable par l'utilisateur. Par défaut, sa valeur est fixée à COMMA[0:9] = 0011111010 qui est le caractère K28.5 défini spécifiquement pour cette utilisation par le protocole Fibre Channel.

Quand le désérialiseur reçoit les premières données, la calibration de la boucle à verrouillage de retards et l'alignement des mots ne sont pas réalisés. Le signal COM_DET est à 0. Ce signal a deux significations possibles :

- Dans le cadre du protocole Fibre Channel, il passe à 1 à chaque fois qu'un caractère COMMA est détecté en sortie du désérialiseur. Le fait de détecter une dizaine de COMMA permet de conclure que le désérialiseur est calibré.

- Hors de ce cadre, COM_DET passe à 1 lorsque le désérialiseur a reçu dix caractères COMMA et reste à 1 ensuite.

Le passage d'une signification à l'autre se fait par l'intermédiaire d'un bit du chemin de 'scan'.

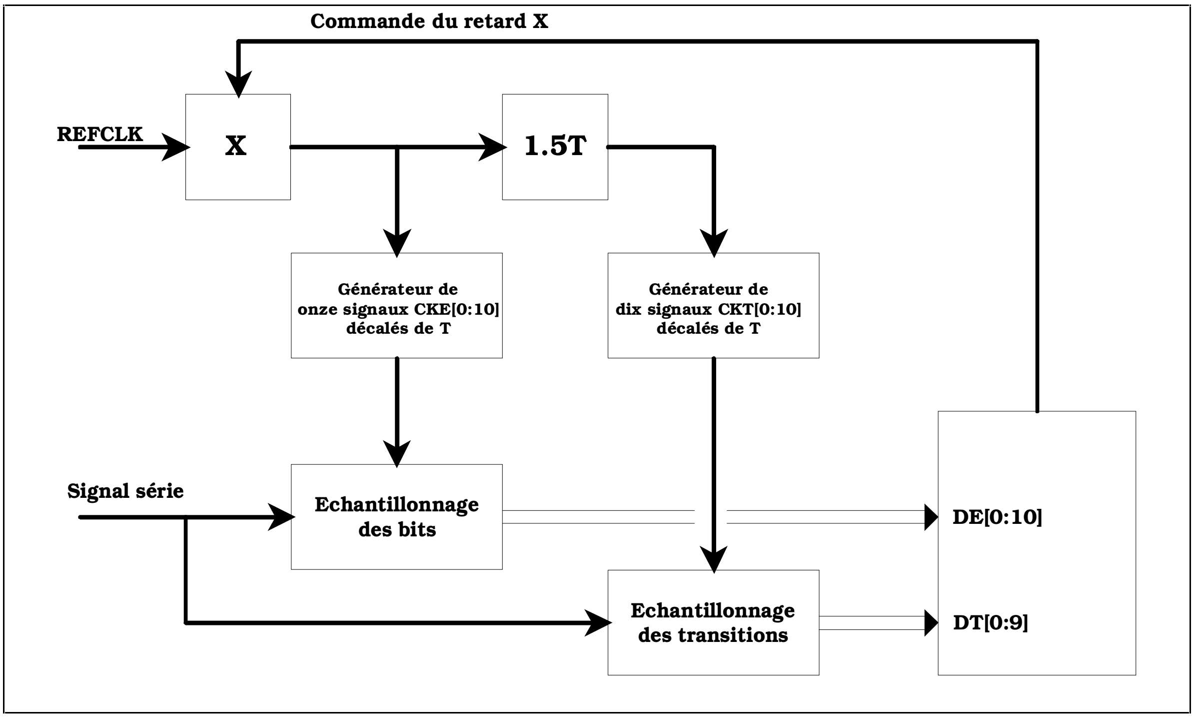
Le désérialiseur de FIBULE est une application directe des principes exposés au chapitre III. Le signal série est échantillonné par deux séries de signaux d'horloge générées par une boucle à verrouillage de retards. Ces signaux d'horloge sont alignés sur le signal série par l'intermédiaire du retard X dont la valeur modulo T (où T est la durée moyenne d'un bit dans le signal série) fluctue dans le temps pour suivre le décalage en phase entre le signal d'horloge de référence du récepteur et celui de l'émetteur.

Pour approfondir le fonctionnement du désérialiseur, nous allons détailler :

- la réalisation de la commande du retard X
- la réalisation du signal d'horloge récupéré CKR
- le traitement des données
- le principe d'alignement des mots à l'aide du mot de référence COMMA

1/ Réalisation de la commande du retard X

La détermination de la valeur du retard X se fait en comparant les résultats de l'échantillonnage du signal série pour les données et pour les transitions (cf. chapitre III).



Le retard X sert à compenser la différence de phase entre le signal d'horloge de référence REFCLK et le signal série. Cette différence de phase évolue dans le temps puisque la précision des quartz qui génèrent les signaux d'horloge de référence de l'émetteur et du récepteur est de 10^{-4}.
Pour atteindre le débit standard de 1.0625 Gbits/sec défini par le protocole Fibre Channel, la liaison série fonctionne avec un signal d'horloge REFCLK de fréquence 106.25 MHz. Dans ces conditions, nous allons chercher à déterminer quelle différence de phase le retard X doit compenser par cycle d'horloge.

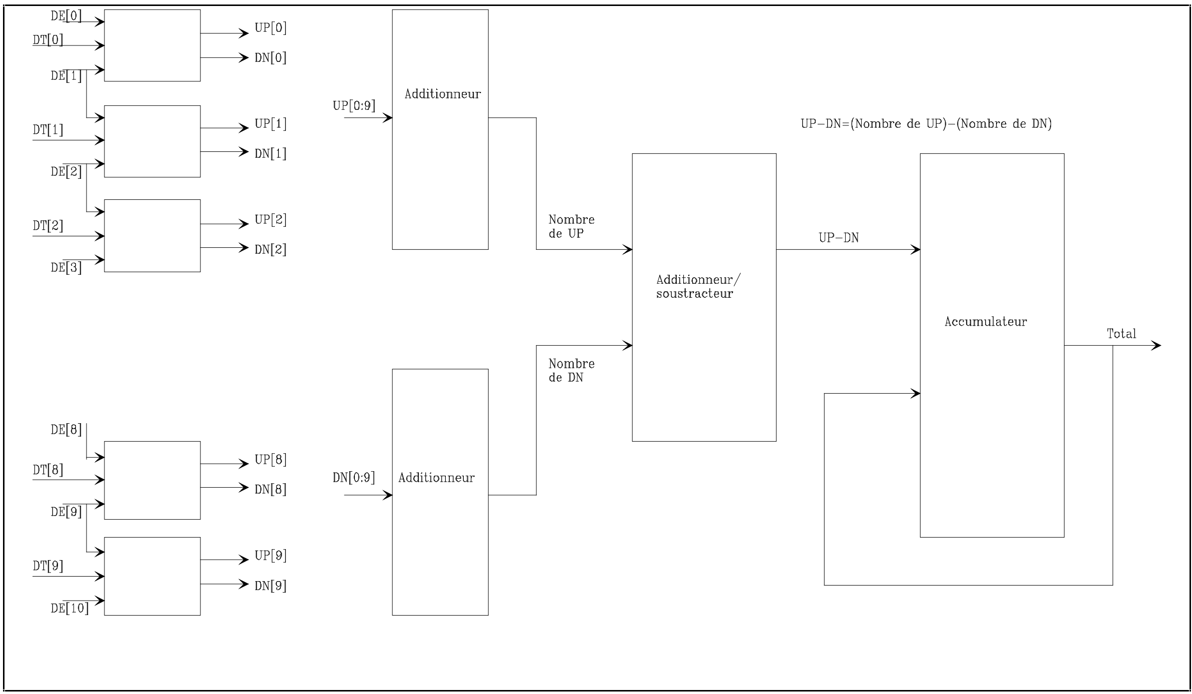
Soit F1 la fréquence du signal d'horloge de l'émetteur, et T1 la période associée
et F2 la fréquence du signal d'horloge du récepteur, et T2 la période associée
l'imprécision sur la valeur en fréquence des quartz s'écrit : $\frac{F1-F2}{F1} \leq 10^{-4}$

Elle s'écrit aussi : $1-\frac{T1}{T2} \leq 10^{-4}$ c'est-à-dire $T2-T1 \leq T2 \times 10^{-4}$

L'évolution du décalage de phase entre F1 et F2 pour un cycle d'horloge est donc au maximum de :

$$T2-T1 = 0.941 \times 10^{-12} \text{ sec} = 0.941 ps$$

Sachant que le pas minimum du retard X est de 40 ps, on voit qu'il suffit de faire varier le retard X une fois tous les 40 cycles pour garantir que le récepteur restera en phase avec le signal série.
En pratique la valeur du retard X est réévaluée tous les dix cycles. Le schéma suivant détaille le processus employé.



Pour chaque bit d’une série de 10, on compare la valeur DE[i] du bit, celle du bit suivant DE[i+1] et la valeur échantillonnée au moment de la transition virtuelle entre ces deux bits DT[i]. Trois cas sont possibles :

DE[i] = DE[i+1]	Pas de transition		UP[i]=0	DN[i]=0
DE[i] ≠ DE[i+1]	Transition réelle	Si DT[i] = DE[i],	UP[i]=0	DN[i]=1
		Si DT[i] = DE[i+1],	UP[i]=1	DN[i]=0

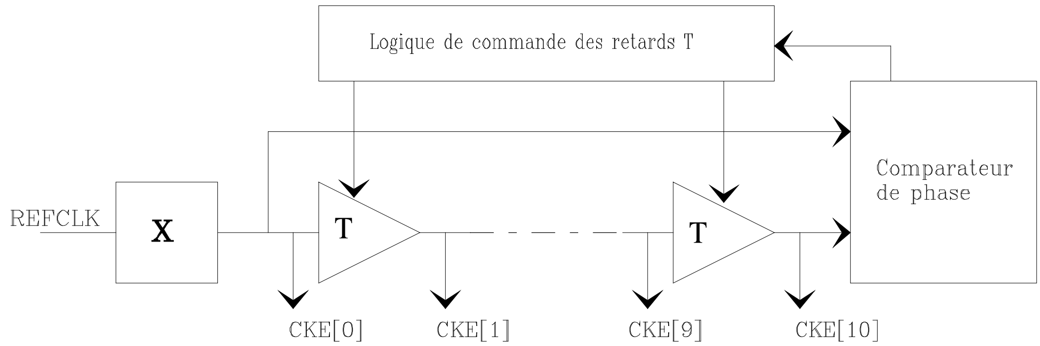
On utilise 10 bascules UP[i] et 10 bascules DN[i].
Au cycle suivant, on effectue la somme séparément des UP[0:9] et des DN[0:9]. Puis on soustrait le nombre de DN au nombre de UP et le nombre ainsi obtenu est accumulé sur une dizaine de cycles.

A ce terme, si le total accumulé est supérieur à 2, le retard X est incrémenté d’une unité et si le total est inférieur à -2, le retard X est décrémenté d’une unité.
Si le total obtenu est compris entre +2 et -2, la valeur du retard n’est pas modifiée. Le décalage détecté n’est pas suffisamment significatif pour être pris en compte. Ce mécanisme permet de ne pas tenir compte des effets de jitter observés sur le train de bits entrant.

Comme expliqué au chapitre III, la valeur du retard X varie dans un intervalle compris entre Xmin et Xmax où Xmin est fixé par l’utilisateur et Xmax = Xmin + T,
T étant la période-bit du signal série.

2/ Réalisation des signaux d’horloge récupérés

Le signal d'horloge récupéré CKR est choisi parmi les dix signaux CKE[0:10] générés par la boucle à verrouillage de retards.

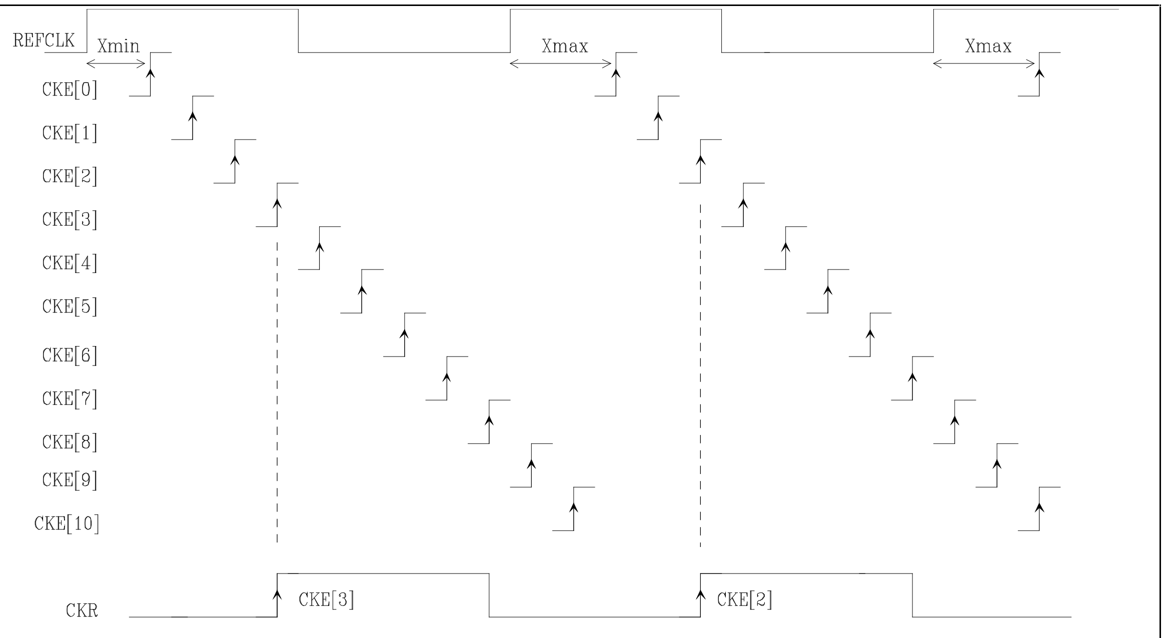


Génération des signaux d'horloge CKE[0:10]

Chacun des signaux CKE[0:10] a la même fréquence que le signal d'horloge de référence du sérialiseur et est aligné en phase sur le signal série (grâce à l'ajustement de la valeur du retard X). Potentiellement chacun des CKE[0:10] pourrait donc être utilisé comme signal d'horloge récupéré.
Cependant la période des signaux d'horloge CKE[0:10] n'est pas constante. Quand la valeur du retard X passe de Xmin à Xmax ou inversement, la période des signaux CKE[0:10] varie pendant un cycle de la durée d'une période-bit.

La méthode adoptée est donc la suivante : On définit au départ que CKR est égal à l'un quelconque des signaux CKE[i]. Puis à chaque fois que se produit un saut de Xmin vers Xmax le choix de CKR passe de CKE[i] à CKE[i-1], et pour un saut de Xmax vers Xmin de CKE[i] à CKE[i+1].
On utilise un registre à décalage cyclique (CK[10] = CK[0]) dans lequel un 1 est décalé à chaque saut de la valeur du retard X. Ce registre commande un multiplexeur qui sélectionne le signal CKE[i] choisi comme CKR.

Bien entendu, le changement de choix de CKR d'un signal CKE[i] à un autre doit se produire pendant l'état bas du signal CKE[i] considéré.



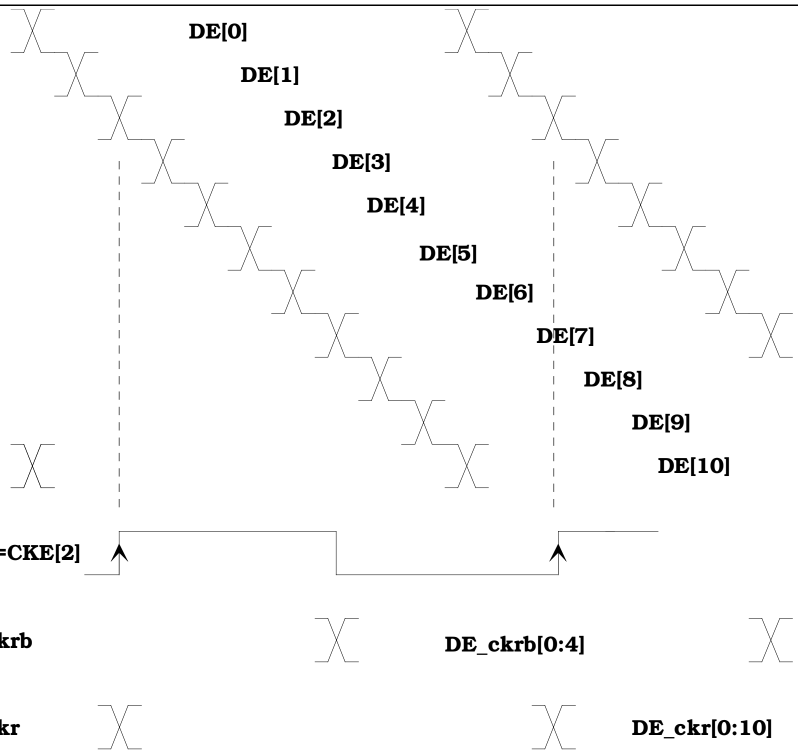
Exemple du changement de choix de CKR
pour un saut de la valeur du retard X de Xmin à Xmax

A partir de CKR, on réalise RBC[0] et RBC[1] en utilisant un diviseur de fréquence par 2. Le choix de ces deux signaux comme signaux d'horloge récupérés respecte l'interface standard définie pour les liaisons série compatibles avec le protocole Fibre Channel.

3/ Traitement des données

Le signal série est échantillonné par la série de signaux d'horloge CKE[0:10] générés par la boucle à verrouillage de retards. Une série de signaux CKE[0:10] échantillonne 11 bits consécutifs DE[0:10] du signal série. Les valeurs obtenues sont resynchronisées sur CKR. En sortie du désérialiseur, les données doivent être synchrones avec RBC[0] ou RBC[1].

La synchronisation est un peu difficile à réaliser dans la mesure où CKR est un des signaux CKE[0:9] et que CKR peut être successivement égal à chacun des ces signaux. Suivant la valeur de CKR, l'échantillonnage de la donnée DE[i] se fait directement par CKR (la donnée DE[i] est stable sur le front montant de CKR) ou on pré-échantillonne la donnée avec le signal inverse de CKR (la donnée DE[i] est stable sur le front descendant de CKR).



Chronogramme d’échantillonnage des données DE[i]
avec CKR=CKE[2]

Dans l’exemple ci-dessus, les données DE[5] à DE[9] sont échantillonnées directement par CKR; les données DE[0] à DE[4] sont échantillonnées successivement par l’inverse de CKR et CKR.

Comme la répartition des bits entre l’échantillonnage simple et l’échantillonnage double évolue dans le temps, on réalise les deux traitements sur chaque bit et un multiplexeur sélectionne la valeur adaptée en fonction de la valeur de CKR.

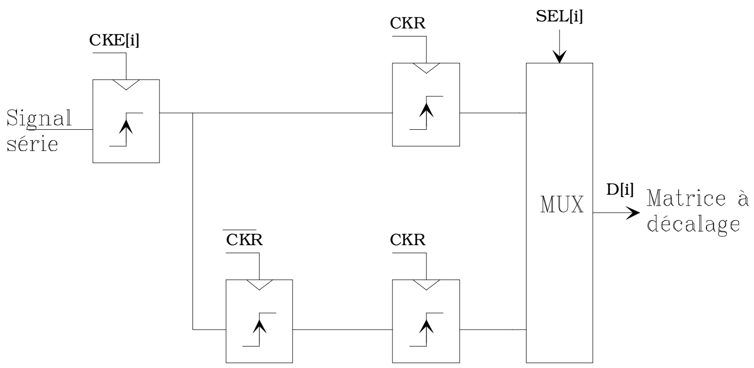


Schéma de synchronisation des données sur CKR

Les données synchronisées sur CKR sont présentées en entrée de la matrice à décalage qui réorganise l’ordre des bits en fonction de la valeur de CKR pour retrouver l’ordre original des bits dans le mot.

4/ Principe d’alignement des mots à l’aide de COMMA

Le signal série est échantillonné par des séries de signaux d’horloge CKE[0:10]. On obtient ainsi des séries de 11 bits consécutifs. Ces 11 bits ne constituent pas forcément un mot tel qu’ils ont été émis en entrée du sérialiseur. Pour retrouver les mots d’origine, on effectue une recherche des limites des mots. Cette procédure est lancée en plaçant le signal EN_CDET à 1.

Etant donné que l’on ne veut imposer aucune contrainte sur le codage, le procédé choisi est ponctuel. Il s’effectue dans un temps donné jusqu’à obtention d’un regroupement correct des bits en mots. Aucun contrôle n’est exercé sur cet alignement une fois qu’il a été trouvé. Si l’utilisateur détecte une erreur d’alignement, il peut toutefois relancer la procédure de recherche des limites des mots en plaçant EN_CDET à 0 pendant un cycle de REFCLK au moins puis de nouveau à 1.

Pour regrouper correctement les bits, on cherche à identifier dans le signal série un mot spécifique de 10 bits nommé COMMA tel que la série de bits qui compose ce caractère ne peut pas survenir par hasard pendant la phase de recherche d’alignement. Le caractère COMMA est programmable par l’utilisateur. Par défaut, sa valeur est fixé à COMMA[0:9]=0011111010 qui est le caractère K28.5 défini spécifiquement pour cette utilisation par le protocole Fibre Channel.

Le groupe de 10 bits proposé en sortie de la matrice à décalage est comparé avec le mot COMMA.

- Si, au bout de 32 groupes de 10 bits, le mot COMMA n’a pas été reconnu le signal d’horloge CKR est décalé de la durée d’un bit (CKR passe de CKE[i] à CKE[i+1]). La recherche reprend sur les 32 groupes de 10 bits suivants.
  Quand CKR parcourt toutes les valeurs de CKE[0:9], on obtient tous les regroupements possibles de 10 bits consécutifs.

- Lorsque le mot COMMA est identifié dans le signal série, la procédure s’arrête; COM_DET signale à l’utilisateur la détection du mot COMMA et la transmission de données peut commencer.

5/ Schéma récapitulatif du désérialiseur

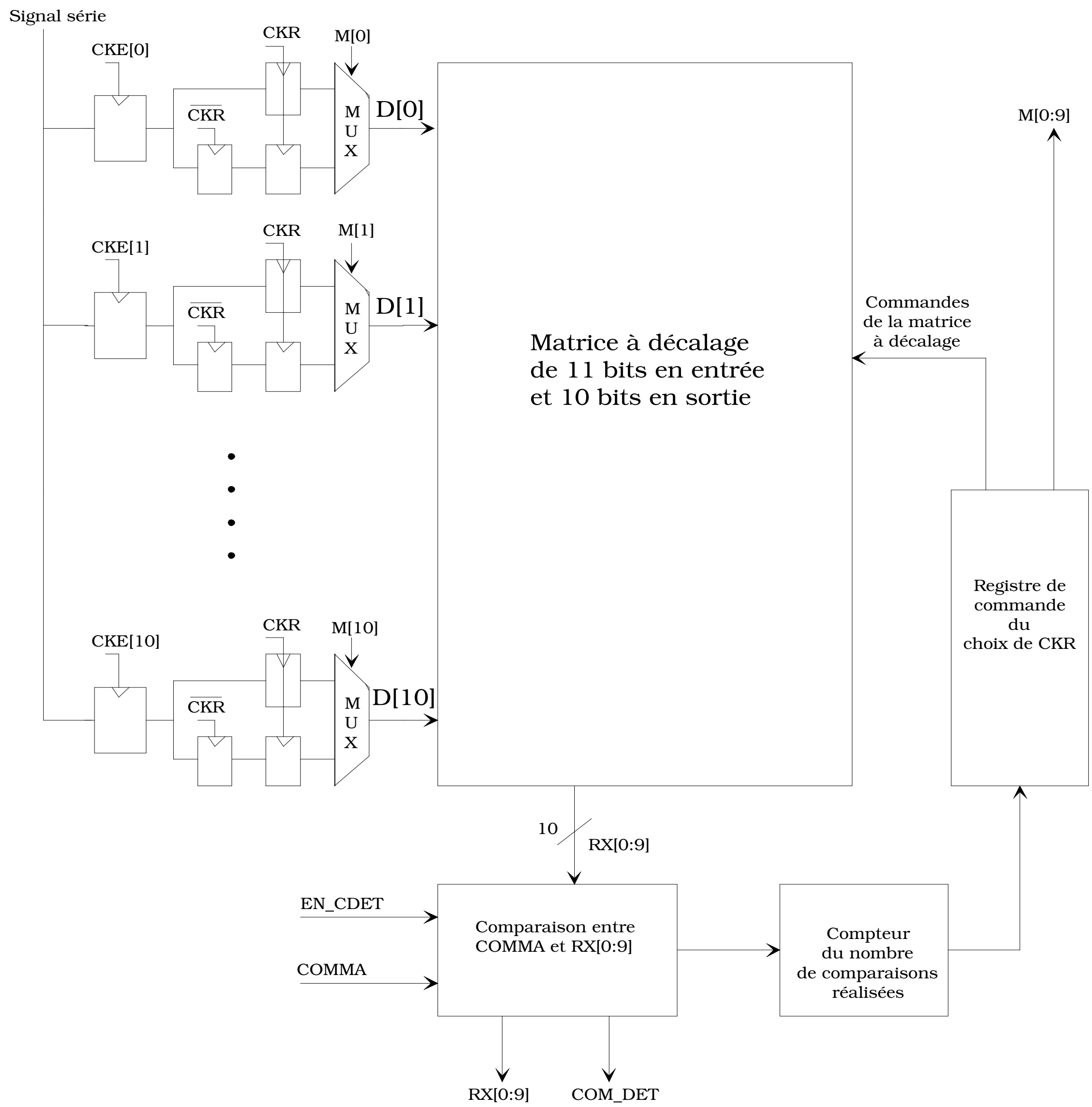


Schéma du désérialiseur

Dans ce schéma, la partie qui concerne le traitement des données c'est-à-dire l'échantillonnage du signal série, la synchronisation des données sur CKR et la matrice à décalage sont réalisées en cellules 'custom'.
Les commandes logiques sont réalisées en cellules pré-caractérisées.

D/ Le bloc auto-test

Ce bloc a pour but de permettre de tester les principales fonctionnalités de la liaison série au niveau "wafer" c'est-à-dire avant la mise en boîtier.
Il comprend un générateur de données aléatoires, un oscillateur à commande digitale qui génère le signal d'horloge système CKosc et un comparateur de données.

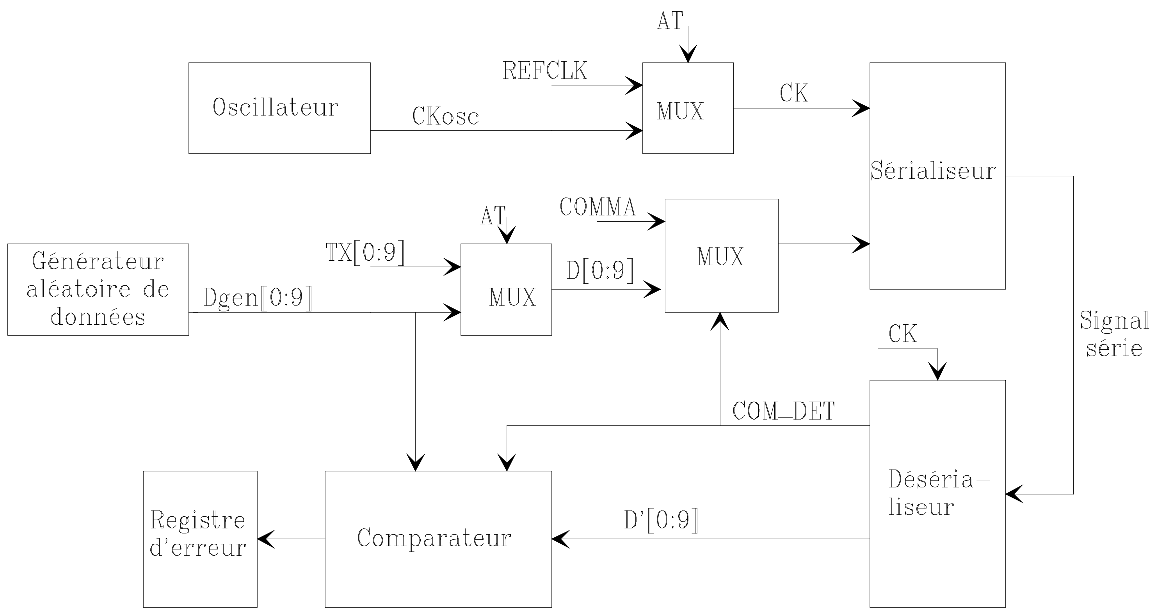


Schéma du bloc auto-test

Le principe de fonctionnement de l'auto-test utilise le rebouclage interne de la sortie du sérialiseur sur l'entrée du désérialiseur.
Pendant la phase de calibration de la liaison, le mot de synchronisation est envoyé en continu.
Quand ce mot est détecté par le désérialiseur le générateur de données aléatoires se met en marche. Les mots générés sont présentés en entrée du sérialiseur. A la sortie du désérialiseur, on effectue la comparaison entre les mots reçus et les mots émis. Si une erreur est détectée, le fait est mémorisé dans un registre. Ce registre est accessible à l'utilisateur par le chemin de SCAN. Il est donc facile de venir vérifier le bon fonctionnement de chaque circuit.

L’oscillateur est réalisé par un ensemble d’éléments de retard contrôlés numériquement (identiques à ceux des boucles DLL) et connectés en anneau. La fréquence de l’oscillateur est commandée par un registre accessible par le chemin de SCAN. Cette fréquence peut donc être ajustée par l’utilisateur en fonction de la rapidité intrinsèque du circuit.

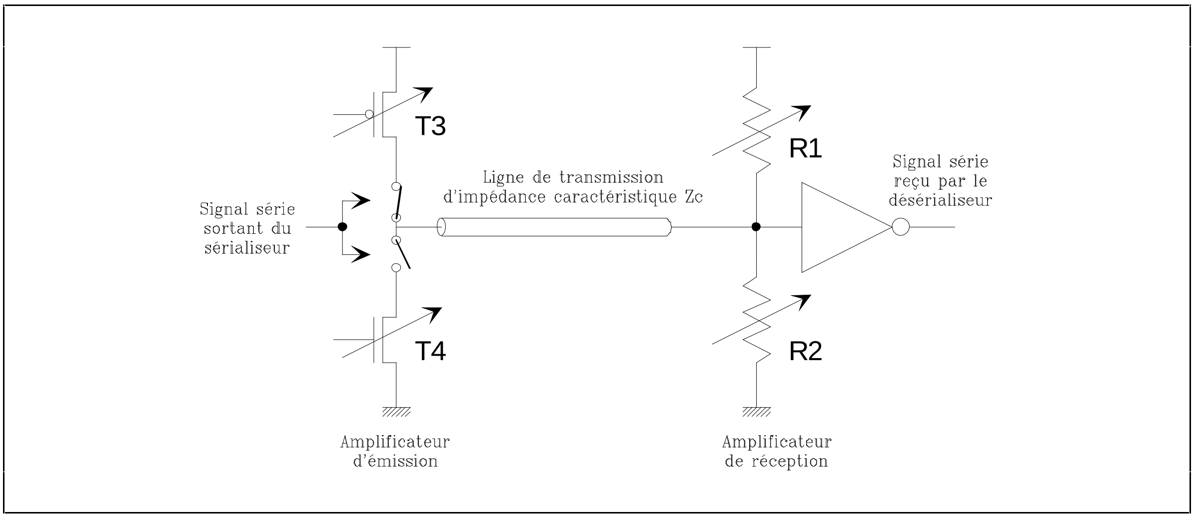
Le nombre d'octets transmis pendant la phase d'auto-test est comptabilisé dans un registre accessible par le SCAN. Il suffit donc de maîtriser la durée de cette phase pour connaître à quel débit de données s'est effectué le test.

E/ Le bloc d'adaptation d'impédance

La technologie CMOS utilisée pour FIBULE présente une grande dispersion des caractéristiques d'un circuit à un autre. Il faut donc utiliser une circuiterie d'émission et de réception peu sensible à cette dispersion.

D'autre part, dans le cas de FIBULE, il est impératif que l'émetteur et le récepteur du signal série soient compatibles avec les autres circuits aux normes Fibre Channel. La norme en ce domaine sont les signaux émis par les circuits en technologie bipolaire de type ECL (Emitter-Coupled Logic) qui sont caractérisés par une faible excursion des signaux de sortie qui est typiquement de 0.8 Volt.
Cette adaptation est d'autant plus nécessaire que FIBULE se place dans la gamme des transmissions sur fibre optique et doit s'interfacer avec des transformateurs électrique/optique. Ceux disponibles actuellement sur le marché pour des applications dans la gamme du Gbits/sec ont des entrées/sortie ECL.

Pour résoudre ces difficultés, il a fallu mettre en place un procédé d'adaptation d'impédance de la ligne de transmission.



Le schéma ci-dessus présente l'adaptation d'impédance déjà utilisée dans le circuit BULLIT.

R1, R2, T3 et T4 sont des ensembles de transistors MOS en parallèle : le nombre de transistors actifs dans chaque ensemble est déterminé par une boucle intégrée dans le circuit qui garantit que

- l'inverseur qui agit comme un amplificateur du signal série pour le récepteur est correctement équilibré
- la ligne de transmission est adaptée suivant son impédance caractéristique Zc
- l'amplitude des signaux sur la ligne de transmission est de 0.8 Volts.

De cette façon la ligne de transmission est adaptée en permanence et le mécanisme en boucle qui la commande permet de compenser les variations de la température et de la tension d'alimentation.
Le mécanisme de transmission de la macro-cellule sérialiseur/désérialiseur permet donc une transmission sur un seul fil avec un bon taux de réjection du mode commun.

Dans le cas du circuit FIBULE, pour respecter la norme Fibre Channel, la transmission se fait en différentiel. Il faut adapter le principe de BULLIT à l'interface standard définie pour le protocole Fibre Channel.
L'émission se fait donc en différentiel sur deux fils TX+ et TX- qui ont chacun leur amplificateur d'émission.
La réception se fait sur un seul fil en tenant compte uniquement de l'entrée RX+.

F/ Schéma récapitulatif de FIBULE

Le schéma ci-contre présente le circuit FIBULE avec toutes ces fonctionnalités.

FIBULE est conçu pour nous permettre de tester la macro-cellule liaison série. A ce titre, différents éléments ont été intégrés dans le circuit pour offrir un maximum de liberté à l'utilisateur pour le choix des conditions de test :

- directement au niveau du wafer, on réalise le test fonctionnel de transmission de données en interne

- sur carte de test, le circuit est adapté, après application du reset, à l'environnement Fibre Channel ce qui nous permet de vérifier l'adéquation de notre macro-cellule à ce protocole

- sur carte de test, on peut utiliser le chemin de scan pour venir initialiser à la valeur souhaitée les principaux registres de commande de la liaison série. On peut ainsi modifier le mot de référence COMMA, la valeur initiale des retards variables, ouvrir les boucles de contrôle des retards T et 1.5T, etc.

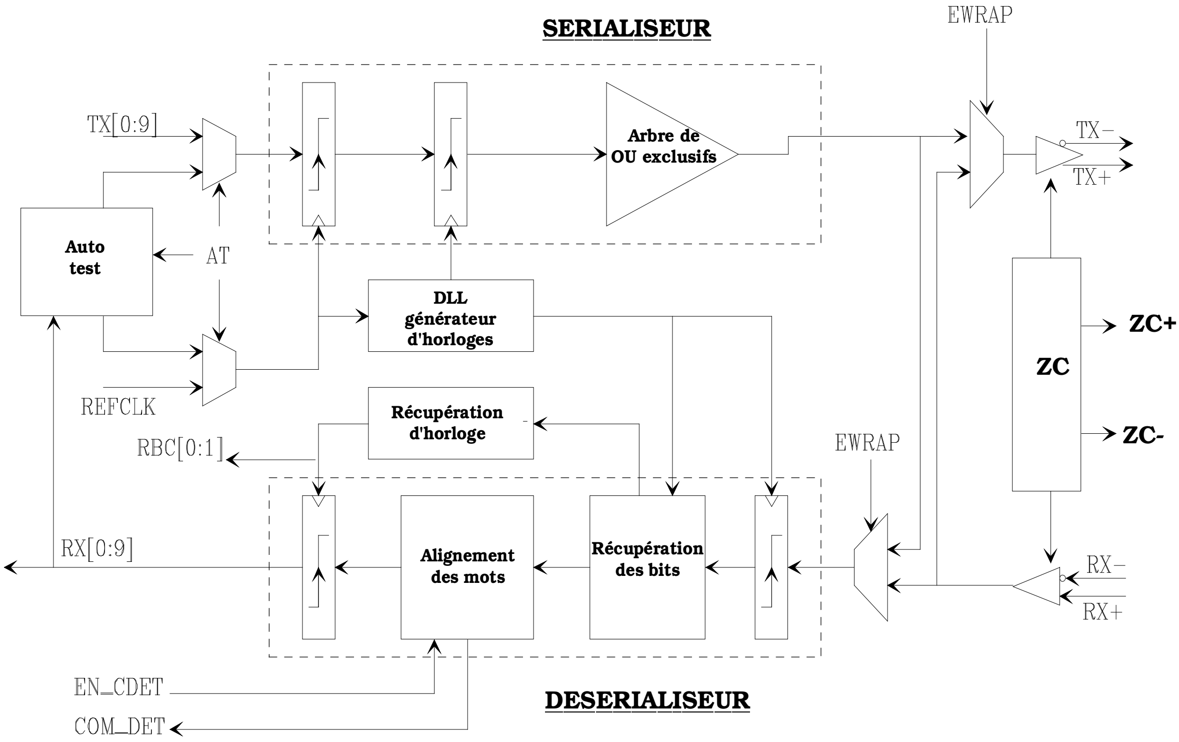


schéma de la liaison série FIBULE

Chapitre VI : Résultats expérimentaux

Cette thèse a abouti à la fabrication du circuit FIBULE qui comprend une macro-cellule liaison série et un circuit d’adaptation d’impédance. L’étude expérimentale de FIBULE doit permettre :

- de valider les principes originaux mis en oeuvre dans la macro-cellule et décrits dans les chapitres III et IV en particulier pour la désérialisation.
- d’évaluer le coût en surface de la macro-cellule liaison série et sa consommation.
- d’identifier les facteurs limitant la fréquence de fonctionnement de la macro-cellule.

Rappelons que FIBULE est conçu pour répondre aux spécifications de la norme Fibre Channel. A ce titre le circuit doit pouvoir transmettre et recevoir un signal série à 1.0625 Gbit/sec.
Le circuit a été fabriqué en deux versions : la première est fabriquée dans un process 0.5µ typique, la deuxième dans un process 0.35µ.
FIBULE a été testé successivement sur la carte de test de BULLIT puis sur une carte de test adaptée à ses fonctionnalités et réalisée par la société SINTEF à partir des spécifications que nous avons fournies.

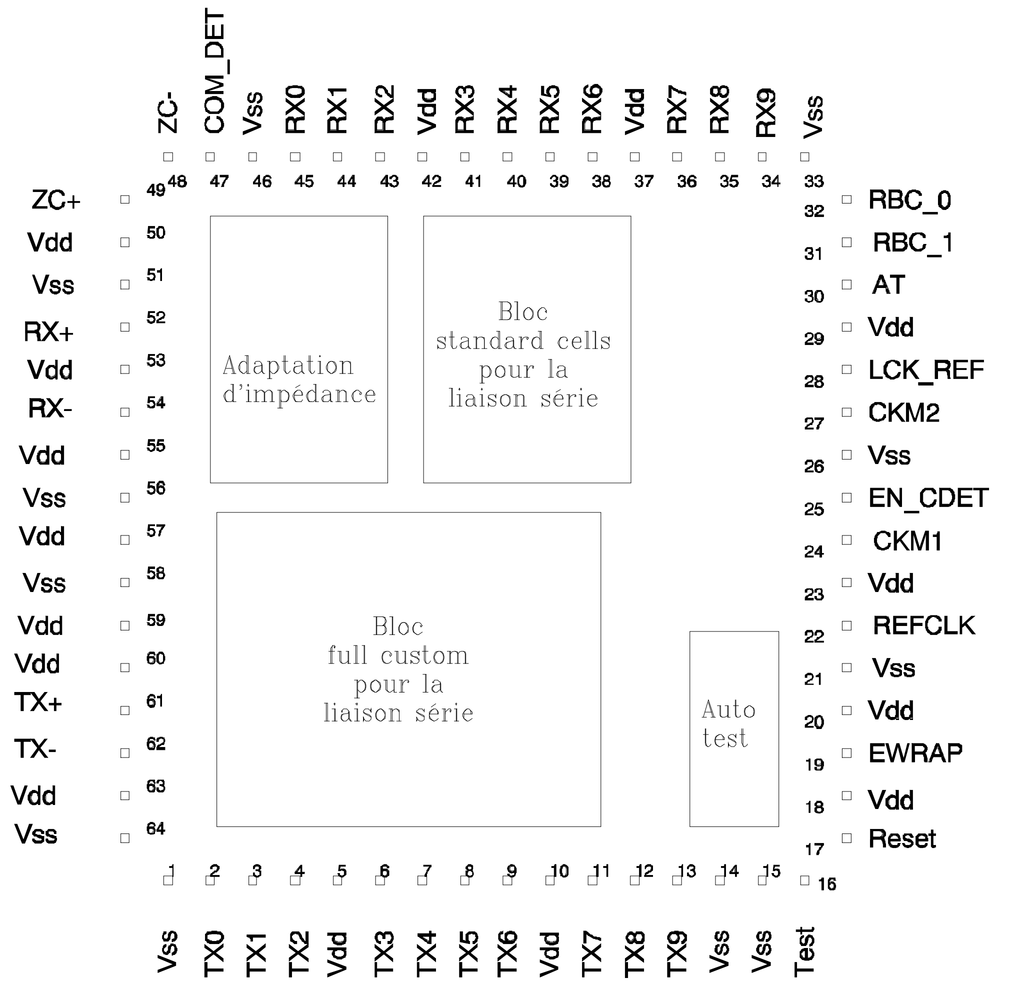
A/ Résultats avant fabrication

FIBULE se présente sous la forme d'un circuit d'environ 22000 transistors. Il comprend :

- un bloc *full custom* qui regroupe la boucle à verrouillage de retards, les différents retards variables et les chemins de données du sérialiseur et du désérialiseur,

- deux blocs constitués de cellules précaractérisées qui gèrent d'une part la logique du désérialiseur (commande du retard X, choix de l'horloge récupérée, synchronisation sur le mot COMMA), d'autre part la logique d'adaptation d'impédance,

- le bloc d'auto-test intégré dans la version 0.35µ de FIBULE qui est également constitué de cellules précaractérisées.

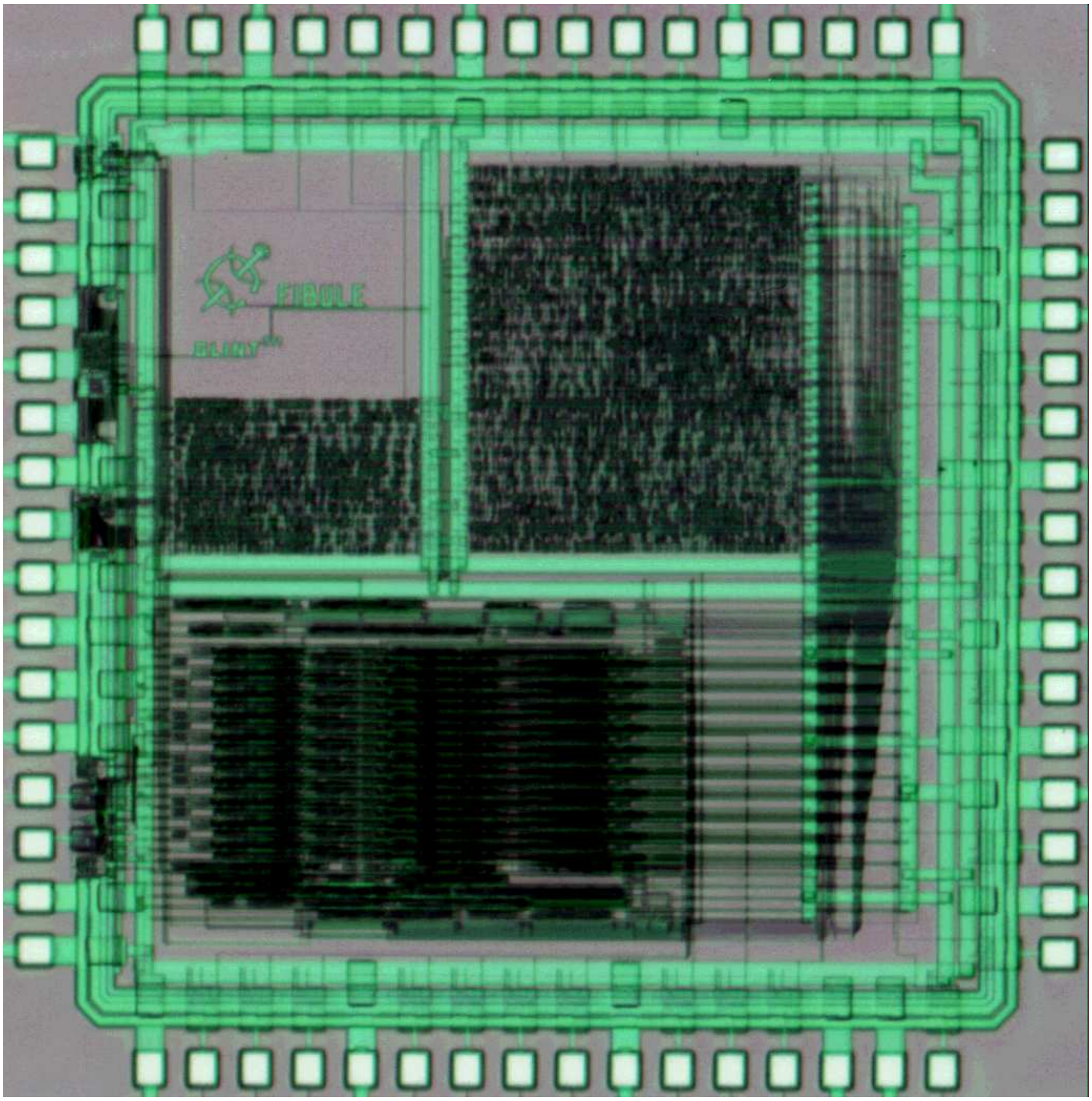


Description et interface du circuit FIBULE version 0.35µ

La surface de la version 0.5µ est de 4.25 x 4.25 mm^2.

La surface de la version 0.35µ est de 3.08 x 3.08 mm^2.

Une photographie du chip dans sa version 0.5µ est présentée ci-dessous:



Les deux versions sont encapsulées dans un boîtier CERQUAD Kyocera (Ref QC-064340 WZ). Les dimensions du boîtier sont 14 x 14 mm^2.

Dans ses deux versions, FIBULE est alimenté en +3.3V. Sa consommation est évaluée (par simulation ELDO) à 0.3 W.

La version 0.35µ de FIBULE est un circuit de test pour l'implantation de circuits intégrés dans ce process. L'analyse de la tranche de silicium contenant les circuits a mis en évidence des défauts de fabrication principalement au niveau des métallisations. Cette version de FIBULE n'a donné aucun résultat fonctionnel.

B/ Modélisation du jitter

La réalisation d'une liaison série comporte un aspect analogique : le traitement du signal série. A cet égard, l'influence des éléments qui réalisent la fabrication, le transport, la réception et l'échantillonnage du signal série est fondamentale.

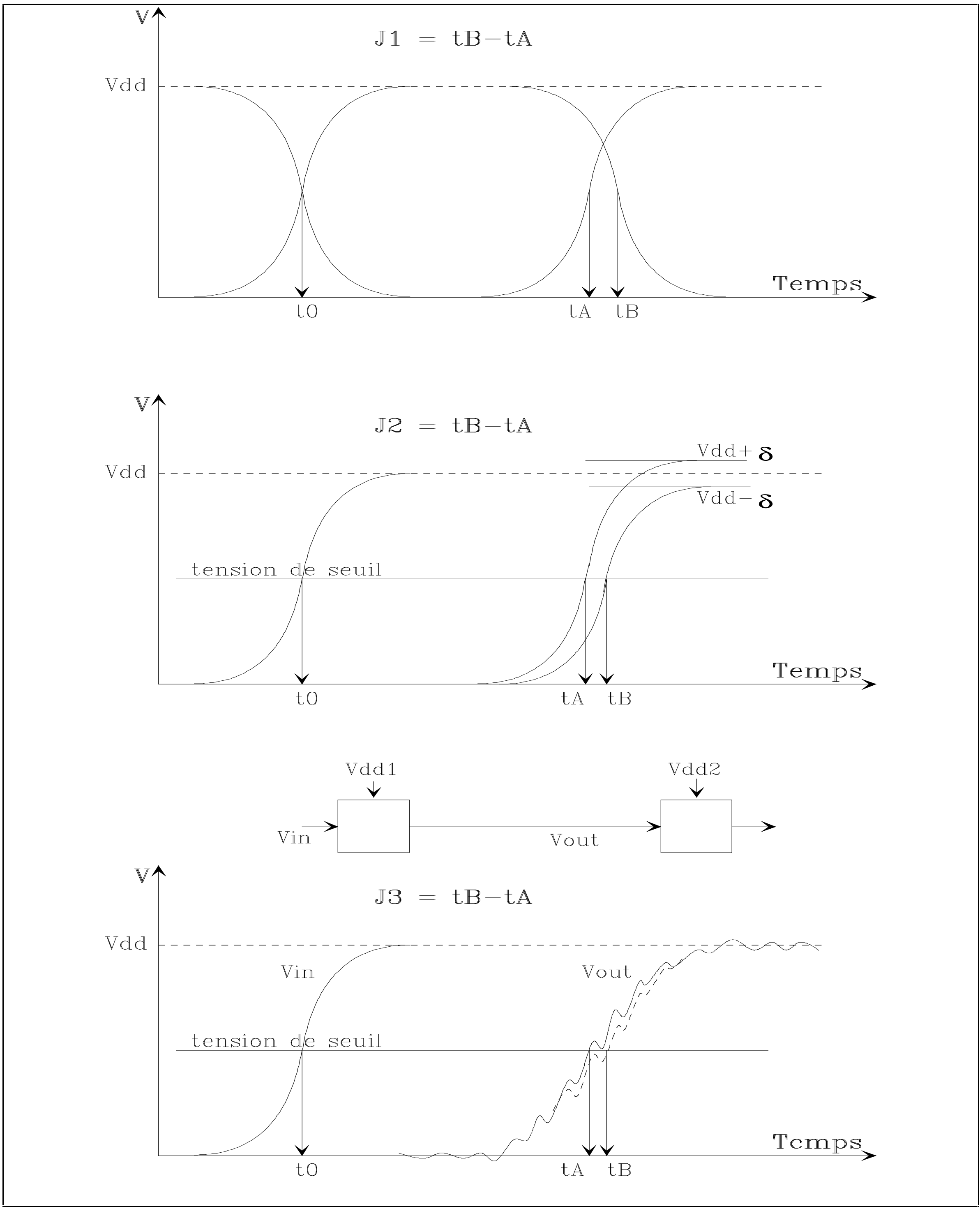
Le principal problème est de maîtriser le 'jitter' associé au signal série.
On décrit sous le terme 'jitter' l'incertitude sur la position d'un front en sortie d'un élément de circuiterie. Le 'jitter' traduit l'imprécision du temps de propagation à travers l'élément.
Plus ce 'jitter' est petit par rapport à la durée moyenne d'un bit, plus le débit transmissible par la liaison série est élevé. A performances égales en terme de débit, réduire le 'jitter' permet de transmettre le signal série sur une plus longue distance ou dans un environnement plus bruité.

Le 'jitter' global du circuit est la somme des contributions des éléments placés dans le chemin critique. Il n'est malheureusement pas possible de mesurer directement chacune de ces contributions. Nous allons donc identifier les différents éléments qui interviennent soit dans la réalisation, soit dans le traitement du signal série et évaluer la contribution de chaque élément en terme de 'jitter'.

L'objectif de cette étude est de construire, sur certaines hypothèses, un modèle du 'jitter' associé au signal série, puis de comparer les résultats de ce modèle aux résultats expérimentaux, enfin d'en déduire les parties de la macro-cellule liaison série susceptibles d'être améliorées dans une version future du circuit.

1/ Généralités

Le 'jitter' induit par la traversée d'un élément de circuiterie provient de la superposition de plusieurs effets



Dans notre évaluation du jitter, les effets pris en compte sont les suivants :

- *Jitter intrinsèque* J1. C'est le 'jitter' introduit par la différence de temps de traversée entre un front montant et un front descendant.

- *Jitter de propagation* J2. Il est défini comme le produit de la sensibilité du temps de propagation d'un élément à la variation de la tension d'alimentation par le bruit local sur la tension d'alimentation.

Les variations de la tension d'alimentation des transistors CMOS déplacent l'instant où le signal de sortie franchit la tension de seuil.

- *Jitter de front* J3. Il est défini comme le produit de la durée d'un front en sortie par le bruit différentiel sur la tension d'alimentation.
  J3 traduit l'effet dû aux écarts de tension d'alimentation entre deux points du circuit.

Le *bruit local sur la tension d'alimentation* représente les variations de cette tension autour de la valeur moyenne +3.3V.
Le *bruit différentiel sur la tension d'alimentation* représente le rapport entre la différence de tension entre deux fils de tension d'alimentation placés à des endroits différents dans le circuit et la valeur moyenne de la tension d'alimentation.

Pour chacun des éléments du chemin critique, on a évalué au moyen de simulations SPICE:

- la sensibilité du temps de propagation à une variation de tension d'alimentation : S
- le temps de montée (ou de descente) du front en sortie de l'élément : Tf
- le 'jitter' intrinsèque : J1

Soit dV le bruit local sur la tension d'alimentation,
dV_diff le bruit différentiel sur la tension d'alimentation,

On évalue à priori dV entre 50 et 300 mV et on pose dV_diff = (2*dV / Vdd)

On obtient ainsi les valeurs de J2 et J3 tels que : J2 = S * dV J3 = Tf * dV_diff

L'évaluation du 'jitter' associé à un élément du chemin critique est donc

$$\begin{aligned} \mathbf{J} &= J1 + J2 + J3 \\ &= J1 + S * dV + Tf * dV\_diff \end{aligned}$$

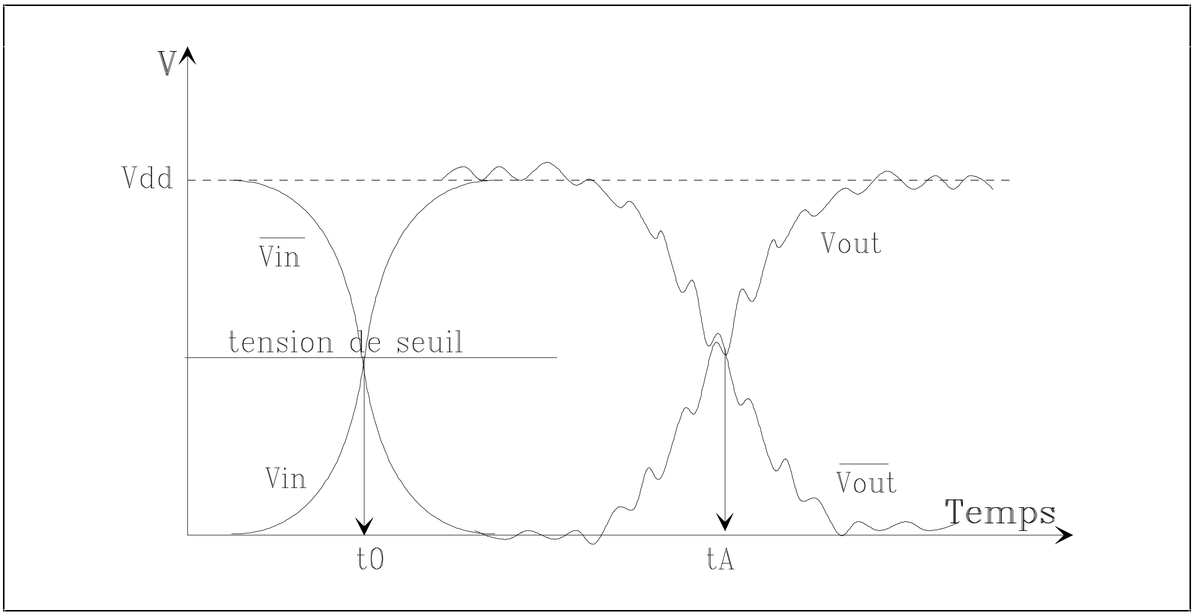
La comparaison pour un élément ou une série d'éléments du chemin critique, du jitter évalué et du jitter mesuré lors des tests doit nous permettre d'ajuster correctement la valeur de dV.

2/ Description du chemin critique

Le chemin critique est divisé en deux parties : la génération et l'émission du signal série au niveau du sérialiseur, la réception et l'échantillonnage du signal série au niveau du désérialiseur.

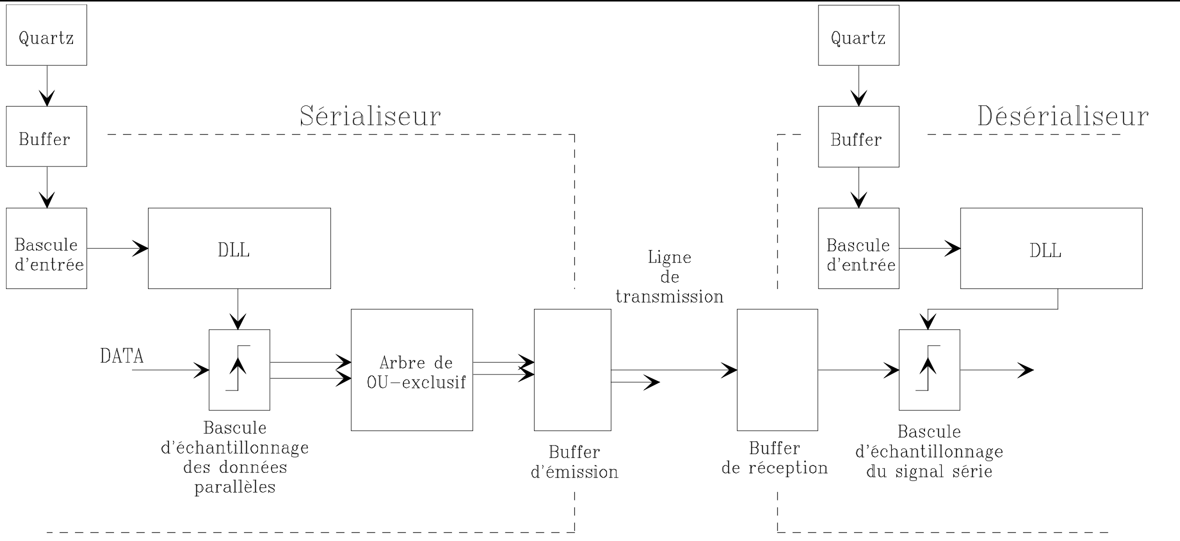
Au niveau du sérialiseur, la sortie de la bascule d’échantillonnage des données parallèles se fait sur deux fils qui transportent en parallèle un signal et son inverse. A partir de cette bascule, le traitement est réalisé en différentiel. Cela comprend l’arbre de OU-exclusifs et l’amplificateur de sortie. Cet amplificateur contient le dernier OU-exclusif qui génère le signal série à 1Gbits/sec. On émet le signal série en différentiel.
Au niveau du désérialiseur, la réception se fait sur un fil unique.

Lorsque le signal série est propagé en différentiel dans le circuit, le signal et son inverse sont affectés de la même façon par le bruit différentiel de tension d’alimentation. On limite ainsi les effets de ‘jitter’ dus au bruit différentiel de tension d’alimentation (J3).



On se place dans les conditions typiques et pour un process CMOS 0.5µ ce qui correspond au circuit de test qui va nous servir de référence.

Le circuit associé à la réalisation des horloges d’échantillonnage du DLL est le même pour le sérialiseur et le désérialiseur. Pour cette raison, on définit un ‘jitter total DLL’ qui somme les effets de ‘jitter’ en sortie du DLL. Cette valeur additionne les ‘jitters’ du quartz, du buffer d’entrée de l’horloge de référence, de la bascule d’entrée de cette horloge ainsi que le ‘jitter’ engendré par la boucle DLL (qui correspond à la valeur du pas d’ajustement des retards c’est-à-dire ici 40 ps).



Eléments du chemin critique pour le signal série

On cherche à associer les valeurs évaluées à des valeurs de 'jitter' mesurables expérimentalement.
La seule mesure directe du 'jitter' possible est celle du 'jitter' en émission: c'est le 'jitter' associé au signal série en sortie du sérialiseur. On évalue le 'jitter' en émission de la manière suivante :

J(émission) = J(total DLL) + J(bascule d'échantillonnage des données) + J(arbre de OU-exclusifs) + J(amplificateur d'émission)

L'amplificateur d'émission inclut le dernier OU-exclusif de manière à ce que le signal de fréquence maximale ait le trajet le plus court possible dans le circuit.

D'autre part, les tests sur FIBULE nous permettent de mesurer la fréquence maximale de fonctionnement de la liaison série en rebouclage interne ou externe. Nous proposons de comparer la valeur de la fréquence maximale de fonctionnement avec la valeur évaluée pour le 'jitter' total associé au signal série en rebouclage interne ou externe.

Le 'jitter interne' c'est le 'jitter' engendré par le chemin critique lorsque l'on reboucle en interne la sortie du sérialiseur sur l'entrée-série du désérialiseur. Il est évalué de la manière suivante :

J(interne) = J(total DLL)+ J(bascule d'échantillonnage des données) + J(arbre de OU-exclusifs) + J(OU-exclusif) + J(total DLL) + J(bascule d'échantillonnage du signal série)

Le 'jitter externe' c'est le 'jitter total' engendré par le chemin critique (on inclut dans ce chemin une ligne de transmission constituée par un mètre de câble coaxial). Ce 'jitter' est évalué de la manière suivante :

J(externe) = J(total DLL)+ J(bascule d'échantillonnage des données) + J(arbre de OU-exclusifs) + J(amplificateur d'émission) +

J(ligne de transmission) + J(amplificateur de réception) +
J(total DLL) + J(bascule d’échantillonnage du signal série)

La valeur mesurée pour le ‘jitter’ en émission nous permet d’évaluer dV.
La valeur de dV ainsi obtenue pourra être reportée dans l’évaluation du ‘jitter interne’ et du ‘jitter externe’.

C/ Environnements et modes de test

Nous présentons ici l’environnement de test utilisé pour tester et évaluer les performances du circuit FIBULE.
Nous décrivons ensuite les modes de fonctionnement de la liaison série accessibles avec cet environnement.

1/ Description de la carte de test de BULLIT

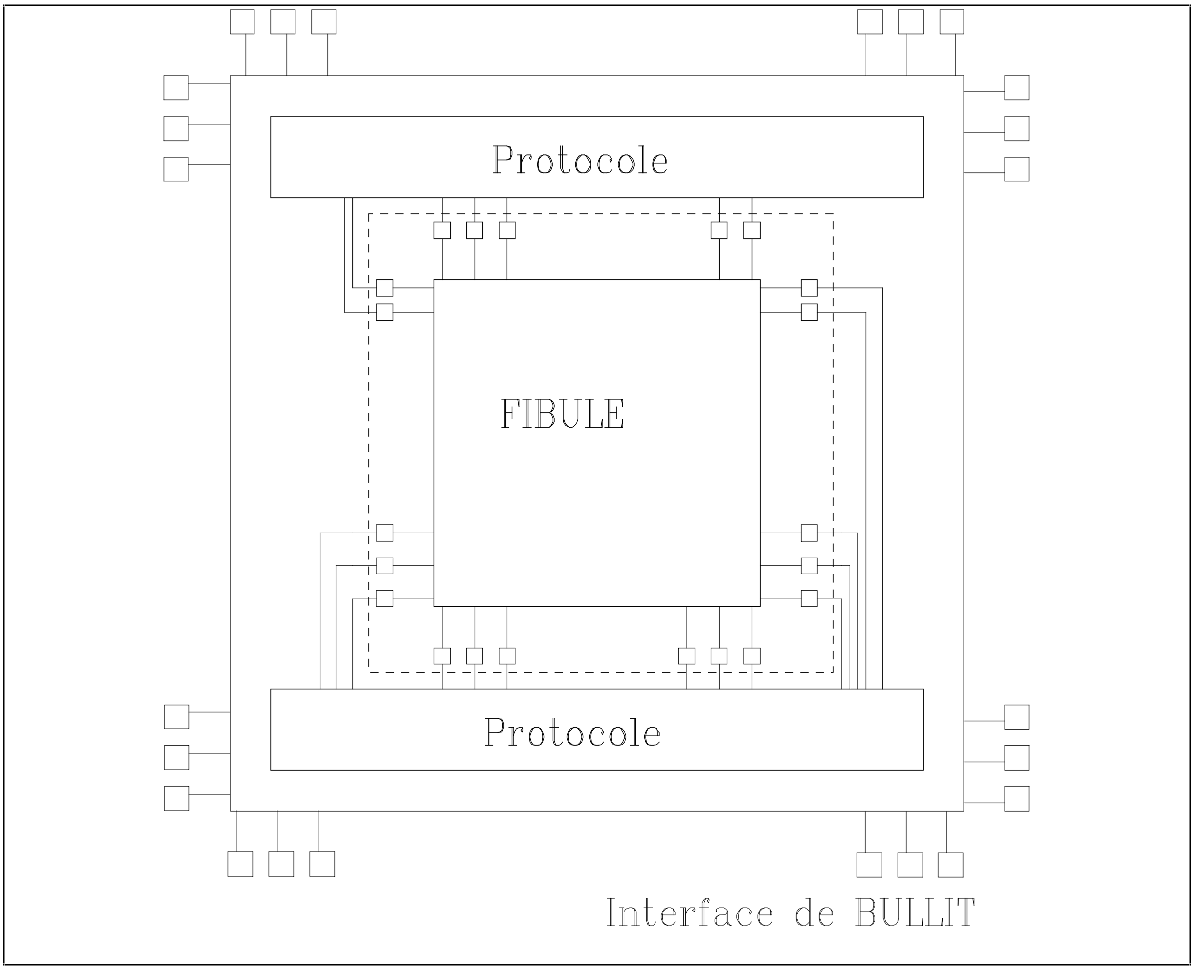
Pour des raisons pratiques, nous n’avons pas pu disposer en même temps du circuit FIBULE et de la carte de test associée.

Nous avons choisi d’utiliser une carte de test existante (celle de BULLIT) pas tout à fait adaptée à nos besoins pour tester notre circuit. Nous avons conçu un circuit de la même taille et avec la même interface que BULLIT. Ce circuit est constitué d’un circuit FIBULE complet avec ses plots d’entrée/sortie et d’un enrobage extérieur qui réalise la transition entre l’interface de FIBULE et l’interface de BULLIT. Le circuit réalisé a également les mêmes fonctionnalités principales que BULLIT. La macrocellule de FIBULE réalise la sérialisation et la désérialisation des données. BULLIT reçoit en entrée des mots de 12 bits constitués de un 1, dix bits de données et un 0. Il suffit de supprimer les bits des extrémités pour obtenir des mots de 10 bits adaptés à l’interface de FIBULE.

La maquette de test de BULLIT est constituée d’une carte de test sur laquelle on place deux circuits de type BULLIT et d’un logiciel de gestion et de contrôle sur PC. Cette maquette comprend deux modes de fonctionnement :

- le mode ‘transmission de données’ utilise un générateur aléatoire de mots de 8 bits qui sont ensuite codés suivant le protocole de BULLIT en 8B/12B. Les mots émis sont comparés avec les mots reçus par le logiciel qui fournit à l’écran le taux d’erreur. On peut aussi remplacer les données du générateur aléatoire par une série de mots transmis en boucle.

- le mode ‘scan’ permet de lire et d’écrire dans le chemin de scan. Ce mode est difficilement accessible avec un FIBULE encapsulé BULLIT car les chemins sont de longueur et d’organisation différentes.



Pour tester FIBULE sur sa propre carte de test, il suffit de découper le circuit suivant les pointillés pour disposer du circuit FIBULE avec sa propre interface.

La carte de test de BULLIT permet de valider les principes mis en oeuvre dans la macrocellule liaison série et d'évaluer les performances du circuit FIBULE.

Toutefois les tests réalisés avec cette carte sont insuffisants pour permettre de déterminer les limites de fonctionnement de la liaison série. En effet, le passage des données et du signal série par les blocs d'adaptation à l'interface de BULLIT crée des perturbations sur le signal. Le signal série, fonctionnant à 1 Gbits/sec, tant en entrée qu'en sortie, se trouve propagé sur un fil d'environ 1mm sur la puce. Il est, de plus, surchargé par les deux plots. La constante de temps RC introduite par ce montage a été minimisée mais son effet n'est pas négligeable.
De plus le chemin de 'scan' est accessible en lecture uniquement et avec beaucoup de difficultés. Ceci limite considérablement notre étude du fonctionnement interne de la macrocellule.

Une carte de test a été conçue spécifiquement pour FIBULE. A ce jour, la carte est fonctionnelle. Cependant le logiciel de contrôle est encore trop limité pour permettre d'obtenir des résultats supplémentaires par rapport à ceux obtenus sur la carte de BULLIT. Pour cette raison, la carte de test de FIBULE est décrite en annexe.

2/ Modes de fonctionnement de la liaison série

Le circuit est testé dans trois modes de transmission :

- *le mode interne* valide l'option de rebouclage interne de la sortie du sérialiseur sur l'entrée du désérialiseur. Ce branchement est réalisé à l'intérieur du circuit. On élimine ainsi le problème d'adaptation d'impédance et les imprécisions liées à la traversée des amplificateurs d'émission et de réception. Un seul circuit est utilisé.

- *le mode externe sur une liaison série* réalise l'option de rebouclage de la sortie du sérialiseur sur l'entrée du désérialiseur en externe à travers un mètre de câble coaxial. La comparaison de ce mode avec le précédent permet d'évaluer l'importance des imprécisions liées à la traversée des amplificateurs d'émission et de réception.

- *le mode externe entre deux équipements* implémente un lien bidirectionnel dans un environnement système réel. Ce mode nous permet, dans le cas de FIBULE, de tester le fonctionnement de la macrocellule dans le cas où le sérialiseur et le désérialiseur fonctionnent avec des horloges générées par deux quartz différents.

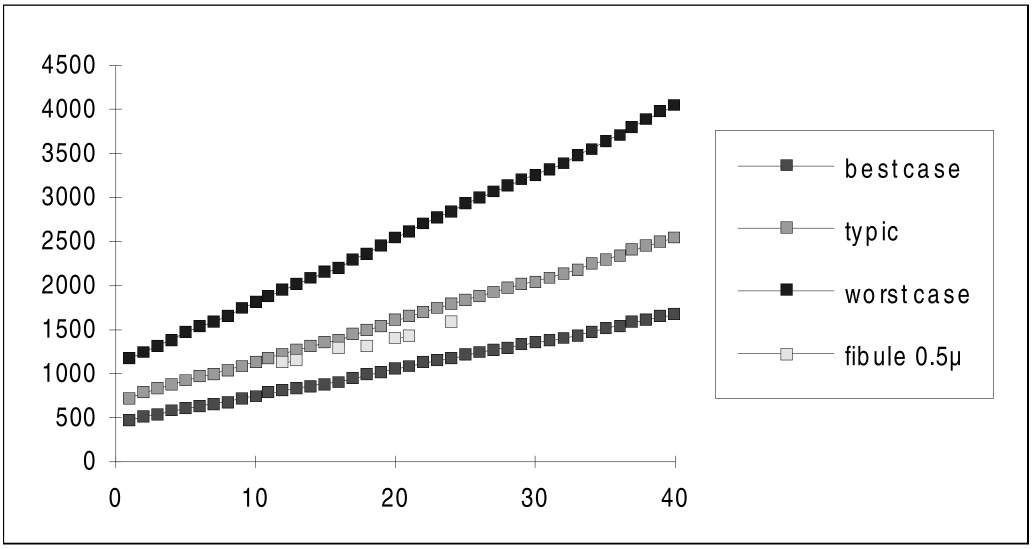
Les conditions de test sont les suivantes :

- température ~25°C,
- tension d'alimentation +3.3V

Le test qui permet de valider les fonctionnalités du circuit consiste pour une fréquence donnée à calibrer la liaison série et à transmettre des données sans erreur pendant quelques minutes. La détection de la calibration est caractérisée par le passage à l'état haut du signal COM_DET qui indique que le mot de référence COMMA a été reconnu en sortie du désérialiseur. La calibration de la liaison déclenche la transmission des données. Le taux d'erreur en temps réel apparaît sur l'écran du PC.
La transmission de données sans erreur doit être obtenue pour une plage continue de fréquence et dans les trois modes définis ci-dessus.

Le test qui définit la limite de fonctionnement est fait sur les fréquences comprises entre 50 et 110 MHz en variant de 1 MHz à chaque fois.
Le circuit a été testé avec trois types de données : des données Fibre Channel, des données quelconques garantissant un taux de plus de 10% de transitions dans le signal série et des données sans aucune contrainte. Le taux de transition dans le signal série détermine la capacité de la liaison série à suivre le déphasage du signal série par rapport à l'horloge de référence du désérialiseur.

Pour analyser le comportement interne de la liaison série, on a accès à tous les registres de contrôle des retards variables et à certains registres d'état par l'intermédiaire du 'scan'. Avec la carte de test de BULLIT, nous pouvons lire le 'scan'. Nous avons ainsi obtenu quelques valeurs de commande des retards variables à différentes fréquences de fonctionnement. En comparant la valeur réelle du retard T de la boucle DLL pour une commande donnée avec la valeur obtenue par simulation SPICE, on peut évaluer le process dont fait partie le lot de circuits testés. Il s'agit d'un process quasi typique.



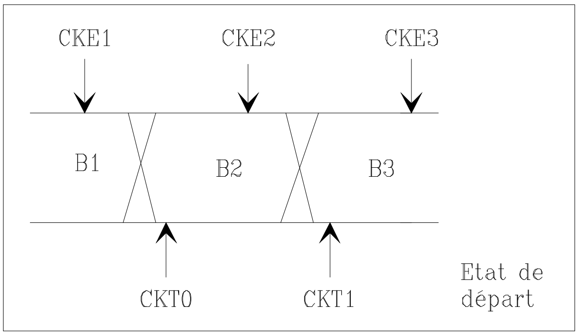
Le tableau ci-dessus présente en abscisse la commande appliquée au retard variable et en ordonnée la valeur du retard en pico-seconde.
Un débit de 100 MHz correspond à un retard de durée 1000 ps. On constate que le débit n'est pas limité par la valeur que peut prendre le retard.

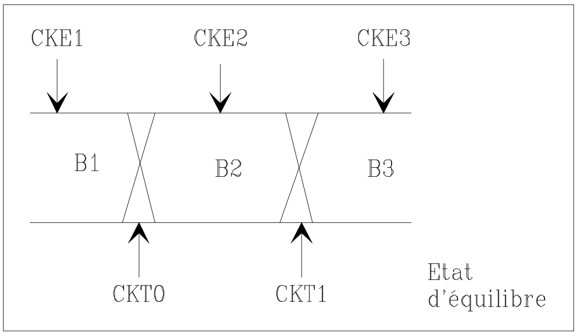
D/ Validation fonctionnelle

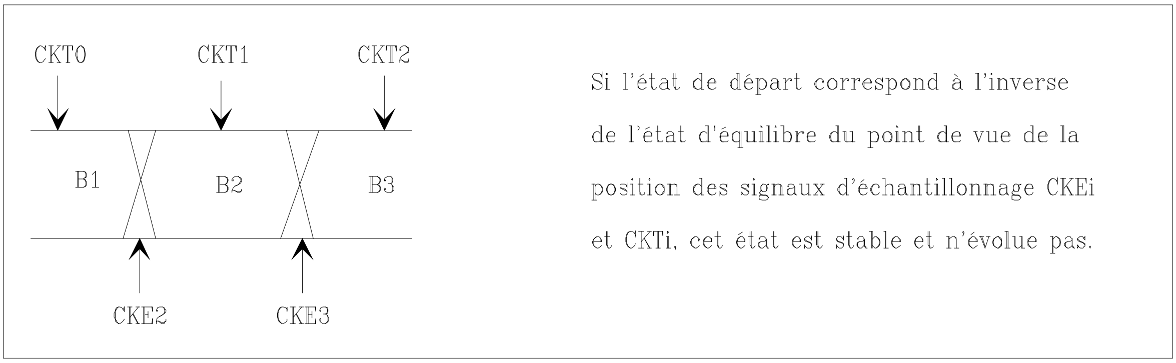
La macro-cellule se calibre et transmet des données sans erreur pour différentes valeurs de fréquence comprises entre 55 et 100 MHz. Toutefois l'exploration de la plage de fonctionnement en fréquence met en évidence un problème dans le procédé de calibration. Dans la zone de fonctionnement, la liaison ne se calibre pas pour la plage de fréquence comprise entre 76 et 80 Mhz (cette plage se déplace d'un circuit à l'autre). Si l'on calibre la liaison sur une autre fréquence (70 Mhz par exemple), on peut alors faire varier lentement la fréquence (au moyen du générateur) et l'amener dans la zone de non-calibration sans provoquer d'erreur dans la réception des données.

L'existence d'une zone de non-calibration s'explique probablement par la construction même de la boucle de contrôle de la valeur du retard X.

Le contrôle du retard X est basé sur la détection des transitions du signal série. Pour cela on compare les valeurs échantillonnées par les CKE[i] et les CKT[i].







Si par malchance, la valeur initiale du retard X est telle que la configuration des horloges par rapport au signal série au départ est celle ci-dessus, les valeurs échantillonnées par CKEi peuvent aussi bien prendre la valeur qui précède la transition que celle qui la suit. En sommant les résultats, les détections réalisées au niveau de chaque transition indiquent autant de demandes d'incrémentation du retard X que de décrémentation. Aucune décision ne peut être prise et le retard X ne se calibre pas. Cette hypothèse a été vérifiée par la lecture du scan après une mise sous tension de la liaison série pour une des fréquences de non-calibration. La commande du retard X reste bloquée.
Une solution a ce problème est l'initialisation de la commande du retard X à une valeur différente par le chemin de scan. Cette solution n'a pas pu être expérimentée puisque le scan n'est pas accessible en écriture à travers la carte de test de BULLIT.

Une transmission bidirectionnelle sans erreur entre deux circuits sur deux cartes différentes alimentés par deux quartz indépendants de fréquence 80 MHz (800 Mbits/sec) a été obtenue pendant une heure. Ne disposant pas de quartz à une fréquence supérieure, c'est le plus haut débit atteint dans cette configuration. Ce test est fondamental car c'est la seule configuration où l'on teste la liaison dans son environnement typique c'est-à-dire avec un émetteur et un récepteur qui ne travaillent pas avec la même horloge.
Le mécanisme de suivi du décalage de phase permanent entre le signal série entrant et le signal d'horloge du désérialiseur fonctionne à 80 MHz : le principe de l'utilisation du retard X pour compenser ce déphasage est validé.

E/ Mesures et performances

1/ Mesure de la fréquence maximale de fonctionnement

Mode interne

La liaison série transmet les données sans erreur entre 55 et 100 MHz. Lorsque l’on dépasse 100 MHz, la calibration s’effectue mais le taux d’erreur de transmission augmente rapidement avec la fréquence.

Mode externe sur une liaison série rebouclée sur elle-même

La plage de fréquence de fonctionnement sans erreur est comprise entre 55 et 95 MHz. On observe une différence de 5% entre la plus haute fréquence de fonctionnement en mode interne et celle en mode externe. L’influence des amplificateurs d’émission et de réception du signal série en terme d’accroissement du jitter est donc faible.

Mode externe entre deux liaisons série

Ce mode a été validé à 80 MHz.

2/ Mesure directe du jitter

La seule mesure concrète du jitter est celle du jitter en émission : il est mesuré en sortie du sérialiseur après propagation du signal série sur un mètre de câble sur le diagramme de l’oeil. La largeur du signal série sur l’oscilloscope est égale à 350ps.

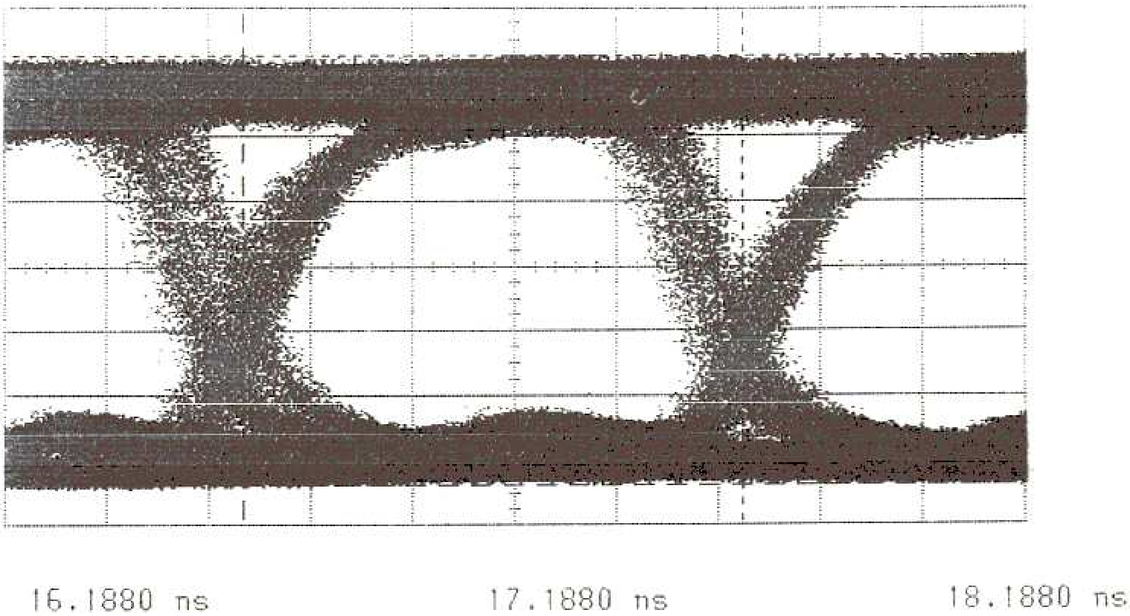


Diagramme de l'oeil
après 1 m de câble coaxial KX22 de diamètre externe 2.54 mm

Rappelons le calcul réalisé au paragraphe B pour évaluer le jitter en émission :

	S (ps/mV)	Tf (ps)	dV (mV)	dV_diff	J1 (ps)	J2 (ps)	J3 (ps)	Jitter (ps)
Oscillateur	0	250	50	0,03	0	0	7,58	7,58
Buffer entree	0,1	215	100	0,06	0	10	13,03	23,03
Latch entree DLL	0,3	300	30	0,02	0	9	5,45	14,45
DLL	0,17	360	30	0,02	40	5,1	6,55	51,65
Latch différentiel	0,11	245	30	0,02	0	3,3	4,45	7,75
Arbre XOR	0,39	150	50	0,03	100	19,5	4,55	124,05
XOR+ Buffer sortie	0,33	300	50	0,03	40	16,5	9,09	65,59
ligne de trans.	0,3	200	100	0,06	20	30	12,12	62,12
Jitter émission					180,00	63,40	50,70	356,22
Jitter mesuré								350,00

On ajuste dV pour rapprocher les valeurs calculée et mesurée pour le jitter en émission. Les valeurs choisies sont finalement dV=50mV à l'intérieur du circuit et dV=150mV pour les buffers.

Cette différentiation reflète le fait que les effets de bruit d'alimentation sont plus importants au niveau de la périphérie du circuit où la présence d'autres buffers crée localement des appels de courant qui font varier la tension d'alimentation.
Le sérialiseur est globalement peu sensible aux effets de bruit différentiel sur la tension d'alimentation.

On reporte ensuite les valeurs obtenues pour dV dans le sérialiseur dans les tableaux concernant le jitter interne et le jitter externe et pour l'ensemble du chemin critique.

	S (ps/mV)	Tf (ps)	dV (mV)	dV_diff	J1 (ps)	J2 (ps)	J3 (ps)	Jitter (ps)
Oscillateur	0	250	50	0,03	0	0	7,58	7,58
Buffer entree	0,1	215	100	0,06	0	10	13,03	23,03
Latch entree DLL	0,3	300	30	0,02	0	9	5,45	14,45
DLL	0,17	360	30	0,02	40	5,1	6,55	51,65
Latch différentiel	0,11	245	30	0,02	0	3,3	4,45	7,75
Arbre XOR	0,39	150	50	0,03	100	19,5	4,55	124,05
XOR de sortie	0,33	300	50	0,03	40	16,5	9,09	65,59
trans. interne	0,3	200	100	0,06	20	30	12,12	62,12
Total DLL					40	24,1	32,61	96,71
Latch échantill.	0,17	180	300	0,18	40	51	32,73	123,73
jitter interne					280,00	168,50	128,15	576,65

	S (ps/mV)	Tf (ps)	dV (mV)	dV_diff	J1 (ps)	J2 (ps)	J3 (ps)	Jitter (ps)
Oscillateur	0	250	50	0,03	0	0	7,58	7,58
Buffer entree	0,1	215	100	0,06	0	10	13,03	23,03
Latch entree DLL	0,3	300	30	0,02	0	9	5,45	14,45
DLL	0,17	360	30	0,02	40	5,1	6,55	51,65
Latch différentiel	0,11	245	30	0,02	0	3,3	4,45	7,75
Arbre XOR	0,39	150	50	0,03	100	19,5	4,55	124,05
XOR+ buffer sortie	0,33	300	50	0,03	40	16,5	9,09	65,59
ligne trans. 1m	0,3	200	100	0,06	20	30	12,12	62,12
buffer entree	0,12	245	150	0,09	10	18	22,27	50,27
Total DLL					40	24,1	32,61	96,71
Latch échantill.	0,17	180	300	0,18	40	51	32,73	123,73
jitter externe					290,00	186,50	150,42	626,92

On obtient une évaluation du jitter total en interne et du jitter total en externe à travers 1 m de câble coaxial. L'évaluation du jitter donne pour le jitter total en interne une valeur de 575 ps et en externe de 625 ps.

Soit les fréquences maximales de fonctionnement obtenues sur la carte de test de BULLIT c'est-à-dire :
en interne Fmax = 100 MHz ce qui correspond à une période bit de 1000ps
en externe Fmax = 95 MHz ce qui correspond à une période bit de 1050ps

Nous cherchons à comparer les valeurs de jitter total obtenues par évaluation avec la période-bit correspondant à la fréquence maximale de fonctionnement en interne et en externe.
La limite de fonctionnement est atteinte pour un jitter égal à 60% de la période-bit.

F/ Etude critique des résultats

Les résultats obtenus lors des tests sur la carte de BULLIT permettent de valider les principes mis en place dans FIBULE avec une réserve concernant le contrôle de la boucle de commande du retard X.
Une étude approfondie du mécanisme de la boucle de commande du retard X est nécessaire.
La carte de test de FIBULE devrait permettre d'atteindre ce but en utilisant les possibilités offertes par le chemin de 'scan'.

Les performances en débit atteignent 95 MHz en rebouclage externe à travers un mètre de câble. Les performances obtenues en débit sont insuffisantes pour satisfaire la norme Fibre Channel à 1.0625 Gbits/sec.

Augmenter le débit revient à diminuer le ‘jitter’ total induit sur le signal série. Une façon de réduire le jitter est de traiter toute les parties qui véhiculent le signal série en différentiel. Cela est réalisé dans le sérialiseur, il faudrait le mettre en place également dans le désérialiseur en modifiant l’amplificateur de réception et la bascule d’échantillonnage du signal série.

Conclusion

La montée en puissance de traitement des microprocesseurs engendre de fait un défi technologique : transmettre une grande quantité de données sur des distances courtes et à très grande vitesse.
A cette demande, nous proposons une réponse qui regroupe de nombreux avantages en terme de performance, consommation, fiabilité et prix. Il s’agit d’une liaison série proposée sous la forme d’une macro cellule intégrable sur le pourtour d’un circuit quelconque et qui permet de transmettre les données d’un bus sur un seul fil.

Nous avons tenté de concevoir une liaison série qui soit :

- adaptée à n’importe quel type de codage des données
- capable de délivrer un débit de 1 Gbits/sec en CMOS
- intégrable dans un circuit utilisateur sous la forme d’une macrocellule
- adaptée aux transmissions en mode plésiochrone
- capable de s’interfacer avec un câble coaxial ou une fibre optique
- de consommation faible pour ne pas perturber le fonctionnement du circuit principal
- de superficie aussi réduite que possible pour qu’un nombre important de liaisons série puissent être implantées sur un même circuit intégré.

Les tests réalisés sur le circuit FIBULE ont permis de démontrer la validité des principes mis en oeuvre dans la macrocellule liaison série :

- le principe de désérialisation du signal série n'impose aucun codage particulier des données

- la gestion du déphasage évolutif entre les horloges de référence de l'émetteur et du récepteur a été démontrée à 80 MHz

- la transmission du signal série sur 1 m de cable coaxial à 95 MHz s'effectue sans erreur

Toutefois le circuit FIBULE ne répond pas parfaitement à la spécification Fibre Channel auquel il est associé.

- le débit de 1.0625 Gbits/sec standard pour Fibre Channel n'est pas atteint par FIBULE

- on a mis en évidence une déficience de la boucle de commande du retard X : cette boucle possède deux états d'équilibre différents dont l'un seulement est valide

Ces déficiences peuvent être traitées de la manière suivante :

- pour augmenter le débit accessible à la macrocellule, on envisage de réaliser la réception et l'échantillonnage du signal série en différentiel au niveau du récepteur. Ceci devrait permettre de diminuer le jitter associé au signal série et donc d'augmenter la valeur du débit que l'on peut transmettre sans erreur

- l'existence de deux états d'équilibre est inhérent au principe d'évaluation de la commande du retard X. Pour éliminer ce problème, il faudrait inclure dans le circuit une procédure de recalibration de la liaison série en modifiant la valeur du retard X au démarrage à travers le chemin de scan. Cette procédure se mettrait automatiquement en marche si la liaison n'est pas calibrée au bout d'un nombre prédéfini de cycles d'horloge.

Bibliographie

10-Bit Interface Specification
Rev 0.30a Dec 5, 1994
10-Bit Study Group sous la responsabilité du comité ANSI X3T11 Fibre Channel

Un générateur paramétrable de multiplieur de fréquence utilisant des techniques numériques
Michel Combes, thèse de doctorat de l'université Pierre et Marie Curie, 1994

VSC7105/VSC7106
VITESSE, présentation des caractéristiques des circuits, octobre 1994

A CMOS serial link for fully duplexed data communication
IEEE Journal of Solid State Circuits, Vol 30 n°4, avril 1995
Kyeongho Lee, Sungjoon Kim, Gijung Ahn, Deog-Kyoon Jeong.

Hots Pads : Gigabit serial Link I/O
David D. Lee & Neil Wilhem
Sun Microsystems Laboratories, 1995

A two-chip 1.5 GBd Serial Link Interface
IEEE Journal of Solid State Circuits, Vol 27 n°12, décembre 1992
Richard C. Walker, Cheryl L. Stout, Jieh-Tsorng Wu, Benny Lai, Chu-Sun Yen, Tom Hornak, Patrick T. Petruno.

Integration of multiple point to point serial links in the gigabits per second range
Hot Interconnects Symposium 1993
R.Marbot, J.C.Lebihan, A.Cofler, R.Nezamzadeh-Moosavi

HS Link : Delay locked loop noise immunity
OMI/HIC, rapport interne de juin 1994
R.Marbot, A.Cofler

BULLIT Data Sheet
(version alimentée à 3.3V) Rev 2.0 mai 1995

BULL Serial Link Technology
R.Marbot, J.C.Lebihan, A.Cofler, R.Nezamzadeh-Moosavi

A DC-balanced, Partitioned-Block, 8B/10B Transmission Code
IBM Journal of Research and Development   Vol 27   n°5   Septembre 1983
A.X. Widmer, P.A. Franaszek

Fibre Channel Transceiver Chip
Preliminary Technical Data,   Mars 1995 (Rev. 3.2)
Bob Mayer,   Hewlett-Packard

Standard for Heterogeneous InterConnect (HIC)
Low Cost Low Latency Scalable Serial Interconnect for Parallel System Construction
IEEE Draft Std P1355,   June 1995 (Rev. D1.5)

Procédé d'échantillonnage d'un signal numérique en série
Brevet n°94 09683 déposé le 4 août 1994 au nom de BULL S.A.
R.Marbot, J.C.Lebihan, A.Cofler, R.Nezamzadeh-Moosavi, A.Pierre-Duplessix, P.Couteaux

Arbre de portes OU-Exclusif et multiplieur de fréquence l'incorporant.
Brevet n°93 400644.6 déposé le 17 décembre 1993 au nom de BULL S.A.
R.Marbot, J.C.Lebihan, A.Cofler, R.Nezamzadeh-Moosavi

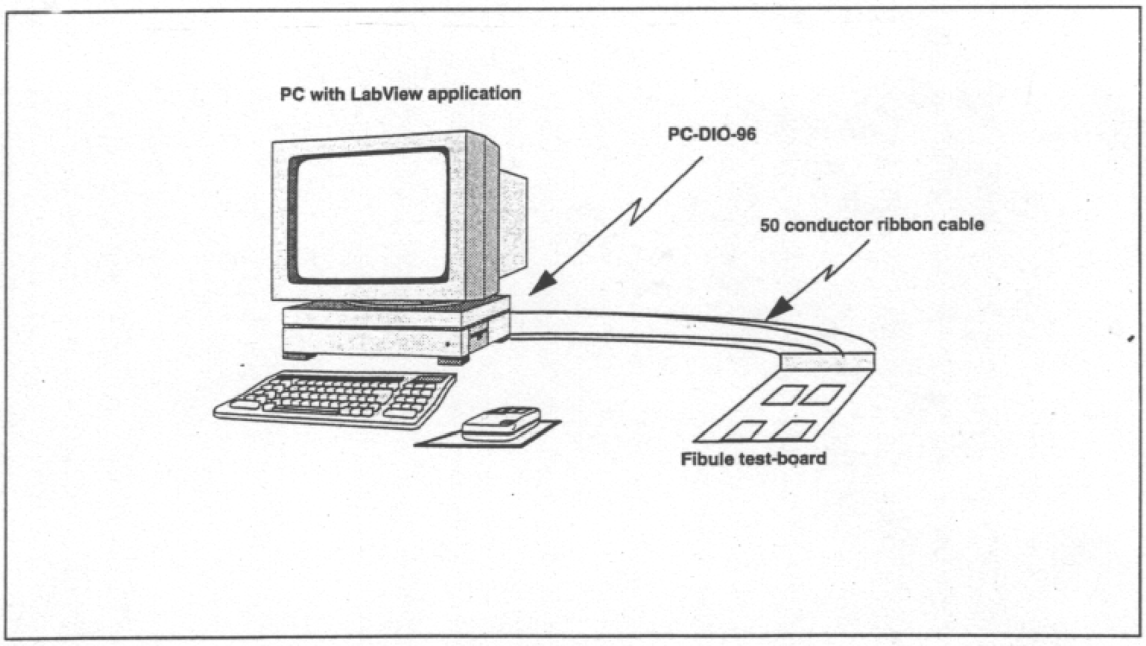
A CMOS Low Voltage Swing Transmission System for 1 Gbaud Serial Point-to-Point Datacommunications
IEEE International Solid State Circuit Conference 1994
R.Marbot, J.C.Lebihan, A.Cofler, R.Nezamzadeh-Moosavi

Procédé et dispositif d'adaptation d'impédance pour un émetteur et/ou un récepteur, circuit intégré et système de transmission les mettant en oeuvre
Brevet déposé le 17 août 1993 au nom de BULL S.A.
R.Marbot, J.C.Lebihan, A.Cofler, R.Nezamzadeh-Moosavi

Annexe I : Présentation de la carte de test réalisée spécifiquement pour le chip FIBULE

Le système de test de FIBULE est composé de trois éléments principaux :

- une maquette de test comprenant deux circuits FIBULE ainsi que des circuits de contrôle
- cette maquette communique avec le PC par l'intermédiaire d'une carte I/O digitale (PC-DIO-96 de National Instruments)
- une application du logiciel LabView permet à l'utilisateur de contrôler le fonctionnement de la carte.



Trois modes de fonctionnement sont proposés par le logiciel de contrôle. Ces modes peuvent être réalisés en mode interne et en mode externe avec une ou deux liaisons série.

- *Mode calibration des liaisons série* : le circuit émetteur envoie en permanence la même série de quatre mots dont l'un est le mot COMMA.

- *Mode transmission d'une série de mots* : le circuit émetteur transmet une série de 1 à 64 mots stockés dans une FIFO puis repasse en mode calibration. Ce mode nous permet de tester le fonctionnement de la liaison série avec des données spécifiques par exemple des données encodées par le protocole Fibre Channel. Les mots sont récupérés en sortie du désérialiseur, stockés dans une FIFO et peuvent être consultés dans un fichier.

- *Mode transmission de données aléatoires* : un générateur de données aléatoires sur 10 bits fonctionne en continu et alimente la liaison série en données. Ce mode permet de déterminer un taux d'erreur sur des temps de fonctionnement de plusieurs heures. Le nombre d'erreurs détectées ainsi que le nombre de mots transmis s'affichent à l'écran du PC.

A cela s'ajoute un mode de lecture/écriture du chemin de scan qui permet d'avoir accès aux registres de contrôle du circuit FIBULE.

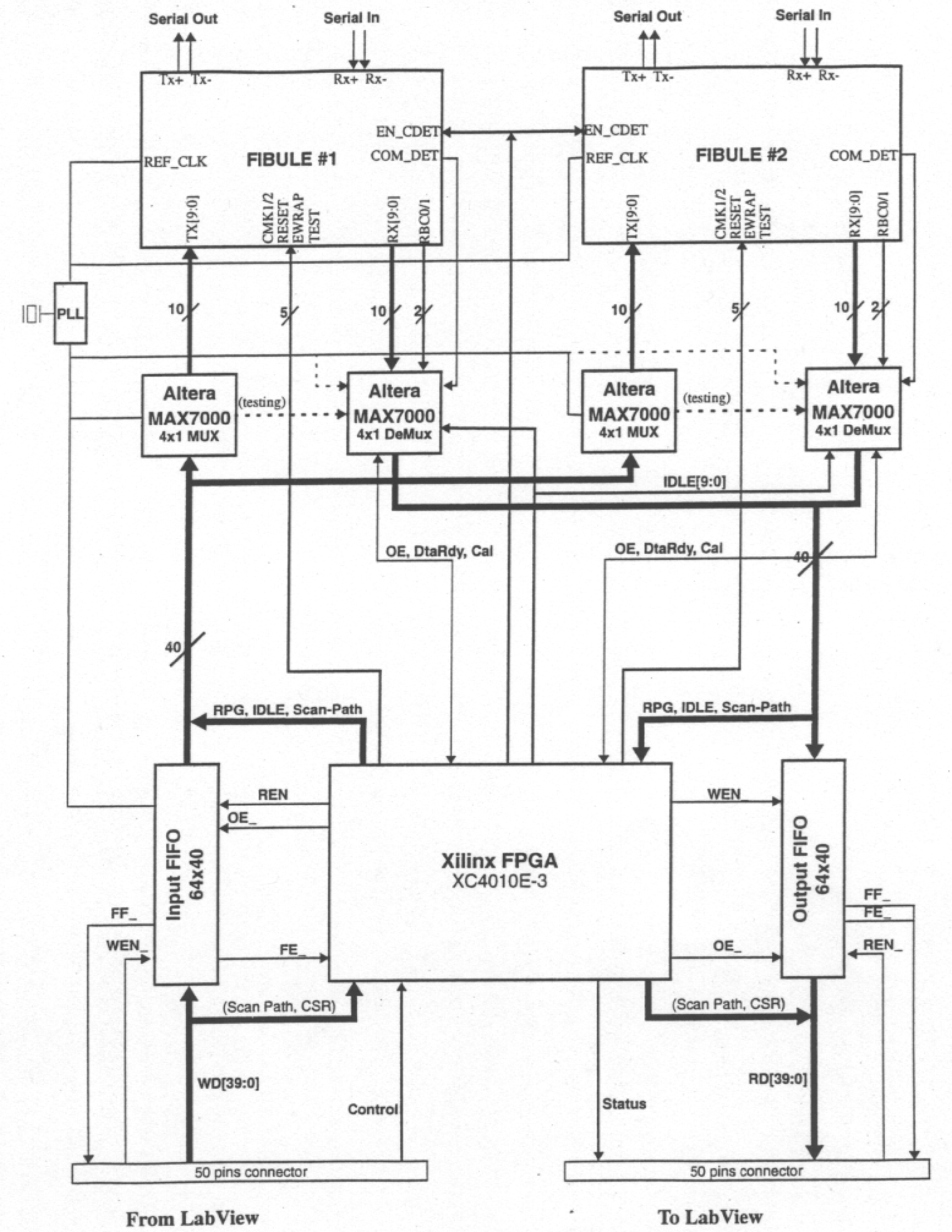


Schéma de la carte de test de FIBULE