

UNIVERSITÉ DE MONTRÉAL

UTILISATION DE LA RECONFIGURATION DYNAMIQUE DES FPGA POUR
LE CONTRÔLE PRÉCIS ET EXACT DES DÉLAIS DANS LES
CONVERTISSEURS TEMPS À NUMÉRIQUE

MARC-ANDRÉ DAIGNEAULT
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
DÉCEMBRE 2009

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

UTILISATION DE LA RECONFIGURATION DYNAMIQUE DES FPGA POUR
LE CONTRÔLE PRÉCIS ET EXACT DES DÉLAIS DANS LES
CONVERTISSEURS TEMPS À NUMÉRIQUE

présenté par : M. DAIGNEAULT Marc-André
en vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées
a été dûment accepté par le jury constitué de :

M. SAVARIA Yvon, Ph.D., président.

M. DAVID Jean Pierre, Ph.D., membre et directeur de recherche.

M. BOIS Guy, Ph.D., membre.

*« Le temps nous égarre, le temps nous étreint,
le temps nous est gare, le temps nous est train. »,
Jacques Prévert.*

Remerciements

J'aimerais profiter de cette occasion pour remercier mon directeur de recherche, le professeur Jean Pierre David, de m'avoir permis de réaliser mes travaux de recherches sous sa supervision. Bien plus qu'un directeur, de par sa rigueur, sa créativité, son ardeur au travail, et ses conseils, le professeur David à été pour moi une grande source d'inspiration. Au cours de cette grande aventure qu'a été la maîtrise en sciences appliquées où nous avons appris à nous connaître, j'ai eu la chance de découvrir qu'au delà du chercheur remarquable se trouve une personne d'une grande sagesse, bonté, et compréhension.

Je me souviendrai toujours à mes premiers pas dans son équipe, en marchant dans un couloir, lorsqu'un de ces anciens étudiants lui avait dit : « Ouin, il est pas pire ton nouvel étudiant ! », Jean Pierre avait répondu précipitamment, sur un ton d'humour qui me semblait mal cacher un léger embarras : « *Shhhhhh*, il ne faut pas lui dire ». Bien qu'anodin, ce geste témoignait sans équivoque de son désir d'obtenir le meilleur de chacun. Il n'y en aurait pas de faciles !

J'aimerais également profiter de l'occasion pour remercier les professeurs du Groupe de Recherche en Microélectronique et Microsystèmes (GR2M) Pierre Langlois, Guy Bois, et Yvon Savaria, pour la qualité et la pertinence de leurs enseignements au niveau des études supérieures en génie électrique et informatique. À ce titre, j'aimerais tout particulièrement souligner l'initiative du professeur Langlois d'initier les étudiants à l'analyse, la rédaction, et la présentation d'articles scientifiques dans le domaine des processeurs configurables.

Résumé

La mesure d'intervalles de temps est importante dans différents domaines d'applications scientifiques. Un convertisseur numérique de temps (TDC) est un circuit électronique permettant de mesurer des intervalles de temps avec des résolutions de l'ordre de la picoseconde. Jusqu'à la dernière décennie, ces circuits étaient implémentés exclusivement sous forme de circuits dédiés (ASIC), mais depuis, plusieurs implémentations sur circuits programmables (FPGA) ont été proposées. Bien qu'à ce jour de telles implémentations accusent toujours des performances inférieures, il existe un intérêt réel pour réduire l'importance de cet écart. En effet, le progrès fulgurant de la technologie FPGA, en termes de densité logique et de fréquence d'opération, en fait aujourd'hui un candidat de choix pour l'implémentation de nombreux circuits et systèmes. Au delà du grand degré d'intégration qu'elle permet d'obtenir, ce type d'implémentation se démarque des circuits dédiés en ce qu'elle permet à la fois des temps de développement et des coûts non-récurrents nettement moins importants. C'est donc dans ce contexte que ce travail se penche sur l'implémentation FPGA d'un convertisseur numérique de temps.

Alors que les implémentations de TDC sur circuits dédiés permettent d'obtenir des résolutions avoisinant la picoseconde, les implémentations FPGA les plus récentes sont limitées à quelques dizaines de picosecondes. Puisque l'implémentation d'un TDC est intimement liée à la notion de délai, les FPGAs sont handicapés d'une part par l'irrégularité des délais d'interconnexions programmables générées par les outils de synthèse, et d'autre part par des délais d'interconnexions plus importants. C'est ainsi que les implémentations FPGA offrant les meilleures performances dans la littérature reposent sur une architecture permettant d'exploiter la présence des structures d'interconnexions dédiés, telle la chaîne de retenue rapide. En effet, ces structures d'interconnexions dédiées offrent à la fois des délais réduits et une régularité accrue. Toutefois, la résolution atteignable avec cette architecture est limitée par les délais minimaux du circuit, et ces derniers sont sensiblement plus importants sur un FPGA que sur un circuit dédié. Néanmoins, cette architecture bénéficie directement des nouvelles générations de FPGA qui sont produites avec des procédés de fabrication permettant d'obtenir des délais minimaux réduits.

Afin de palier à cette limitation imposée par les délais minimaux disponibles, cet ouvrage propose une approche basée sur l’ajustement fin et précis des délais. L’originalité de l’approche proposée réside en ce qu’elle tire profit de deux caractéristiques propres aux FPGA : la richesse du système d’interconnexions programmables, conjointement avec la possibilité de reconfigurer dynamiquement le circuit. En effet la reconfiguration dynamique, supportée par certains FPGA, permet de reconfigurer partiellement le circuit lors de son opération. Ainsi, puisqu’il existe un nombre important de chemins possibles entre deux points du circuit, la reconfiguration dynamique du système d’interconnexions programmables permet potentiellement d’ajuster le délai entre ceux-ci avec une résolution autrement inaccessible. C’est ainsi que nos résultats font état d’une ligne à délai programmable ajustable avec une résolution de 1ps sur une plage de plus de 2,5ns.

Sur la base de ces lignes à délais programmables à fine résolution, une nouvelle architecture de TDC est proposée. Cette architecture se caractérise par l’utilisation de plusieurs lignes à délais en parallèles, et permet d’obtenir une résolution non pas limitée par les délais minimaux du circuit, mais plutôt par la différence minimale d’ajustement entre deux lignes à délais. Ainsi, au moyen d’une telle architecture, il devrait être possible de produire un TDC offrant des résolutions inférieures à celles obtenues à l’état de l’art. Néanmoins, l’ajustement fin des délais étant à lui seul insuffisant, une méthode de calibration permettant un ajustement précis de ces derniers est également proposée. Les délais d’un circuit sont affectés par les variations de températures, ainsi que les variations dues au procédé de fabrication de la puce. Afin de rendre possible un ajustement précis des délais, le processus de calibration proposé tient compte de ces deux facteurs, en plus de tenir compte du biais de l’horloge. En appliquant la méthodologie proposée, un TDC avec une résolution de 50ps et une précision de 48ps a été obtenue, et les résultats montrent que des résolutions aussi faibles que 10ps peuvent être atteintes. A notre connaissance, à ce jour cette résolution correspond à la plus fine valeur rapportée dans la littérature pour une implémentation sur FPGA.

L’approche basée sur l’ajustement fin et précis des délais est également appliquée à une architecture préexistante de TDC utilisant la méthode de Vernier avec deux oscillateurs. Avec cette architecture, la résolution atteignable correspond en fait à

la différence de périodes entre deux oscillateurs démarrables. Notre méthode permettant d'ajuster précisément la période d'un oscillateur en anneau, l'application de l'approche proposée à cette architecture devrait permettre d'atteindre des résolutions plus fines que celles rapportées dans la littérature. Effectivement, nos résultats font état de résolutions inférieures à 5ps. Toutefois, étant donnée l'instabilité des oscillateurs, on observe que la précision obtenue à de telles résolutions n'est pas meilleure que 100ps. Néanmoins, des résolutions de l'ordre de 25ps sont obtenues avec des précisions équivalentes avec cette implémentation. Comparativement aux implémentations statiques proposées dans la littérature, notre approche utilisant la reconfiguration dynamique apporte une contribution intéressante en permettant de tenir compte des variations de température, aussi bien que des variations dues au procédé de fabrication.

Abstract

Time interval measurement is important in various scientific and engineering applications. A Time-to-Digital Converter (TDC) is an integrated circuit allowing measurement of time intervals with resolutions and precisions down to a picosecond. Until the last decade or so, these circuits were implemented exclusively as application specific integrated circuits (ASIC), but since then, various implementations targeting field-programmable gate arrays (FPGA) have been proposed. While these implementations still deliver reduced performances in terms of resolution and precision, there is a growing interest within the scientific community to reduce this gap. Indeed, with the dazzling progress of the FPGA technology over the past decade, both in terms of logic density and operation frequency, it is becoming a implementation target of choice for an ever growing range of circuit and systems. Two key benefits from such implementations are considerably reduced development times and non-recurring costs. It is therefore in this context that this work is focused on the FPGA implementation of time-to-digital converters.

While ASIC implementations of TDC can enable resolutions neighbouring a single picosecond, most recent FPGA implementations are still limited to a few tens of picoseconds. As the implementation of a TDC is closely related to the notion of delay, FPGAs are handicapped both by the irregularity of interconnection delays, and increased minimal delays. Therefore, to this day, the most successful FPGA implementations that have been proposed in the literature rely on architectures allowing to take advantage of dedicated interconnection structures, such as the carry-chain used in arithmetic circuits. Indeed, these dedicated interconnection structures provide both reduced delays and increased interconnection delay regularity. However, the resolution achievable with such architecture is limited by the minimal delays available on the circuit, which are substantially more important on an FPGA than on an ASIC. Nevertheless, this architecture directly benefits from newer generations of FPGAs, produced with fabrication processes that enable reduced minimal delay.

In order to address the limitation imposed by minimal delays, this work proposes an approach based on the fine and precise adjustment of delays. The novelty of this

approach resides in its ability to exploit two characteristics of FPGAs, namely the richness of the programmable interconnections fabric and the possibility to reconfigure the circuit dynamically. The latter, dynamic reconfiguration, is supported by some FPGAs and enables the user to reconfigure partially the circuit while in operation. As there exist a considerable amount of possible routes between two points of an FPGA, dynamic reconfiguration of the programmable interconnection fabric potentially enables delay adjustments with unprecedented resolutions. Indeed, our results show that the proposed delay line enables adjustments with a resolution of 1ps on a range that can span over 2,5ns.

On the basis of these fine resolution delay lines, a novel TDC architecture is proposed. Relying on the utilization of multiple parallel delay lines, the resolution achievable is now limited by the ability to adjust delay differences, instead of the previous limitation imposed by minimal delays. Therefore, such an architecture can potentially deliver resolutions lower than those found in state-of-the-art FPGA implementations. However, fine delay adjustment is itself insufficient to achieve the desired goal, and a calibration process enabling precise adjustment is also proposed. Delays are affected by both temperature and fabrication process variations of the chip. In order to enable precise delay adjustment, the proposed calibration take into account these variations, as well as clock skew. Using the proposed methodology, a TDC featuring 50ps resolution and 48ps precision was obtained. Moreover, results show that the methodology can be used to achieve resolutions as fine as 10ps. To our knowledge, to this day, such resolution is the lowest ever reported in the literature for an FPGA implementation.

Our approach based on fine and precise delay adjustment is also applied to a pre-existing TDC architecture based on the Vernier method with two oscillators. With this architecture, the achievable resolution corresponds to the period difference between the two oscillators. As our methodology enables precise period adjustment of a ring oscillator, the proposed implementation can deliver considerable resolution improvement when compared to those proposed in the literature. Indeed, our results show that resolutions below 5ps can be obtained. However, because of the instability of the oscillators, precision at such resolutions exceeds 100ps. Nevertheless, precisions of 25ps can be obtained at equivalent resolutions. Moreover, when compared

to static approaches proposed in the literature, our approach using dynamic reconfiguration is remarkable as it takes into account both temperature and fabrication process variations.

Table des matières

Dédicace	iii
Remerciements	iv
Résumé	v
Abstract	viii
Table des matières	xi
Liste des tableaux	xiii
Liste des figures	xiv
Liste des algorithmes	xvii
Liste des annexes	xviii
Liste des sigles et abréviations	xix
Chapitre 1 INTRODUCTION	1
Chapitre 2 REVUE DE LITTÉRATURE	8
2.1 Introduction	8
2.2 Résolution, Précision, et Exactitude	9
2.3 Critères de performance d'un TDC	13
2.4 Méthode d'interpolation	15
2.5 Architecture basée sur la méthode du vernier avec deux oscillateurs	16
2.6 Architecture basée sur la ligne à délais simple	19
2.7 Architecture basée sur les lignes à délais de type vernier	24
2.8 Conclusion	29
Chapitre 3 APPROCHE PROPOSÉE	31
3.1 Introduction	31

3.2	Une nouvelle architecture d'interpolateur basée sur l'ajustement fin des délais	32
3.2.1	Calibration de l'interpolateur proposé	33
3.2.2	Une ligne à délai ajustable avec une résolution de l'ordre de la picoseconde	36
3.2.3	Un oscillateur à fréquence variable précis à l'ordre de la picoseconde	41
3.2.4	Avantages de la reconfiguration dynamique	42
3.2.5	Implémentation sur un FPGA Virtex-II Pro	43
3.3	Une nouvelle implémentation de la méthode du vernier avec deux oscillateurs	49
3.3.1	Conception de l'oscillateur démarrable	49
3.3.2	Calibration de l'interpolateur	49
3.3.3	Implémentation sur un FPGA Virtex-II Pro	50
3.4	Conclusion	51
Chapitre 4 RÉSULTATS EXPÉRIMENTAUX		53
4.1	Introduction	53
4.2	Lignes à délais programmables	53
4.3	Oscillateur à fréquence variable	54
4.4	Une nouvelle architecture de TDC basée sur l'ajustement fin des délais	55
4.5	Une nouvelle implémentation de la méthode du vernier avec deux oscillateurs	59
4.6	Conclusion	64
Chapitre 5 CONCLUSION		67
5.1	Synthèse des travaux	67
5.2	Limitations de la solution proposée	68
5.3	Améliorations futures	69
Références		71
Publications de l'auteur		75
Annexes		76

Liste des tableaux

TABLEAU 2.1	Performances des interpolateurs utilisant la méthode du vernier avec deux oscillateurs.	30
TABLEAU 2.2	Performances des interpolateurs utilisant une ligne à délais simple.	30
TABLEAU 2.3	Performances des interpolateurs utilisant des lignes à délais de type vernier.	30
TABLEAU 4.1	Comparaison des TDC dans la littérature offrant un temps mort d'un cycle ou moins.	66
TABLEAU 4.2	Comparaison des interpolateurs utilisant la méthode du vernier avec deux oscillateurs.	66

Liste des figures

FIGURE 1.1	Capture d'une séquence vidéo à 30 images/seconde en utilisant des appareils photographiques, limités à 1 prise/seconde, reliés en série.	3
FIGURE 1.2	Illustration de l'approche proposée appliquée à l'exemple des appareils photographiques.	6
FIGURE 2.1	Analogie de la cible pour représenter la précision (a), la justesse (b) et l'exactitude (c) d'un instrument de mesure.	12
FIGURE 2.2	Méthode d'interpolation de Nutt, utilisant un compteur binaire et deux interpolateurs fins.	16
FIGURE 2.3	Diagrammes temporels (a), et architecture (b) de l'interpolateur basé sur la méthode du vernier avec deux oscillateurs. . .	17
FIGURE 2.4	Oscillateur à anneau utilisant une porte XOR proposé par Lin <i>et al.</i> (2006).	19
FIGURE 2.5	Échantillonnage de copies régulièrement retardées d'un même signal pour augmenter la fréquence d'échantillonnage efficace.	20
FIGURE 2.6	Architecture d'un interpolateur basé sur une ligne à délais sur écoute.	21
FIGURE 2.7	Implémentation d'une ligne à délais sur écoute avec la chaîne de retenue d'un additionneur proposée par Song <i>et al.</i> (2006).	22
FIGURE 2.8	Méthode d'union d'onde attaquant le problème des divisions larges.	23
FIGURE 2.9	Interpolateur basé sur les lignes à délais de type vernier.	24
FIGURE 2.10	Implémentation d'un interpolateur basé sur les lignes à délais de type vernier proposée par (Kalisz <i>et al.</i> , 1997).	26
FIGURE 2.11	Implémentation d'un interpolateur basé sur les lignes à délais de type vernier proposée par Minas <i>et al.</i> (2007).	27
FIGURE 2.12	Interpolateur utilisant plusieurs lignes à délais de type vernier proposé par Amiri <i>et al.</i> (2009).	28
FIGURE 3.1	Architecture d'interpolateur basé sur l'utilisation de lignes à délais programmables parallèles.	32

FIGURE 3.2	Architecture d'interpolateur proposée utilisant plusieurs ensembles de lignes à délais programmables parallèles.	33
FIGURE 3.3	Processus de calibration proposé pour ajuster les délais de l'interpolateur en tenant compte des variations de procédés et de température, ainsi que du biais d'horloge.	35
FIGURE 3.4	Reconfiguration dynamique de la ligne à délai en oscillateur en anneau, de manière à pouvoir mesurer les demi-périodes (proportionnelles aux délais) associées à chaque route.	37
FIGURE 3.5	Implémentation proposée de la ligne à délai programmable. . .	39
FIGURE 3.6	Implémentation du TDC proposé dans un Virtex-II Pro. Le design est partitionné en une région dynamique et une région statique.	44
FIGURE 3.7	Processus de génération du design statique.	46
FIGURE 3.8	Processus de génération et calibration du design dynamique du TDC proposé.	48
FIGURE 3.9	Implémentation proposée d'un oscillateur démarrable.	50
FIGURE 4.1	Demi-périodes associés à chaque route de la liste générée pour la ligne à délai programmable proposée.	54
FIGURE 4.2	Dérive de la demi-période de l'oscillateur à fréquence variable sur une période de 10 minutes.	55
FIGURE 4.3	Distribution des mesures effectuées en injectant un signal d'horloge à période constante, avec une moyenne de 124,9997ns et un écart-type de 40,3ps.	56
FIGURE 4.4	Influence d'un chauffage sur la précision pour différents intervalles de temps allant de 62,5ns à 2,048ms.	57
FIGURE 4.5	Non-linéarité différentielle (a) et intégrale (b), à une résolution de 50ps sur une portée dynamique de 10ns.	58
FIGURE 4.6	Non-linéarité différentielle (a) et intégrale (b), à une résolution de 25ps et sur une portée dynamique de 5ns.	60
FIGURE 4.7	Non-linéarité différentielle (a) et intégrale (b), à une résolution de 10ps et sur une portée dynamique de 2ns.	61
FIGURE 4.8	Stabilité sur 30 secondes des résolutions calibrées pour 25ps, 10ps, et 5ps.	62
FIGURE 4.9	Précision du TDC en fonction de la résolution.	63

FIGURE 4.10	Moyenne des intervalles de temps mesurés en fonction de la résolution du TDC pour des résolutions allant de 5ps à 50ps. .	64
FIGURE 4.11	Distribution de mesures d'intervalles de temps constantes montrant une moyenne de 1347ps et un écart type de 25,1ps pour une résolution de 26ps.	65
FIGURE A.1	Architecture de la bibliothèque DXCL.	77

Liste des algorithmes

ALGORITHME 3.1	Génération de routes reconfigurables	41
ALGORITHME A.1	Exemple simple d'application C avec la bibliothèque DXCL	79

Liste des annexes

ANNEXE A Bibliothèque DXCL	76
--------------------------------------	----

Liste des sigles et abréviations

ASIC	Application Specific Integrated Circuit
CLB	Configurable Logic Block
CMOS	Complementary Metal-Oxide-Semiconductor
CPLD	Complex Programmable Logic Device
DNL	Differential Non-Linearity
FPGA	Field Programmable Gate Array
ICAP	Internal Configuration Access Port
INL	Integral Non-Linearity
IOB	Input-Output Block
LSB	Least Significant Bit
LUT	Look-Up Table
OPB	On-chip Peripheral Bus
PIP	Programmable Interconnection Point
PLB	Processor Local Bus
TDC	Time-to-Digital Converter

Chapitre 1

INTRODUCTION

Les instruments de mesure et la science expérimentale sont intimement liés. En effet, au cœur même de la méthode scientifique, l'observation et l'expérimentation reposent sur des moyens d'observer et de quantifier certaines propriétés physiques. D'autre part, les avancées scientifiques et technologiques rendent possible la création de nouveaux appareils de mesures et l'amélioration de la précision de ceux qui existent. Les instruments de mesure sont omniprésents dans nos vies, autant que dans les laboratoires scientifiques. Par exemple, une balance nous permet de mesurer notre masse, un thermomètre nous permet de connaître la température, une règle à mesurer de connaître notre taille, jusqu'à l'horloge qui nous indique le temps présent.

De toutes les mesures possibles, le temps a ceci de particulier qu'il ne peut être fixé, s'écoulant toujours dans la même direction, seconde par seconde. La mesure du temps par l'homme remonte à l'antiquité, le cadran solaire en étant souvent considéré le plus ancien instrument. Son principe était tout simple, basé sur le déplacement du soleil, la position de l'ombre projetée par un objet indiquait l'heure du jour. À la même époque existait également la clepsydre, que l'on peut décrire comme un sablier à base d'eau. Le problème de la clepsydre était malheureusement que le débit d'écoulement de l'eau n'était pas constant, la pression exercée par l'eau diminuant au fil du temps. Le sablier qui apparut plus tard n'avait pas ce problème, son principe reposant sur l'écoulement de sable au travers d'un orifice, et possédait également l'avantage d'être portable, en plus de ne pas geler au froid ! Bien qu'il soit aujourd'hui utilisé comme objet de décoration plutôt que pour le temps qu'il indique, le sablier demeure toujours un symbole largement répandu du temps. Il fallu attendre le 14^e siècle pour voir l'apparition d'horloges mécaniques. Ces dernières utilisaient un mécanisme d'échappement afin de contrôler la rotation d'un engrenage souvent relié à un ressort où un poids suspendu. Le principe des montres de poche du 17^e siècle n'était pas si différent, sinon que la période d'oscillation du mécanisme d'échappement était

contrôlée par un balancier plutôt qu'un pendule. Les montres Quartz tant répandues aujourd'hui ont vu le jour après l'avènement de l'électronique. Elles tiennent leur nom du fait qu'elles utilisent un oscillateur électronique régulé par un cristal quartz. L'avantage principal de l'oscillateur à quartz est que sa période est bien plus précise que celle de son prédécesseur. Plutôt que d'utiliser un train d'engrenage pour compter les oscillations et les transformer en heures, minutes et secondes, un circuit numérique simple (compteur binaire) prend généralement en charge cette responsabilité.

L'électronique est plus omniprésente que jamais dans nos vies, que ce soit dans notre montre quartz, notre cafetière, notre voiture, jusqu'à notre ordinateur personnel, il semble que rien ne puisse freiner la prolifération des transistors dans les objets que nous utilisons quotidiennement. Les instruments de mesure n'y échappent pas non plus, bien au contraire ! Dans un monde où les circuits intégrés se retrouvent presque partout, il n'est pas surprenant de s'intéresser à la mesure de signaux électriques. Dans le domaine de la microélectronique, un signal numérique peut prendre deux valeurs possibles, c'est ainsi que l'on s'intéresse tout particulièrement aux moments de transition de ceux-ci, de même qu'à leur synchronisation. Au-delà du domaine de la micro-électronique, la mesure d'intervalles de temps à de très fines résolutions est essentielle dans plusieurs domaines d'applications scientifiques. Dans le domaine de la physique des particules, le temps de vol d'une particule sur une distance connue permet d'en connaître la masse. Un radar laser requiert la mesure du temps de vol d'aller-retour d'une pulsation lumineuse vers un objet pour connaître la distance à laquelle il se trouve. En imagerie médicale, la tomographie par émission de positrons requiert la détection de coïncidences des temps d'émission de paires de photons. Or, bien que le temps de course d'un coureur de 100m soit mesuré au centième de seconde, des applications de pointes telles que celles mentionnées précédemment requièrent des résolutions inférieures au milliardième de seconde !

Un convertisseur numérique de temps (TDC) est un circuit intégré permettant de mesurer des intervalles de temps à de très fines résolutions avec une haute précision. À l'ère des circuits numériques cadencés par des horloges dont la fréquence se chiffre en gigahertz (un milliard d'oscillations par seconde et plus), en principe un convertisseur numérique de temps pourrait se contenter d'appliquer le principe de la montre Quartz, utilisant un oscillateur et un compteur d'oscillations très rapides. Ainsi, à

des fréquences d'oscillations allant de 1 à 10 GHz, il serait possible d'obtenir des résolutions temporelles se situant entre un milliardième et un dixième de milliardième de secondes. Néanmoins, en pratique, cette approche pose problème. D'une part, la conception et fabrication de circuits opérant à de telles fréquence comporte de nombreuses difficultés et exige des procédés de fabrication de pointe très onéreux. D'autre part, de manière plus fondamentale, afin de mesurer ou d'observer des signaux numériques dont les fréquences peuvent atteindre les gigahertz, il est impératif d'échantillonner ces signaux à des fréquences plus élevées encore. Le problème de conception d'un convertisseur numérique de temps est ainsi d'offrir une résolution de mesure d'un ordre au moins dix fois plus fine par rapport aux périodes d'oscillations atteignables avec les procédés les plus récents. C'est un peu comme si quelqu'un désirait utiliser des appareils photos limités à une prise de photo par seconde pour capturer une séquence vidéo, à raison de 30 images par secondes. Après réflexion, une approche possible serait d'en utiliser 30 et de les déclencher les uns après les autres, à chaque 30ième de seconde. Malheureusement, aucun être humain n'est assez rapide pour suivre une telle cadence. Pour y parvenir, il pourrait être suffisant de propager un signal de capture à chaque appareil au travers d'un seul fil en série, de sorte à ce que le temps de propagation du signal entre chaque appareil soit de $1/30$ de seconde (voir Figure 1.1). Il ne resterait qu'à déclencher le signal à chaque seconde. Effectivement, le principe d'opération d'un convertisseur numérique de temps s'apparente souvent à celui exposé dans cet exemple des appareils photos, de sorte à surmonter les limitations imposées en termes de fréquence d'opération.

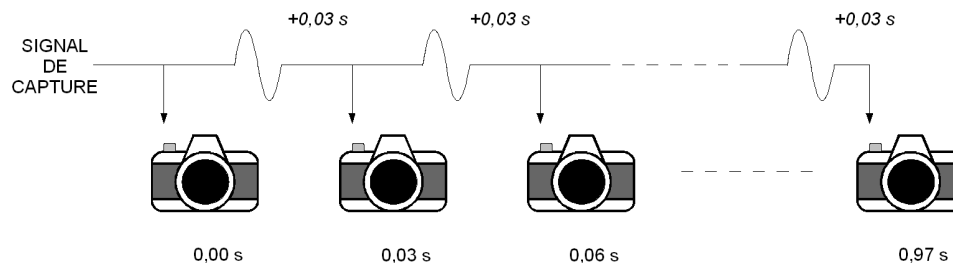


FIGURE 1.1 Capture d'une séquence vidéo à 30 images/seconde en utilisant des appareils photographiques, limités à 1 prise/seconde, reliés en série.

En matière de circuits intégrés, on distingue deux grandes classes : les circuits dédiés (*ASIC*), et les circuits (re-) programmables (*FPGA*, *CPLD*, *PLA*). La dis-

inction fondamentale entre ces deux classes est que les circuits (re-) programmables sont en fait des circuits dédiés à permettre, par programmation, l'implémentation de différents circuits. Cette capacité de programmer le circuit pour implémenter différents designs numériques a toutefois un prix. En effet, par rapport aux circuits dédiés, les circuits programmables offrent des performances nettement moindre, en termes de densité logique efficace, de fréquence d'opération maximale (délais d'interconnexions plus grands), et de consommation de puissance. En contrepartie, les puces (re-) programmables requièrent un effort de conception nettement moindre, entraînant des temps de mise en marché réduits. De plus, en termes économiques, la fabrication d'un circuit dédié implique des coûts non-récurrents massifs, d'autant plus que le processus de fabrication ciblé est récent. Il en résulte que les procédés de fabrication de pointe sont essentiellement réservés aux compagnies concevant des circuits vendus en très grands volumes, et disposant de capitaux importants, tel qu'est le cas pour les fabricants de processeurs *Intel* et *AMD*, ainsi que les fabricants de FPGA *Xilinx* et *Altera*. Lorsque des plus petits volumes de production sont ciblés, il faut alors se tourner vers des procédés de fabrication plus anciens, offrant des performances réduites, ou bien vers des puces programmables. Puisque les FPGA permettent de bénéficier des procédés de fabrication les plus récents, la différence de performance est alors moindre que lorsqu'un FPGA et un circuit dédié de procédés de fabrication équivalents sont comparés l'un à l'autre.

Historiquement, les convertisseurs numériques de temps étaient exclusivement implémentés sous forme de circuits intégrés dédiés. À ce jour, les meilleurs convertisseurs numériques de temps sont encore des circuits dédiés, atteignant des résolutions avoisinant la picoseconde. Néanmoins, avec les progrès fulgurants de la technologie FPGA, plusieurs se sont intéressés au courant de la dernière décennie à l'implémentation de ces circuits avec cette technologie, si bien qu'à ce jour des résolutions inférieures à 100ps (0,1 milliardième de seconde) sont ainsi atteintes. Bien que l'implémentation de TDC sur puces programmables bénéficie des connaissances développées pour les circuits dédiés, cette dernière est également confrontée à de nouveaux problèmes. Essentiellement, la problématique est que contrairement au circuit dédié qui permet une grande flexibilité de conception, le concepteur pour FPGA doit composer avec ce que le circuit programmable met à sa disposition. Mais revenant aux avantages et inconvénients discutés, la difficulté additionnelle d'une telle implémentation pro-

vient principalement du fait que les délais d'interconnexions sont sensiblement plus élevés, avec pour corollaire que la fréquence d'opération maximale est moindre.

Ce travail de recherche porte sur l'implémentation d'un convertisseur numérique de temps sur un FPGA reconfigurable dynamiquement. À sa plus simple expression, la reconfiguration dynamique, ou reconfiguration active partielle, consiste à reconfigurer une partie du circuit lors de son opération. Normalement, un FPGA est configuré avant sa mise en opération, mais certains supportent également la reconfiguration partielle en cours d'opération. La première contribution de ce travail est de proposer une méthode, tirant profit du système d'interconnexions programmables, permettant de réaliser des lignes à délais programmables avec une résolution de 1ps sur une plage d'environ 3ns. Une extension de cette méthode consiste en la réalisation d'un oscillateur en anneau à fréquence variable dont la période est précise à ± 1 ps sur une plage de 60ns. La deuxième contribution de ce travail est de proposer une nouvelle architecture de TDC basée sur l'utilisation de lignes à délais programmables, ainsi qu'un processus de calibration, utilisant un oscillateur à fréquence variable. Cette approche, basée sur la reconfiguration dynamique d'un FPGA, présente 3 avantages intéressants :

- Le processus de calibration tient compte des variations de températures ainsi que des variations physiques liées au procédé de fabrication, qui affectent tous deux les délais des interconnexions.
- Le processus de calibration n'est pas affecté par le biais de l'arbre d'horloge, utilisé dans l'implémentation de l'architecture proposée, lors de l'ajustement des délais.
- Les résolutions atteignables avec l'architecture proposée sont limitées par la finesse à laquelle peuvent être ajustés les délais du circuit, plutôt que les délais minimaux du circuit.

Il est possible d'établir un parallèle entre la méthodologie proposée et l'exemple des appareils photos présenté plus tôt. Dans cet exemple, afin de prendre 30 photos par seconde, les appareils avaient été reliés en série par un fil. En considérant la situation où le temps de propagation sur le fil du signal de capture, d'un appareil à l'autre, est supérieur à $1/30$ de seconde, un problème se pose. En fait, dans une telle situation, cette approche ne permet pas d'atteindre les performances visées. Cela s'apparente à

l'importance des délais dans un FPGA comparativement à ceux d'un circuit dédié (de même procédé). La méthodologie proposée permet de contourner ce problème, sous l'hypothèse d'une capacité d'ajuster précisément le délai d'un fil, même si ce dernier ne peut être réduit sous sa valeur minimale. Plutôt que de relier les appareils en série avec un long fil, nous proposons simplement d'utiliser 30 fils parallèles. Ainsi, afin d'obtenir les 30 photos par seconde, il suffit d'ajuster les délais de sorte à ce que les différences entre ceux-ci correspondent à $1/30$ de secondes, les uns après les autres (voir Figure 1.2).

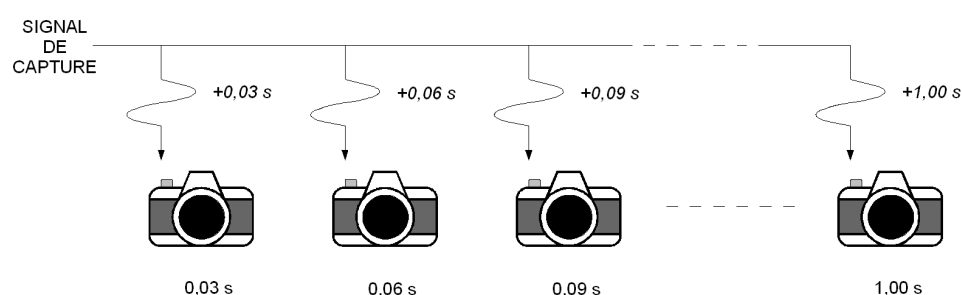


FIGURE 1.2 Illustration de l'approche proposée appliquée à l'exemple des appareils photographiques.

Avec la méthodologie et l'architecture proposée, nous avons réalisé un convertisseur numérique de temps fonctionnant à une résolution de 50ps avec une précision de 45ps. De plus, les résultats expérimentaux obtenus montrent que la méthodologie peut être utilisée pour atteindre des résolutions aussi fines que 10ps. À notre connaissance, cette résolution est la meilleure atteinte à ce jour sur FPGA. En troisième lieu, notre méthodologie basée sur l'ajustement fin des délais est appliquée à une architecture préexistante de TDC. Les résultats ainsi obtenus montrent une certaine amélioration par rapport à ceux proposés dans la littérature.

Le présent document est divisé en 5 chapitres. Afin de bien situer ce travail dans l'état de l'art, le deuxième chapitre présente une revue de la littérature en matière d'implémentation de convertisseurs numériques de temps sur FPGA. Le troisième chapitre de ce travail est consacré à la méthodologie proposée, tirant profit de la reconfiguration dynamique pour réaliser des délais programmables à fine résolution, ainsi qu'à l'architecture novatrice de TDC basée sur l'ajustement fin des délais et son

processus de calibration. Ce même chapitre présente également l'application du savoir-faire développé à une architecture de TDC basée sur l'utilisation de deux oscillateurs à fréquences légèrement différentes. Le chapitre 4 rapporte les résultats expérimentaux obtenus, tandis que le chapitre 5 vient conclure et suggérer de futures directions à ce travail.

Chapitre 2

REVUE DE LITTÉRATURE

2.1 Introduction

L'aube des premières implémentations sur FPGA de convertisseurs numériques de temps (TDC) à haute résolution remonte à la fin des années 90. Depuis, le sujet a été considérablement étudié et les implémentations de pointe offrent des résolutions et précisions sous les 20ps. Ce progrès est également lié à celui de la technologie FPGA, qui a été fulgurant lors de la dernière décennie. Néanmoins, les implémentations de TDC sur circuits dédiés surpassent toujours les implémentations programmables. Par exemple, dans l'implémentation proposée par Lee et Abidi (2008) une résolution de 1,25ps est obtenue en utilisant un procédé de fabrication à 90nm. Il est toutefois intéressant d'observer que dans un appareil de mesure de pointe tel le *U1050A* d'Agilent Technologies offrant des résolutions de 50ps à 5ps, bien que le TDC soit un circuit dédié, le circuit d'acquisition de l'appareil est implémenté dans un FPGA (Virtex-II Pro de Xilinx).

Certaines méthodes de conception de TDC font usage de circuits analogiques, tel un convertisseur analogique à numérique, et par conséquent ne permettent pas une intégration sur FPGA. Il existe 3 méthodes de conception numériques qui reviennent fréquemment dans la littérature : la méthode du vernier avec deux oscillateurs, la méthode de la ligne à délais simple « sur écoute », et la méthode des lignes à délais avec vernier. Ces méthodes, développées initialement pour les circuits dédiés sont ainsi applicables aux circuits programmables. L'implémentation FPGA de chacune de ces méthodes pose différents problèmes qui lui sont spécifiques. Dans ce chapitre, nous proposons au lecteur une revue des différentes architectures et implémentations FPGA de convertisseurs numériques de temps. Bien que cette dernière ne soit pas exhaustive, l'étude de l'état de l'art proposée expose les différents problèmes spécifiques à l'implémentation FPGA de chacune des trois méthodes mentionnées, et explore les

solutions développées afin de les adresser.

Les convertisseurs numériques de temps mesurent des intervalles de temps, typiquement définis par les moments d'arrivées d'une impulsion de départ et d'une impulsion d'arrêt. Bien qu'un compteur rapide d'oscillations ne soit pas une solution viable au problème, afin d'obtenir de bonnes résolutions pour de longs intervalles, un compteur grossier est typiquement utilisé. Afin d'augmenter la résolution offerte par ce compteur, de tels TDC comprennent également un *interpolateur fin* de temps, pour « interpoler » le temps entre les valeurs données par ce compteur. Il en découle que l'interpolateur fin est le circuit critique d'un TDC, déterminant pour sa résolution.

Dans ce chapitre, nous débutons par une discussion générale sur les termes et définitions utilisés dans la littérature afin de caractériser la qualité d'un instrument de mesure. La pertinence d'une telle section provient de l'existence de certaines ambiguïtés dans la terminologie. Nous poursuivons ensuite avec la présentation des critères de performances relatifs aux convertisseurs numériques de temps. Une fois ces caractéristiques permettant d'apprécier la qualité d'un convertisseur numérique de temps présentées, nous présentons une méthode d'interpolation fréquemment employée dans la littérature, consistant à utiliser un compteur binaire conjointement avec un interpolateur fin. À ce point, les 3 méthodes les plus fréquemment utilisées pour réaliser ces interpolateurs fins (sur FPGA) sont présentées, ainsi que les résultats obtenus pour différentes implémentations FPGA.

2.2 Résolution, Précision, et Exactitude

Dans cet ouvrage, les termes *résolution*, *précision*, et *exactitude* (*accuracy*) sont utilisés avec un sens bien précis. Néanmoins, les définitions de ces termes comportent certaines ambiguïtés. C'est ainsi que cette section propose une brève analyse des définitions de ces termes telles que retrouvées dans deux dictionnaires de renommée internationale (Merriam-Webster (2009), Kaplan (2004)). La pertinence d'une telle section, loin d'être anodine, est d'exposer les différents sens donnés à ces termes et de mettre en évidence certaines ambiguïtés quant à leurs définitions. L'objectif ainsi poursuivi est de permettre au lecteur d'apprécier pleinement les distinctions importantes qui existent entre ces termes, tels qu'employés dans ce travail.

Dans un premier lieu, voici les définitions des termes *precision* (précision), *resolution* (résolution), et *accuracy* (exactitude) tirées du dictionnaire de la langue anglaise Merriam-Webster (2009) :

precision : **1** : the quality or state of being precise : EXACTNESS.
2 a : the degree of refinement with which an operation is performed or a measurement stated – compare ACCURACY **2 b** : the accuracy (as in binary or decimal places) with which a number can be represented usually expressed in terms of the number of computer words available for representation <double precision arithmetic permits the representation of an expression by two computer words>. [...]

precise : **1** : exactly or sharply defined or stated. **2** : minutely exact.
3 : strictly conforming to a pattern, standard, or convention. [...]

exact : **1** : exhibiting or marked by strict, particular, and complete accordance with fact or a standard. **2** : marked by through consideration or minute measurement of small factual details.

sharp : **1** : adapted to cutting or piercing : as **a** : having a thin keen edge or fine point **b** : briskly or biting cold [...]. **6 a** : terminating in a point or edge <sharp features> [...]

resolution : **1** : the act or process of resolving : as [...] **6 a** : the process or capability of making distinguishable the individual parts of an object, closely adjacent optical images, or sources of light. **b** : a measure of the sharpness of an image usually expressed as the total number or density of pixels in the image <a resolution of 1200 dots per inch>

accuracy : **1** : freedom from mistake or error : CORRECTNESS. **2 a** : conformity to truth or to a standard or model : EXACTNESS **b** : degree of conformity of a measure to a standard of a true value – compare PRECISION 2a.

Merriam-Webster (2009)

En deuxième lieu, les définitions de ces termes, tirées du dictionnaire sur l'ingénierie électrique et électronique Kaplan (2004), sont les suivantes :

precision : **1.** A measure of the repeatability of a given result or procedure, such as measurement. For example, a measurement made several times in which the same, or nearly the same value is obtained, indicates precise measurement. This does not necessarily mean that such measurement specify or approximate a real value, thus a precise measurement may not be accurate. Still, even in technical usage the term is often used synonymously with **accuracy**. **2.** The number of significant digits provided in a measurement or other value. The more significant digits, the more precise. [...]

resolution : Its abbreviation is **res**. **1.** The minimum difference between two values, quantities, or entities, which can be distinguished by, separated, or acted upon, by a detector, sensor, measuring device, or the like. Also, such differentiation. **2.** The smallest detail which can be distinguished by, or acted upon, by a detector, sensor, measuring device, or the like. Also, such differentiation. [...] **6.** The precision with which a quantity or value can be measured or indicated. [...]

accuracy : The extent to which a measured or calculated value approximates the real value. May be expressed as a percentage, as in accurate within 2%.

Kaplan (2004)

Ironiquement peut-être, le terme dont le sens semble être le moins précis est la précision. Dans le premier sens donné au mot dans le dictionnaire *Merriam Webster*, on indique que la précision est définie comme étant quelque chose d'exactly défini, ce qui renvoie directement à la notion d'exactitude (*accuracy*), ou alors comme quelque chose de défini de manière pointue, ce qui renvoie plutôt à la notion de chiffres significatifs. Cette notion correspondant au deuxième sens donné au terme précision dans Kaplan (2004). En définissant le terme précision comme quelque chose de défini de manière pointue, la définition est également en accord avec le premier sens donné au terme précision dans Kaplan (2004). Dans le deuxième sens donné au mot précision dans le dictionnaire *Merriam Webster*, en 2.a) on indique le degré de raffinement avec lequel une opération est réalisée ou une mesure est énoncée, ce qui renvoie en quelques sortes à la notion d'incertitude compatible avec le premier sens donné au terme précision dans Kaplan (2004). En 2.b), on définit la précision comme

l'exactitude avec laquelle un nombre peut être représenté (précision double ou simple en calcul sur ordinateur par exemple). On observe ainsi que la définition du terme précision dans Merriam-Webster (2009) renvoie aussi bien à la notion de précision telle que définie dans Kaplan (2004), qu'à la notion d'exactitude. Tout cela explique bien pourquoi ces deux termes sont souvent utilisés comme synonymes.

Pour ce qui est de la définition du terme résolution, le sens 6.a) donné dans Merriam-Webster (2009) apparaît être en accord avec les deux premiers sens tirés de Kaplan (2004), faisant référence au plus petit détail pouvant être distingué. Le mot résolution possède toutefois plusieurs sens qui ne sont pas de nature scientifique ou technologique.

Bien qu'une ambiguïté liée à l'utilisation des termes *précision* et *résolution* puisse exister, dans le domaine de la métrologie il apparaît dans la littérature (Webster (1999a), Webster (1999b), AFNOR (1994)) que ces termes sont employés avec leur premier sens, tels qu'énoncés dans Kaplan (2004). À cet effet, une analogie pour illustrer la signification de ces termes au moyen d'une cible est fréquemment employée. La figure 2.1 illustre une telle analogie, où la résolution est représentée par la largeur des anneaux de la cible. La figure 2.1(a) illustre une bonne précision (fidélité) des mesures puisque ces dernières sont fortement regroupées. Or, bien que les valeurs obtenues dans ce cas d'une mesure à l'autre soient très similaires, en moyenne la valeur est considérablement éloignée du centre de la cible. La figure 2.1(b) illustre le concept de *justesse*. C'est à dire qu'en moyenne, la valeur obtenue est centrée sur

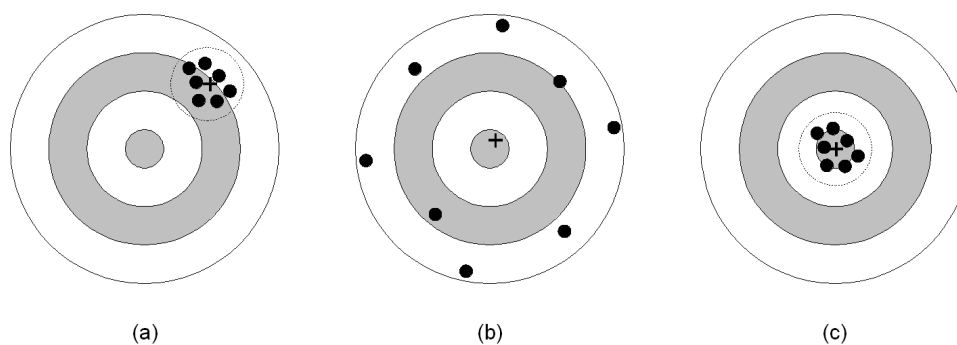


FIGURE 2.1 Analogie de la cible pour représenter la précision (a), la justesse (b) et l'exactitude (c) d'un instrument de mesure.

la cible (croix), mais sans pour autant que la précision soit bonne, de sorte que les mesures individuelles diffèrent fortement les unes des autres. L'exactitude est finalement représentée à la figure 2.1(c), par une combinaison de justesse et de précision, de sorte à ce que d'une mesure à l'autre, l'erreur avec le centre de la cible est faible.

2.3 Critères de performance d'un TDC

Un convertisseur numérique de temps est caractérisé par plusieurs paramètres, et comme c'est généralement le cas pour tout circuit électronique, certains avantages sont souvent obtenus au détriment d'une dégradation de certains autres paramètres. Il importe donc de bien comprendre leurs significations pour en avoir une appréciation juste. Les caractéristiques les plus importantes et les plus fondamentales d'un convertisseur numérique de temps sont les suivantes :

- La résolution (du bit le moins significatif)
- La précision des mesures
- L'exactitude des mesures
- La non-linéarité, différentielle et intégrale
- Le temps mort
- La durée maximale d'une mesure
- La largeur minimale d'impulsion

Sans surprises, les termes résolution, précision, et exactitude se retrouvent parmi les critères de performance les plus fondamentaux d'un TDC. Puisque le TDC est un instrument de mesure, ce travail utilise ces termes avec leurs sens tel que couramment utilisés dans le domaine de la métrologie (illustrés par l'analogie avec la cible de la Figure 2.1). Bien que le sens donné à chacun de ces mots dans ce travail ait été exposé à la section précédente, la signification de ces termes dans le cas précis de la caractérisation d'un TDC est ici présentée, en plus des définitions des autres paramètres mentionnés ci-haut.

La *résolution* du bit le moins significatif détermine le pas de quantification de l'interpolateur de temps utilisé. La résolution correspond ainsi au plus petit interval de temps pouvant être **théoriquement** résolu par un TDC pour une mesure donnée. Typiquement la précision d'un TDC devrait être meilleure que sa résolution, sans

quoi le seul intérêt lié à la haute-résolution est une erreur de quantification réduite (voir Kalisz, 2003).

La *précision* est définie par l'écart type des mesures réalisées en mesurant des temps d'arrivées d'impulsions équidistantes. La précision est fortement liée à la qualité du circuit, ainsi qu'à la qualité de l'horloge de référence qui est utilisé par ce dernier, notamment lorsque les intervalles de temps sont grands. En effet, en mesurant des impulsions équidistantes, une certaine variation est observée, ce qui est attribuable en partie à la gigue de l'horloge, mais également à d'autres phénomènes aléatoires (non-contrôlables) comme le bruit thermique par exemple. On prend ainsi généralement l'écart type par rapport à la moyenne comme critère de précision d'un TDC. La précision est également utilisée afin de déterminer le plus petit interval de temps pouvant être **réellement** résolu par un TDC pour une mesure donnée (chiffres significatifs).

L'*exactitude* d'un TDC définit l'erreur séparant la valeur d'une mesure de la quantité réelle qui est mesuré. L'exactitude est ainsi affectée par les variations aléatoires autour d'une valeur moyenne (précision), en plus de la présence d'erreurs systématiques occasionnant un biais entre la valeur de la moyenne mesurée et la valeur réelle qui est mesurée (justesse). L'exactitude peut ainsi être considérablement améliorée par compensation des erreurs systématiques, suite à un processus de calibration.

La *non-linéarité différentielle* du TDC mesure la déviation d'un pas de quantification par rapport à la valeur idéale correspondant à celle du bit le moins significatif. Par exemple, dans un TDC avec une résolution de 100ps, il est peu probable que chacun des bits soit distancé d'exactly 100ps. On aurait certains espacement de 110ps, d'autres de 90ps, ce qui correspondrait à des non-linéarités différentielles de +/-10ps pour ces bits respectifs. Afin de mesurer la non-linéarité différentielle, une méthode très répandue consiste à injecter des impulsions dont les espacements temporels sont pseudo-aléatoires. Considérant un nombre élevé d'impulsions, la taille des sous-divisions peut être estimée en observant le nombre d'impulsions ayant été détectées dans chacune des divisions de l'interpolateur par rapport au nombre total d'impulsions.

La *non-linéarité intégrale* fait plutôt référence à la déviation par rapport à la droite idéale qui représente la relation d'entrée et sortie de l'interpolateur (voir Dyer, 2001). Cette dernière est simplement obtenue en intégrant les erreurs différentielles des divisions.

Le *temps mort* indique le taux maximal auquel les impulsions peuvent se présenter à l'entrée du convertisseur (voir Hewlett Packard, 1997). Dans le cas d'un convertisseur mesurant le temps entre une impulsion de départ et une impulsion d'arrêt, le temps mort correspond au temps qui doit séparer l'arrivée d'une impulsion d'arrêt avec l'arrivée de la prochaine impulsion de départ.

La *durée maximale d'une mesure* correspond simplement à la distance maximale séparant deux impulsions qui peut être mesurée. Lorsque le convertisseur fonctionne avec un signal de départ et d'arrêt, ce paramètre indique le temps maximal pouvant être mesuré entre ces deux impulsions.

La *largeur minimale d'impulsion* correspond à la plus petite taille d'une impulsion qui peut être perçue par le circuit de conversion. Ce paramètre est important car toute impulsion de taille inférieure sera ou pourrait être ignorée, faussant aussi les mesures.

2.4 Méthode d'interpolation

Afin de réaliser des mesures qui s'étendent sur une longue période de temps avec de fines résolutions et de hautes précisions, une méthode d'interpolation de temps consiste à utiliser un compteur binaire offrant une résolution grossière et une grande portée, conjointement avec un ou des interpolateurs fins offrant de très fines résolution sur de courtes portées. La méthode la plus fréquemment employée est la méthode de Nutt (voir Dyer, 2001), illustrée en Figure 2.2, utilisant 2 interpolateurs fins. Le premier interpolateur est utilisé pour mesurer le court temps T_1 séparant l'arrivée du signal de départ avec le front d'horloge montant suivant. Sur ce même front, le compteur binaire mesure le temps T_{12} jusqu'au prochain front montant d'horloge qui suivra l'arrivée du signal d'arrêt. Le deuxième interpolateur est utilisé pour mesurer le temps T_2 séparant l'arrivée du signal d'arrêt avec ce dernier front d'horloge. On

obtient alors que le temps de mesure entre l'arrivée des signaux de départ et d'arrêt est donné par :

$$T = T_1 + T_{12} - T_2 \quad (2.1)$$

Cette méthode permet ainsi d'étendre la portée de mesure d'un interpolateur fin tout en conservant sa résolution et précision. Toutefois, d'autant plus que les mesures sont grandes, l'influence de la gigue (*jitter*) de l'horloge pourra réduire la précision de ces dernières.

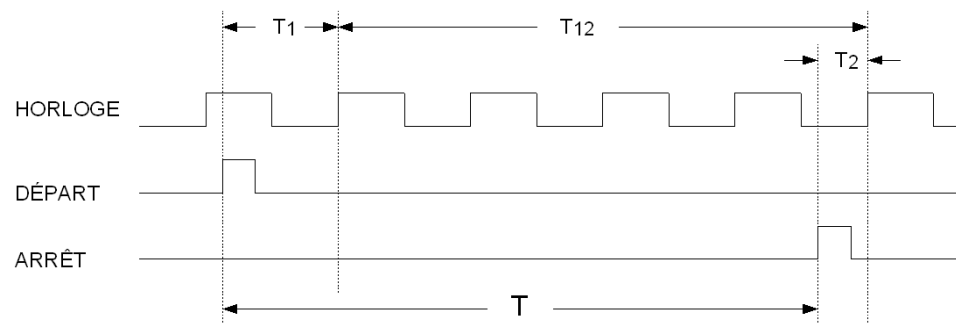


FIGURE 2.2 Méthode d'interpolation de Nutt, utilisant un compteur binaire et deux interpolateurs fins.

2.5 Architecture basée sur la méthode du vernier avec deux oscillateurs

Afin de produire une interpolation avec une fine résolution, la méthode du vernier se base sur l'utilisation de deux oscillateurs pouvant être démarrés et arrêtés, dont les périodes sont sensiblement différentes. La résolution qu'il est possible d'atteindre avec cette méthode correspond en fait à la différence entre les périodes des deux oscillateurs. Tel qu'illustré à la Figure 2.3b, chacun de ces oscillateurs est connecté à un compteur binaire, de même qu'à un circuit pouvant détecter la coïncidence des fronts montants des horloges produites par chaque oscillateur.

Lors de son opération, le signal de départ vient activer l'oscillateur le plus lent tandis que le signal d'arrêt vient activer l'oscillateur le plus rapide. Ainsi, lorsqu'une coïncidence est détectée entre les fronts montants des deux oscillateurs, ceux-ci sont

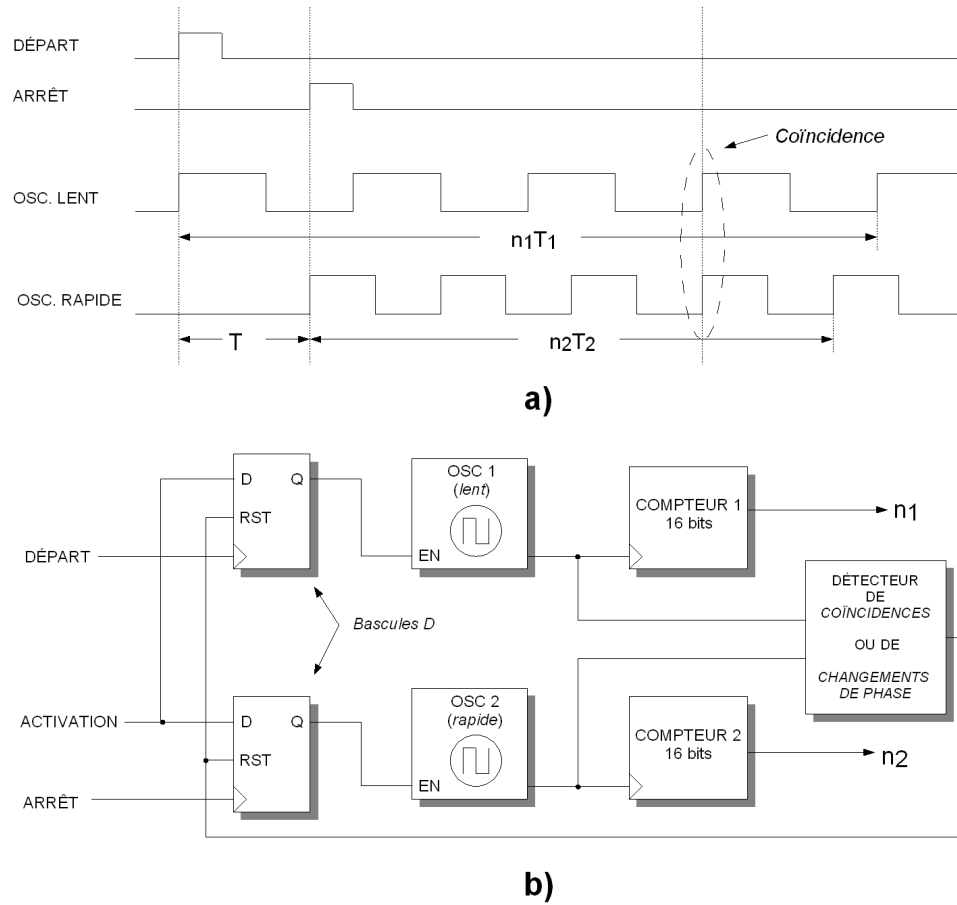


FIGURE 2.3 Diagrammes temporels (a), et architecture (b) de l'interpolateur basé sur la méthode du vernier avec deux oscillateurs.

désactivés de sorte à arrêter l'incrémentation des compteurs binaires. Maintenant, soit τ_1 et τ_2 les périodes des oscillateurs, avec $\tau_1 < \tau_2$, et n_1 , n_2 les valeurs des compteurs binaires, il est possible de montrer (Figure 2.3a) que le temps mesuré sera :

$$T = (n_1 - 1)\tau_1 - (n_2 - 1)\tau_2 \quad (2.2)$$

De même, lorsque les mesures réalisées sont inférieures à τ_2 , $n_1 = n_2$ et le temps mesuré est donné par :

$$T = (n_1 - 1)(\tau_1 - \tau_2) \quad (2.3)$$

On peut alors remplacer le circuit de détection de coïncidences par un détecteur de changement de phase, et se contenter d'un seul compteur binaire. Ainsi lorsque

le signal d'horloge rapide, initialement en retard, devance l'horloge plus lente, les oscillateurs sont arrêtés et la mesure est donnée par l'Équation 2.3.

Un des grands avantages offert par cette méthode est sans contredit sa simplicité ainsi que le faible coût matériel qui lui est associé, résultant en un circuit de très petite taille. Cependant cette méthode possède certains inconvénients. D'abord, il est essentiel que les oscillateurs employés soient stables, sans quoi la fiabilité des mesures ne peut être assurée, surtout lorsque les mesures s'étendent sur une longue durée (longue durée séparant l'arrivée des impulsions de départ et d'arrêt). Un autre inconvénient majeur avec cette approche est le temps mort du convertisseur. En effet, pour des périodes τ_1 et τ_2 , on obtient que dans le pire cas (voir Kalisz, 2003), le temps de conversion est donné par :

$$T = \frac{(\tau_1 \tau_2)}{(\tau_1 - \tau_2)} \quad (2.4)$$

En observant que le dénominateur correspond à la résolution, on voit qu'il existe un compromis entre la résolution et le temps de conversion.

Utilisant cette approche, des chercheurs de l'université Kun Shan de Taiwan on publié en 2006 des résultats quant à la réalisation de deux oscillateurs à anneau modifiés avec des différences de périodes aussi faibles que 30ps (voir Lin *et al.*, 2006). Afin de pouvoir en contrôler le départ et l'arrêt, ces oscillateurs sont conçus avec des portes de type ou-exclusif plutôt que des inverseurs, tel qu'illustré à la Figure 2.4, et leurs périodes se situent autour de 4ns. L'obtention de telles différences de périodes se base sur l'utilisation astucieuse des contraintes de placement et les différences de délais produits lors de la phase de placement et routage avec les outils de synthèse de Xilinx en ciblant les FPGA Spartan-3 et Virtex-4. Les auteurs proposent également un circuit pour la détection de phase, cependant aucun résultat expérimental supplémentaire d'implémentation pour le convertisseur numérique de temps n'est présenté.

Dans une conférence sur la science nucléaire, en 2007, une équipe de chercheurs de l'Institut de Technologie Industrielle de Corée à également publié des résultats de travaux similaires (voir Jin et Moon, 2007). Les deux oscillateurs produits avec des périodes de 2,25ns et 2,34ns permettent d'obtenir une résolution inférieure à 100ps. En activant les oscillateurs, une détection de coïncidence à lieu toutes les 61ns, les compteurs binaires indiquant alors des valeurs de 26 et 27.

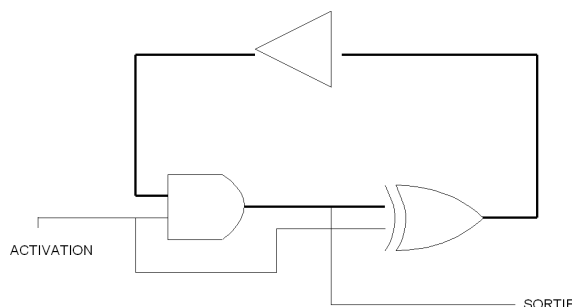


FIGURE 2.4 Oscillateur à anneau utilisant une porte XOR proposé par Lin *et al.* (2006).

2.6 Architecture basée sur la ligne à délais simple

La méthode d'interpolation basée sur la ligne à délais simple « sur écoute » est sans aucun doute la plus intuitive. En plus d'être la méthode la plus utilisée, c'est celle qui produit les meilleurs résultats d'implémentations FPGA. L'idée consiste à diviser une ligne à délais en N délais égaux de sorte à produire autant de copies retardées du signal. En échantillonnant ces copies au même moment avec une horloge par exemple, on mémorise alors différents points (Figure 2.5) équidistants les uns des autres du signal source. Pour autant que la ligne à délais produise un retard total au moins aussi grand que la période d'horloge, on parvient effectivement à accroître la résolution d'échantillonnage d'un facteur N . La Figure 2.6 illustre un circuit permettant ce type d'interpolation fine. Dans un convertisseur numérique de temps fonctionnant avec des signaux de départ et d'arrêt, il serait suffisant de remplacer le signal data de la figure par le signal de départ, et de remplacer l'horloge par le signal d'arrêt. Une autre variante de ce circuit est de placer la ligne à délais chaînés sur le signal d'horloge plutôt que le signal de donnée, ce qui est équivalent d'un point de vue fonctionnel.

Dans une telle architecture d'interpolateur, la résolution obtenue correspond au délai d'un maillon de la chaîne. Dans un FPGA, l'utilisation de tampons (*buffers*) dans la chaîne pose toutefois problème puisque ceux-ci, réalisés au moyen de tables de correspondances, possèdent alors un délai de l'ordre de quelques centaines de picosecondes, sans compter le délai de routage des connexions entre ces derniers. Un des problèmes importants avec les délais liés au routage entre les éléments logiques d'un FPGA est que ces derniers sont généralement produits de manière imprévisible

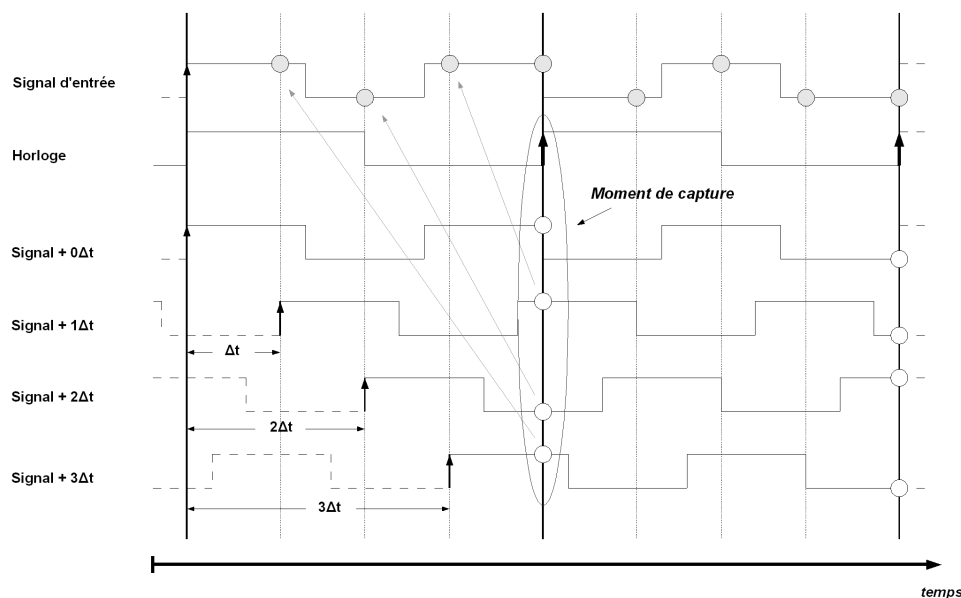


FIGURE 2.5 Échantillonnage de copies régulièrement retardées d'un même signal pour augmenter la fréquence d'échantillonnage efficace.

(hors du contrôle de l'utilisateur) par les outils de synthèses, pouvant ainsi entraîner d'importantes irrégularités. Cela s'explique notamment par le fait qu'il existe un nombre important de routes pouvant servir à relier une sortie d'un élément logique à l'entrée d'un autre via le système d'interconnexions programmables. Il est bien entendu possible de router manuellement ces interconnexions, mais l'effort associé en sera d'autant plus important, et cette approche ne permet toujours pas de tenir compte des variations de délais dues au procédé de fabrication et à la température. De plus, le routage au travers d'un système d'interconnexion programmable entraîne des délais non-négligeables pouvant limiter la résolution atteignable, puisque cette programmabilité est elle-même obtenue par une multitude de multiplexeurs.

Il existe toutefois des types d'interconnexions dédiées dans les FPGA modernes permettant de chaîner la logique avec des délais réduits. Deux exemples répandus et bien connus sont la chaîne de sommes de produit, et la chaîne de retenues qui peut aussi bien être utilisée pour la logique arithmétique que pour l'expansion de produits. Du point de vue du concepteur de ligne à délais chaînés, ces structures présentent un double avantage : d'une part, ces structures dédiées permettent d'obtenir de plus faibles délais que ceux obtenus au travers du système d'interconnexions program-

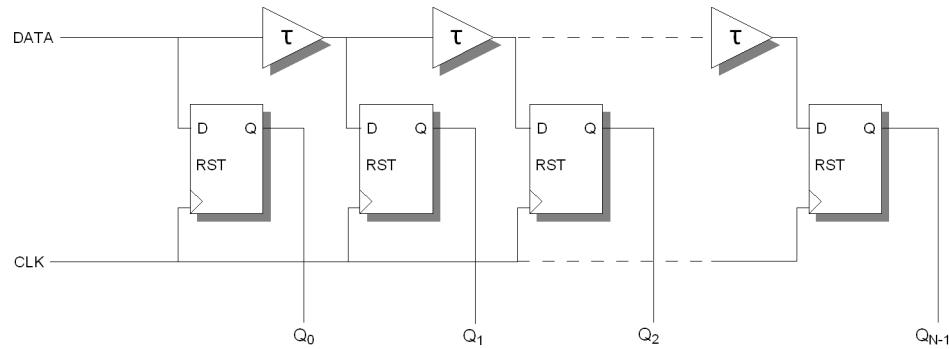


FIGURE 2.6 Architecture d'un interpolateur basé sur une ligne à délais sur écoute.

mables, et d'autre part, puisqu'elles ne sont pas programmables, elles permettent d'obtenir une meilleure régularité. Une conception soignée, ciblant ces structures de routage pour l'implémentation d'une ligne à délais chaînés, peut ainsi mener à des résolutions intéressantes dans un FPGA.

Utilisant cette approche, en 2003 une équipe de chercheurs du laboratoire de l'accélérateur national de Fermi (*Fermilab*) a développé un convertisseur numérique de temps avec une résolution de 400ps et une erreur maximale sur les mesures de 300ps (voir Wu *et al.*, 2003). Leur implémentation se base sur la chaîne d'expansion logique rapide présente dans le FPGA ACEX 1K d'Altera, divisée en 48 chaînons. Les tests ont ensuite été effectués avec le convertisseur cadencé à des fréquences d'horloges de 35MHz et 70MHz. La précision mesurée se situe à moins de 0,5 LSB (bit le moins significatif), toutefois le graphe de non-linéarité différentielle présente des pics réguliers (0,5 à 0,6 LSB) toutes les 8 divisions. Cette observation est facilement expliquée par les sauts entre les différents blocs logiques (Logic Array Block) qui impliquent un délai plus long.

En 2006, une équipe de recherche du laboratoire d'électronique rapide de l'Université de Science et Technologie de Chine (USTC) publie une autre architecture intéressante (voir Song *et al.*, 2006). L'idée consiste à utiliser la chaîne de retenue rapide d'un additionneur afin de produire la ligne à délais (Figure 2.7). La résolution ainsi obtenue est équivalente au délai de propagation de la retenue au travers un additionneur complet de 1bit. L'architecture de l'interpolateur proposé a été implémenté dans un FPGA ACEX 1K d'Altera ainsi que dans un Virtex-II de Xilinx.

Avec l'implémentation sur le FPGA d'Altera, la résolution obtenue est de 112,5ps avec une précision (déviatoin standard) de 129,4ps, soit de 91,5ps rms. En observant le graphe de non-linéarité différentielle obtenu, on retrouve le même phénomène de pics périodiques et cette dernière est comprise entre -0,416 et +0,783 divisions. La non-linéarité intégrale est ainsi comprise entre -0,567 et +0,687 divisions. Pour l'implémentation sur le FPGA de Xilinx, la taille d'une division est de 69,5ps et la précision obtenue est de 93,1ps, soit 65,8ps rms. En ce qui a trait aux graphes de non-linéarité différentielle et intégrale, le premier offre un écart de -0,954 à +1,051 divisions, et le deuxième un écart allant de -2,003 à +1,855 divisions. Sur les deux implémentations, le temps mort peut être réduit à un seul cycle d'horloge, soit 5ns où 10ns respectivement.

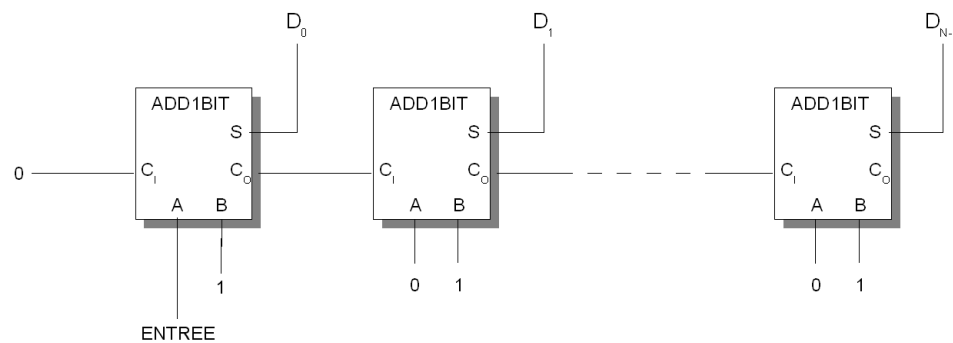


FIGURE 2.7 Implémentation d'une ligne à délais sur écoute avec la chaîne de retenue d'un additionneur proposée par Song *et al.* (2006).

En 2009, une équipe de l'École Polytechnique de Lausanne a publié des résultats d'une implémentation d'un TDC basée sur la chaîne de retenue d'un FPGA fabriqué en technologie 65nm (Virtex-5) (voir Favi et Charbon, 2009). Leurs résultats font état d'une résolution de 17ps avec une précision d'au moins 25ps. La ligne à délais contient 207 divisions et le circuit opère à une fréquence de 300MHz. À cette résolution, la non-linéarité différentielle observée se situe entre -1 et 3,55 divisions, tandis que la non-linéarité intégrale est comprise entre -3 et 2,58 divisions. Ces résultats mettent en évidence une des grandes forces de cette architecture : l'implémentation bénéficie directement des nouveaux procédés de fabrication des FPGA modernes, qui offrent typiquement des délais réduits (portes logiques et interconnexions) et des fréquences d'opération supérieures (requiert une portée dynamique réduite).

Voulant s'attaquer au problème de non-linéarité des architectures basées sur les structures d'interconnexions dédiées pour le chaînage de logique sur FPGA, en 2008 des chercheurs du laboratoire Fermi ont publié une méthode originale permettant de sous-diviser les divisions relativement larges qui apparaissent périodiquement dans ces implémentations (voir Wu et Shi, 2008). La méthodologie dénommée *Wave Union* (union d'onde) consiste en l'envoi d'un train d'impulsions dont l'espacement est connu plutôt que d'une seule impulsion, avec pour objectif d'assurer qu'une de ces impulsions voit systématiquement sa transition avoir lieu dans une des divisions plus fine. La Figure 2.8 illustre le principe avec un train de 2 impulsions, on y voit bien qu'en ajustant soigneusement la distance entre les impulsions du train, considérant la présence périodique de grandes divisions, si une des impulsions tombe dans une division plus large, alors l'autre a lieu dans une division normale.

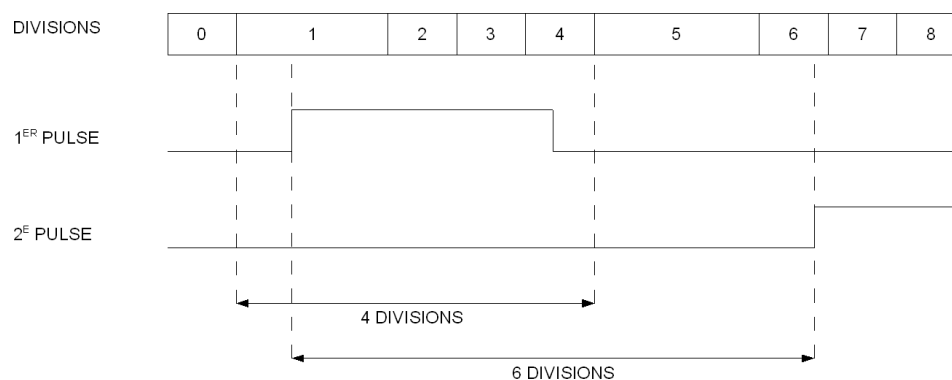


FIGURE 2.8 Méthode d'union d'onde attaquant le problème des divisions larges.

Ainsi cette méthode permet de sous-diviser les divisions très larges, ou alors d'améliorer la sensibilité du convertisseur numérique de temps. Il est important de noter toutefois que cette amélioration de la résolution se fait au dépend du temps mort du convertisseur, puisque pour la mesure du temps d'arrivée de chaque impulsion en entrée, la ligne à délais doit être monopolisée par chacun des « wagons » du train d'impulsions émis en substitution. Donc, l'accroissement du nombre d'impulsions augmente la résolution des mesures en même temps qu'il augmente le temps mort du convertisseur. Ainsi les auteurs présentent deux variantes de leur architecture : la première utilise un train de 2 impulsions et la deuxième un train de 16 impulsions.

Dans leur première implémentation sur un FPGA Cyclone II d'Altera, le train d'impulsions se répète à une fréquence de 53,11MHz, alors que le convertisseur numérique est cadencé à 400MHz et fonctionne avec 64 divisions. Au départ, la résolution obtenue se situe à 60ps, et la précision atteinte est de 58ps. En employant la technique d'union d'ondes, la résolution efficace est réduite à aussi peu que 30ps avec une précision de 25ps, mais le temps mort passe de 2,5ns à 5ns. Dans leur deuxième implémentation, en augmentant le nombre d'impulsions la précision est réduite à 10ps, alors que le temps mort passe de 2,5ns à 45ns.

2.7 Architecture basée sur les lignes à délais de type vernier

La méthode de type vernier avec des lignes à délais différentielles est un raffinement intéressant de la méthode utilisant une ligne à délais simple. Le point fort de cette approche est qu'elle permet d'obtenir une résolution correspondant à la différence de délais qui existe entre un maillon d'une chaîne avec un maillon de l'autre. La Figure 2.9 présente un exemple d'implémentation d'une telle ligne à délais, avec τ_1 supérieur à τ_2 .

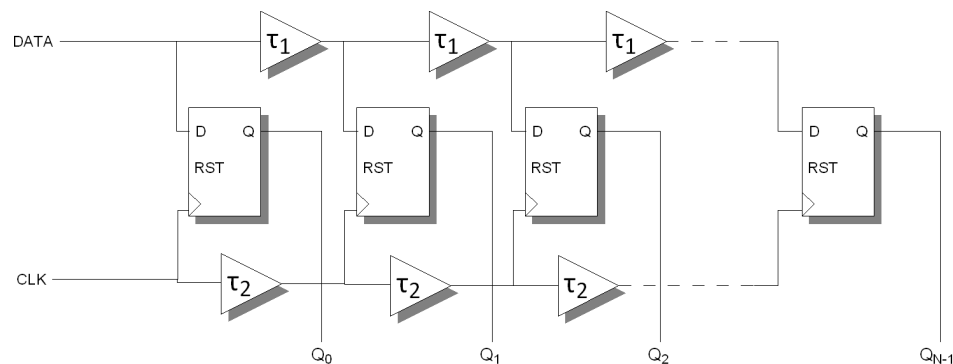


FIGURE 2.9 Interpolateur basé sur les lignes à délais de type vernier.

Dans le cas d'une ligne à délais chaînés simple, la résolution atteignable est limitée au délai minimal τ_1 qu'il est possible d'obtenir, mais dans le cas d'une ligne à délais différentielle de type vernier, c'est plutôt la différence de ce dernier avec un autre délai

τ_2 qui importe. Ainsi, cette méthode offre potentiellement la capacité de produire de meilleures résolutions que la méthode des délais simples. Toutefois, en vue d'une implémentation sur FPGA, cette approche pose problème. Il a déjà été mentionné dans la section précédente que le routage des interconnexions posait un problème important de régularité des délais des divisions. Avec la méthode de la ligne à délais de type vernier, le problème semble doublé, puisque l'arbre d'horloge (dédié) est remplacé par une nouvelle série d'interconnexions. Néanmoins, différentes implémentations sur FPGA ont été réalisées avec cette architecture et sont ainsi présentées dans cette section.

En 1997, à l'aube des premières implémentations de TDC sur FPGA, une équipe de chercheurs de l'Université Militaire de Technologie de Pologne publie des résultats d'une implémentation offrant une résolution de 200ps (voir Kalisz *et al.*, 1997). Le FPGA utilisé du fabricant QuickLogic utilise une structure anti-fusible et est produit en technologie CMOS de $0,65\mu\text{m}$, offrant un délai minimal de 1,7ns pour une porte ! Il n'est ainsi peut-être pas si surprenant que l'approche basée sur les lignes à délais de type vernier ait été choisie. L'architecture de l'interpolateur proposé, illustré en Figure 2.10, utilise la différence entre le délai d'un tampon (*buffer*) et le délai d'un verrou (*latch*) afin d'obtenir la résolution mentionnée. Le temps mesuré par ce circuit est celui séparant la montée des signaux *IN* et *EN**. Lorsque le signal *IN* devient haut, il se propage au travers des verrous qui sont initialement transparents, et pendant que le '1' logique se propage d'un verrou à l'autre, le verrou précédent est réinitialisé. En parallèle, lorsque le signal *EN** devient haut, celui-ci se propage au travers des tampons de sorte à verrouiller les bistables D. Il en résulte un mot dans lequel la position du '1' indique le délai mesuré. Une caractéristique intéressante du convertisseur produit est que la non-linéarité différentielle est restreinte à un intervalle de -0,47 à +0,44 incréments de 200ps. La non-linéarité intégrale résultante s'étend de -0,2 à +1,28 incréments. Sur un échantillon de six FPGA programmés, la résolution obtenue se situe entre 200ps et 238,1ps, et le convertisseur possède une portée dynamique de 10ns correspondant à la période de l'horloge de référence employée.

Dans des travaux plus récents, des scientifiques de l'Université de Newcastle au Royaume-Uni ont publié en 2007 une architecture reposant sur une méthode de calibration permettant de prendre en compte les variations liées aux procédés de fa-

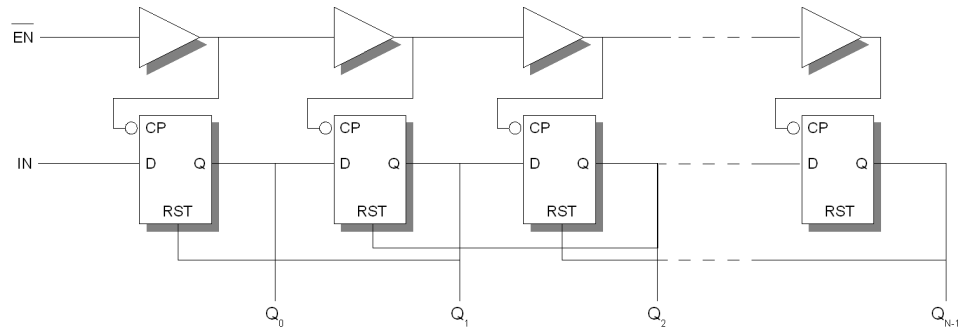


FIGURE 2.10 Implémentation d'un interpolateur basé sur les lignes à délais de type vernier proposée par (Kalisz *et al.*, 1997).

brication (voir Minas *et al.*, 2007). L'architecture est basée sur les lignes à délais chaînés différentielles tel qu'illustré en début de section à la Figure 2.9, mais à ceci de particulier que les délais utilisés se limitent à ceux des interconnexions. Puisque les différences de délais ainsi obtenues sont imprévisibles, un processus de calibration, basé sur l'utilisation d'autant de compteurs binaires que de divisions dans les lignes à délais, est présenté. Dans l'implémentation proposée, illustrée à la Figure 2.11, 32 divisions sont produites et la sortie de chacun des 32 registres est reliée à un compteur binaire de 20 bits. La sortie de chacun de ces compteurs passe ensuite par un multiplexeur afin d'en permettre la lecture. Le processus de calibration consiste à solliciter le convertisseur numérique de temps avec des impulsions arrivant à de moments pseudo-aléatoire afin de collectionner suffisamment de donnée sur les transitions montantes des registres, comptabilisées avec les compteurs binaires. Par la suite, une analyse statistique permet de déterminer la portée dynamique, la résolution, ainsi que la précision du convertisseur. Les résultats obtenus avec cette implémentation sur un Virtex-II de Xilinx indiquent une résolution de 61,9ps sur une portée dynamique limitée à 1,64ns.

En 2009, des chercheurs de l'École Polytechnique de Montréal ont publié une architecture pour le moins intéressante faisant un usage astucieux des lignes à délais différentielles de type vernier (voir Amiri *et al.*, 2009). L'innovation de cette architecture consiste à utiliser plusieurs lignes différentielles retardées les unes par rapport aux autres, plutôt que d'avoir recours à une seule longue chaîne. L'architecture est organisée sous forme d'une matrice rectangulaire où chaque rangée est une ligne à délais

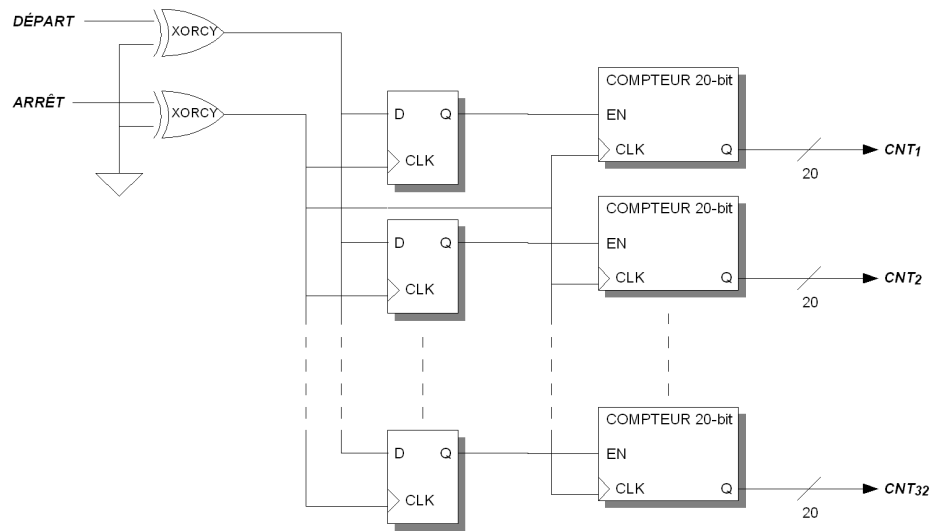


FIGURE 2.11 Implémentation d'un interpolateur basé sur les lignes à délais de type vernier proposée par Minas *et al.* (2007).

différentielle de type vernier, et la sortie de la première cellule de délais de chaque rangée est connectée à l'entrée de la première cellule de la rangée suivante. La cellule de délai employée est identique à celle présentée dans l'interpolateur de la Figure 2.10, composée d'un tampon et d'un verrou, cependant les interconnexions entre ces dernières sont différentes tel qu'illustré dans la Figure 2.12. Le circuit proposé fait également appel à des générateurs d'impulsions de tailles fixes, transformant un front montant en une courte impulsion, afin de produire les signaux de réinitialisation des verrous.

Afin d'adresser le problème de non-uniformité du routage, dans l'implémentation proposée sur un FPGA Spartan-3 de Xilinx, le placement des cellules à délai est réalisé à raison d'une cellule par bloc combinatoire logique (*CLB*). Cette approche est toutefois coûteuse en espace, puisqu'un *CLB* contient 4 tranches possédant chacune deux tables de correspondance (*LUT*) et deux verrous. Utilisant cette architecture basée sur les lignes à délais différentielles de type vernier avec une matrice de 25 lignes et 15 colonnes, la résolution obtenue est de 75ps, et le convertisseur numérique de temps possède une précision oscillant autour de 300ps. La portée dynamique est de 56,32us. Dans les résultats présentés, l'analyse de non-linéarité est réalisée dans un premier lieu sur les incréments grossiers entre les premières cellules de chaque ligne.

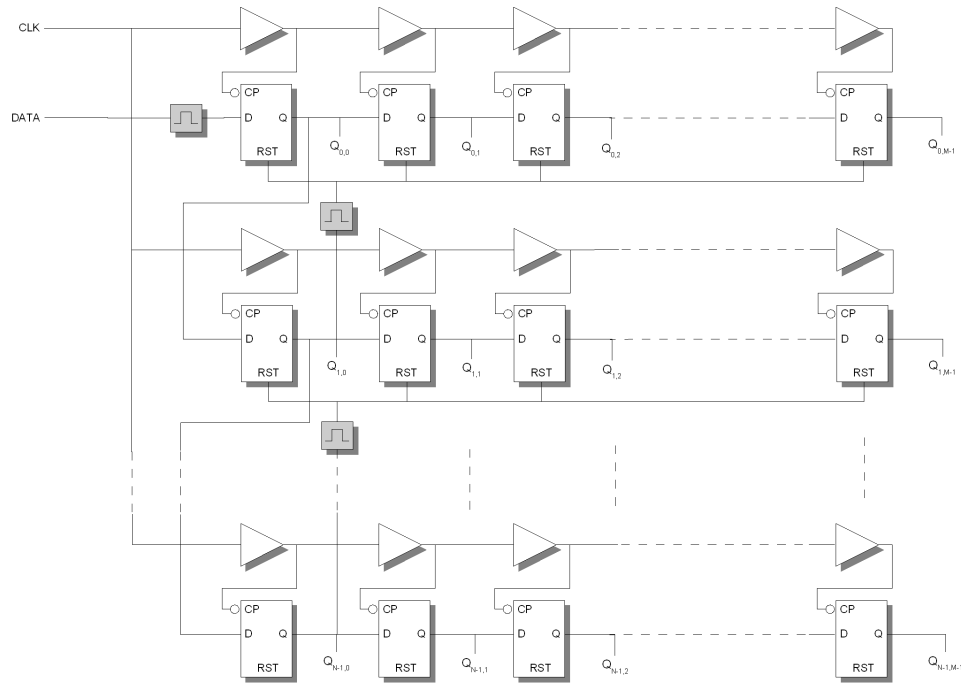


FIGURE 2.12 Interpolateur utilisant plusieurs lignes à délais de type vernier proposé par Amiri *et al.* (2009).

La valeur de ces incréments est d'environ 1ns. Le graphe de non-linéarité différentielle présente un écart se situant entre -0,3 et 0,5 divisions, tandis que pour la non-linéarité intégrale, l'erreur se situe entre -1 et 0 LSB. Dans un deuxième lieu, l'exercice est répété sur une région de la matrice correspondant à 25 lignes par 8 colonnes, ce qui est justifié par le fait qu'une analyse statistique permet d'observer que 95% des impulsions se retrouvent dans les huit premières colonnes. Les graphes ainsi obtenus présentent une forte non-linéarité. Dans le graphe de non-linéarité différentielle, l'écart sur 200 cellules oscille entre des valeurs s'étendant de -1 à 2,5 divisions, tandis que la non-linéarité intégrale s'étend sur la plage de -2,5 à 3 divisions. Ces résultats témoignent bien des problèmes d'irrégularité du routage produit par les outils de synthèse et des variations liées aux procédés de fabrication des puces lorsque des résolutions aussi fines sont recherchées.

2.8 Conclusion

Dans ce chapitre, une revue de littérature en matière d'implémentation FPGA de TDC à été présentée. Contrairement à l'implémentation sur circuits dédiés, l'implémentation sur FPGA est plus contraignante et requiert une approche numérique (par opposition à analogique). La principale limitation d'une telle implémentation, en termes de performances, provient des délais d'interconnexions plus grands et de leur irrégularité.

Un TDC est typiquement composé d'un compteur et d'un interpolateur fin. Le compteur réalise un décompte grossier du temps, tandis que l'interpolateur fin vient accroître la résolution en interpolant entre les incréments du compteur. Le défi de conception d'un TDC réside essentiellement dans l'implémentation de l'interpolateur fin. Au courant de la dernière décennie, 3 architectures classiques ont eu un certain succès dans leurs implémentations FPGA : la méthode du vernier avec deux oscillateurs, la ligne à délai simple « sur écoute », et les lignes à délais de type vernier. Les implémentations basées sur la ligne à délai sur écoute sont sans contredits les plus répandues, et sont également celles ayant eu le plus de succès. Un point faible de cette architecture, la limitation imposée par les délais minimaux du circuit, est également un point fort en ce sens que les nouvelles technologies de FPGA permettent d'obtenir directement de meilleurs résultats. L'intérêt de cette approche est également qu'elle permet une utilisation simple des structures d'interconnexions dédiées, comme la chaîne de retenue et l'arbre de distribution d'horloge, qui considèrent le problème d'irrégularité des délais d'interconnexion et des délais minimaux.

Bien que les implémentations de TDC sur circuits dédiés atteignent des résolutions sous les 10 picosecondes, les implémentations FPGA parviennent rarement à atteindre les 50 picosecondes. À ce jour, la meilleure implémentation FPGA est basée sur l'utilisation de la chaîne de retenue d'un Virtex-5, un circuit de génération très récente, et permet d'obtenir une résolution aussi faible que 17 picosecondes. Les implémentations FPGA de TDC proposées dans la littérature sont typiquement caractérisées en termes de résolution, de précision, et de non-linéarité. Un autre paramètre non-négligeable est le temps mort de l'interpolateur. Il a été discuté qu'avec la méthode de type vernier basée sur deux oscillateurs, le temps mort est variable et peut être considérable. Une distinction importante est que pour les deux autres architectures, le temps mort

obtenu est typiquement d'un cycle d'horloge, bien que la méthode d'union d'onde permette d'obtenir une meilleure résolution en augmentant le temps mort. Pour fins de comparaisons de ces architectures, on s'intéresse ainsi à la résolution de l'interpolateur en considérant un temps mort d'un cycle. Les tableaux 2.1, 2.2, et 2.3 résument les caractéristiques de performance importantes des implémentations présentées dans ce chapitre, organisées par architectures d'interpolateurs fins.

TABLEAU 2.1 Performances des interpolateurs utilisant la méthode du vernier avec deux oscillateurs.

Auteurs	FPGA	Résolution	Précision
Lin et al. (2006)	Xilinx Spartan-3/Virtex-4	27ps	N-D
Jin et al. (2007)	Xilinx	100ps	N-D

TABLEAU 2.2 Performances des interpolateurs utilisant une ligne à délais simple.

Auteurs	FPGA	Résolution	Précision	DNL	INL
Wu et al. (2003)	ACEX 1K	400ps	300ps	-0,4 à 1,7	N-D
Song et al. (2006)	ACEX 1K	112,5ps	129,4ps	-0,4 à 0,8	-0,6 à 0,7
Song et al. (2006)	Virtex-II	69,5ps	93,1ps	-1 à 1	-2 à 2
Wu et al. (2008)	Cyclone-II	60ps	40ps	-1 à 2,7	N-D
Favi et al. (2009)	Virtex-5	17ps	24ps	-1 à 3,55	-3 à 2,58

TABLEAU 2.3 Performances des interpolateurs utilisant des lignes à délais de type vernier.

Auteurs	FPGA	Résolution	Précision	DNL	INL
Kalisz et al. (1997)	QuickLogic	200ps	N-D	-0,5 à 0,5	-0,2 à 1,4
Minas et al. (2007)	Virtex-II	69,3ps	N-D	N-D	N-D
Amiri et al. (2009)	Spartan-3	75ps	300ps	-1 à 2,5	-2,5 à 3

Chapitre 3

APPROCHE PROPOSÉE

3.1 Introduction

Dans des travaux antérieurs, notre équipe de recherche s'est intéressée à la reconfiguration dynamique à grains fins des FPGA. C'est ainsi que sous la direction des professeurs Jean Pierre David et Marc Feeley, les travaux de doctorat d'Etienne Bergeron ont donné naissance à la bibliothèque *DXCL* (voir Annexe A), programmée en langage C, permettant d'atteindre cet objectif. Cette bibliothèque, supportant le Virtex-II Pro et d'autres FPGA de Xilinx, offre le niveau de reconfiguration dynamique le plus fin qui soit, offrant l'équivalent de ce qui est offert par l'outil *FPGA Editor* de Xilinx. La différence est que la bibliothèque *DXCL* permet d'accéder directement et dynamiquement à la configuration du FPGA à partir d'une application C, tandis que *FPGA Editor* offre une interface graphique pour créer ou modifier un circuit.

La bibliothèque *DXCL* fût présentée dans la littérature dans Bergeron *et al.* (2008), qui présente également une application de la reconfiguration dynamique à grain fin pour la génération de délais programmables à fines résolutions. Les résultats présentés font état d'une ligne à délai dont la résolution est de 36ps sur une portée de 947ps. Les travaux présentés dans ce chapitre se fondent sur ce travail et apportent une contribution permettant d'affiner la résolution d'une ligne à délais jusqu'à la picoseconde, et sur une portée pouvant atteindre plusieurs nanosecondes.

Sur la base de ces lignes à délais programmables à très fine résolution, nous proposons une nouvelle architecture de TDC. Les points forts de cette architecture sont qu'elle tient en compte des variations de température aussi bien que celles causées par les procédés de fabrication. Au moyen du processus de calibration proposé, l'architecture tient aussi compte du biais d'horloge induit par l'arbre de distribution. Ce

chapitre présente également une implémentation de TDC basée sur la méthode du vernier avec deux oscillateurs, tirant profit de l'ajustement fin des délais.

3.2 Une nouvelle architecture d'interpolateur basée sur l'ajustement fin des délais

Dans la revue de littérature, il a été observé que souvent les implémentations FPGA de TDC à l'état de l'art sont limitées en performance soit par les délais minimaux d'interconnexions, soit par l'irrégularité des délais de routage produits avec les outils de synthèse classiques. Afin de surpasser ces limitations, l'architecture proposée dans cette section utilise plusieurs lignes à délais programmables en parallèle (Figure 3.1). Dans une telle architecture, la résolution atteignable est limitée par la capacité d'ajuster précisément les différences de délais entre chaque ligne à délai. Il s'ensuit que la résolution est limitée par la résolution de la ligne à délai programmable, et que la portée dynamique d'une telle configuration est limitée par la portée d'ajustement de la ligne.

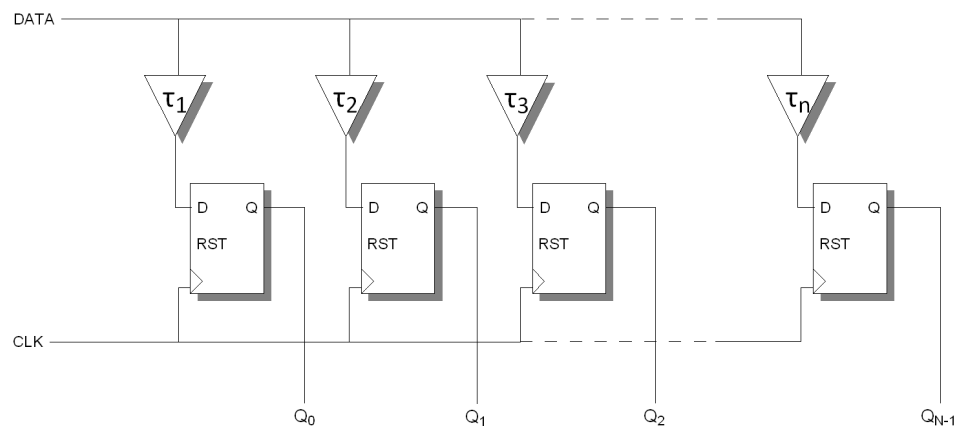


FIGURE 3.1 Architecture d'interpolateur basé sur l'utilisation de lignes à délais programmables parallèles.

Puisqu'il n'est pas réaliste de viser une portée d'ajustement correspondant à la période d'horloge du système (de l'ordre de 10ns), l'architecture de l'interpolateur fin proposé utilise plusieurs ensembles de lignes à délais parallèles. Ainsi, l'interpolateur

est composé de M colonnes de N lignes à délais parallèles (Figure 3.2). Bien que les délais entre chaque colonne soient ajustés au moyen de lignes à délais parallèles, une ligne à délais simple « sur écoute » aurait pu être utilisée puisque la résolution requise entre les colonnes est M fois moins fine que la résolution du TDC. De plus, l'irrégularité de la ligne à délai sur écoute peut être compensée par les N lignes à délais parallèles d'une colonne.

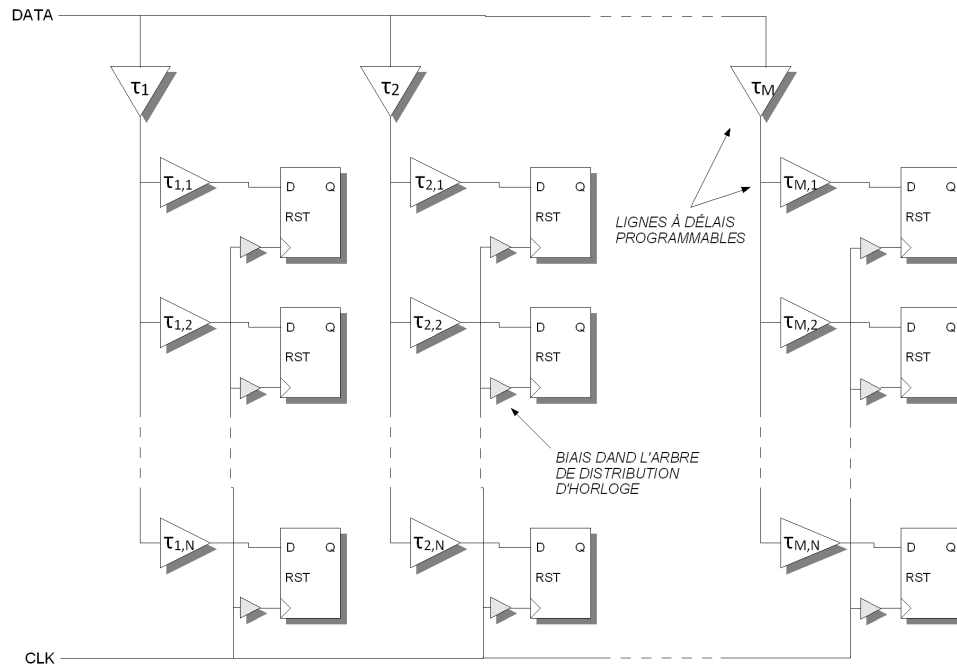


FIGURE 3.2 Architecture d'interpolateur proposée utilisant plusieurs ensembles de lignes à délais programmables parallèles.

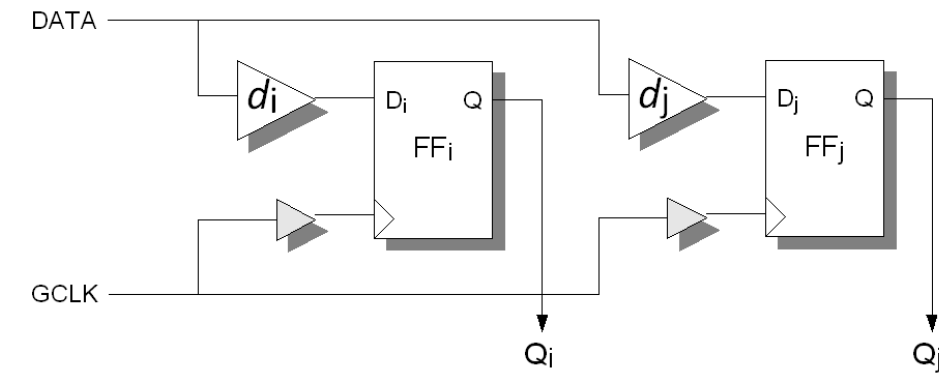
3.2.1 Calibration de l'interpolateur proposé

L'architecture proposée se fonde sur la possibilité d'ajuster précisément les délais programmables. Différents facteurs affectent les délais d'un circuit, notamment les variations dues au procédé de fabrication, ainsi que les variations de température. Afin de produire de bons résultats, le processus de calibration de l'interpolateur doit prendre en compte ces deux facteurs. De plus, puisque l'architecture a recours à l'arbre de distribution d'horloge, la calibration doit aussi tenir compte du biais de l'horloge.

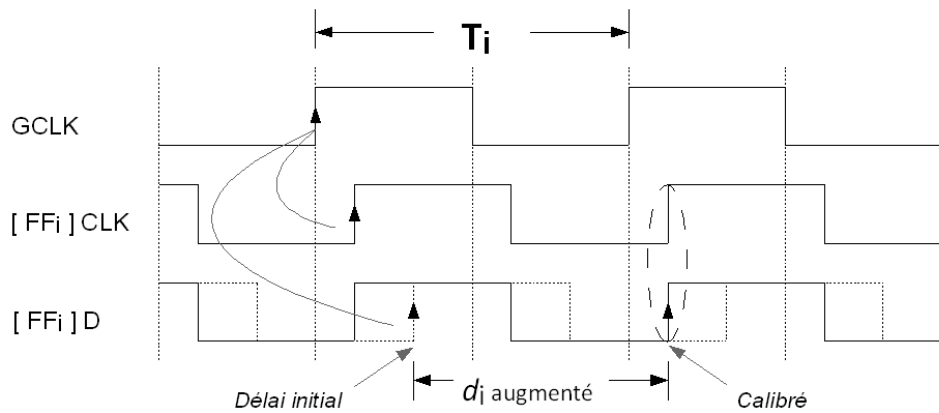
Afin d'atteindre conjointement ces trois objectifs, la méthode de calibration se base sur la détection de changement de phase, et requiert un oscillateur à fréquence variable dont la période d'oscillation peut être ajustée précisément. L'idée consiste à envoyer le signal d'horloge généré à la fois vers l'entrée D et l'entrée d'horloge d'une bascule D. De cette manière, le même signal est envoyé vers les deux entrées d'une même bascule, mais les deux copies accusent un certain décalage de phase. La procédure consiste à débiter avec une période d'horloge T_0 , et d'ajuster le premier délai de sorte à obtenir un décalage de phase de 2π radians de l'entrée D par rapport à l'entrée d'horloge du premier registre. L'idée consiste ensuite à répéter le processus pour chaque délais, en prenant soin d'incrémenter la période de l'oscillateur variable de la résolution désirée r de l'interpolateur. En s'approchant d'un déphasage de 2π par augmentation du délai, le bit échantillonné change de 1 à 0. Toutefois, étant donné le bruit thermique et la gigue du signal de l'oscillateur, près du point de transition le bit échantillonné oscille aléatoirement entre 0 et 1. Le point de transition ciblé est donc celui pour lequel le rapport du nombre d'occurrences de 1 vis-à-vis du nombre d'occurrences de 0 est unitaire, pour un certain nombre N_E d'échantillons.

La figure 3.3 illustre la calibration de deux lignes à délai parallèles, d_i et d_j , pour une résolution r . Dans un premier temps, l'oscillateur à fréquence variable est ajusté pour une période T_i , et sa sortie est dirigée à la fois aux entrées D et CLK de la bascule FF_i . Le délai programmable d_i est ensuite augmenté jusqu'à ce qu'un retard de 2π soit observé entre les deux entrées de la bascule. Il s'agit ainsi de lire le bit échantillonné par la bascule FF_i pour détecter le point de transition de 1 à 0. Ce point indique le délais d_i permettant d'obtenir un retard de $2\pi = T_i$. Dans un second temps, l'oscillateur à fréquence variable est ré-ajusté pour une période $T_j = T_i + r$. L'ajustement du délai d_j se fait comme l'ajustement de d_i , sauf que la transition 1 à 0 est recherchée sur la bascule FF_j .

Au terme de ce processus, les différences entre les moments d'échantillonnage de chacun des $N \times M$ registres consécutifs sont calibrées pour correspondre à la résolution r désirée. Puisque la différence de phase est mesurée entre l'arrivée d'un signal aux entrées D et d'horloge (CLK) des registres, l'ajustement des délais tient compte du biais de l'horloge. De plus, étant donné que l'ajustement prend en compte les délais réels, affectés par les variations dues au procédé de fabrication et la température



A) Calibration du délai d_j :



B) Calibration du délai d_j :

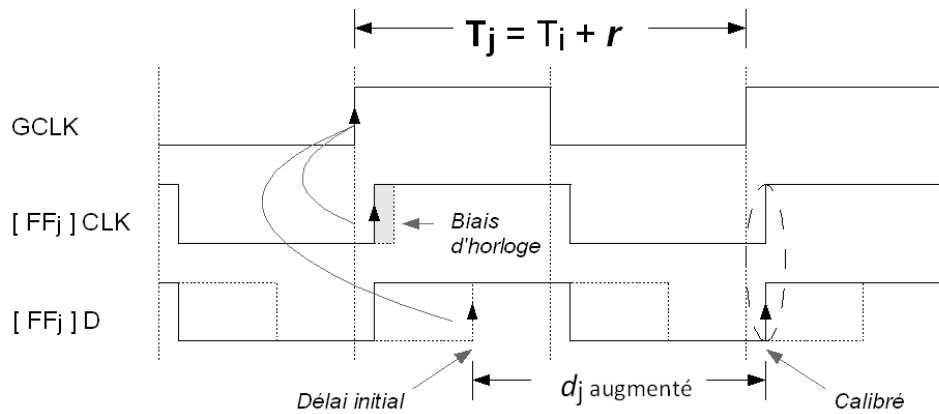


FIGURE 3.3 Processus de calibration proposé pour ajuster les délais de l'interpolateur en tenant compte des variations de procédés et de température, ainsi que du biais d'horloge.

du circuit, la méthode de calibration proposée a le potentiel de rendre possible des résolutions très fines.

3.2.2 Une ligne à délai ajustable avec une résolution de l'ordre de la picoseconde

Afin de produire des délais programmables de l'ordre de la picoseconde, il s'agit de prendre avantage du nombre important de routes possibles entre deux points d'un circuit offert par le système d'interconnexions programmables. Au moyen de la reconfiguration dynamique, il est possible de modifier le délai d'une route en modifiant le chemin emprunté par celle-ci. L'approche utilisée pour mesurer les différences de délais entre les différentes routes possibles consiste à boucler la ligne à délai en connectant sa sortie à son entrée, afin de créer un oscillateur en anneau. Pour y parvenir on insère des tampons (*buffers*) dans la ligne à délai, et on reconfigure un de ceux-ci en inverseur pour activer l'oscillateur. On obtient alors que la différence de délais peut-être observée à même des changements de fréquence de l'oscillateur en anneau. Puisque la demi-période d'oscillation de l'oscillateur correspond au délai de la boucle, celle-ci varie directement selon la différence de délais des routes reconfigurables. Il s'agit donc de produire un ensemble de routes, tel que pour toute route (sauf la première et la dernière) on trouve une route qui excède son délai de résolution souhaitée, et une autre dont le délai est moindre de cette résolution.

La figure 3.4 illustre cette approche, proposée dans Bergeron *et al.* (2008). On y présente l'implémentation d'une ligne à délai programmable dans un bloc logique configurable (CLB) d'un FPGA Virtex-II Pro, et sa reconfiguration en oscillateur en anneau. Dans un FPGA Virtex-II Pro, chaque CLB contient une matrice de routage ainsi que 4 tranches (*slices*) contenant chacune 2 tables de correspondance (LUT). La route reconfigurable permettant de modifier le délai de la ligne programmable est implémentée dans la matrice de routage d'un CLB, reliant deux tables de correspondance configurées en tampons. La matrice de routage est composée de plusieurs points d'interconnexions programmables (PIP), qui permettent notamment à une route d'atteindre d'autres CLB. Une particularité intéressante de ces PIP est qu'ils permettent également de faire rebondir la route vers d'autres PIP de la matrice. Puisque la majorité des PIP de sortie (cercles foncés sur la figure) permettent un rebond vers différents

sous-ensembles des autres PIP de sortie de la matrice, le nombre de routes possibles permettant de relier les deux LUT est important. Afin de mesurer les différences de délais entre chaque routes, la ligne à délai programmable est reconfigurée dynamiquement en oscillateur en anneau. Pour réaliser cette transformation, il suffit de reconfigurer le PIP de sortie reliant l'entrée de la ligne à une LUT, de sorte à relier l'entrée de cette LUT à la sortie de l'autre, ce qui crée une boucle fermée. La reconfiguration est complétée en modifiant la configuration d'une LUT pour implémenter un inverseur plutôt qu'un tampon.

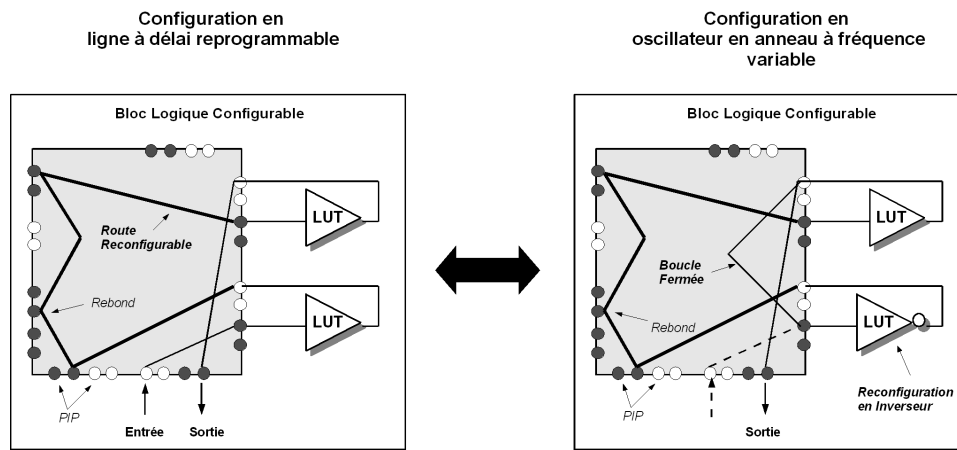


FIGURE 3.4 Reconfiguration dynamique de la ligne à délai en oscillateur en anneau, de manière à pouvoir mesurer les demi-périodes (proportionnelles aux délais) associées à chaque route.

Afin de mesurer les différences de délais entre les routes, la fréquence (moyenne) à la sortie de l'oscillateur est mesurée. Or, la demi-période $T_{osc}/2$ d'oscillation d'un oscillateur en anneau est donnée par l'équation 3.1, où d_{boucle} correspond au délai de la boucle, et $d_{inverseur}$ au délai de l'inverseur. On peut également décomposer d_{boucle} en une composante statique $d_{statique}$ et une autre dynamique $d_{dynamique}$. La composante $d_{dynamique}$ correspond au délai la route reconfigurable, et puisque le délai de la composante statique est quasi-constant, les différences de délais entre les routes peuvent être obtenue à partir des mesures de demi-périodes de l'oscillateur. Afin de mesurer ces demi-périodes, la sortie de l'oscillateur est reliée à un compteur binaire, et la sortie d'un oscillateur de référence est reliée à un deuxième compteur. Ainsi, soit CNT_{osc} et CNT_{ref} les valeurs des compteurs binaires reliés à l'oscillateur à fréquence variable

et l'oscillateur de référence, respectivement, et soit T_{ref} la période de l'oscillateur de référence, alors la demi-période $T_{osc}/2$ est donnée par l'équation 3.2.

$$T_{osc}/2 = d_{boucle} + d_{inverseur} \quad (3.1)$$

$$T_{osc}/2 = (T_{ref}/2) \times (CNT_{ref}/CNT_{osc}) \quad (3.2)$$

Dans le travail présenté dans Bergeron *et al.* (2008) la résolution de 36ps est obtenue au moyen d'une liste de 546 routes entre la sortie et l'entrée de deux tables de correspondance (LUT) configurées en tampons. L'implémentation de la ligne à délais sur un FPGA Virtex-II Pro occupe un bloc logique combinatoire (CLB), et chaque route possible est confinée à la matrice de routage de ce bloc. Un tel nombre de routes est atteint en reconfigurant les points d'interconnexion programmables (PIP) de la matrice de routage. Les PIP de sortie (cercles foncés) implémentent la fonction d'un multiplexeur, et permettent également de faire rebondir la route à l'intérieur de la matrice vers un autre PIP de sortie. On définit la *profondeur* d'une route par le nombre de PIP de sortie qu'elle utilise. Dans cette première approche, la profondeur maximale obtenue est de 9. Dans cette section, nous proposons une méthode pour diminuer considérablement la résolution atteignable, jusqu'à la picoseconde, tout en augmentant la portée d'ajustement de la ligne à délais.

Implémentation de la ligne à délais configurable proposée

L'implémentation proposée de la ligne à délais programmable permettant d'obtenir une résolution d'une picoseconde sur une plage de quelques nanosecondes est illustrée à la figure 3.5. Contrairement à l'approche précédente (Figure 3.4), cette implémentation permet à la route configurable de sortir et de ré-entrer dans la matrice de routage, via d'autres LUT configurées en tampons. Dans un Virtex-II Pro, un CLB contient 8 tables de correspondance, et donc il est possible d'utiliser jusqu'à 6 LUT supplémentaires. En procédant de la sorte, le bénéfice est double. En premier lieu, le délai typique d'une LUT est de l'ordre de 250ps pour le FPGA ciblé, ainsi en insérant ces dernières dans la route reconfigurable, la portée est augmentée. Pour les 8 LUT présentes dans un CLB, on obtient environ 2ns de portée, en excluant les délais d'interconnexions. En deuxième lieu, en utilisant des LUT additionnelles, on

permet d'utiliser un nombre plus important de PIP dans la route reconfigurable, ce qui augmente considérablement le nombre de routes possibles. La route reconfigurable illustrée à la figure 3.5 utilise ainsi 1 des 6 LUT disponibles.

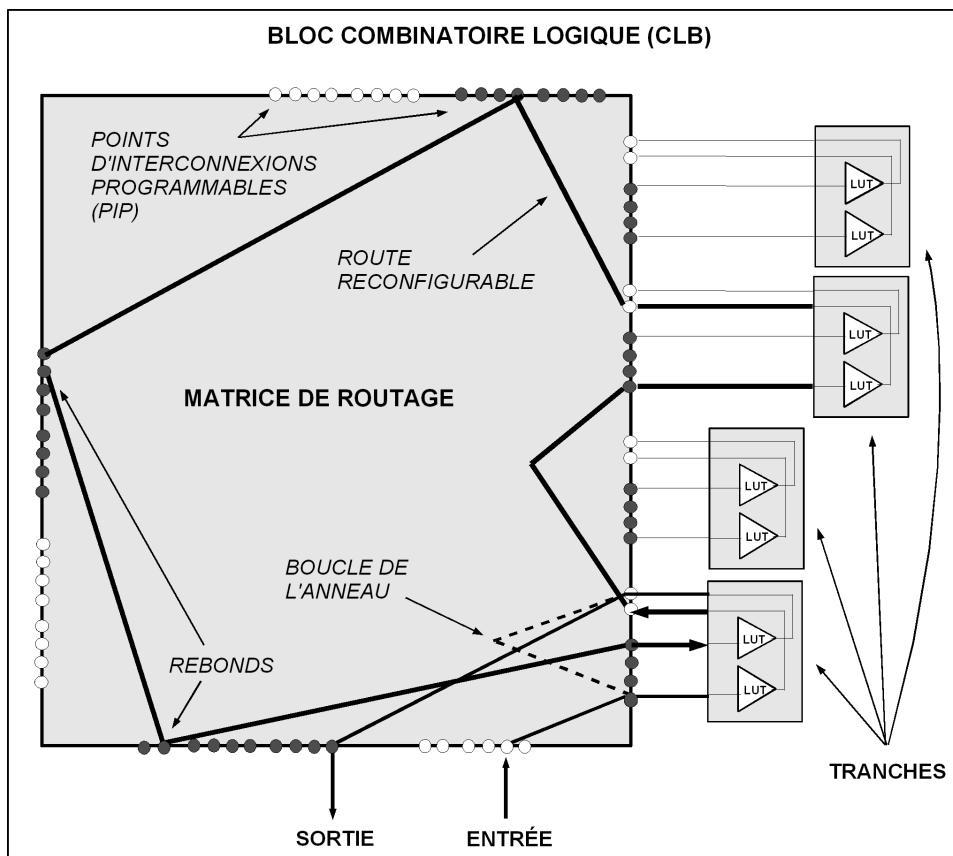


FIGURE 3.5 Implémentation proposée de la ligne à délai programmable.

Génération des routes reconfigurables

La génération des routes possible requiert la connaissance des interconnexions entre chacun des points d'interconnexions programmables. Afin d'obtenir cette structure il est possible d'utiliser le programme XDL fournit par Xilinx dans la suite *Integrated Software Environment (ISE)*, mais notre approche utilise directement la bibliothèque DXCL (voir Annexe A). La connaissance des interconnexions existantes entre les PIP d'un CLB permet ainsi de chercher les différents ensembles de PIP (routes) permettant de relier la sortie et l'entrée de deux LUT différentes.

En utilisant la nouvelle implémentation de ligne à délai programmable proposée, le nombre de routes reconfigurables possibles augmente considérablement et semble n'avoir pratiquement plus de limites. En effet, il devient alors possible de générer des millions de routes reconfigurables confinées à l'intérieur du CLB, et la profondeur maximale d'une route excède alors 30. Malheureusement, une liste contenant un nombre de routes aussi important n'est pas souhaitable. En effet, en assumant une portée de 4ns pour la ligne à délais, afin d'obtenir une résolution de 1ps il serait théoriquement suffisant de disposer d'une liste de 4000 routes. En pratique, afin de garantir la résolution sous l'influence de variations de températures, une certaine redondance est tout de même désirée. En effet, toutes les routes ne sont pas affectées (en termes de délai) de la même manière par les variations de températures. Ainsi, une liste de 40000 routes serait plus certainement suffisante, pour autant que les délais soient uniformément distribués ! Une approche plus directe serait de mesurer la demi-période associée à chaque route produite par l'algorithme de recherche, et d'ajouter les routes requises à la liste. Toutefois, étant donnée la redondance intrinsèquement élevée existante entre ces routes, un temps énorme serait perdu à mesurer des routes pour lesquelles le délai associé est déjà obtenu avec une redondance suffisante. Afin de palier à ce problème, une autre approche est de mise.

L'algorithme utilisé pour la génération de routes exécute une recherche récursive en profondeur, pour des profondeurs allant de 1 à 31. Pour chaque profondeur, les routes peuvent être classifiées par le nombre de LUT qu'elles utilisent. Cependant, pour des profondeurs plus élevées, le nombre de routes possibles peut devenir important, et une heuristique est désirable afin de déterminer un moment à partir duquel on peut sauter directement à la profondeur suivante. Il serait possible d'attribuer simplement un nombre de routes maximal à produire pour chaque profondeur, néanmoins cette approche pose problème. En effet, pour chaque profondeur à partir d'un certain point, deux classes de routes (nombre différents de LUT) ou plus peuvent exister. Or en prenant les N premières routes produites, il est possible de ne pas générer un nombre appréciable de routes de l'une ou plusieurs des classes. L'heuristique que nous proposons pour palier à ce problème est exposée dans l'Algorithme 3.1. Un nombre de route maximal M pour une classe d'une profondeur donnée est fixé, de sorte que lorsqu'aucune des K dernières routes générées n'a été ajouté dans une classe non-pleine, l'algorithme saute à la profondeur suivante. En procédant de la sorte, et en

ajustant adéquatement les paramètres M et K , on parvient à contrôler la taille de la liste tout en obtenant une distribution plus uniforme.

ALGORITHME 3.1 GÉNÉRATION DE ROUTES RECONFIGURABLES

```

1: for profondeur  $p = 1$  jusqu'à 31 do
2:    $cnt = 0$ ;
3:   while autresRoutesPossibles( $p$ ) do
4:      $route = genererRouteSuivante(p)$ ;
5:     if ( classeDeRouteRemplie( $route$ ) ) then
6:        $cnt ++$ ;
7:       if (  $cnt > K$  ) then
8:         break;
9:       end if
10:    else
11:      ajouterRouteDansListe( $route$ );
12:       $cnt = 0$ ;
13:    end if
14:  end while
15: end for

```

3.2.3 Un oscillateur à fréquence variable précis à l'ordre de la picoseconde

Dans la sous-section précédente, le besoin d'un oscillateur précis à fréquence variable a été souligné pour mettre en œuvre le processus de calibration proposé. En fait, la précision de cet oscillateur vient effectivement limiter la résolution atteignable au moyen de ce processus. La réalisation d'un tel oscillateur se fonde sur l'utilisation de la ligne à délais programmable proposée. Le sujet de la transition d'une ligne à délais vers un oscillateur à anneau a déjà été discuté à la section 3.2.2. Néanmoins, comme la portée d'ajustement de la période de l'oscillateur doit minimalement égaliser et préférablement excéder la période de l'horloge du système, la ligne à délai proposée est insuffisante. Afin d'accroître cette portée d'ajustement, la méthode choisie est simplement d'insérer des LUTs supplémentaire provenant de CLBs voisins dans la boucle de l'oscillateur. Puisque le délai d'une LUT correspond à environ 250ps, cette approche permet facilement d'étendre la portée à plusieurs dizaines de nanosecondes.

En considérant la liste de routes reconfigurables, triée par ordre croissant de délais, il est peu probable qu'en cours d'opération, sous l'influence des variations de température, la liste conserve sa monotonie. En effet, chaque route n'est pas affectée de la même façon par ces variations, et le besoin d'une méthode de calibration se fait sentir. Ainsi en premier lieu, un ajustement grossier de la période est fait en modifiant le nombre de tampons dans la chaîne. Ensuite, afin d'obtenir la précision désirée, une recherche par approche dichotomique vers la période cible est réalisée. Cette recherche itérative débute en mesurant les fréquences actuelles pour des routes à faibles et grands délais de la liste de référence, et en réalisant une interpolation linéaire pour déterminer l'indice de la route cible. La fréquence associée à la route de l'indice obtenu est ensuite mesurée et le processus itératif continue jusqu'à ce que la fréquence désirée soit obtenue, ou bien que les indices des routes à faibles et grands délais soient suffisamment près l'un de l'autre pour démarrer une recherche exhaustive entre ceux-ci. Pour autant que la période ainsi calibrée soit suffisamment stable pour une brève période de temps, un tel oscillateur rend possible la calibration du TDC proposé.

3.2.4 Avantages de la reconfiguration dynamique

Bien que la méthodologie de conception de la ligne à délai programmable proposée présume l'utilisation de la reconfiguration dynamique d'un FPGA, cette dernière n'est pas strictement nécessaire. Cependant, l'utilisation de la reconfiguration dynamique offre certains avantages qui seront discutés dans cette section.

Pour réaliser l'implémentation de la ligne à délai programmable proposée au moyen d'une méthodologie *statique*, il serait d'abord requis de produire un fichier de programmation (*bitstream*) pour chaque configuration possible de la ligne à délais (une configuration par route reconfigurable). En considérant un ensemble de 40000 routes, il faudrait autant de fichiers de programmations, et à raison d'environ 1Mo par fichier (pour le Virtex-II Pro ciblé) cela peut représenter jusqu'à 40 Go de mémoire. Toutefois, étant donné que ces fichiers de programmations seraient pratiquement identiques, il serait possible de compresser considérablement l'information. Le problème réel qui se pose avec une telle approche est que la reconfiguration complète du FPGA sera nécessairement beaucoup plus longue à réaliser qu'une reconfiguration partielle. Ainsi, plutôt que de reconfigurer partiellement une colonne de CLB du FPGA, la

reconfiguration complète de 46 colonnes de CLB serait requise pour modifier le délai d'une ligne.

Afin d'illustrer un autre problème qui se pose lorsque l'utilisation de la reconfiguration partielle (dynamique) n'est pas considérée, il suffit de prendre l'exemple d'ajustement d'une ligne à délai au moyen d'un oscillateur à fréquence variable tel que proposé à la section 3.2.1. Dans ce cas, soit N_1 le nombre de configurations possibles de la ligne à délai programmable, et N_2 le nombre de configurations requises de l'oscillateur à fréquence variable, le nombre de configurations possibles pour un circuit composé d'une ligne à délai et d'un oscillateur est donné par le produit $N_1 N_2$. Face à cette explosion de configurations possibles, il serait nécessaire de générer dynamiquement (à partir d'un ordinateur) les fichiers de configurations du FPGA. Une telle approche est possible en considérant l'utilisation des commandes *XDL* et *BITGEN* de Xilinx, mais le temps de génération serait alors de l'ordre de 10 secondes et plus par configuration ! Ainsi en considérant que $N_1 N_2$ est de l'ordre de 40000^2 , il faudrait considérer un temps de l'ordre 185000 jours pour générer l'ensemble des configurations possibles. En utilisant une bibliothèque de reconfiguration dynamique comme la bibliothèque DXCL, l'avantage est que la génération d'une configuration à partir d'une autre ne requiert que la modification d'un sous-ensemble des bits de configuration, ce qui représente des temps nettement moindres, de l'ordre de la microseconde. Ainsi, les avantages liés à l'utilisation de la reconfiguration dynamique (partielle) des FPGA, conjointement avec la bibliothèque DXCL, se résument par :

- La capacité de reconfigurer plus rapidement le FPGA (reconfiguration partielle).
- La capacité de générer plus rapidement l'ensembles des configurations requises (bibliothèque DXCL).

3.2.5 Implémentation sur un FPGA Virtex-II Pro

L'architecture de TDC proposée à été implémentée dans un FPGA Virtex-II Pro de Xilinx (XC2VP30), supportant la reconfiguration dynamique et supporté par la librairie DXCL. L'interpolateur fin est implémenté par une matrice de 20 colonnes par 11 lignes à délais parallèles. Avec une fréquence d'opération cible de 100MHz, une telle implémentation permet d'obtenir une résolution de 50ps. La 11^e colonne est utilisée pour fins de validation, en permettant d'observer la superposition d'échantillons consécutifs. Dans cette implémentation, chaque échantillon comporte 220-bits, et

ces derniers sont transmis vers un périphérique spécialisé. Ce périphérique contient un compteur binaire 36-bits, ainsi qu'une FIFO (512x256bits) pour mémoriser les échantillons. Ainsi, lorsqu'une transition est détectée dans un échantillon ou entre deux échantillons consécutifs, les échantillons de 220-bits sont enregistrés avec la valeur du compteur binaire, sous forme d'échantillons composites de 256-bits. Pour des raisons expérimentales, le décodage des temps de transition des échantillons est réalisé par une application logicielle embarquée.

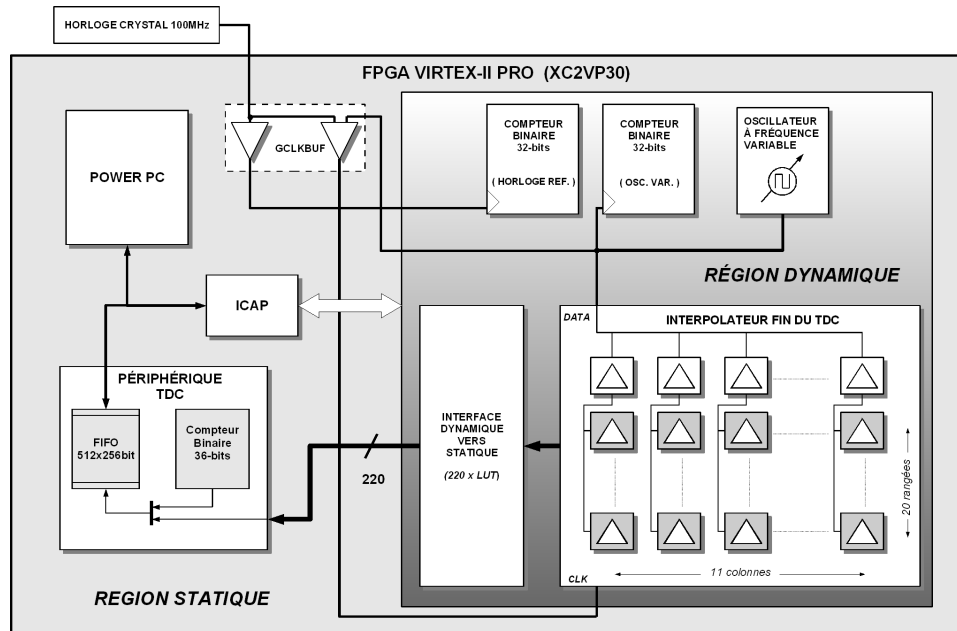


FIGURE 3.6 Implémentation du TDC proposé dans un Virtex-II Pro. Le design est partitionné en une région dynamique et une région statique.

Considérations liées à la reconfiguration dynamique

La reconfiguration dynamique est offerte sur certains FPGA depuis plusieurs années. Néanmoins, cette dernière est rarement utilisée par les concepteurs et les outils disponibles sont contraignants en termes de possibilités offertes. Puisque notre approche requiert une reconfiguration du FPGA à son plus fin niveau de granularité, notre approche utilise la librairie C DXCL plutôt que le processus (*flow*) de conception modulaire proposé par Xilinx (voir Xilinx, 2004).

Lors de la conception d'un design utilisant la reconfiguration dynamique, une étape importante est de partitionner le design en une région statique et une région dynamique. Dans l'implémentation proposée (Figure 3.6), la région statique comprend un système embarqué basé sur le processeur PowerPC présent dans le Virtex-II Pro, avec des périphériques standards comme un contrôleur mémoire (DDR), un périphérique de lien sériel UART, et un périphérique d'accès interne à la configuration du FPGA (ICAP), le tout interconnecté au moyen des bus OPB et PLB. En plus de ces périphériques standards, le design statique contient également le périphérique personnalisé d'acquisition du TDC. Ce périphérique sert également d'interface statique à la sortie de la zone dynamique, contenant l'interpolateur fin, l'oscillateur à fréquence variable, et les compteurs binaires de calibration. Il est à noter que l'interface dynamique est simplement composée de LUT configurées en tampons.

Normalement, l'interface entre les régions statiques et dynamiques dans les FPGA Virtex de Xilinx est réalisée au moyen de macros de bus (*bus macros*). Dans un Virtex-II, ces macros de bus sont implémentées avec des tampons à 3 états (*tri-state buffers*) et utilisent les longues lignes horizontales. Or, puisque chaque rangée de ce FPGA contient 4 de ces lignes horizontales, le nombre de signaux composant l'interface est limité. Dans le XC2VP30 utilisé, les 80 rangées de CLB disponibles limitent cette interface à 320 signaux. Bien que l'implémentation proposée requiert 220 signaux pour obtenir une résolution de 50ps, cette approche ne permet pas d'atteindre des résolutions de 25ps ou moins, qui demanderaient des interfaces composées de 400 signaux ou plus. De plus, étant donné que le nombre de signaux permis par une telle interface dépend directement du nombre de rangées du FPGA, l'implémentation proposée peut poser problème lorsque des FPGA de plus petites tailles sont considérés. L'approche originale utilisée afin de palier ce problème, basée sur l'utilisation de LUTs, fait l'objet de la prochaine section.

Génération du design statique

La première étape dans la réalisation du convertisseur numérique de temps consiste à générer le fichier de programmation qui sera utilisé lors de la mise sous tension du FPGA. Le processus de génération du fichier de programmation requis est illustré à la Figure 3.7. Le logiciel EDK 9.1 de Xilinx est utilisé pour produire le système embarqué basé sur le processeur PowerPC. Afin de réserver une région dynamique

suffisamment grande (sans ressources utilisées), des contraintes de placement sont utilisées. Dans le fichier de programmation produit par EDK, les entrées du périphérique personnalisé d'acquisition du TDC sont reliées à des blocs d'entrées (IOB). Il faut donc déconnecter ces signaux et les reconnecter à l'interface dynamique. En effet, il convient de mentionner que bien que la librairie DXCL possède un routeur, ce dernier ne peut être utilisé dans une région statique sans risques d'altérer les connexions préexistantes. Les signaux passant par la région statique doivent ainsi être routés de manière statique. Pour ce faire, notre approche utilise le routeur disponible dans le logiciel FPGA Editor, et l'opération de routage est automatisée au moyen de scripts.

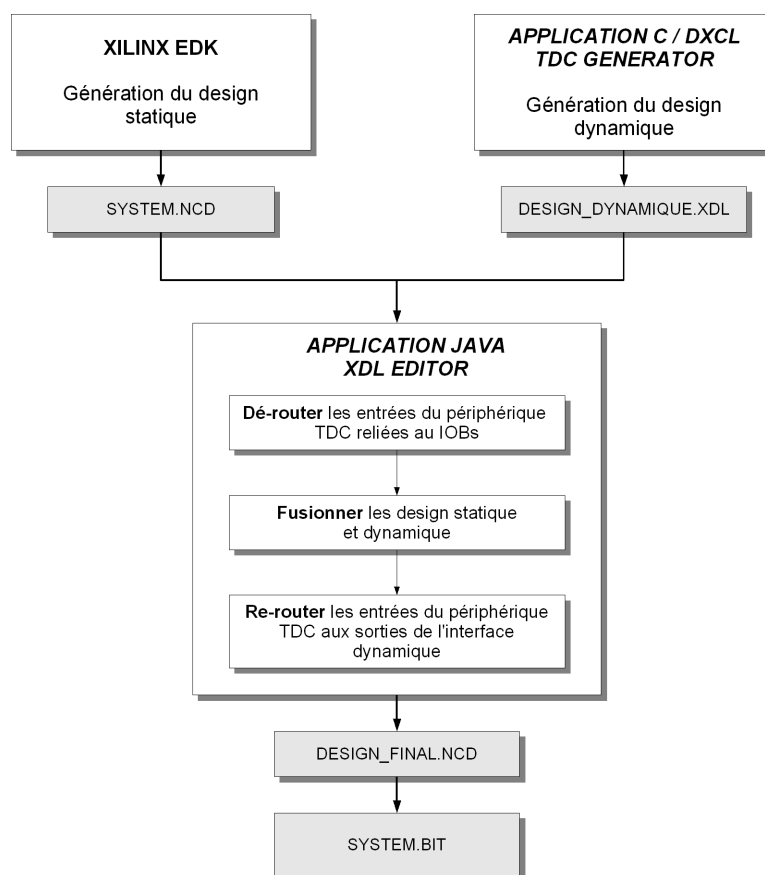


FIGURE 3.7 Processus de génération du design statique.

Afin de relier l'interface statique et dynamique ensemble, au moyen de FPGA Editor, il faut au préalable que l'interface dynamique composée de LUT soit présente dans le design. Pour y parvenir, la capacité de la librairie DXCL d'exporter en format

XDL (*Xilinx Description Language*) un circuit généré dynamiquement est exploitée. Le format XDL est une représentation textuelle d'un circuit implémenté sur un FPGA de Xilinx. Il existe dans l'outil ISE (*Integrated Software Environment*) de Xilinx des fonctions de conversion permettant de passer du format XDL vers NCD et inversement. Le format NCD (*Native Circuit Description*), est utilisé pour représenter un circuit sur FPGA de Xilinx, et les fichiers de ce type peuvent être convertis en fichier de programmation FPGA de format BIT (*Bitstream*). Dans l'implémentation proposée, l'ensemble des connexions du circuit généré dynamiquement, à l'exception des routes reconfigurables, sont statiques. (Il en va de même pour la plupart des points de configuration des éléments logiques.) En effet, cet ensemble de connexions est généré au moyen de la bibliothèque DXCL, mais n'est pas modifié en cours d'opération. Ainsi, le design dynamique réalisant l'interpolateur fin est exporté sous format XDL.

À partir du design statique sous format NCD et du design généré dynamiquement en format XDL, la suite du processus est réalisée par une application logicielle Java permettant d'éditer un circuit représenté en XDL. Ainsi cette application déconnecte en premier lieu les connexions reliant le périphérique du TDC aux blocs d'entrée (IOB). Une fois cette opération complétée, les deux design statique et dynamique sont fusionnés en un seul circuit, bien que disjoints. Le processus est essentiellement complété en connectant les interfaces statiques et dynamiques au moyen d'un script FPGA Editor généré par l'application Java.

Application logicielle de génération du TDC

Bien que l'implémentation du convertisseur numérique de temps proposé soit un circuit matériel, sa mise-en-œuvre requiert plusieurs étapes qui sont réalisées par une application logicielle écrite en langage C que nous appellerons *TdcGenerator*. Tel que mentionné à la section précédente, une des premières étapes de conception du TDC consiste à exécuter *TdcGenerator* afin de produire une représentation XDL du circuit qui sera implémenté dans la région dynamique. Certaines étapes, réalisées par l'application, telle celle qui vient d'être mentionnée, doivent être lancées depuis un ordinateur personnel, tandis que d'autres sont exécutées sur le processeur PowerPC du système embarqué. La Figure 3.8 illustre les différentes étapes réalisées par l'application ainsi que les nœuds d'exécution qui leurs sont associées.

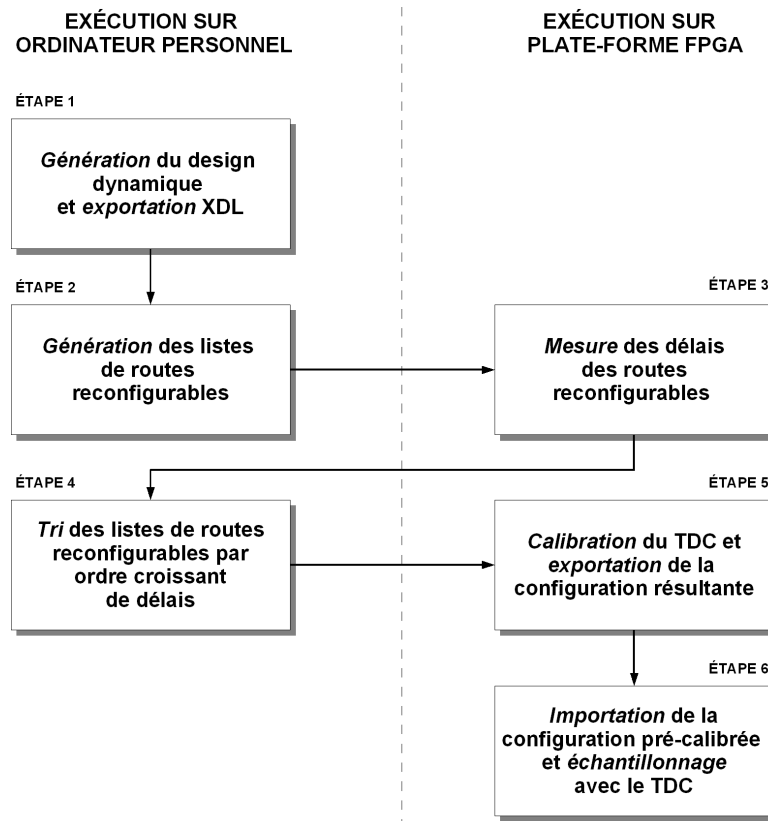


FIGURE 3.8 Processus de génération et calibration du design dynamique du TDC proposé.

Une fois le design statique programmé dans le FPGA, l'application *TdcGenerator* est utilisée afin de réaliser les différentes étapes requises pour générer, calibrer, et opérer le TDC. Il est à noter que chacune des tâches présentées est réalisée lors d'une exécution différente. La deuxième tâche qu'accomplit l'application est la génération des listes de routes reconfigurables de référence pour les lignes à délais, ainsi que l'oscillateur à fréquence variable. Ensuite, la première exécution sur le Virtex-II Pro consiste à lancer la mesure des délais associés aux routes reconfigurables. Une fois les délais mesurés, l'exécution subséquente consiste simplement à procéder au tri des listes de routes par ordre croissant de délais. C'est après avoir réalisé toutes ces étapes que la calibration du TDC peut avoir lieu, en utilisant l'oscillateur à fréquence variable. Une fois la calibration terminée, la configuration résultante du TDC est exportée de sorte à permettre son importation lors d'une prochaine exécution, sans qu'il soit requis de re-calibrer le circuit à chaque fois. À ce stade, il devient possible

d'utiliser l'application *TdcGenerator* afin de réaliser la mesure d'intervalles de temps, ou bien de caractériser le TDC. Ces opérations peuvent être réalisés directement après la calibration du TDC, ou alors ultérieurement en important une configuration pré-calibrée.

3.3 Une nouvelle implémentation de la méthode du vernier avec deux oscillateurs

Nous avons vu dans la section précédente comment l'ajustement fin des délais permet de proposer et d'implémenter une architecture de TDC novatrice. Dans cette section, la méthode d'ajustement des délais est appliquée à une architecture préexistante d'interpolateur fin. Étant donnée la précision atteignable avec l'oscillateur à fréquence variable, il semble pertinent d'explorer l'implémentation de la méthode du vernier avec deux oscillateurs démarrables.

3.3.1 Conception de l'oscillateur démarrable

Bien que l'implémentation d'un oscillateur à fréquence variable ait été discutée, la méthode du vernier ciblée requiert que les oscillateurs soient démarrables. La modification à apporter est minimale (Figure 3.9), il s'agit simplement d'insérer un multiplexeur dans l'anneau afin de pouvoir l'ouvrir et le fermer. L'oscillateur résultant dispose donc d'une entrée afin de contrôler les oscillations au moyen de ce multiplexeur, et d'une sortie pour le signal périodique à produire. Dans l'implémentation proposée, seules les tables de correspondances (LUT) d'un seul bloc combinatoire logique (CLB) sont utilisées. Puisque c'est la différence entre les périodes qui prime, il n'est pas nécessaire dans cette application d'obtenir une portée d'ajustement de la période s'étendant sur une grande plage.

3.3.2 Calibration de l'interpolateur

La calibration de l'interpolateur fin basé sur la méthode du vernier se résume essentiellement à la calibration des oscillateurs lent et rapide. Puisqu'aucune LUT externe au CLB n'est utilisée, la calibration de l'oscillateur démarre directement avec la recherche par approche dichotomique. Lors de la calibration de l'oscillateur, le

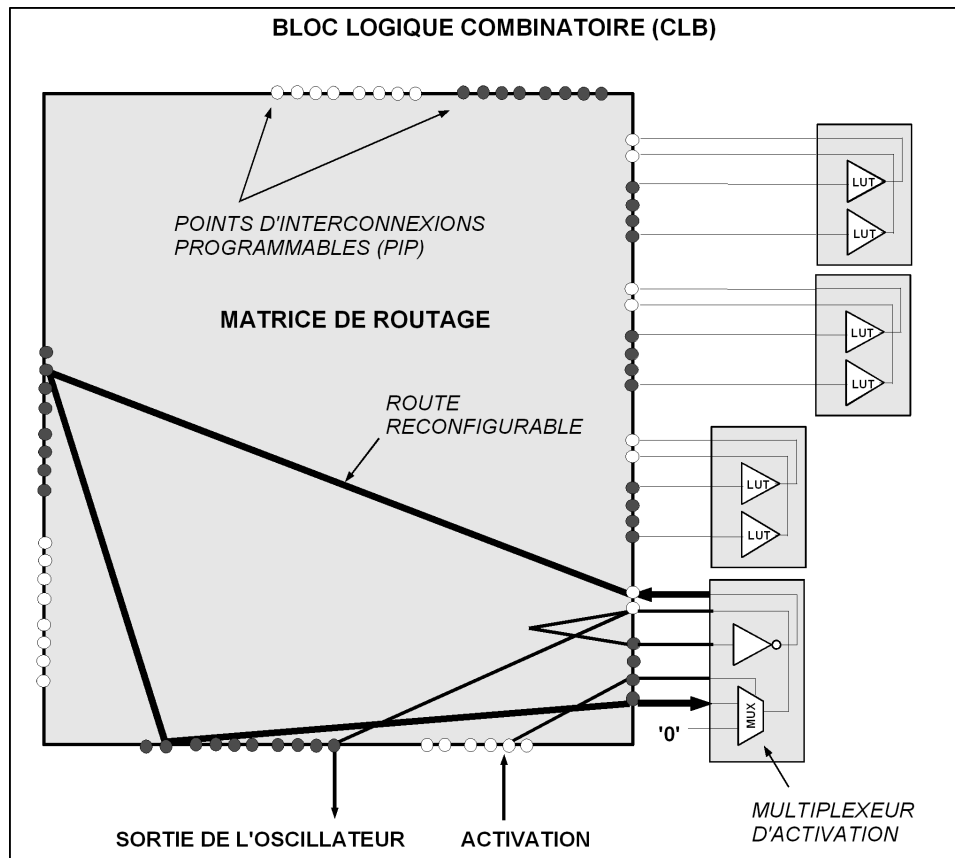


FIGURE 3.9 Implémentation proposée d'un oscillateur démarrable.

Le multiplexeur est reconfiguré dynamiquement en tampon et l'oscillateur en anneau oscille librement. Puisque la résolution de l'interpolateur est donnée par la différence de périodes des deux oscillateurs, il suffit de s'assurer que la différence est stable pour une brève période de temps. Un aspect intéressant de notre approche basée sur la reconfiguration dynamique est qu'il devient possible de réajuster la résolution de l'interpolateur au fil du temps.

3.3.3 Implémentation sur un FPGA Virtex-II Pro

L'implémentation de l'interpolateur a également été réalisée sur un Virtex-II Pro XC2VP30. Le système statique est composé d'une plate-forme PowerPC produite avec l'outil EDK, tandis que l'interpolateur est généré dynamiquement. Dans cette implémentation, il n'y a pas de connections entre les portions statiques et dynamiques

(à l'exception de l'horloge système). Les compteurs binaires de l'interpolateur sont lus au moyen de la reconfiguration dynamique (par lecture de la mémoire de configuration du FPGA). L'interpolateur utilise deux compteurs binaires de 16-bits.

Il convient de mentionner que l'implémentation place également chacun des oscillateurs à fréquence variable à une bonne distance l'un de l'autre. Autrement, nos expériences ont montré que les oscillateurs ont tendance à se synchroniser ensemble lorsque la différence de période ciblée est inférieure à 15ps. Ce phénomène, loin d'être banal, est expliqué par le couplage existant entre les deux oscillateurs (voir Strogatz, 2001). Partant de l'hypothèse que ce couplage provient de l'alimentation, en séparant spatialement de 10 CLB les deux oscillateurs, une telle synchronisation n'a plus été observée.

3.4 Conclusion

Dans ce chapitre, une méthode basée sur l'ajustement fin des délais est proposée afin de résoudre le problème de l'implémentation sur FPGA de convertisseurs numériques de temps. Cette méthodologie, basée sur la reconfiguration dynamique à grain fin du FPGA, permet d'ajuster les délais d'un circuit en modifiant la configuration des points d'interconnexions programmables (PIP), de sorte à modifier le chemin emprunté par une route. Dans l'approche proposée, la route reconfigurable d'une ligne à délais est obtenue à partir d'une liste d'environ 40000 routes différentes.

Sous la base de lignes à délais programmables ajustables à de fines résolutions, une nouvelle architecture de TDC est proposée. Cette architecture se caractérise par son utilisation de lignes à délais en parallèles. Afin d'ajuster chacun des délais de l'interpolateur, un processus de calibration est présenté, prenant en compte les variations dues au procédé de fabrication et les variations de températures, ainsi que le biais causé par l'arbre de distribution d'horloge. Ce processus de calibration a recours à un oscillateur en anneau à fréquence variable, dont l'ajustement de la période peut être réalisé avec une grande précision. C'est ainsi que nous proposons également une méthode d'ajustement précis de la période, en reconfigurant dynamiquement un segment de la route d'un oscillateur en anneau. Au-delà des qualités déjà mentionnées, un point fort de l'architecture proposée est qu'elle permet d'obtenir des résolutions

plus faibles que les délais minimaux du circuit programmable.

Notre approche basée sur le contrôle fin et précis des délais a également été appliquée à la conception d'un interpolateur basé sur la méthode du vernier avec deux oscillateurs. Ainsi, l'implémentation d'un oscillateur à fréquence variable démarrable est proposée. Puisque la période de cet oscillateur peut être ajustée précisément, la résolution peut être modifiée ou re-calibrée en cours d'opération. De plus, cette méthode permet potentiellement d'atteindre de très fines résolutions. Néanmoins, en appliquant notre méthodologie, la résolution sera vraisemblablement limitée par la stabilité des oscillateurs plutôt que leurs précisions d'ajustement. Seule l'expérience pourra montrer jusqu'à quelles résolutions la précision obtenue demeure acceptable. Les résultats expérimentaux obtenus font justement l'objet du prochain chapitre.

Chapitre 4

RÉSULTATS EXPÉRIMENTAUX

4.1 Introduction

L'implémentation de la nouvelle architecture de TDC sur FPGA est le fruit de nombreux efforts dans le domaine de la reconfiguration dynamique. Cette dernière est notamment rendue possible par notre méthodologie permettant un ajustement fin et précis des délais d'un circuit programmable dynamiquement. Dans ce chapitre, nous présentons les différents résultats expérimentaux obtenus dans l'application de la méthodologie proposée à la conception de convertisseurs numériques de temps. En premier lieu, nous présentons les caractéristiques obtenues pour les lignes à délais programmables et l'oscillateur à fréquence variable à grande portée. En deuxième lieu, les résultats de l'implémentation de l'architecture novatrice proposée de TDC sont exposés. En troisième et dernier lieu, les résultats obtenus pour l'implémentation de l'interpolateur utilisant la méthode du vernier sont présentés.

4.2 Lignes à délais programmables

Les listes de routes générées afin de rendre possible une résolution d'un picoseconde contiennent environ 40000 routes, utilisant de 0 à 6 tables de correspondances (configurées en tampons) et des profondeurs atteignant les 30 points d'interconnexions programmables (PIP). La Figure 4.1 présente les demi-périodes mesurées pour l'ensemble de ces routes. On observe notamment que la résolution de 1ps est obtenue avec une redondance d'environ 10 routes par incrément (de 1ps), et que la portée d'ajustement à cette résolution s'étend sur plus de 2,5ns. La mesure de la demi-période associée à chaque route (3.2.2) se fait à raison de 5 à 10 routes par secondes, et la mesure des 40000 routes est complétée en quelques heures. Il existe une liste (de routes) de référence pour l'ensemble des lignes à délais, et une deuxième pour l'oscillateur à fréquence variable.

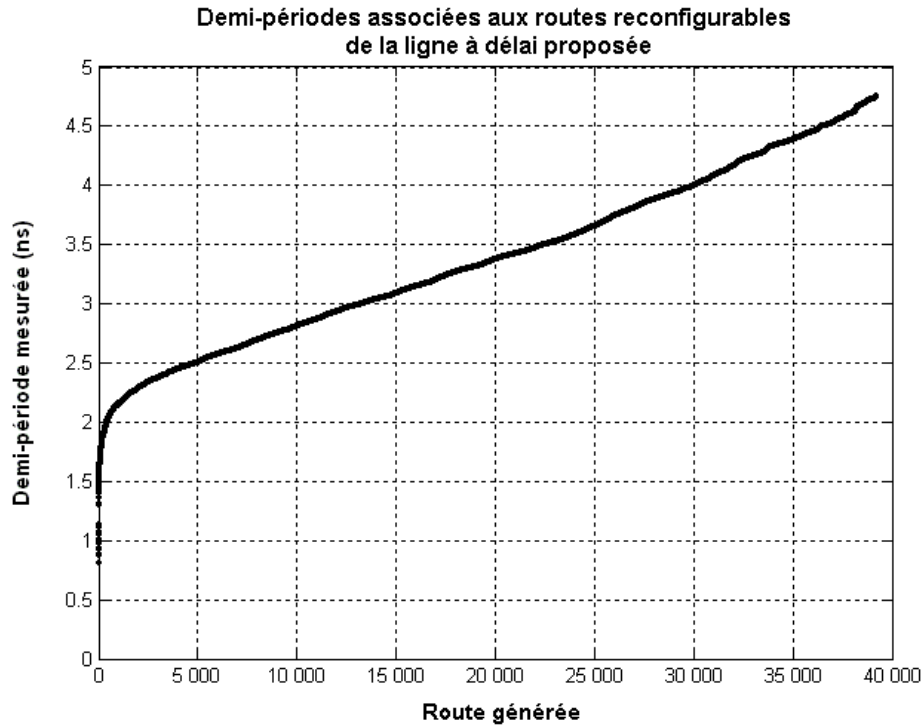


FIGURE 4.1 Demi-périodes associés à chaque route de la liste générée pour la ligne à délai programmable proposée.

4.3 Oscillateur à fréquence variable

L'oscillateur à fréquence variable est une pièce clé de la méthode de calibration de l'architecture de TDC proposée. Avec la méthodologie proposée, basée sur l'ajustement fin des délais, la période de l'oscillateur produit est ajustable avec une précision de 1ps. La portée d'ajustement de cette période va au-delà des 60ns, en utilisant jusqu'à 64 LUT voisines supplémentaires, ce qui est largement suffisant pour l'application ciblée. En moyenne, le processus de calibration requiert environ une dizaine de secondes, et moins encore lorsque la précision requise est inférieure. La Figure 4.2 présente une mesure de stabilité de l'oscillateur sur une période de 10 minutes. On observe notamment que la période augmente sensiblement après configuration, ce qui est attribuable à l'augmentation de la température de l'anneau en oscillation.

Avec l'oscillateur proposé, les résultats permettent d'aspirer à de bons résultats pour la calibration de l'interpolateur fin du TDC. Dans ce contexte, la période de

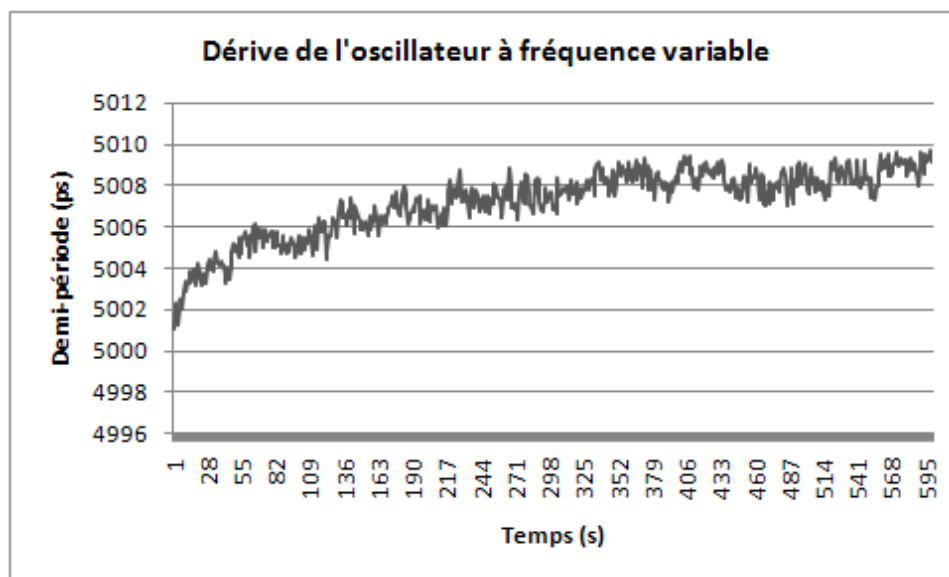


FIGURE 4.2 Dérive de la demi-période de l'oscillateur à fréquence variable sur une période de 10 minutes.

stabilité sur 10 minutes est largement excessive. En effet, la calibration de chacune des lignes devrait se faire en quelques secondes, une fois l'oscillateur lui-même calibré.

4.4 Une nouvelle architecture de TDC basée sur l'ajustement fin des délais

Les résultats obtenus avec l'implémentation de l'architecture de TDC proposée sur le Virtex-II Pro sont présentés dans cette section. Différents paramètres et critères de performances ont été mesurés, notamment la non-linéarité, différentielle et intégrale, et la précision de mesures. Dans les résultats présentés, l'interpolateur a été calibré pour des résolutions de 50, 25, et 10 picosecondes. Cependant, seule une résolution de calibration a mené à l'implémentation d'un TDC complet, soit celle de 50ps. La limitation est essentiellement liée à la grandeur de la région dynamique dans le XC2VP30, et à la présence d'un second PowerPC (non-utilisé) dans le milieu de cette dernière. Néanmoins, la non-linéarité à ces résolutions peut être observée en calibrant l'interpolateur à des résolutions plus fines, mais la portée dynamique devient insuffisante pour couvrir la période de l'horloge système de 10ns. Bien que la précision ne puisse être

mesurée, la non-linéarité nous permet tout de même d'estimer la qualité du processus de calibration à ces résolutions.

Pour l'implémentation complète du TDC, calibré à 50ps de résolution, la précision a été mesurée en injectant un signal d'horloge à fréquence fixe dans l'entrée de l'interpolateur, et en mémorisant les intervalles de temps séparant des fronts montants consécutifs. En calculant l'écart type de la distribution résultante, la précision du TDC est obtenue. À la Figure 4.3, une distribution d'environ 1000 mesures est obtenue avec une moyenne de 124,9997ns et un écart-type de 40,3ps. La fréquence du signal d'entrée étant de 8MHz, la justesse de la moyenne est de 0,3ps.

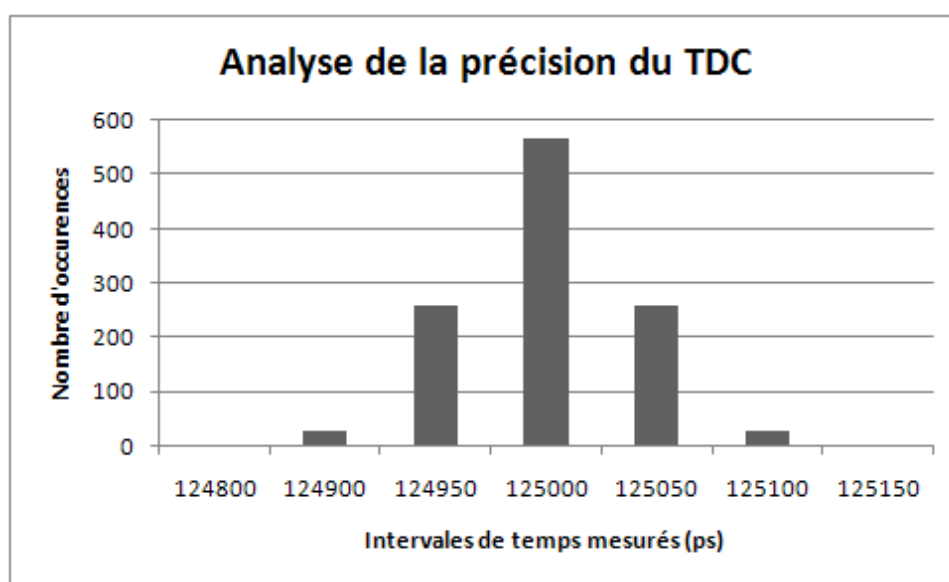


FIGURE 4.3 Distribution des mesures effectuées en injectant un signal d'horloge à période constante, avec une moyenne de 124,9997ns et un écart-type de 40,3ps.

La figure 4.4 présente les précisions mesurées pour différents intervalles de temps, allant de 62,5ns à 2,048ms. Afin d'observer l'influence d'un chauffage sur la précision, le test est également reproduit en soumettant le circuit à un courant d'air chaud (séchoir à cheveux). Les résultats obtenus présentent une dégradation moyenne de la précision de l'ordre de 10ps.

Afin de mesurer la non-linéarité de l'implémentation produite, un signal d'horloge dont la fréquence n'est pas un multiple de la fréquence d'horloge du système (100MHz)

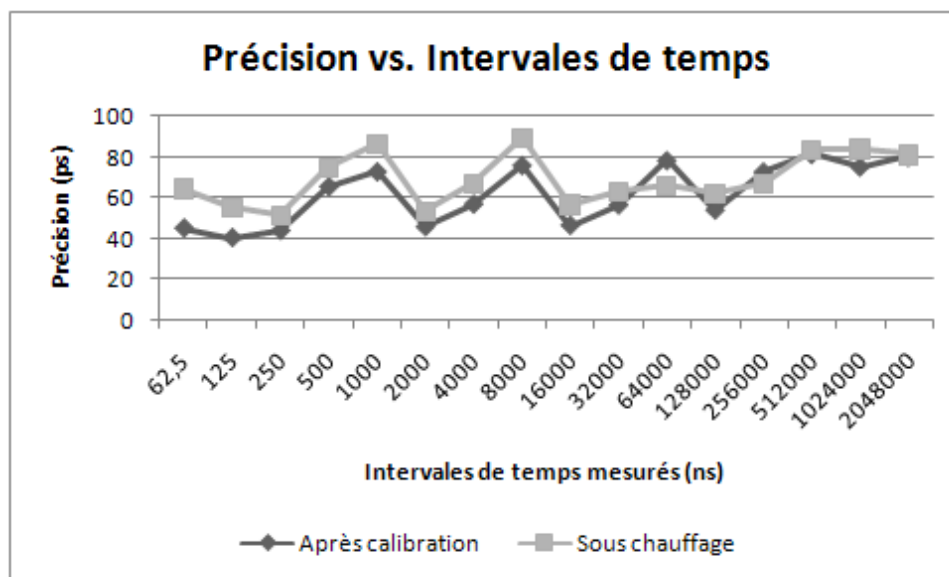
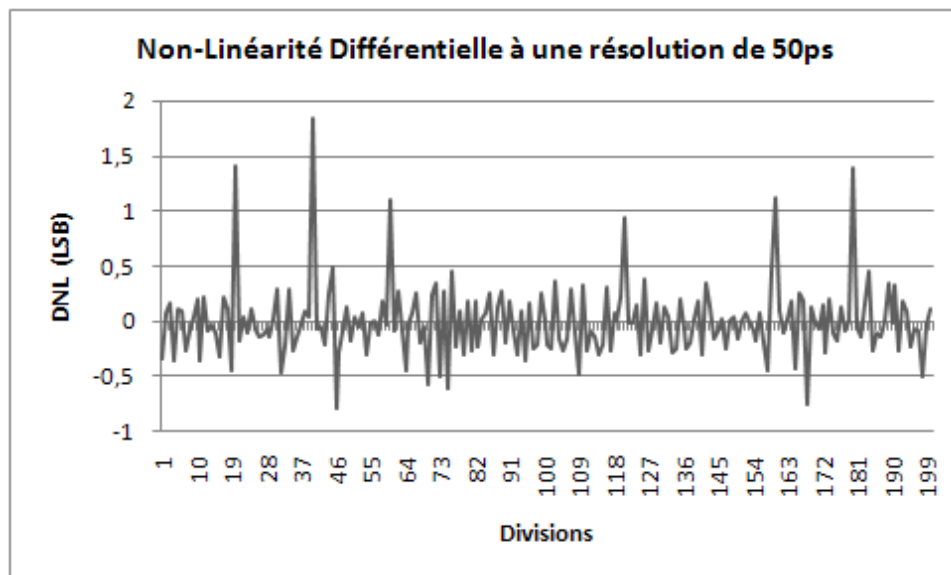


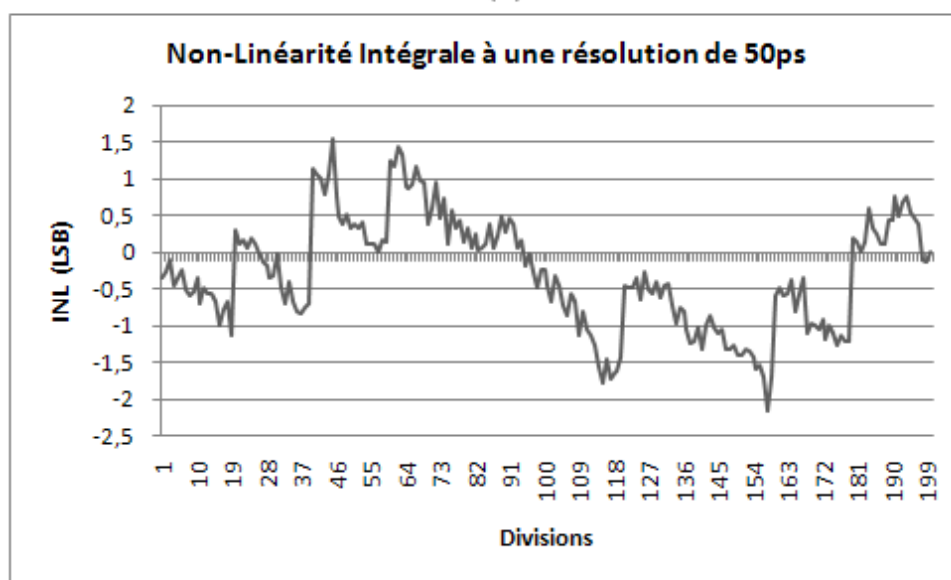
FIGURE 4.4 Influence d'un chauffage sur la précision pour différents intervalles de temps allant de 62,5ns à 2,048ms.

est injecté dans l'interpolateur. L'idée est de stimuler l'interpolateur avec des temps de transitions (fronts montants) distribués uniformément de manière pseudo-aléatoire. Dans le cas où l'interpolateur serait idéal, chacune des divisions serait stimulée autant de fois, mais en pratique les divisions de l'interpolateur ne sont pas toutes identiques ; certaines sont plus larges et d'autres plus fines. En injectant des impulsions pseudo-aléatoires dans l'interpolateur, la distribution obtenue est représentative de la taille de chacune de ses divisions. L'implémentation considérée offre une plage de 220bits, qui sont ici calibrés à une résolution de 50ps pour couvrir une plage dynamique de 11ns. Les résultats des non-linéarités, différentielle et intégrale, sont présentés dans la Figure 4.5, on n'y considère que les 200 premiers bits correspondants à la période d'horloge de 10ns. On peut y observer que la non-linéarité différentielle est contenue dans un intervalle allant de -0,75 à 2 divisions. On observe également que dans plus de 90% des divisions, celle-ci est contenue dans un intervalle de -0,5 à 0,5 divisions, exposant la force de la méthodologie proposée. La non-linéarité intégrale obtenue s'étend d'environ -2 à 2 divisions.

Dans les Figures 4.6 et 4.7, l'interpolateur de 220 lignes à délais est calibré pour des résolutions de 25 et 10 picosecondes. Pour de telles calibrations, les portées dyna-



(a)



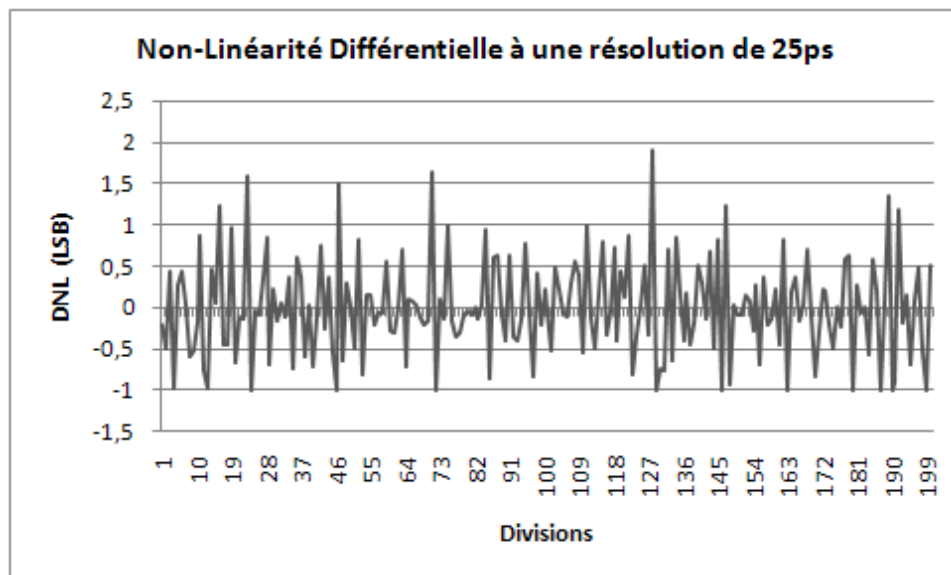
(b)

FIGURE 4.5 Non-linéarité différentielle (a) et intégrale (b), à une résolution de 50ps sur une portée dynamique de 10ns.

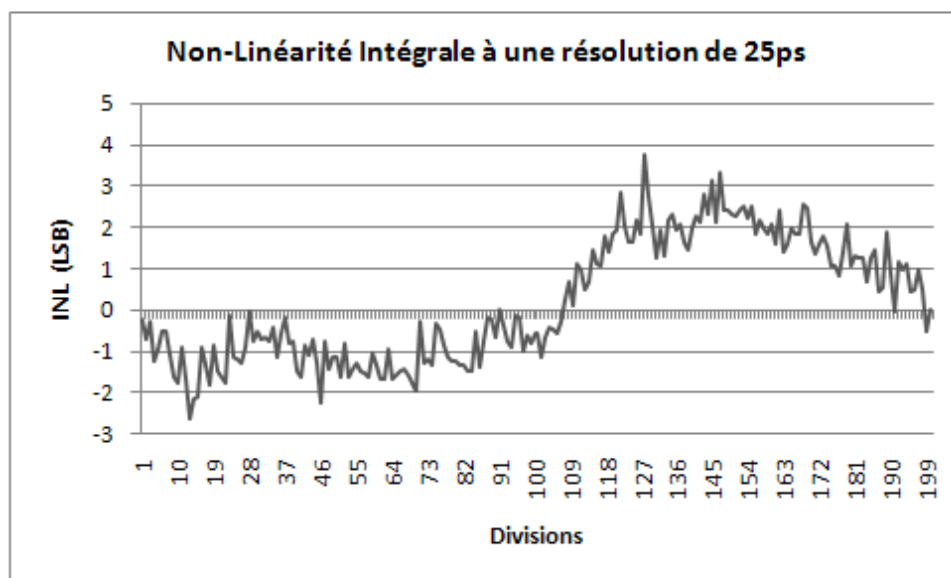
miques obtenues sont respectivement de 5,5ns et 2,2ns, mais seulement celles de 5ns et 2ns ont été considérées pour l'analyse de non-linéarité. Dans les résultats obtenus, on observe que la non-linéarité différentielle augmente sensiblement avec la diminution de la résolution. En considérant le nombre de divisions pour lesquelles la taille relative excède l'unité, on décompte 5 divisions à 50ps, 8 à 25ps, et 31 à 10ps. Même à une résolution aussi faible que 10ps, on observe que la non-linéarité différentielle n'excède pas 3,5 divisions, alors que la non-linéarité intégrale est contenue dans un intervalle allant de -2 à 2,5 divisions. Ces résultats montrent clairement que le processus de calibration est en mesure de délivrer un interpolateur de qualité, basé sur notre architecture, à des résolutions aussi faibles que 10 picosecondes, ce qui à notre connaissance représente le meilleur résultat obtenu à ce jour pour une implémentation FPGA.

4.5 Une nouvelle implémentation de la méthode du vernier avec deux oscillateurs

Dans cette section, les résultats liés à l'application de notre méthodologie à l'implémentation FPGA d'un interpolateur basé sur la méthode du vernier avec deux oscillateurs démarrables sont présentés. Puisque la résolution d'un tel interpolateur est déterminée par la différence des périodes entre les deux oscillateurs, la stabilité des oscillateurs est un point critique de conception. Dans la section précédente, la stabilité d'un oscillateur ajustable à été rapportée en oscillation libre sur une longue période de temps. Dans cette section, les résultats de stabilité présentés illustrent la variation de la résolution, liée aux variations de périodes des oscillateurs, au fil du temps (Figure 4.8). La période de variation considérée est de 30 secondes, pendant laquelle les oscillateurs sont en mode d'oscillations libres. Les oscillateurs ont également été calibrés pour des périodes d'oscillations centrées à 5ns. Les résolutions considérées pour cette analyse sont de 5, 10, et 25 picosecondes. Les périodes de chaque oscillateur sont mesurées à chaque seconde, et la stabilité estimée de la résolution est ainsi obtenue. Pour l'ensemble des résolutions considérées, on observe une légère variation de la résolution, de l'ordre de quelques picosecondes.

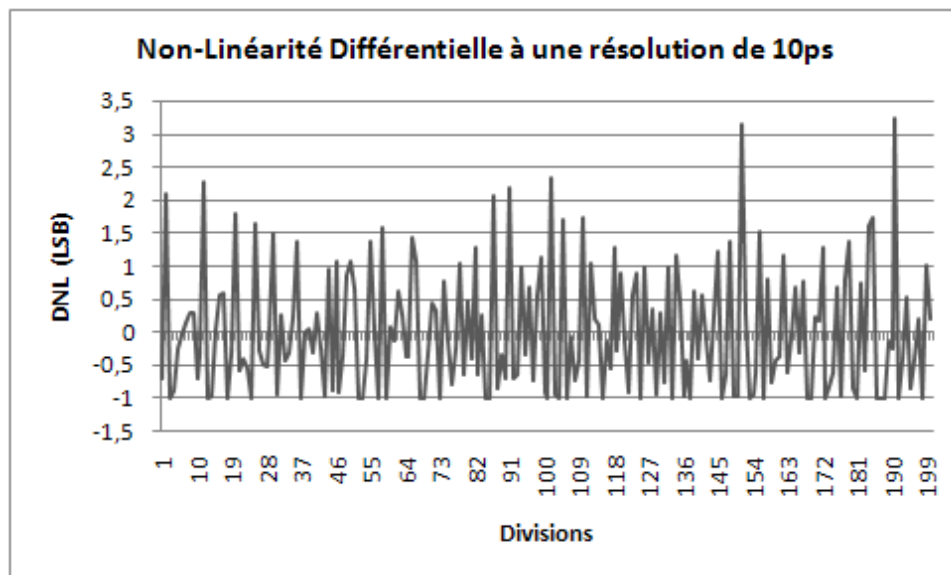


(a)

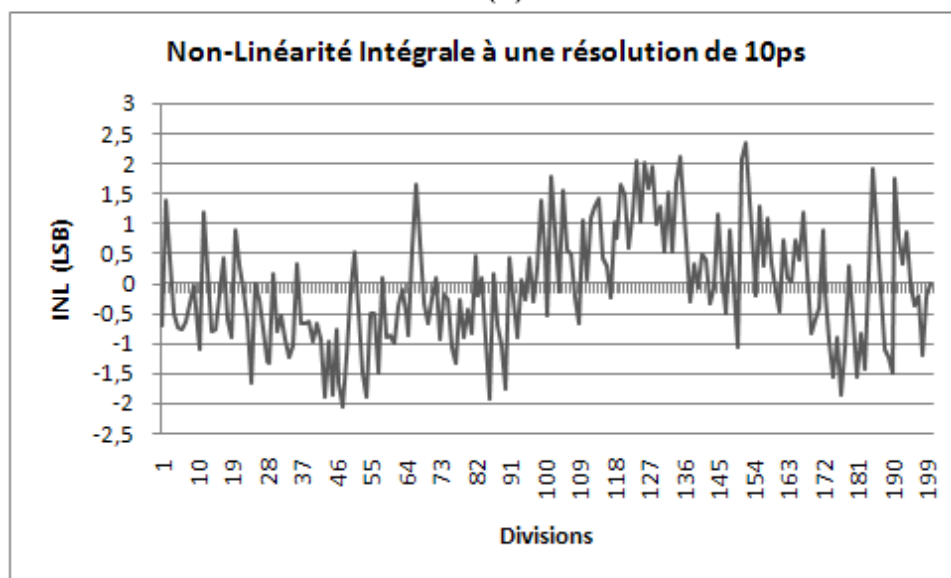


(b)

FIGURE 4.6 Non-linéarité différentielle (a) et intégrale (b), à une résolution de 25ps et sur une portée dynamique de 5ns.



(a)



(b)

FIGURE 4.7 Non-linéarité différentielle (a) et intégrale (b), à une résolution de 10ps et sur une portée dynamique de 2ns.

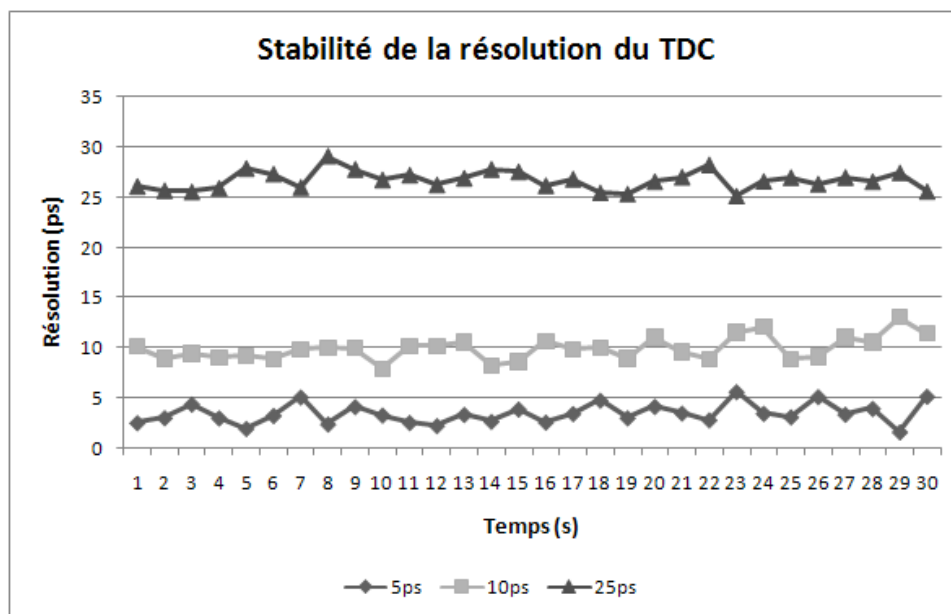


FIGURE 4.8 Stabilité sur 30 secondes des résolutions calibrées pour 25ps, 10ps, et 5ps.

Bien que la différence de périodes entre les deux oscillateurs puisse être aussi faible que 5 picosecondes, étant donné l'instabilité qui occasionne des fluctuations de l'ordre de quelques picosecondes, il est prévisible que l'interpolateur proposé aura d'importants problèmes de précision à de telles résolutions. La précision de l'interpolateur est le deuxième point d'intérêt à avoir été étudié pour cet interpolateur. Afin d'obtenir la précision, l'interpolateur a été stimulé avec des impulsions de départ et d'arrêt distancés par des intervalles constants. Encore, l'écart type de la distribution des mesures obtenues est pris comme précision. La Figure 4.9 illustre les précisions obtenues en fonction de la résolution du TDC. La précision est obtenue à partir d'une distribution de 1000 mesures, et pour chaque résolution reproduit la mesure de précision 10 fois. Il devient alors possible d'observer la variation de la précision en cours d'opération.

On observe clairement dans la Figure 4.9 que la précision du TDC se dégrade avec l'affinement de la résolution. En effet, jusqu'à une résolution de 30ps, notre implémentation a une précision inférieure à 1 LSB (pas de quantification), mais en diminuant la résolution sous cette valeur, la précision passe de 2LSB à 18ps jusqu'à 20LSB à 6ps. On observe également dans ce graphe que la variation de la précision augmente avec la diminution de résolution, ce qui est principalement attribuable à

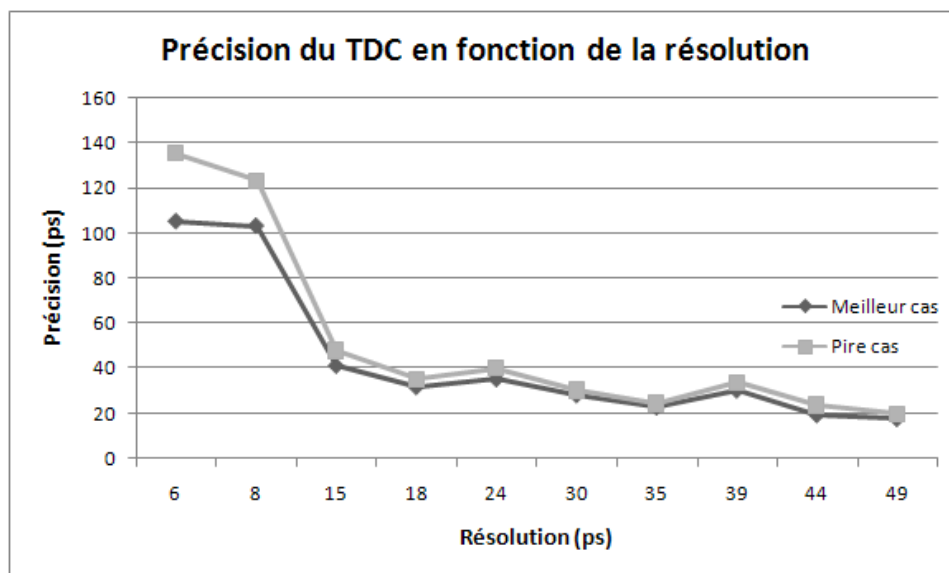


FIGURE 4.9 Précision du TDC en fonction de la résolution.

l'instabilité des oscillateurs.

Dans ce test, un autre paramètre d'intérêt est la moyenne des intervalles de temps mesurés. La Figure 4.10 résume ainsi les mesures moyennes obtenues à chacune des résolutions considérées. Pour chacune de ces dernières, la moyenne est calculée à partir de 10000 mesures d'intervalles de temps. Sur la base des moyennes obtenues aux résolutions supérieures à 30ps, il est raisonnable de postuler que la mesure exacte se situe aux environs de l'intervalle compris entre 1450ps et 1475ps. Or il apparaît qu'avec la diminution de la résolution, la mesure moyenne s'écarte fortement de cette valeur. Ce constat est une autre conséquence de l'instabilité des oscillateurs, et donc de la résolution. En effet, selon l'Équation 2.3, pour une mesure donnée la valeur d'un compteur sera d'autant plus grande que la résolution sera plus faible. Or comme la valeur de la mesure est donnée par le produit de la valeur du compteur avec la valeur de la résolution, il en résulte que l'influence d'une variation de résolution devient d'autant plus importante que la résolution est faible.

À la lumière des résultats obtenus, il apparaît qu'afin d'obtenir une résolution permettant une précision au moins aussi bonne que cette dernière, la résolution minimale se situe quelque part entre 25ps et 30ps. Dans la Figure 4.11, on présente une

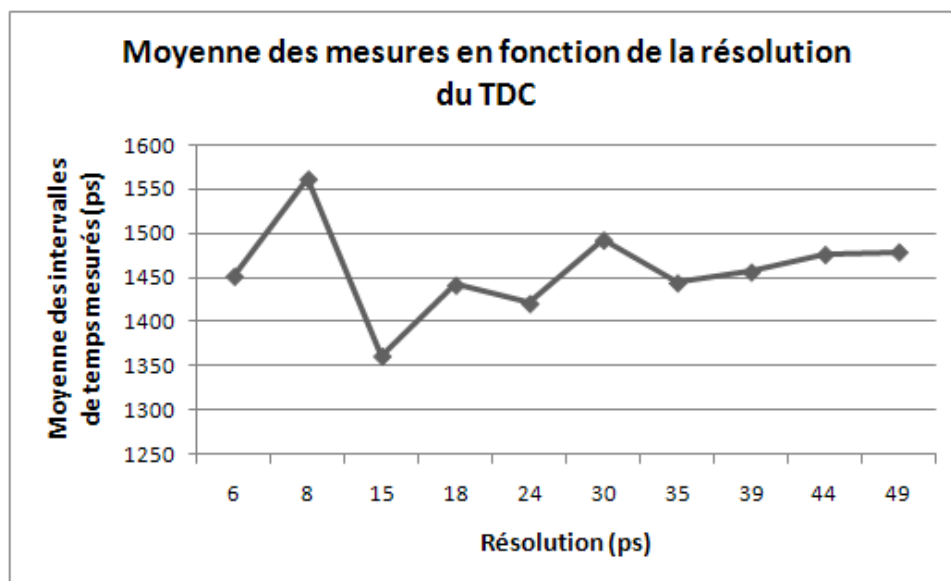


FIGURE 4.10 Moyenne des intervalles de temps mesurés en fonction de la résolution du TDC pour des résolutions allant de 5ps à 50ps.

distribution obtenue avec un millier de mesures présentant un écart type de 25,1ps pour une résolution de 26,4ps.

4.6 Conclusion

Dans ce chapitre, les résultats expérimentaux découlant de nos travaux de recherche ont été présentés. Notre méthodologie basée sur la reconfiguration dynamique d'un FPGA rend possible l'implémentation de lignes à délais ajustables avec une résolution de 1ps sur une plage de plus de 2,5ns. Sur la base d'une telle résolution d'ajustement, un oscillateur dont la période est ajustable avec une précision de 1ps est également présenté.

Sur la base de la capacité d'ajuster finement et précisément les délais d'un circuit, une nouvelle architecture de TDC est présentée. Dans une implémentation complète, l'implémentation de cette architecture sur un Virtex-II Pro permet d'obtenir une résolution de 50ps avec une précision de 45ps. Les résultats d'analyse de non-linéarité à des résolutions de 25ps et 10ps montrent que la méthodologie peut être utilisée pour atteindre ces résolutions. Alors que l'architecture basée sur la ligne à délai simple sur

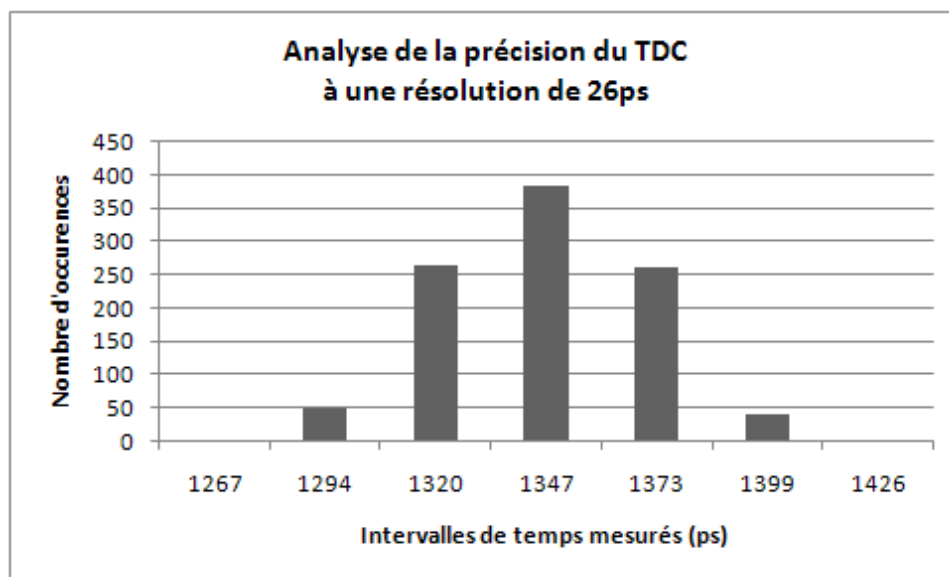


FIGURE 4.11 Distribution de mesures d'intervalles de temps constantes montrant une moyenne de 1347ps et un écart type de 25,1ps pour une résolution de 26ps.

écoute permet d'atteindre des résolutions de 17ps sur un Virtex-5 fabriqué en 65nm, notre approche permet d'atteindre jusqu'à 10ps sur un Virtex-II Pro fabriqué en 130nm.

La méthodologie développée pour réaliser l'architecture proposée à également été appliquée à l'implémentation d'un interpolateur basé sur la méthode du vernier avec deux oscillateurs. Les résultats montrent que bien que des résolutions de l'ordre de 5ps sont atteignables, à ces résolutions la précision n'est simplement pas au rendez-vous. Néanmoins, il est observé que pour des résolutions de l'ordre de 25ps, la précision est adéquate, équivalente à la valeur de la résolution. En termes de résolutions et de précisions atteintes avec cette architecture, notre implémentation amène une certaine amélioration par rapport aux résultats présentés dans la littérature. Notamment, cette implémentation tient compte à la fois des variations de températures et des variations dues au procédé de fabrication.

Au terme de ce travail, les performances des TDC implémentés au moyen de notre approche basée sur l'ajustement fin des délais sur un Virtex-II Pro peuvent être résumées dans le Tableau 4.1 pour la nouvelle architecture proposée, et dans le

Tableau 4.2 pour l'implémentation de la méthode du vernier avec deux oscillateurs.

TABLEAU 4.1 Comparaison des TDC dans la littérature offrant un temps mort d'un cycle ou moins.

Auteurs	FPGA	Résolution	Précision	DNL	INL
Kalisz et al. (1997)	QuickLogic	200ps	N-D	-0,5 à 0,5	-0,2 à 1,4
Wu et al. (2003)	ACEX 1K	400ps	300ps	-0,4 à 1,7	N-D
Song et al. (2006)	ACEX 1K	112,5ps	129,4ps	-0,4 à 0,8	-0,6 à 0,7
Song et al. (2006)	Virtex-II	69,5ps	93,1ps	-1 à 1	-2 à 2
Minas et al. (2007)	Virtex-II	69,3ps	N-D	N-D	N-D
Wu et al. (2008)	Cyclone-II	60ps	40ps	-1 à 2,7	N-D
Amiri et al. (2009)	Spartan-3	75ps	300ps	-1 à 2,5	-2,5 à 3
Favi et al. (2009)	Virtex-5	17ps	24ps	-1 à 3,55	-3 à 2,58
Ce travail (2009)	Virtex-II Pro	50ps	45ps	-0,8 à 1,9	-2,2 à 1,6
Ce travail (2009)	Virtex-II Pro	25ps	N-D	-1 à 2	-2,8 à 4
Ce travail (2009)	Virtex-II Pro	10ps	N-D	-1 à 3,4	-2,1 à 2,5

TABLEAU 4.2 Comparaison des interpolateurs utilisant la méthode du vernier avec deux oscillateurs.

Auteurs	FPGA	Résolution	Précision
Lin et al. (2006)	Xilinx Spartan-3/Virtex-4	27ps	N-D
Jin et al. (2007)	Xilinx	100ps	N-D
Ce travail (2009)	Xilinx Virtex-II Pro	26,4ps	25,1ps
Ce travail (2009)	Xilinx Virtex-II Pro	15ps	~40ps
Ce travail (2009)	Xilinx Virtex-II Pro	5ps	~110ps

Chapitre 5

CONCLUSION

5.1 Synthèse des travaux

Dans ce travail, une approche permettant d'ajuster les délais d'un circuit reprogrammable au moyen de la reconfiguration dynamique a été présentée. Afin de rendre possible des résolutions de 1ps, les délais configurables sont implémentés par les points d'interconnexions programmables d'une matrice de routage. La méthodologie proposée rend ainsi possible la génération de millions de routes différentes entre deux points du circuit. Notre méthodologie permet également d'ajuster précisément les délais en tenant compte des variations de températures et de procédés de fabrication. Il nous a ainsi été possible de réaliser un oscillateur à fréquence variable dont la période peut être ajustée avec une précision de 1ps.

Sur la base de cette capacité d'ajustement des délais d'un circuit, une nouvelle architecture de TDC est proposée. Cette architecture se caractérise par l'utilisation de nombreuses lignes à délais programmables en parallèles. Une telle architecture surmonte ainsi les limitations liées aux délais minimaux du circuit reprogrammable. Le processus de calibration que nous proposons pour cette architecture présente 3 forces importantes :

- Le processus de calibration tient compte des variations de température ainsi que des variations physiques liées au procédé de fabrication, qui affectent tous deux les délais des interconnexions.
- Le processus de calibration n'est pas affecté par le biais de l'arbre d'horloge, utilisé dans l'implémentation de l'architecture proposée, lors de l'ajustement des délais.
- Le processus de calibration permet d'atteindre des résolutions à la fine pointe de l'état de l'art et mieux, en utilisant des circuits programmables d'anciennes générations, tel le Virtex-II Pro fabriqué à l'aide d'un procédé CMOS à 130nm.

C'est ainsi que notre implémentation d'un TDC basé sur cette architecture à été en mesure d'offrir une résolution de 50ps, avec une précision de l'ordre de 45ps. De plus, nos résultats montrent que le processus de calibration est en mesure de supporter des résolutions aussi faibles que 10ps. À notre connaissance, cette résolution représente la meilleure valeur obtenue avec une implémentation FPGA.

L'approche proposée a également été appliquée à l'implémentation d'un interpolateur basé sur la méthode du vernier avec deux oscillateurs. La résolution qu'il est possible d'atteindre avec un tel interpolateur correspond à la différence entre les périodes des deux oscillateurs démarrables. Ainsi, puisque notre approche permet d'ajuster précisément la période d'un oscillateur, il semblait tout pertinent d'explorer cette application. Bien que les résultats obtenus montrent que des résolutions aussi faibles que 5ps sont atteignables, ils montrent également que la précision à ces résolutions est au mieux médiocre, dépassant les 100ps. Néanmoins, notre implémentation permet d'obtenir des précisions acceptables (précision inférieure ou égale à la résolution) à des résolutions de l'ordre de 25ps (ou plus). Par surcroît, notre méthodologie basée sur l'ajustement fin des délais se démarque en ce qu'elle permet encore de tenir compte des variations de températures et des variations dues au procédé de fabrication.

5.2 Limitations de la solution proposée

L'approche proposée comporte certaines limitations. En premier lieu, on peut souligner le besoin d'un système embarqué basé sur un processeur afin d'assurer la reconfiguration dynamique du circuit reprogrammable, ce qui représente un coût supplémentaire en termes de surface. De plus, la reconfiguration dynamique n'est pas pleinement supportée par les outils commerciaux actuels, et en outre ne permet pas la reconfiguration dynamique à grain fin rendue possible par la librairie DXCL. Bien que cette librairie supporte les FPGA des familles Virtex-II et Spartan-3, les familles plus récentes Virtex-4 et Virtex-5 ne sont pas (encore) supportées. L'implémentation présente requiert approximativement un bloc combinatoire logique (CLB) par ligne à délais, ce qui représente un coût de surface considérable considérant qu'à 100MHz une résolution de 10ps requiert 1000 de ces lignes pour couvrir une portée de 10ns! L'intérêt lié à l'utilisation de familles de FPGA plus récentes, permettant des fréquences d'opération supérieures, serait donc de réduire la portée dy-

namique requise réduisant du même coup le nombre de lignes à délais requises.

D'autre part, bien que l'architecture proposée puisse supporter des résolutions aussi faibles que 10ps, on observe que cette dernière valeur semble approcher la limite atteignable. En fait, le processus de calibration de l'architecture est limité par la stabilité de l'oscillateur à fréquence variable, ainsi que par les variations thermiques des points d'interconnexions programmables qui causent des variations de délais.

En respect de l'implémentation proposée d'un interpolateur basé sur un vernier avec deux oscillateurs, on rappelle la limitation liée à la stabilité des deux oscillateurs. Étant donnée les fluctuations des deux périodes d'oscillation, lorsque que la résolution cible s'approche de l'amplitude des fluctuations, la précision des mesures réalisées est sérieusement atteinte.

5.3 Améliorations futures

Il a été mentionné plus haut que le coût en surface d'un bloc combinatoire logique par ligne à délai, bien qu'acceptable, est loin d'être négligeable. Afin de réduire le nombre de lignes requis pour une résolution donnée, il est possible d'augmenter la fréquence d'horloge du système jusqu'à un certain point. Pour aller plus loin, il serait intéressant de considérer d'augmenter la densité des lignes à délais. Une approche envisageable serait d'utiliser également les bistables D (*D-Latch*) présents dans chaque CLB pour implémenter une deuxième ligne à délai programmable.

Dans les résultats de non-linéarité présentés pour l'architecture proposée, certaines divisions présentent une taille relativement plus grande. Il serait intéressant d'étudier la possibilité de re-calibrer localement les lignes à délais associées, sur la base de la non-linéarité différentielle mesurée. Une telle approche pourrait ainsi permettre de réduire sensiblement l'ampleur de cette dernière. Il serait également intéressant de considérer le remplacement des M lignes à délai parallèles à la tête des colonnes de N lignes parallèles par une ligne à délais chaînés. En effet, une telle approche permettrait théoriquement de réduire l'influence de la température sur le biais entre les colonnes du TDC proposé.

En ce qui a trait à l'implémentation de la méthode du vernier avec deux oscillateurs, notre méthodologie rend possible des résolutions sans précédents. Néanmoins, il a été observé que pour de très fines résolutions, la précision est considérablement dégradée. La pierre angulaire de cette précision étant la stabilité des oscillateurs, des travaux futurs devraient se pencher sur ce problème.

Références

- [1] AFNOR (1994). NF ISO-5725-1, Exactitude (justesse et fidélité) des résultats et méthodes de mesure – Partie 1 : Principes généraux et définitions.
- [2] AGILENT TECHNOLOGIES (2009). Agilent U1050A Acquisiris Time-to-Digital Converter. *Technical Overview*. pp. 1–5.
- [3] AMIRI, A. M., BOUKADOUM, M. et KHOUAS, A. (2009). A Multihit Time-to-Digital Converter Architecture on FPGA. *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 530–540.
- [4] AMIRI, A. M., KHOUAS, A. et BOUKADOUM, M. (2007). On the Timing Uncertainty in Delay-Line-based Time Measurement Applications Targeting FPGAs. *IEEE International Symposium on Circuit and Systems (ISCAS) 2007*. pp. 3772–3775.
- [5] BERGERON, E., FEELEY, M., DAIGNEAULT, M.-A. et DAVID, J. P. (2008). Using Dynamic Reconfiguration to Implement High-Resolution Programmable Delays on an FPGA. *IEEE Northeast Workshop on Circuit and Systems and TAISA Conference 2008*. pp. 265–268.
- [6] CHAN, A. H. et ROBERTS, G. W. (2004). A jitter characterization system using a component-invariant Vernier delay line. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, pp. 79–95.
- [7] DYER, S. A. (2001). *Wiley Survey of Instrumentation and Measurement (first edition)*. Wiley-IEEE Press.
- [8] FAVI, C. et CHARBON, E. (2009). Time-to-Digital Converter Implemented in 65nm FPGA Technology. *FPGA '09 : Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays*. pp. 113–120.
- [9] HAJIMIRI, A., LIMOTYRAKIS, S. et LEE, T. H. (1999). Jitter and Phase Noise in Ring Oscillators. *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 790–804.
- [10] HEWLETT PACKARD (1997). Fundamentals of Time Interval Measurement. *Application Note*. pp. 6–52.
- [11] JIN, K.-C. et MOON, M.-K. (2007). TDC Module for Time-Of-Flight. *2007 IEEE Nuclear Science Symposium Conference Record*. pp. 761–764.

- [12] K. PARK AND J. PARK (1998). 20 ps resolution time-to-digital converter for digital storage oscilloscopes. *1998 IEEE Nuclear Science Symposium Conference Record*.
- [13] KALISZ, J. (2003). Review of methods for time interval measurements with picosecond resolution. *Metrologia*, vol. 41, pp. 17–32.
- [14] KALISZ, J., SZPLET, R., PASIERBINSKI, J. et PONIECKI, A. (1997). Field-Programmable-Gate-Array-Based Time-to-Digital Converter with 200-ps Resolution. *IEEE Transactions on Instrumentation and Measurement*, vol. 46, pp. 51–55.
- [15] KAPLAN, S. M. (2004). *Wiley Electrical and Electronics Engineering Dictionary*. Wiley-Interscience IEEE Press.
- [16] KARADAMOGLOU, K., PASCHALIDIS, N. P., SARRIS, E., STAMATOPOULOS, N., KOTTARAS, G. et PASCHALIDIS, V. (2004). An 11-bit High-Resolution and Adjustable-Range CMOS Time-to-Digital Converter for Space Science Instruments. *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 214–222.
- [17] KINNIMENT, D. J., DIKES, C. E., HERON, K., RUSSELL, G. et YAKOVLEV, A. V. (2007). Measuring Deep Metastability and its effect on synchronizer performance. *IEEE Transactions on Very Large Scale Integration Systems*, vol. 15, pp. 1028–1039.
- [18] LEE, M. et ABIDI, A. A. (2008). A 9b, 1.25ps resolution coarse-fine time-todigital converter in 90nm CMOS that amplifies a time residue. *IEEE Journal of solid-state circuits*, vol. 43, pp. 769–777.
- [19] LEVINE, P. M. et ROBERTS, G. W. (2004). A High-Resolution Flash Time-to-Digital Converter and Calibration Scheme. *IEEE International Test Conference (ITC) 2004*. pp. 1148–1157.
- [20] LIN, M.-C., TSAI, G.-R., LIU, C.-Y. et CHU, S.-S. (2006). FPGA-Base High Area Efficient Time-To-Digital IP Design. *IEEE TENCON 2006 Conference*. pp. 1–4.
- [21] LYSAGHT, P., BLODGET, B., MASON, J., YOUNG, J. et BRIDGFORD, B. (2006). Invited Paper : Enhanced Architectures, Design Methodologies and CAD Tools for Dynamic Reconfiguration of Xilinx FPGAs. *Conference on Field Programmable Logic and Applications (FPL) 2006*. pp. 1–6.

- [22] MAATTA, K. et KOSTAMOVARA, J. (1998). A high-precision time-to-digital converter for pulsed time-of-flight laser radar applications. *IEEE Transactions on Instrumentation and Measurement*, vol. 47, pp. 521–536.
- [23] MERRIAM-WEBSTER (2009). Merriam-Webster Online English Dictionary. <http://www.merriam-webster.com/>.
- [24] MINAS, N., KINNIMENT, D., HERON, K. et RUSSELL, G. (2007). A High Resolution Flash Time-to-Digital Converter Taking Into Account Process Variability. *13th IEEE International Symposium on Asynchronous Circuits and Systems*. pp. 163–174.
- [25] NISSINEN, I. et KOSTAMOVARA, J. (2009). On-chip Voltage Reference-based Time-to-Digital Converter for Pulsed Time-Of-Flight Laser Radar Measurements. *IEEE Transactions on Instrumentation and Measurement*, vol. 58, pp. 1938–1948.
- [26] PALOJARVI, P., MAATTA, K. et KOSTAMOVARA, J. (2002). Pulsed time-of-flight laser radar module with millimeter-level accuracy using full custom receiver and TDC ASICs. *IEEE Transactions on Instrumentation and Measurement*, vol. 51, pp. 521–536.
- [27] SONG, J., AN, Q. et LIU, S. (2006). A High-Resolution Time-to-Digital Converter Implemented in Field-Programmable-Gate-Arrays. *IEEE Transactions on Nuclear Science*, vol. 53, pp. 236–241.
- [28] STROGATZ, S. H. (2001). Exploring Complex Networks. *Nature*, vol. 410, pp. 268–276.
- [29] SWANN, B. K., BLALOCK, B. J., CLONTS, L. G., BINKLEY, D. M., ROCHELLE, J. M., BREEDING, E. et BALDWIN, K. M. (2004). A 100-ps time-resolution CMOS time-to-digital converter for positron emission tomography imaging applications. *IEEE Journal of Solid-State Circuits*, vol. 39, pp. 1839–1852.
- [30] SZPLET, R., KALISZ, J. et SZYMANOWSKI, R. (2000). Interpolating time counter with 100 ps resolution on a single FPGA device. *IEEE Transactions on Instrumentation and Measurement*, vol. 49, pp. 879–883.
- [31] WEBSTER, J. G. (1999a). *The Measurement, Instrumentation and Sensors Handbook*. CRC Press and IEEE Press.

- [32] WEBSTER, J. G. (1999b). *Wiley Encyclopedia of Electrical and Electronics Engineering*, vol. 22. Wiley-Interscience IEEE Press.
- [33] WIKIPEDIA (2009). Timeline of time measurement technology. http://en.wikipedia.org/wiki/Timeline_of_time_measurement_technology.
- [34] WU, J. et SHI, Z. (2008). The 10-ps Wave Union TDC : Improving FPGA TDC Resolution beyond Its Cell Delay. *2008 IEEE Nuclear Science Symposium Conference Record*. pp. 3440–3446.
- [35] WU, J., SHI, Z. et WANG, I. Y. (2003). Firmware-only Implementation of Time-to-Digital Converter (TDC) in Field-Programmable Gate Array (FPGA). *IEEE 2003 Nuclear Science Symposium Conference*. pp. 177–181.
- [36] XILINX (2004). Two flows for partial reconfiguration : module based or difference based. *Technical Report XAPP290*. pp. 2–12.
- [37] XILINX (2005). Virtex-II Pro and Virtex-II Pro X Platform FPGAs : Complete Data Sheet. *Technical Report DS083*.
- [38] YOUSIF, A. S. et HASLETT, J. W. (2007). A Fine Resolution TDC Architecture for Next Generation PET Imaging. *IEEE Transactions on Nuclear Science*, vol. 54, pp. 1574–1582.

Publications de l'auteur

BERGERON, E., FEELEY, M., DAIGNEAULT, M.-A. et DAVID, J. P. (2008). Using Dynamic Reconfiguration to Implement High-Resolution Programmable Delays on an FPGA. *IEEE Northeast Workshop on Circuit and Systems and TAISA Conference 2008*. pp. 265–268.

DAIGNEAULT, M.-A. et DAVID, J. P. (2009a). A High Resolution Time-to-Digital Converter on FPGA Using Dynamic Reconfiguration. *IEEE Transactions on Instrumentation and Measurement*, (Soumis, Octobre 2009).

DAIGNEAULT, M.-A. et DAVID, J. P. (2009b). Toward 5ps Resolution TDC on a Dynamically Reconfigurable FPGA. *FPGA '10 : Proceeding of the ACM/SIGDA international symposium on Field programmable gate arrays*, (Accepté, Novembre 2009).

DAIGNEAULT, M.-A., LANGLOIS, J. M. P. et DAVID, J. P. (2008). Application Specific Instruction Set Processor Specialized for Block Motion Estimation. *IEEE International Conference on Computer Design (ICCD) 2008*. pp. 266–271.

Annexe A

Bibliothèque DXCL

La bibliothèque DXCL implémentée en langage C permet la reconfiguration dynamique d'un FPGA. Le processus de conception proposé par les outils standards ne permet que la génération statique de modules, or la bibliothèque DXCL permet la génération dynamique de modules. De plus, le niveau de granularité de configuration offert est le plus fin qui soit. Il est ainsi possible de configurer chacun des points de configurations des blocs logiques combinatoires (CLB) du FPGA, et ce de manière complètement dynamique au moyen d'une application C.

La bibliothèque DXCL est essentiellement divisée en 4 entités (Figure A.1). La définition de l'architecture du FPGA contient les fonctionnalités et les paramètres associés aux différentes puces disponibles. Les architectures pleinement supportées s'étendent des Virtex-II et Virtex-II Pro aux Spartan-3, tous du fabricant Xilinx. Le *back-end* de reconfiguration offre un accès aux bits de la mémoire de configuration, et donc aux points programmables du FPGA. La bibliothèque supporte 3 *back-end* différents :

- HWICAP : L'application (utilisant la bibliothèque DXCL) est exécutée sur un processeur embarqué sur le FPGA et accède à la mémoire de configuration au travers du port de configuration interne (ICAP) du FPGA.
- Client Sériel : L'accès à la mémoire de configuration se fait au travers un serveur sériel, exécuté sur un processeur embarqué ayant accès au port de configuration interne du FPGA (ICAP). L'application utilisant la bibliothèque DXCL s'exécute sur un client envoyant les commandes de reconfigurations au serveur.
- Fichier de programmation : La mémoire de configuration du FPGA correspond dans ce cas aux bits d'un fichier de programmation du FPGA (bitstream).

La cache présente dans la bibliothèque sert à contenir les segments de configuration (frames) qui sont lus et écrits lors de la reconfiguration dynamique. Au moyen de l'API de la bibliothèque, les aspects précédents sont abstraits permettant de réaliser

des applications dynamiques plus portables et performantes.

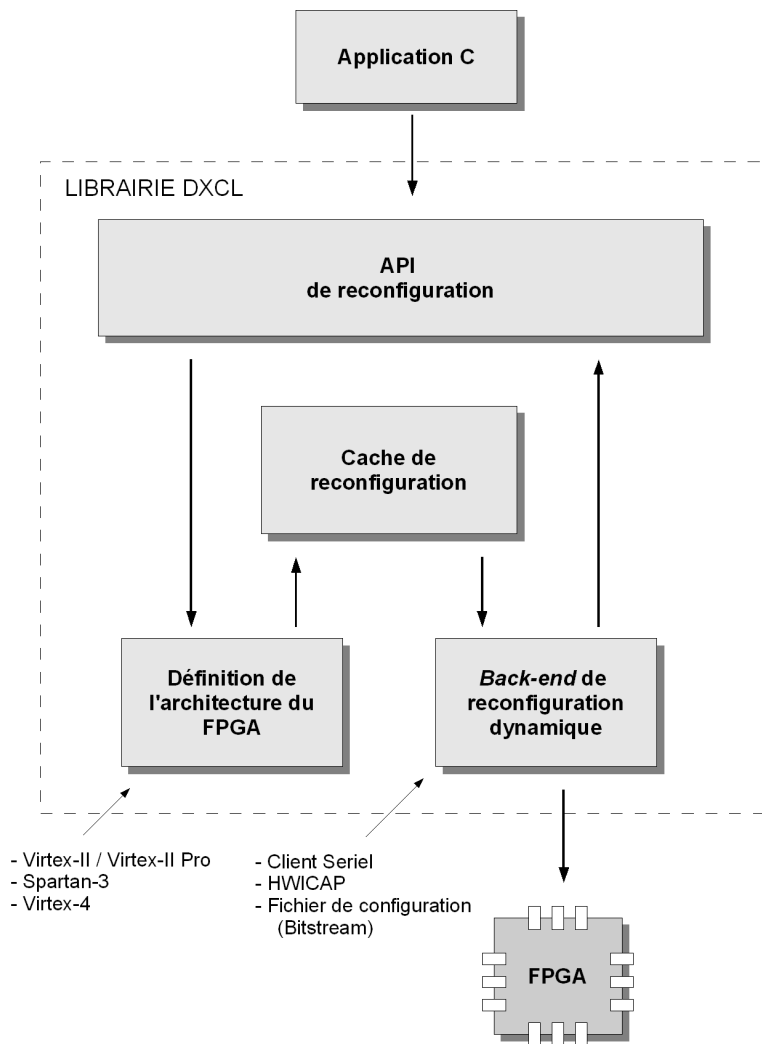


FIGURE A.1 Architecture de la bibliothèque DXCL.

Le fait de disposer de plusieurs back-end est une caractéristique très intéressante, notamment étant donnée la capacité d'exporter une région configurée dynamiquement sous format XDL. Le format XDL est une représentation textuelle d'un design sur FPGA Xilinx. Les outils du fabricant permettent ainsi la conversion vers le format NCD, qui permet de visualiser le design avec l'outil FPGA Editor. Tel qu'implicité, FPGA Editor de Xilinx est un outil graphique permettant de visualiser l'implémentation d'un design au niveau des points de configuration du FPGA.

L'outil permet également de créer ou de modifier une implémentation au moyen de ces points de programmation. Ainsi, en utilisant la bibliothèque avec un fichier de programmation en *back-end*, il est possible en exportant le design généré en format XDL de l'observer par la suite dans FPGA Editor, ce qui est très utile pour fins de validations et vérifications par inspection du design. Bien que cette approche ne soit pas comparable à l'utilisation d'un simulateur, elle est nettement plus efficace que la simple inspection d'un algorithme de génération d'un circuit !

Afin d'illustrer l'interface de programmation offerte par la bibliothèque, l'Algorithme A.1 présente un exemple simple d'application réalisant la reconfiguration dynamique d'un circuit. Dans cet exemple, une table de correspondance (LUT) à 4 entrées est programmée de sorte à inverser l'entrée F2, et d'autres points de programmation d'une tranche (*slice*) reliés à l'utilisation de la LUT. L'application réalise également la configuration dynamique d'un point d'interconnexion programmable (PIP).

Le type *dxcl_system_t* correspond à une structure composée notamment d'une définition d'architecture FPGA ainsi que d'un *back-end* de configuration. Le type *dxcl_tile_t* représente une tuile du FPGA, soit un bloc combinatoire logique (CLB), un bloc d'entré/sortie (IOB), ou un bloc mémoire, néanmoins, seuls les CLB sont supportés pour la reconfiguration dynamique. Le type *dxcl_site_t* permet de spécifier un site d'un CLB, soit l'une des 4 tranches (contenant 2 LUT à 4 entrées et 2 registre). Les commandes *dxcl_open()* et *dxcl_capture()* permettent d'initialiser le système (*dxcl_system_t*) avec une architecture et/ou un *back-end*, et de réaliser une capture de la mémoire de configuration. Dans les lignes 13 et 14, on sélectionne une tranche de la tuile aux coordonnées (x=10,y=10) du FPGA. Ensuite, aux lignes 16-18 la configuration de la LUT *F* et du site sélectionné est réalisée. La ligne 20 du programme configure un point d'interconnexion programmable de la matrice de routage. La commande *dxcl_frames_cache_writeback()* réalise la modification de la mémoire de configuration en fonction des modifications réalisées par l'application. Le programme se termine par un appel à la fonction *dxcl_close()* qui coupe le lien avec le port de configuration interne du FPGA (ICAP).

ALGORITHME A.1 EXEMPLE SIMPLE D'APPLICATION C AVEC LA BIBLIOTHÈQUE DXCL

```
1: #include "dxcl.h"
2: #define LUT_NOT_F2 0x3333
3:
4: int main(int argc, char** argv) {
5:
6:     dxcl_system_t sys;
7:     dxcl_tile_t tile;
8:     dxcl_site_t site;
9:
10:    dxcl_open(&sys, argv[1]);
11:    dxcl_capture(&sys);
12:
13:    tile = dxcl_get_tile(&sys, 10,10);
14:    site = dxcl_get_site(&sys, tile, V2_COMP_SLICE2);
15:
16:    dxcl_arch_set_lut(site, V2_RESS_F, LUT_NOT_F2);
17:    dxcl_arch_set_cfg(site, V2_RESS_FXMUX, V2_VAL_F);
18:    dxcl_arch_set_cfg(site, V2_RESS_XUSED, V2_VAL_0);
19:
20:    dxcl_arch_set_pip(tile, V2_WIRE_X1, V2_WIRE_DX1);
21:
22:    dxcl_frames_cache_writeback(&sys);
23:    dxcl_close(&sys);
24:
25: }
```
