



**HAL**  
open science

# Numérisation rapide d'un système synchronisé en sortie d'antennes multi-réparties tel que le Radiohéliographe de Nançay

El Houssain Ait Mansour

## ► To cite this version:

El Houssain Ait Mansour. Numérisation rapide d'un système synchronisé en sortie d'antennes multi-réparties tel que le Radiohéliographe de Nançay. Astrophysique [astro-ph]. Université Paris sciences et lettres, 2018. Français. NNT : 2018PSLEO001 . tel-01901916

**HAL Id: tel-01901916**

**<https://theses.hal.science/tel-01901916>**

Submitted on 23 Oct 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE DE DOCTORAT

De l'Université de recherche Paris Sciences et Lettres  
PSL Research University

Préparée à l'Observatoire de Paris, Station de  
Radioastronomie de Nançay

NUMÉRISATION RAPIDE D'UN SYSTÈME SYNCHRONISÉ D'ANTENNES  
RADIO MULTI-REPARTIES TEL QUE LE RADIOHÉLIOGRAPHE DE NANÇAY

**Ecole doctorale N° 127**

ASTRONOMIE ET ASTROPHYSIQUE D'ÎLE-DE-FRANCE

**Radioastronomie**

**Soutenue par**  
M. El Houssain **AIT MANSOUR**  
**Le 19 Janvier 2018**

Dirigée par  
M. Karl-Ludwig **KLEIN**

## COMPOSITION DU JURY :

Mme. ANGONIN Marie-Christine  
Observatoire de Paris, Président

M. DZAHINI Daniel  
IPN de Grenoble, Rapporteur

M. GRISEL Richard  
Université de Rouen, Rapporteur

M. KLEIN Karl-Ludwig  
Observatoire de Paris, Membre du jury

M. DA SILVA Bruno  
Observatoire de Paris, Membre du jury

M. GRANADO Bertrand  
Université Pierre et Marie CURIE,  
Membre du jury



# Table des matières

<b>Abstract</b>	<b>6</b>
<b>Remerciements</b>	<b>9</b>
<b>Introduction générale</b>	<b>11</b>
<b>I Contexte et objectifs de la thèse</b>	<b>13</b>
<b>1 La radioastronomie</b>	<b>15</b>
1.1 Introduction . . . . .	15
1.2 Observations en Radioastronomie . . . . .	15
1.3 Observation du soleil par le Radiohéliographe de Nançay (NRH) . . . . .	16
1.4 Importance de l’observation large bande du soleil . . . . .	19
1.5 Limites instrumentales du NRH actuel . . . . .	21
1.6 Objectifs de la thèse . . . . .	22
<b>2 Les radiotélescopes et imageurs radio</b>	<b>25</b>
2.1 Introduction . . . . .	25
2.1.1 LOFAR . . . . .	25
2.1.2 EMBRACE . . . . .	32
2.1.3 GMRT . . . . .	34
2.2 Comparaison globale entre les instruments existants . . . . .	36
2.2.1 La bande de base (bande numérisée) . . . . .	36
2.2.2 La résolution des convertisseurs A/D . . . . .	37
2.2.3 Précision de synchronisation des horloges . . . . .	38
2.3 Conclusion . . . . .	39
<b>3 Instrumentation et synchronisation</b>	<b>41</b>
3.1 Introduction . . . . .	41
3.1.1 Les notions de synchronisation . . . . .	41
3.1.2 Les objectifs de la synchronisation d’horloges . . . . .	41
3.2 Synchronisation et modélisation d’une horloge . . . . .	42
3.2.1 Introduction . . . . .	42

3.2.2	Notion des horloges logique et physique . . . . .	42
3.2.3	Modèle d'une horloge . . . . .	43
3.2.4	Compensation de l'offset et de skew . . . . .	44
3.2.5	Compensation de la gigue d'une horloge . . . . .	48
3.3	Instrumentation du temps/fréquence . . . . .	48
3.3.1	Introduction . . . . .	48
3.3.2	La précision . . . . .	48
3.3.3	La stabilité . . . . .	50
3.3.4	Mesure du temps d'intervalle . . . . .	51
3.4	Techniques de synchronisation . . . . .	57
3.4.1	Algorithme de Cristian . . . . .	57
3.4.2	Algorithme de Berkeley . . . . .	58
3.4.3	Protocole FTSP . . . . .	59
3.4.4	Synchronisation par Ethernet PTP . . . . .	59
3.4.5	Synchronisation par Ethernet WR . . . . .	61
3.5	Conclusion . . . . .	67
<b>4</b>	<b>Étude et conception du système de synchronisation appliqué au Radiohéliographe de Nançay</b>	<b>69</b>
4.1	Introduction . . . . .	69
4.2	Les caractéristiques du convertisseur A/D pour le radiohéliographe de Nançay .	69
4.2.1	Calcul de la puissance du bruit du ciel . . . . .	69
4.2.2	Calcul de la résolution binaire du convertisseur A/D . . . . .	72
4.2.3	Calcul de la précision de la fréquence de l'échantillonneur du convertisseur A/D . . . . .	75
4.3	Précision du système de synchronisation . . . . .	80
4.4	Conclusion . . . . .	82
<b>5</b>	<b>Algorithme de synchronisation et calibration de la chaine</b>	<b>85</b>
5.1	Introduction . . . . .	85
5.2	Algorithme de calcul et compensation des offsets . . . . .	86
5.2.1	Introduction . . . . .	86
5.2.2	Ajustement des offsets symétriques . . . . .	86
5.2.3	Ajustement des offsets asymétriques . . . . .	88
5.2.4	Performances et limitations . . . . .	91
5.2.5	Discussions et conclusion . . . . .	99
5.3	Calibration des retards . . . . .	100
5.3.1	Introduction . . . . .	100
5.3.2	Processus de calibration de la chaine de mesure . . . . .	100
5.3.3	Simulation . . . . .	102
5.3.4	Conclusion . . . . .	104
<b>6</b>	<b>Implémentation du système de synchronisation</b>	<b>105</b>
6.1	Introduction . . . . .	105
6.2	Contrôle et acquisition des retards . . . . .	106
6.2.1	Générateur des intervalles de temps $T_i$ . . . . .	108
6.2.2	Interfaces d'acquisition et de contrôle . . . . .	113

6.3	Calibration, calcul et ajustement des offsets . . . . .	114
6.3.1	Introduction . . . . .	114
6.3.2	Calibration de la chaîne . . . . .	114
6.3.3	Tri des canaux . . . . .	130
6.3.4	Calcul des écarts d'ajustement des horloges . . . . .	132
6.3.5	Calibrage des résolutions TDC/DTC . . . . .	138
6.3.6	Transfert des écarts d'ajustement aux horloges . . . . .	141
6.4	Description de l'architecture matérielle . . . . .	142
6.4.1	Introduction . . . . .	142
6.4.2	Circuit de mesure des intervalles de temps . . . . .	144
6.4.3	Architecture Maître/Esclaves . . . . .	145
6.5	Résultats de test et mesures . . . . .	153
6.5.1	Calibration de la chaîne Maître/Esclaves . . . . .	153
<b>Conclusion générale</b>		<b>170</b>
<b>7</b>	<b>Annexes</b>	<b>171</b>
7.1	Annexe 0 : Interfaces de communication . . . . .	171
7.1.1	Interface série TDC-FPGA . . . . .	171
7.1.2	Interface série TDC-PC . . . . .	176
7.1.3	Interface série FPGA-TDC . . . . .	182
7.2	Annexe I : Programme VHDL du générateur des intervalles $T_i$ . . . . .	190
7.3	Annexe II : Programme VHDL de l'interface série des résultats TDC-FPGA . . . . .	194
7.4	Annexe III : Programme VHDL de l'interface série TDC-PC . . . . .	198
7.5	Annexe IV : Programme VHDL de l'interface de programmation du TDC . . . . .	201
7.6	Annexe V : Programme VHDL de filtre de lissage . . . . .	204
7.7	Annexe VI : Programme VHDL de circuit de tri de trois canaux . . . . .	206
7.8	Annexe VII : Programme VHDL de circuit de calcul des écarts d'ajustement . . . . .	209
7.9	Annexe VIII : Programme VHDL de circuit de transfert des écarts d'ajustement . . . . .	211
<b>bibliographie</b>		<b>212</b>
<b>Liste des figures</b>		<b>228</b>
<b>Liste des tableaux</b>		<b>230</b>
<b>Glossaire</b>		<b>231</b>
<b>Index</b>		<b>235</b>

# Résumé

Le Radiohéliographe de Nançay est le seul instrument dédié à l'imagerie du soleil en ondes décimétriques-métriques. Il fonctionne sur le principe de l'interférométrie, en utilisant 47 antennes essentiellement réparties sur des axes est-ouest (3,2 km) et nord-sud (2,5 km). Cette étude a pour but d'explorer un nouveau concept de numérisation propre à la radioastronomie du futur, appliquée ici à l'interférométrie solaire. Elle porte sur la numérisation rapide d'un système synchronisé en sortie d'antennes. Ces aspects "numérisation rapide" et "synchronisation" sont d'une importance capitale pour les prochains radiotélescopes du futur. Ils permettent de simplifier les chaînes de réception radiofréquence et de diminuer la consommation électrique ainsi que les coûts d'entretien et de la maintenance. L'application à l'observation du soleil comporte cependant des contraintes originales, comme la grande dynamique des signaux, qui ne sont pas prises en compte actuellement dans les études en cours pour les radiotélescopes du futur. Le radiohéliographe actuel a une chaîne de réception analogique avec une numérisation centralisée. La commutation entre les différentes fréquences dans la bande 150-450 MHz est réalisée d'une façon analogique et temporelle. Ceci nécessite beaucoup de calibrations analogiques et oblige de figer la gamme des fréquences (10 fréquences avec une largeur de bande de 1 MHz). De plus, en interférométrie métrique, les très grandes longueurs de câbles coaxiaux sont onéreuses. Les signaux transmis des antennes au récepteur sont toujours sources d'erreurs et des fluctuations importantes réduisent l'information radiofréquence. Apporter une numérisation complète de la bande (300 MHz) permet d'avoir de la souplesse dans le traitement et l'analyse des données (résolution fréquentielle et la possibilité d'observer plusieurs bandes simultanément, traitement des parasites). Ceci engendre la nécessité d'avoir une très grande précision des horloges (0,7 ps d'erreur de phase) pour cadencer des ADC (Analog-to-Digital-Converter) à large bande (de fréquence 1 GHz d'horloge). L'objectif principal de la thèse est d'étudier la synchronisation pour l'application à un réseau d'antennes multi-réparties. Les résultats et les concepts étudiés ont d'autres applications dans les grands projets européens et internationaux.

**Mots clés :** Radiohéliographe, Numérisation rapide, Synchronisation d'horloge, Antennes multi-réparties, Architecture numérique, Observation du soleil

# Abstract

The Nançay Radioheliograph is the only instrument dedicated to imaging of the solar corona imaging in the 150-450 MHz frequency band. It operates on the principle of interferometry, using 47 antennas essentially distributed on east-west (3.2 km) and north-south (2.5 km) axes. This study aims to explore a new technical concept for future radio astronomy, applied to a solar interferometer. It deals with the rapid digitization of a synchronized system at the antennas sides. High speed digitization and high accuracy synchronization are most important aspects for future radio telescopes. They make it possible to simplify radiofrequency reception chains and reduce their power consumption, as well as maintenance costs and complexity. The application to the observation of the sun, however, has some original constraints, such as the great dynamics of the signals, which are not taken into account in the current studies for future radio telescopes. The current radio telescope has an analog receiver with a centralized digitization. The switching time between each frequency (10 frequencies with 1 MHz bandwidth) in the 150-450 MHz band introduce latency in solar images processing, and also decrease the signal-to-noise ratio. In addition, in metric interferometry, the long coaxial cables in which the signal is transported from the antennas to the receiver always cause significant errors and fluctuations in the radiofrequency reception chains. Providing full digitization of the band (300 MHz) allows more flexibility in data processing and analyzing (frequency resolution and the ability to observe multiple bands simultaneously). This required high clock accuracy (0.7 ps of jitter) for ADCs clocks (1 GHz clock frequency). Therefore, the main objective of this thesis is to reach a sub-ns global time synchronization of distributed networks as the Nançay Radioheliograph. The results and the concepts studied here can be of use in major European and international projects.

**Key words :** Radioheliograph, High speed digitizer, Clock synchronization, Multi-Distributed Antennas, Digital Architecture, Solar observation



# Remerciements

Tout d'abord, j'adresse mes remerciements les plus chaleureux à Monsieur Klein Karl-Ludwig, mon directeur de thèse, astrophysicien pour sa disponibilité et ses qualités d'encadrement. Il m'a beaucoup aidé dans les aspects de la physique solaire. Son écoute et ses conseils m'ont permis de cibler ma recherche. Son expérience dans la physique solaire et sa rigueur m'ont été d'un grand soutien tout au long de ces trois années.

Je tiens à remercier vivement mon encadrant technique Monsieur Bruno Da Silva, ingénieur de recherche en microélectronique, pour tous ses efforts, le temps passé ensemble et le partage de son expertise au quotidien. Grâce à sa confiance j'ai pu accomplir totalement mon travail de recherche. Il fut d'une aide précieuse dans les moments les plus délicats.

Je suis très reconnaissant aux Messieurs Stéphane Bosse, Alain Kerdraon, Christian Fabrice et Gabriel Auxepaules pour leur soutien au long de la thèse et durant la récolte des données de puissances liées aux antennes du radiohéliographe. Je remercie également toute l'équipe électronique et en particulier Severin Barth pour leur accueil, leur esprit d'équipe. Ils m'ont beaucoup aidé durant la phase de conception des cartes.

Je suis très reconnaissant aux Messieurs Cédric Viou et Guy Kenfack, ingénieurs de recherche en électronique numérique et traitement du signal sur FPGA pour leur aide précieuse et leur disponibilité pour m'éclairer sur les problèmes liés au développement et au synthèse du VHDL sur FPGA.

Je remercie aussi l'ESEP (Exploration Spatiale des Environnements Planétaires) pour le financement de la thèse.

Ce travail ayant été réalisé à la station de Radioastronomie de Nançay, je tiens à remercier l'ensemble du personnel de la station pour leur accueil chaleureux. Merci aussi au directeur du LESIA (Laboratoire d'études spatiales et d'instrumentation en astrophysique), à Monsieur Pierre Drossart pour son soutien dans la solution des problèmes administratifs.



# Introduction générale

Pour l'observation du ciel, des instruments d'observations dans le spectre visible de plus en plus performants [1, 2] ont été progressivement mis en œuvre. Cependant ce n'est que relativement récemment qu'on s'est aperçu que les astres rayonnaient aussi dans d'autres gammes de longueur d'onde que le visible.

L'astronomie ou l'astrophysique sont des sciences d'observation et non expérimentales. Nous ne pouvons pas fabriquer des étoiles ou des galaxies dans le laboratoire. Cependant, l'astronomie est étroitement liée à la technologie des détecteurs. En effet, en 1933, Karl Jansky avait construit une antenne, pour étudier l'origine des parasites paraissant lors d'une communication. Il a mis en évidence la présence d'un rayonnement permanent en onde radio. Le centre de notre galaxie fut alors la première source cosmique à être détectée grâce à son activité radioélectrique.

Dés lors, mais surtout après la seconde guerre mondiale, de nombreux autres types de radio-sources ont été détectées, des plus proches, nos voisins du système solaire, aux plus éloignées. Du sol, seulement une partie de la fenêtre radio (environ 1 mm à 30 m de longueur d'onde) [3] nous permet d'écouter l'univers.

Le soleil est l'étoile la plus proche et une source radio intense. Il émet un fort rayonnement thermique même en étant calme. Il existe de plus une émission radio intense provoquant des sursauts associés à des phénomènes d'activité solaire et les éjections de masse coronale. Les phénomènes du soleil actif peuvent entraîner des perturbations dans les systèmes de télécommunication. Ses émissions sont caractérisées par leur spectre large allant de quelques MHz aux dizaines de GHz. Le radiohéliographe de Nançay est un instrument dédié à l'observation de la couronne solaire dans une bande comprise entre 150 MHz et 450 MHz.

Les travaux exposés dans ce mémoire s'inscrivent dans le cadre d'étude et de conception d'un nouvel instrument dédié à l'observation solaire. Le premier chapitre présente les concepts fondamentaux liés aux observations en radioastronomie, en particulier l'observation du soleil par le radiohéliographe de Nançay. Les besoins d'explorer un nouveau concept technique propre à la radioastronomie du futur et appliqué à l'interférométrie solaire y sont formulés.

Le deuxième chapitre est consacré à une étude comparative des différents instruments dédiés à la radioastronomie et leurs caractéristiques. En particulier, nous décrivons la composition du numériseur et les techniques utilisées pour la distribution des horloges et leurs systèmes de synchronisation.

La plupart des instruments radio utilisent des références de temps ou de fréquence précises pour la datation et la synchronisation des données. Le chapitre trois sera consacré à l'instrumentation des horloges ainsi qu'aux divers protocoles de synchronisation. Plusieurs techniques seront introduites pour la mesure d'intervalles de temps avec une grande précision allant de la microseconde à la sub-nanoseconde.

Le chapitre quatre est consacré dans un premier temps à l'étude des spécifications du numériseur du nouvel instrument, dans une bande 150 - 450 MHz, pour le calcul de la résolution binaire du convertisseur A/D. L'étude est effectuée en se basant sur les mesures de l'instrument NRH actuel. La deuxième partie de ce chapitre sera consacrée au calcul de la précision temporelle du système de synchronisation avec le principe de l'interférométrie radio.

Une étude théorique de l'algorithme proposé dédié au calcul et l'ajustement des offsets des horloges sera analysé et validé par simulation dans le chapitre V. Le chapitre se termine avec l'étude et des simulations d'une technique de calibration des retards, basée sur l'estimateur gaussien.

Le dernier et sixième chapitre décrira une implémentation matérielle du système de synchronisation, avec la description de diverses unités constituant la chaîne. Nous terminerons ce chapitre par des résultats des tests et mesures obtenus dans la présente recherche.

La conclusion développera la portée ainsi que les perspectives de ce travail.

# Première partie

## Contexte et objectifs de la thèse



# La radioastronomie

## 1.1 Introduction

L'astrophysique est une science qui permet de découvrir l'univers et sa composition (planètes, galaxies, étoiles, etc.). D'énormes progrès ont été réalisés dans le dernier siècle en matière d'observation de l'univers grâce à l'évolution des instruments au sol et les sondes spatiales. La grande partie du spectre est accessible au sol en exploitant deux techniques : L'astronomie optique (longueurs d'ondes comprises entre 390 nm et 710 nm) et la radioastronomie (entre 1 mm et 20 m) [4, 5]. On distingue deux méthodes d'imagerie dans le domaine radio. La première est la cartographie par balayage de la source avec un seul radiotélescope [6]. La seconde consiste à effectuer l'imagerie en utilisant un réseau d'antennes multi-reparties au sol, ce principe s'appelle l'interférométrie radio [7, 8], comme dans le cas du radiohéliographe de Nançay. Dans la suite du chapitre, on présente les caractéristiques des divers types d'émission radio solaire, puis on aborde le fonctionnement de l'interféromètre radio de Nançay dédié à l'observation de la couronne solaire et ses limitations. On termine le chapitre avec la présentation des performances et limitations actuels afin de définir l'importance d'étudier une nouvelle génération d'instruments dotés d'une numérisation large bande distribuée pour la physique solaire.

## 1.2 Observations en Radioastronomie

Les émissions en radioastronomie sont divisées en deux grandes parties en fonction de la largeur de bande occupée. Les émissions en bande étroite (émissions de raies) et le continuum (large bande). Les émissions de type continuum peuvent être d'origine thermique ou non. Les sursauts solaires (I à IV) [9] constituent un exemple de ce type d'émission. Les sursauts sont dus à l'accélération des électrons. Chaque sursaut est caractérisé par le spectre occupé en fréquence et sa durée. En effet, le sursaut de type I occupe une bande de quelques MHz, il dure entre une et quelques secondes. Cependant, le sursaut de type IV s'étale sur une bande de plusieurs centaines de MHz et peut durer plusieurs heures [10]. La raie de l'hydrogène neutre de fréquence égale à 1,42 GHz (21 cm) a été découverte en 1944 par Hendrik van de Hulst [11, 12] et [13]. Le rayonnement synchrotron et le rayonnement de freinage constituent l'exemple des émissions large bande (continuum). Le tableau 1.2 résume une liste non exhaustive des sources et le type d'émission engendrée.

TABLE 1.1 – Source et le type d'émission engendrée

Sources	Type d'émission
Galaxie	Continuum + Raies [14, 15]
Gaz interstellaire	Continuum + Raies [16, 17]
Soleil	Continuum [18]
Quasar	Continuum [19]

### 1.3 Observation du soleil par le Radiohéliographe de Nançay (NRH)

Le Radiohéliographe de Nançay [20] est un interféromètre radio pour la détection des émissions du soleil (voir le tableau 1.1). L'instrument NRH est composé d'un réseau de 47 antennes radio réparties en deux branches perpendiculaires formant un "T" : 19 antennes en Est-Ouest (EW) réparties sur 3200 m, et 25 antennes paraboliques réparties en Nord-Sud (NS) sur 2440 m et 4 antennes d'extension du réseau avec une antenne commune aux réseaux EW et NS. La figure 1.1 montre les images radio du rayonnement freinage de la couronne solaire calme.



FIGURE 1.1 – Observation des images radio de la couronne solaire le 24 février 2006 des fréquences (de gauche à droite) 150,9 MHz, 228 MHz, 327 MHz, 408 MHz et 432 MHz

L'observation de la couronne solaire à différentes fréquences permet d'obtenir des informations sur la température du soleil, la densité des électrons et sur son activité. D'après l'étude récente développée par Mercier et al. en 2015 [21], on peut calculer la densité d'électrons et la température de brillance de la couronne solaire grâce à des observations multifréquences. La figure 1.2 montre les images radio 2D de la couronne solaire obtenues par la synthèse rotationnelle à 410 MHz et 150 MHz observées le 27 mai 2004 par le radiohéliographe de Nançay. Les deux images montrent le gaz chaud de la couronne calme. Elles montrent aussi que l'aspect de la couronne varie en fonction de la fréquence. La figure 1.3 illustre le résultat obtenu par [21] pour la mesure de la température de brillance du soleil  $T_b$  en fonction de la fréquence en se basant sur les observations de dix fréquences obtenues par le NRH. La température  $T_b$  décroît en fonction de la fréquence avec une légère variation par rapport au cycle solaire. La figure 1.4 montre l'estimation de la densité d'électrons en fonction de la distance au centre du soleil à partir de ces observations multi-fréquences. Les deux résultats obtenus sont basés sur un nombre fixe de fréquences (neuf fréquences au maximum). Par conséquent, la précision d'estimation des deux

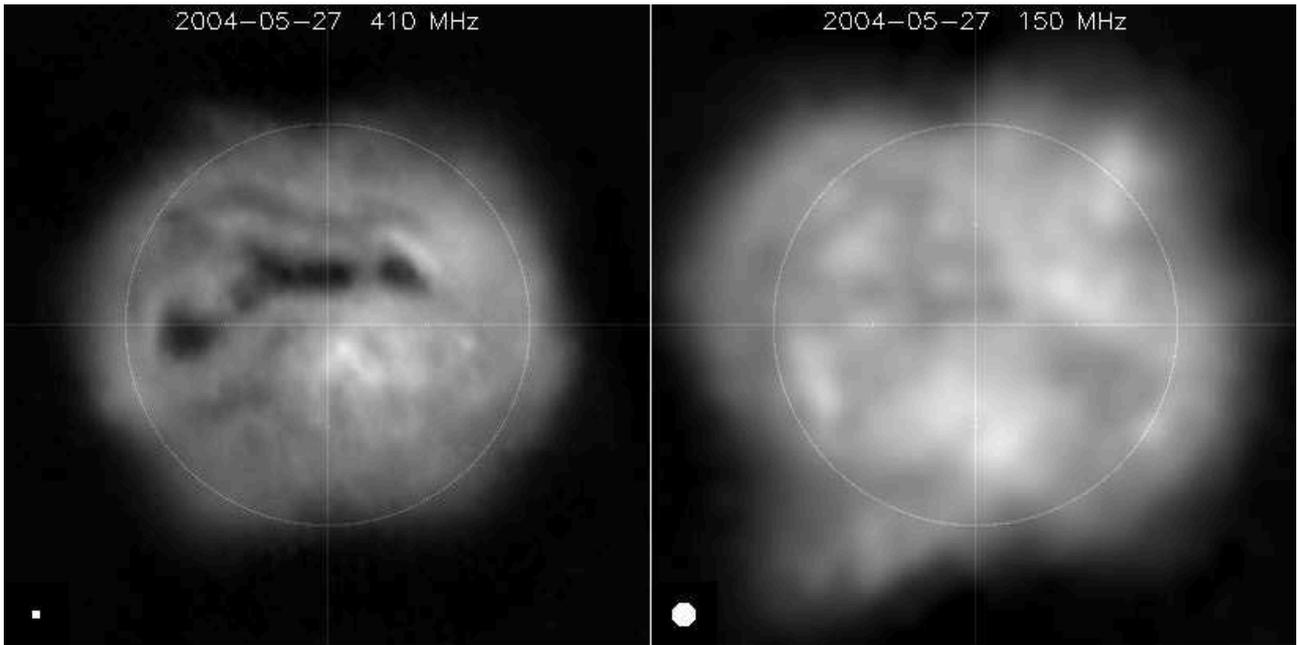


FIGURE 1.2 – Images de la couronne solaire au début du cycle solaire pour les deux fréquences 410 MHz et et 150 MHz observées le 27 mai 2004 [21].

paramètres est réduite (voir la figure 1.3 et 1.4). Cependant dans le cas d’une observation large bande, avec une analyse multi-fréquences, on pourrait mieux déterminer la forme du spectre. On aura une meilleure estimation de la température de brillance  $T_b$  en fonction de la fréquence.

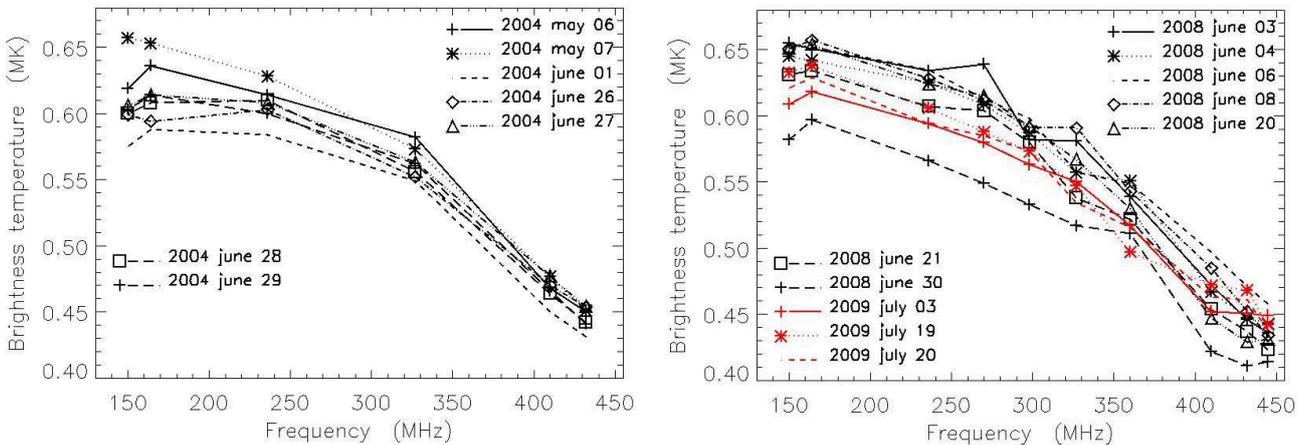


FIGURE 1.3 – La température de brillance du soleil  $T_b$  (MK) en fonction de la fréquence (MHz). A gauche : 7 jours d’observations en 2004 après la maximum du cycle. A droite : 10 jours d’observations en 2008-2009 durant le minimum du cycle solaire [21]

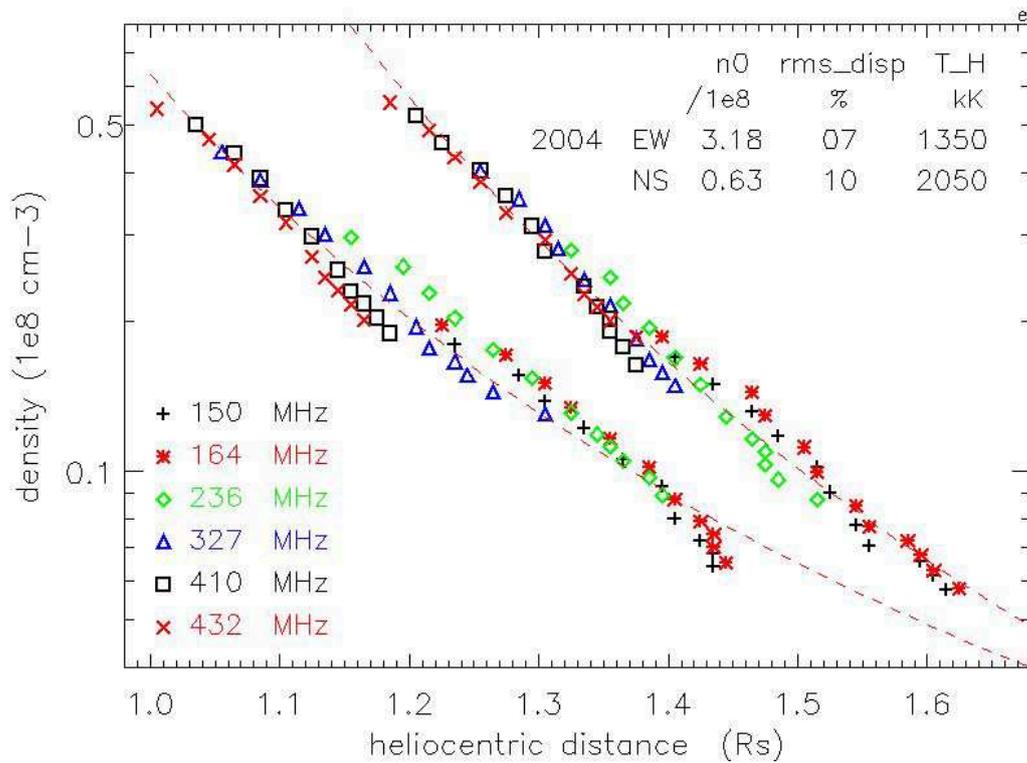


FIGURE 1.4 – La densité d’électron en fonction de la distance au centre du soleil pour les 6 fréquences (comprises entre 150 MHz et 432 MHz) dans les deux directions EW (Est-Ouest) et NS ( Nord-Sud) [21]

## 1.4 Importance de l'observation large bande du soleil

Le Radiohéliographe de Nançay permet d'observer différents types de sursauts solaires [22]. La figure 1.5 montre les sources radio superposées au magnétogramme de la photosphère pour les fréquences 164 MHz, 237 MHz et 327 MHz. Ces sources sont le lieu d'émission, à une densité ambiante donnée, de faisceaux d'électrons. La figure 1.5 montre les faisceaux d'électrons guidés par les lignes de champ magnétique de la basse vers la haute couronne. Ces émissions radio sont près de la fréquence plasma, proportionnelle à la racine de la densité électronique. On utilise l'identification des sources radio pour tracer le chemin des électrons qui traversent la couronne [22]. L'observation à plus de fréquences nous permettra de tracer les faisceaux d'électrons plus finement et donc d'avoir une meilleure description des lignes de champ magnétique. Un exemple est le sursaut de type III du 19/04/2000 qui n'est observé qu'à une seule fréquence du NRH.

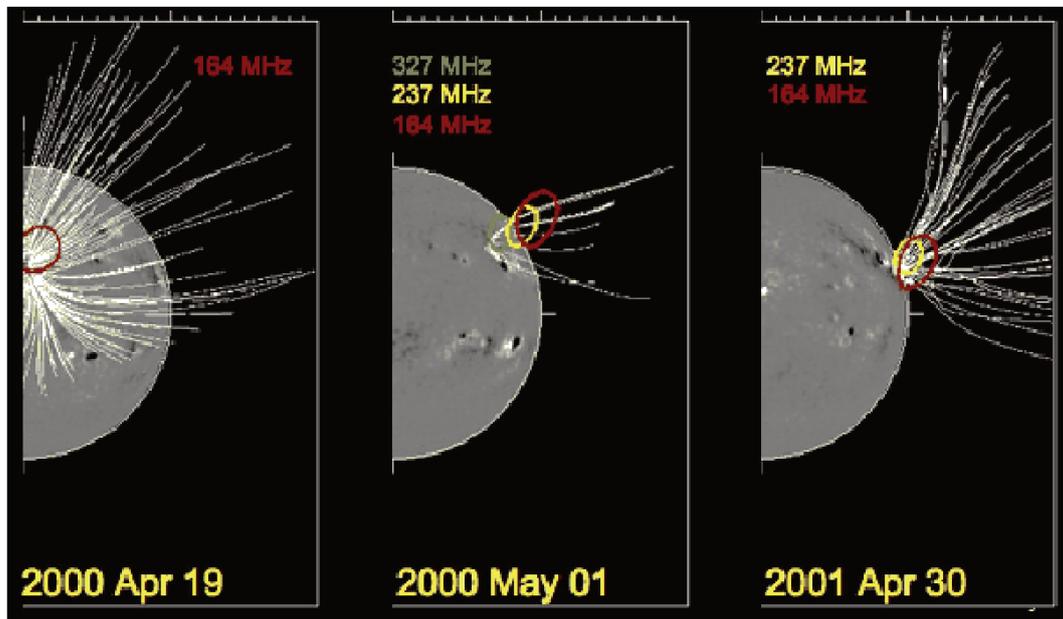


FIGURE 1.5 – Contours d'iso-intensité à mi-puissance (iso-contours colorés) de quelques sursauts solaires de type III observés par le Radiohéliographe de Nançay superposés aux magnétogrammes de la photosphère et à l'extrapolation dans la couronne des lignes de champ ouvertes vers l'espace interplanétaire (courbes en blanc).

La figure 1.6 illustre un exemple de spectre dynamique complexe observé le 3/11/2003, entre 200 et 400 MHz avec un sursaut de type IV, à bande large, et des bandes avec une structuration forte en fonction de la fréquence. Ces bandes font partie d'un sursaut de type II, elles ne sont pas accessibles à l'imagerie actuelle qu'avec une observation à 9 fréquences discrètes. Seulement certaines de ces structures spectrales peuvent être cartographiées par un instrument observant à fréquences fixes. L'imagerie sur l'ensemble de la bande spectrale augmenterait donc beaucoup le potentiel scientifique d'un imageur.

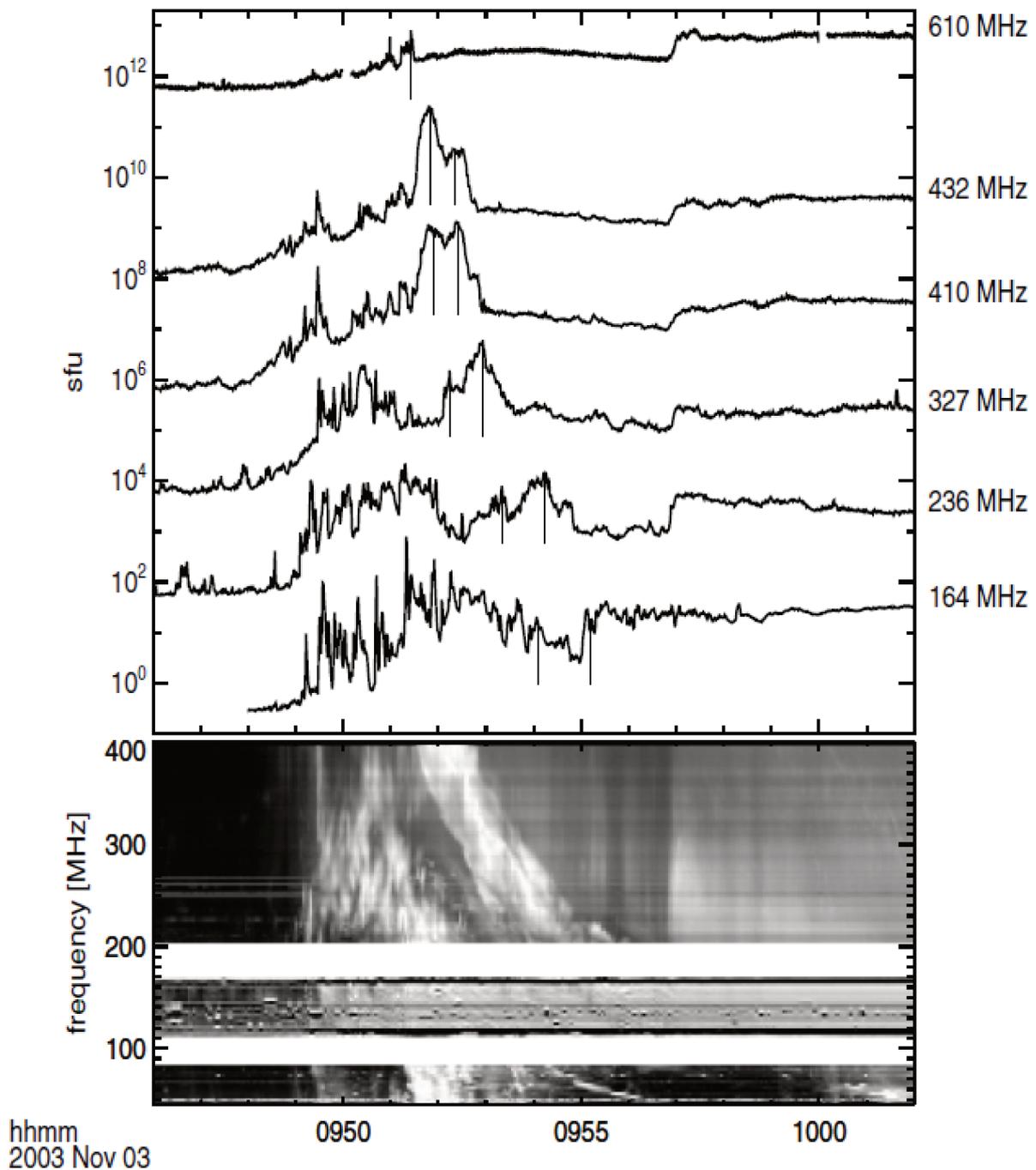


FIGURE 1.6 – Observations des sursauts de types II, III et IV le 3 novembre 2003 aux fréquences discrètes 432 MHz, 410 MHz, 327 MHz, 236 MHz et 164 MHz obtenues par le NRH (en haut) et les observations de spectrographie du soleil entier par OSRA (AIP Potsdam)(en bas)

## 1.5 Limites instrumentales du NRH actuel

La chaîne radiofréquence au pied de chaque antenne est constituée d'un ensemble de composants discrets (filtres, amplificateurs, etc.) (voir la figure 1.7). Le récepteur dispose de 4 entrées RF (entrées HF et BF pour chaque filtre) pour chaque polarisation. Les bandes BF (150-260 MHz) et HF (255-450 MHz) sont séparées en entrée. Suit un filtre sélectif de rejection du Canal+. Les signaux sont amplifiés avant modulation. Un banc de filtres et de commutateurs (C3V et C5V) permet de sélectionner une fréquence parmi dix en fonction de la fréquence d'observation désirée dans la bande 150 MHz-450 MHz. Deux oscillateurs locaux ramènent la fréquence d'observation à 10,7 MHz avec une bande de 1,4 MHz pour la numérisation. La 1<sup>re</sup> limite du système actuel est que le temps de commutation entre deux fréquences d'observation introduit une latence dans les images solaires, il réduit également le rapport signal sur bruit des données d'observations, ce bruit est dû à l'ensemble des composants discrets (amplificateurs, atténuateurs, commutateurs, filtres, etc.). D'autre part, le RH observe 10 fréquences successivement, à raison de 5 ms pour une fréquence, qui sont ensuite moyennées sur plusieurs acquisitions. Ces traitements engendrent des trous dans les observations.

Chaque voie pour chaque antenne est modulée par une fonction de Walsh qui lui est propre pour les deux polarisations. Pour récupérer les deux signaux au niveau du démodulateur, il faut une synchronisation parfaite entre la fonction de modulation et la fonction de démodulation. En effet, une fonction est intégrée au niveau numériseur pour la compensation des retards de démodulation avec une précision de 23 ns. Le système actuel utilise des fonctions qui peuvent avoir 128 transitions sur une période d'intégration de 5 ms. Le nombre de transitions est différent entre les deux antennes. La 2<sup>me</sup> limite est le problème de synchronisation des fonctions de modulation et de démodulation. En effet, le système actuel utilise une technique de compensation manuelle pour la mesure des écarts d'ajustement de ces dernières. Le décalage des fonctions peut réduire le taux de corrélation entre les voies. D'autre part, le rapport des transitions des fonctions de Walsh entre les antennes influence les gains en puissance reçues par chaque antenne, d'où la nécessité de calibrer les antennes en puissance.

Les signaux Radio Fréquence (RF) (sortie de l'amplificateur MF113, voir la figure 1.7) sont transportés dans des câbles coaxiaux avec des longueurs différentes (50 m à 3,2 km) au numériseur. La différence de distance entre les antennes et le numériseur induit des erreurs dues à la compensation manuelle des retards d'offsets. D'où l'importance d'effectuer une numérisation aux pieds des antennes de l'ensemble de la bande analogique observée.

La radioastronomie dispose de quelques bandes protégées. Le nombre de fréquences occupées par des émetteurs terrestres ne fait qu'augmenter, induisant une croissance des observations polluées. En effet, le parasite [23] peut être dans la bande utile de l'instrument, comme le cas de Canal+ (Bande III, canal 8, 200 - 206 MHz, 5 MHz) du radiohéliographe de Nançay. Le système actuel est muni d'un filtre sélectif pour le filtrage du canal. Un parasite peut être en limite de la bande ou une partie de l'information contenue dans les harmoniques (lobes secondaires). Cela concerne les émetteurs à spectre étalé comme les GPS ou les émetteurs à fortes puissances dont les harmoniques générées par le modulateur ou la non linéarité de l'émetteur ne sont pas suffisamment filtrés. Donc, il sera judicieux de pouvoir choisir les fréquences utiles après l'observation, grâce à l'acquisition des données large bande afin d'identifier et éliminer le

ou les parasites.

Le corrélateur utilisé par le système actuel du NRH est sur 1 bit (deux niveaux), avec une fréquence d'échantillonnage de 42,8 MHz. Elle correspond à  $4 \times B$  avec  $B$  la bande analogique numérisée égale à 10,7 MHz. Par conséquent, le facteur de dégradation est réduit à 1,35 (perte de 35% de taux de corrélation). La numérisation avec une résolution binaire supérieure à 3 bits induit une amélioration importante du taux de corrélation (perte inférieure à 5%) [24].

## 1.6 Objectifs de la thèse

L'instrument actuel est contraint en termes du nombre de fréquences observées (dix fréquences dans la bande), de la simultanéité des observations et du bruit additif des divers composants discrets du récepteur, etc. Cette étude a pour objectif d'explorer un nouveau concept technique, appliqué à l'interférométrie solaire. Elle porte sur la faisabilité du système de la numérisation large bande et la synchronisation dédiée en sortie d'antennes. Deux aspects seront approfondis dans cette thèse d'une importance capitale pour les radiotélescopes du futur : la numérisation rapide et la synchronisation dédiée.

L'application à l'observation du soleil comporte cependant des contraintes originales, comme la grande dynamique des signaux dans le cas des sursauts [25] solaires, qui ne sont pas prises en compte dans les études en cours pour les radiotélescopes futurs. Le développement du système permettra de prendre en considération les effets de l'activité solaire sur le numériseur. La finalité de la thèse est d'apporter une numérisation complète de la bande comprise entre 150 MHz et 450 MHz, elle permettra d'avoir la souplesse dans le traitement et l'analyse des observations. Les chapitres suivants seront consacrés à l'étude de conception du nouveau système pour le NRH.

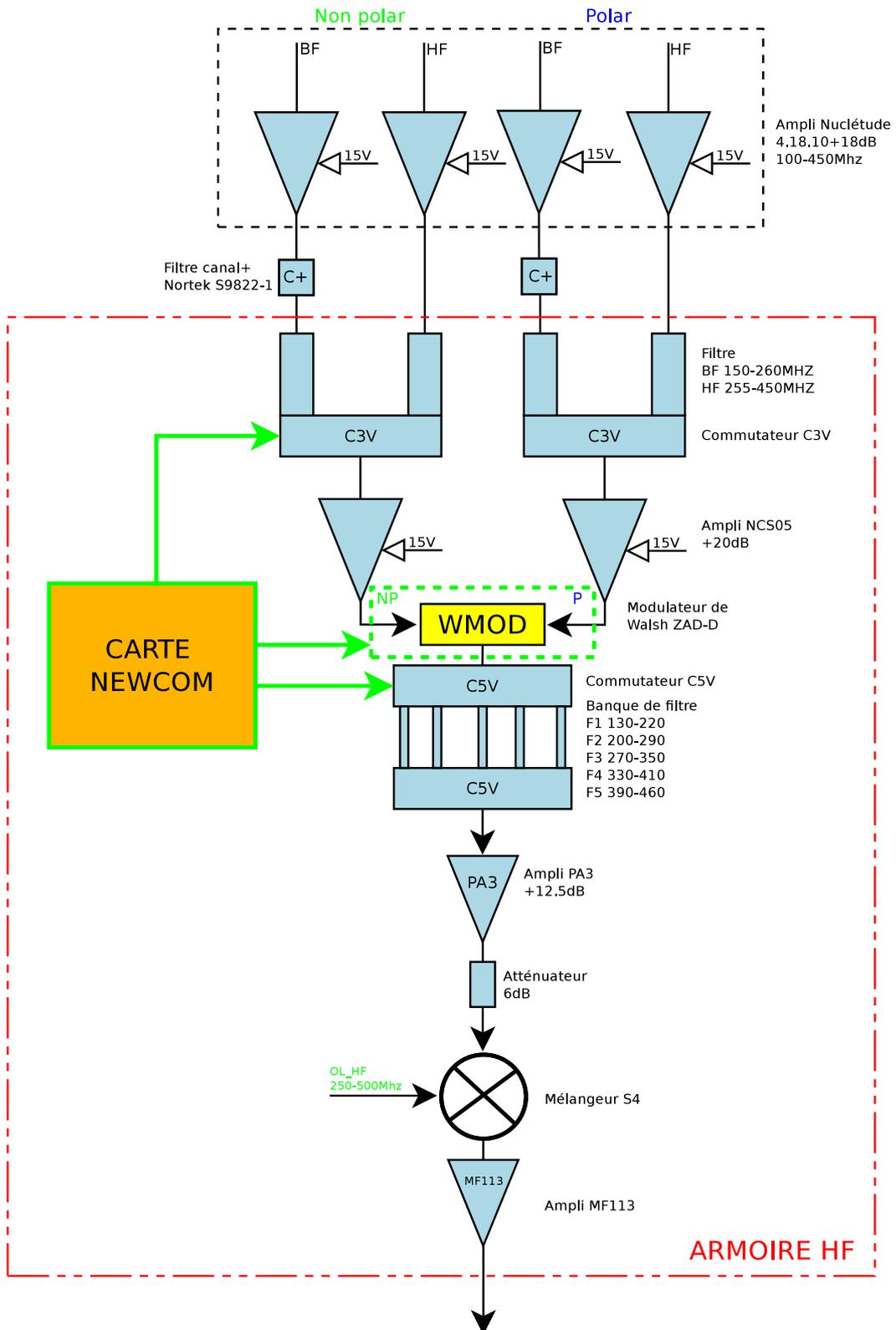


FIGURE 1.7 – Schéma bloc de la chaîne radiofréquence d’une antenne NS (Nord-Sud) du Radiohéliographe de Nançay



# Les radiotélescopes et imageurs radio

## 2.1 Introduction

Ce chapitre est consacré à une étude comparative des instruments radio basés sur des dispositifs à plusieurs antennes : interféromètres ou réseaux phasés. On abordera la comparaison des caractéristiques des interféromètres radio, des dispositifs de synchronisation et de leurs limitations. Le but principal est de déduire, à travers l'analyse des instruments existants, les paramètres principaux qui vont nous permettre d'orienter notre étude pour la conception d'une numérisation large bande au pied des antennes de l'interféromètre NRH. Ci-dessous une liste non exhaustive des principales caractéristiques :

- la bande de base et la bande observée
- le type d'horloge
- la stratégie et précision de synchronisation
- les caractéristiques du numériseur

### 2.1.1 LOFAR

#### 2.1.1.1 Introduction

LOW Frequency ARray (LOFAR) est un instrument radio généraliste permettant d'observer entre autres des pulsars, des exoplanètes, les galaxies, amas de galaxies et des rayonnements cosmiques [26–28] et de cartographier le champ magnétique de Jupiter [29,30]. Cet interféromètre radio international est constitué d'un réseau d'antennes multi-reparties, dont la combinaison des signaux reçus permet de reconstituer les images radio du ciel basse fréquence dans la bande 30-250 MHz (voir la figure 2.1). En effet, il dispose d'une multitude de "stations" distantes. Chacune d'entre elles dispose d'un ensemble d'antennes élémentaires, l'ensemble est regroupé pour former une grande antenne. L'instrument LOFAR est constituée de :

- Un réseau de 96 antennes Basse Fréquence (BF), dans la gamme de fréquence 30-80 MHz (avec une résolution fréquentielle d'environ 10 MHz), chaque antenne est composée de 2 dipôles croisés, afin de détecter deux polarisations linéaires orthogonales des ondes reçues.

- Un réseau de 1536 antennes Haute Fréquence (HF) dans la gamme de fréquence (110-250 MHz) (voir la figure 2.1), soit 96 groupes de 16 dipôles croisés de grandeur petits par rapports au réseau BF.
- Un ensemble de 96 paires de récepteurs numériques (deux polarisations), disposant de convertisseurs A/D à 200 Msps pour construire un ou plusieurs faisceaux simultanés sur le ciel.
- Une liaison Ethernet par fibre optique haut débit (3 Gbps) (figure 2.1), destinée à transporter le signal numérique de chaque station à Groningen (Pays-Bas), où se trouve le système central qui va corrélérer les signaux pour chaque couple d'antennes et construire les images radio.

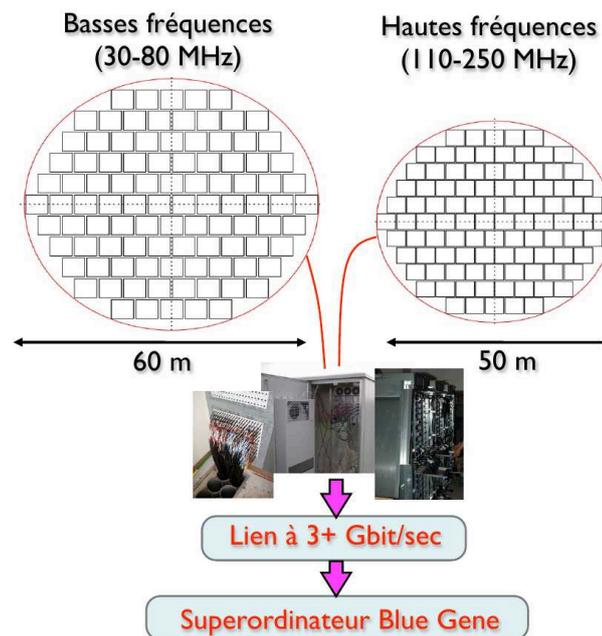


FIGURE 2.1 – Architecture globale du réseau LOFAR : BF (30-80 MHz) d'un diamètre de 60m et HF (110-250 MHz) avec d'un diamètre de 50 m [31].

Après l'acquisition de données d'observation par le système central, elles seront enregistrées sur des disques durs de l'institut ASTRON (Netherlands Institute for Radio Astronomy) au Pays-Bas [32], où seront construites les images radio et les paramètres d'observation pour l'étude et l'analyse scientifique.

### 2.1.1.2 Caractéristiques techniques de LOFAR

L'instrument LOFAR utilise un convertisseur A/D AD9430 [33] de résolution 12 bits, 65 dB du SNR (Signal to Noise Ratio) et 250 ps de gigue. Le numériseur est conçu par ASTRON [34–36]. En radioastronomie, la résolution binaire du convertisseur A/D est majorée par ( $n_s$ ) 1, 2 bits ou plus supplémentaires pour prendre en considération l'effet du bruit du ciel [37]. La localisation des stations dans une portée supérieure à 400 km nécessite une synchronisation des horloges distantes entre les stations (des antennes constituant une station) et globale (entre

les stations). La première synchronisation est réalisée par l'intermédiaire d'un circuit de distribution d'horloge précise basée sur le Rubidium refroidi assisté par le GPS qui peut atteindre des stabilités relatives de fréquence variant entre  $10^{-13}$  et  $10^{-14}$  (voir le chapitre suivant). La liaison filaire (câbles coaxiaux) liant la référence et chaque antenne pour une station donnée, est supposée équidistante (utilisation des câbles de la même longueur pour l'aller et le retour pour chaque antenne). La synchronisation globale entre stations est gérée par le système central en se basant sur les données datées par chaque station distante. Le système central aligne les données issues de chaque station via des fibres optiques haut débit (20 Tb/s) (voir la figure 2.1) avant d'affecter la corrélation numérique en utilisant des calculateurs embarqués à base de CPU/GPU. Le tableau 2.1 résume les spécifications techniques de LOFAR.

TABLE 2.1 – Résumé des caractéristiques techniques de l'instrument LOFAR

La bande passante (MHz)	30-250
La fréquence d'échantillonnage $F_s$ (MHz)	200-160
Caractéristiques du convertisseur A/D	Référence : AD9430 Résolution : 12 bits SNR = 65 dB @ $F_{in} = 70$ MHz @ 210 MSPS Jitter : 0,25 ps (RMS)
Transmission	Par fibre optique à 20 Tb/s entre les stations lointaines
Portée maximale	400 km
Horloge physique	Rubidium + GPS
Synchronisation	Fréquence de référence (10 MHz) en sortie du Rubidium. La fréquence d'échantillonnage de 200 MHz est synchronisée sur 10 MHz La synchronisation est centralisée dans un container maintenu à température constante

Le récepteur LOFAR peut être configuré parmi les quatre modes d'utilisation (voir le tableau 2.2) [38, 39] en fonction de la bande observée. La bande numérisée varie ainsi que le gain du récepteur  $G_{rec}$  en fonction du mode. Les fréquences observées sont ramenées à la bande 40 MHz, 60 MHz ou 80 MHz en fonction du mode d'observation. Cette particularité diminue considérablement (un ratio de 6 entre la bande numérisée et la bande observée pour le Mode 4 (voir le tableau 2.2)) la quantité de données reçues par le système central et par conséquent réduit la complexité du récepteur numérique.

TABLE 2.2 – Les bandes de base utilisées par LOFAR

Bande	Mode	Fréquence(MHz)	BW(MHz)	$G_{rcr}$ (dB)
LB	1	10-90	80	22
HB	2	110-190	80	19
HB	3	170-230	60	24
HB	4	210-250	40	29

### 2.1.1.3 Architecture de distribution d'horloge

La figure 2.2 montre l'architecture globale du système d'acquisition de LOFAR. Elle est constituée de cinq couches. La couche basse est dédiée à la réception du signal RF issue des antennes. Le RCU (ReCeiver Unit) fait l'interface entre les antennes et le processeur de contrôle RSP (Remote Station Processor) [38]. L'unité RCU permet de sélectionner une bande parmi les trois bandes utiles (LBH, LBL ou HB) de LOFAR, ramène le signal à la bande de base puis numérise le signal RF. Les données numérisées sont ensuite transmises aux stations de contrôle RSP, ces dernières contiennent plusieurs circuits FPGA permettant de récupérer et mettre en paquet les divers canaux de données issues du RCU. Les données sont ensuite transmises par un réseau Ethernet à l'unité de contrôle local LCU (Local Control Unit).

### Caractéristiques de la carte RSP

La carte RSP reçoit 4 horloges de 200 MHz et 4 signaux PPS de la carte TDS (Timing Distribution Subrack) (figure 2.3), elle génère un code d'identification pour la trame de CEP (CEntral Processor) sur 10 bits pour chaque signal PPS (chaque canal). Le RSP dispose également d'un circuit de compensation des retards dus aux câbles avec une résolution d'une période d'horloge et une dynamique de 128 périodes d'horloge [40]. Pour une fréquence d'échantillonnage de 200 MHz, la résolution est de 5 ns pour une marge de correction 600 ns. On voit dans la figure 2.3 la présence d'un bloc de retards programmables avec une précision de 78 ps sur 4 bits qui sert à ajuster la phase du signal PPS avec une dynamique d'une période d'horloge (5 ns pour 200 MHz). La synthèse des signaux PPS est assurée en utilisant un diviseur d'horloge 1 : 5 pour chaque carte (4 cartes) avec un "skew" de l'ordre de 500 ps [41].

### Caractéristiques techniques de l'horloge de rubidium

Le circuit de distribution des horloges de LOFAR permet de distribuer une référence d'horloge stable et synchrone pour tous les convertisseurs A/D (en total 192) intégrés dans l'unité RCU. Une horloge Rubidium assisté par un GPS a été choisie pour sa stabilité et sa précision au long terme (voir le chapitre III). Le temps absolu doit être connu par l'unité de contrôle local LCU (Local Control Unit) via le réseau Ethernet (Figure 2.2, 2.4) pour pouvoir synchroniser les données.

Le générateur d'horloge de référence à base de rubidium est stabilisé par le signal PPS (Pulse Per Second) issu du récepteur GPS (Figure. 2.4). L'unité RSP et la carte TDS permettent de synchroniser et dater l'horloge de référence du système complet [42]. En utilisant une référence de 10 MHz à la sortie du GPS (figure 2.5) et une PLL combinée avec un VCXO (Voltage

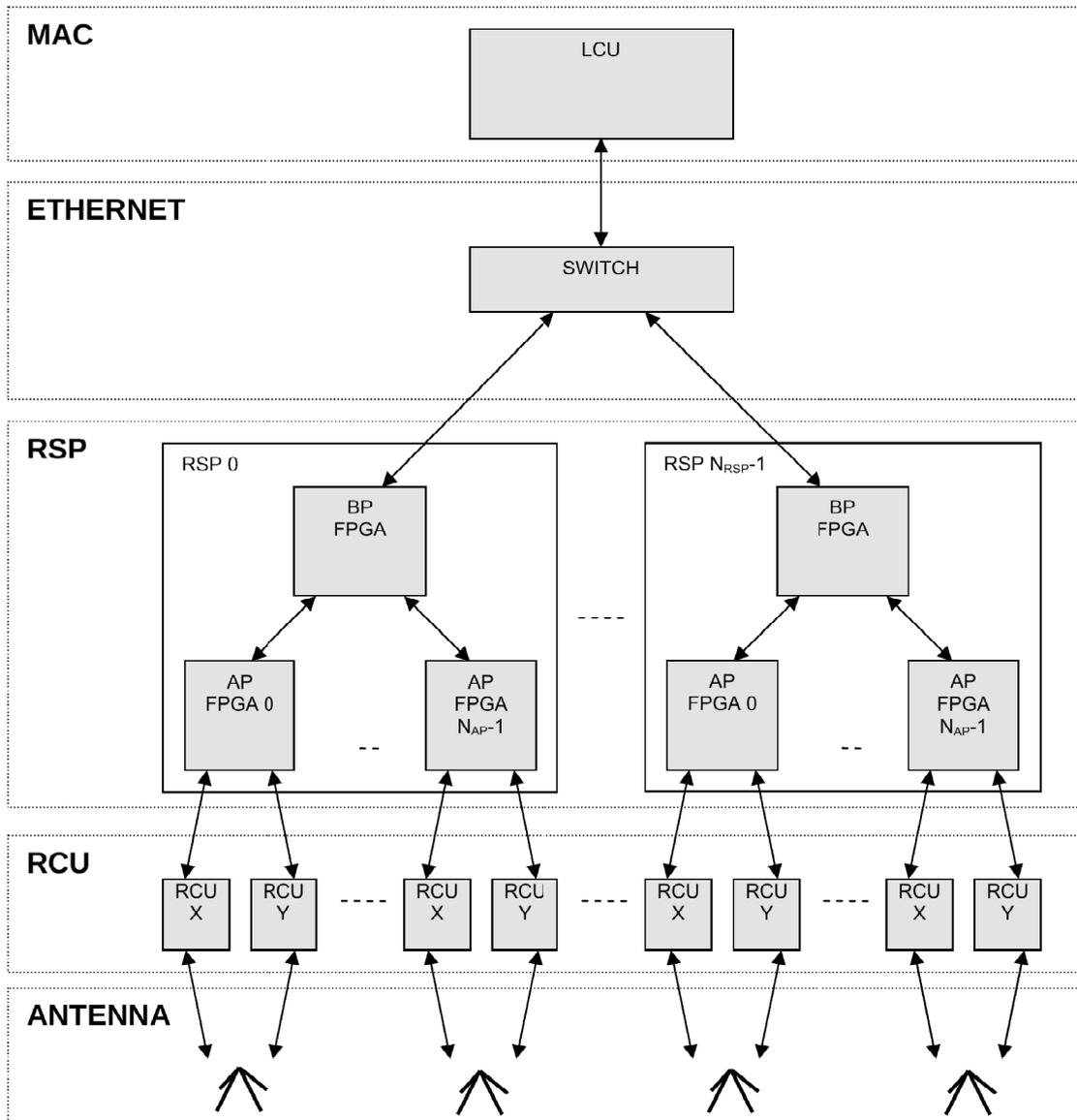


FIGURE 2.2 – Arborecence du système d’acquisition et du transfert de données de LOFAR

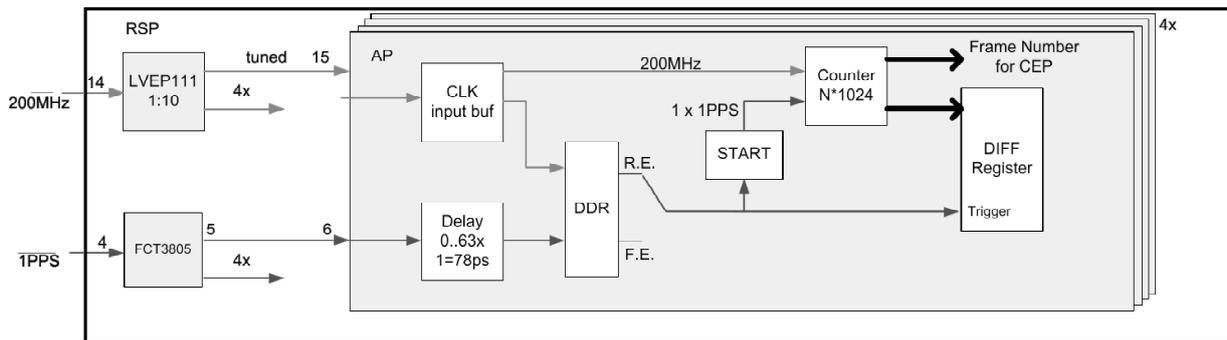


FIGURE 2.3 – Architecture interne de la carte RSP

TABLE 2.3 – Caractéristiques clés de la carte TDS

Caractéristiques	Valeurs
Entrée	3x1 PPS et 3x10 MHz
Sortie	4x1 PPS et 4x200 MHz
Jitter	Jitter < 1 ps (RMS)
Fréquence Maximale (MC100LVEP111)	< 3 GHz
Temps de propagation (MC100LVEP111)	430 ps

Controlled Crystal Oscillator), la gigue du signal de sortie est minimisée. Les sorties 10 MHz sont ensuite couplées avec la carte TDS (Timing Distribution Subrack) pour ensuite générer des horloges d'échantillonnage à 200 MHz avec un "skew" (différence d'offset maximal entre chaque couple d'horloges) de sortie inférieur à la nanoseconde. La transmission simultanée des horloges est assurée par des longueurs de câbles identiques. Le générateur [43] intègre un oscillateur basé sur un rubidium à faible bruit avec une alimentation AC. Il dispose de deux sorties à 10 MHz et une autre à 5 MHz avec un bruit de phase ultra faible (-130 dBc/Hz à 10 Hz) et une variance d'Allan inférieure à  $2 \times 10^{-11}$  [43]. Il peut être asservi avec un PPS de référence (Exemple : GPS). Le générateur est équipé également d'une liaison série pour le réglage et le monitoring de l'instrument.

### Caractéristiques de la carte TDS

L'unité TDS [44] permet de nettoyer l'horloge (gigue et skew), de générer une multitude d'horloges synchrones et d'augmenter la fréquence d'échantillonnage à 200 MHz en utilisant une PLL. La PLL est cadencée par une fréquence de référence de 10 MHz issue de l'horloge de rubidium.

#### 2.1.1.4 Conclusion

Nous avons abordé dans un premier temps les caractéristiques techniques de l'instrument LOFAR. La résolution binaire du convertisseur A/D est due l'ordre de 12 bits pour une bande passante de 80 MHz et une gigue de 0,25 ps. Ensuite, nous avons illustré les composants constituant le système de synchronisation et le circuit de distribution des horloges à 200 MHz. Le

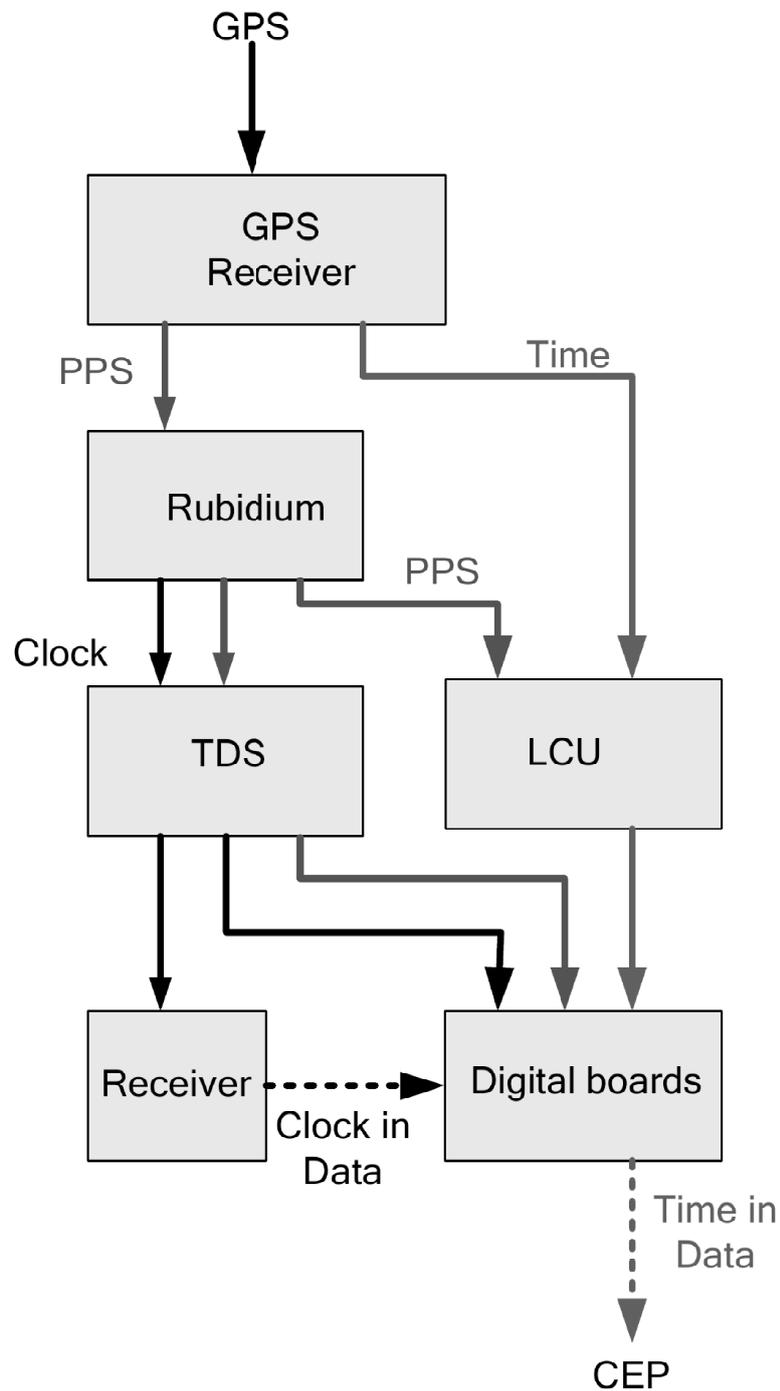


FIGURE 2.4 – Circuit de distribution d’horloge et de datation du récepteur LOFAR

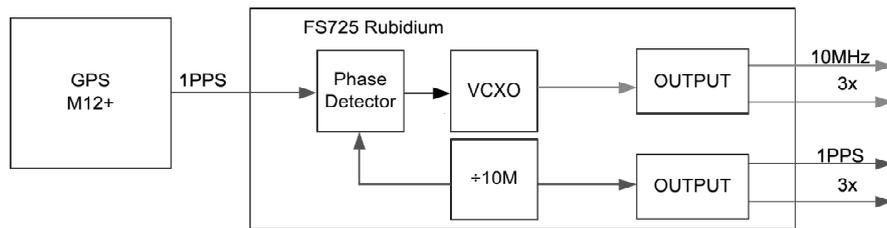


FIGURE 2.5 – Schéma bloc du générateur FS725 dédié de trois sorties à 10 MHz et trois signaux PPS asservi par une source GPS

système de compensation des offsets a une précision de 78 ps avec une dynamique de compensation de 5 ns (une période d'horloge du convertisseur A/D avec une fréquence de 200 MHz). Le circuit de distribution des horloges dispose d'un "skew" de l'ordre de 500 ps. Les liens physiques (câbles) du circuit de distribution sont supposés de même longueur pour éviter des différences de retards entre les horloges.

### 2.1.2 EMBRACE

EMBRACE (Electronic Multi-Beam Radio Astronomy ConcEpt) [45,46] est un démonstrateur technologique pour le concept du projet mondial SKA (Square Kilometre Array) [47–49] avec une superficie de 300 m<sup>2</sup> à Westerbork et 100 m<sup>2</sup> à Nançay (figure 2.4). Les objectifs principaux du projet EMBRACE sont de démontrer le potentiel technique et scientifique du concept des réseaux phasés en utilisant des réseaux à faible coût. La bande passante de l'instrument est comprise entre 400 MHz et 1500 MHz. Le tableau 2.4 illustre les autres caractéristiques techniques de l'instrument.

L'instrument EMBRACE est technologiquement avancé et compte parmi les radiotélescopes les plus complexes au monde. Doté de ses 4608 antennes, ce démonstrateur a permis de faire un grand pas en observant le pulsar B0329+54 [50] et en démontrant pour la première fois le fonctionnement d'EMBRACE dans son intégralité. Le rayonnement périodique du pulsar est ainsi capté par les nombreuses antennes pour être traité par une chaîne électronique. Les signaux des antennes sont combinés en phase pour maximiser la sensibilité dans deux directions différentes et indépendantes pour pouvoir effectuer deux observations simultanées de deux sources astronomiques très distantes dans le ciel. Ce phasage des signaux se fait dans un circuit intégré conçu à Nançay par l'équipe micro-électronique (le "beamformer"), et représente la fonction la plus importante du système EMBRACE. Cette combinaison est par la suite numérisée et traitée par un ordinateur. Il s'agit alors de synthétiser numériquement des centaines de directions de visée très fines, imbriquées dans chacun des deux champs d'observations simultanés de l'instrument.

La numérisation des signaux RF issus des antennes d'EMBRACE est centralisée. Les signaux RF sont transmis par des câbles coaxiaux de mêmes longueur entre les antennes et le système central. Le système de distribution des horloges aux oscillateurs locaux (voir la figure 2.7) utilise également des câbles de la même longueur avec le même système d'acquisition que LOFAR. Le système dans sa globalité suppose des liens identiques pour toutes les voies RF. Le démonstrateur utilise une source à base de rubidium comme fréquence de référence.

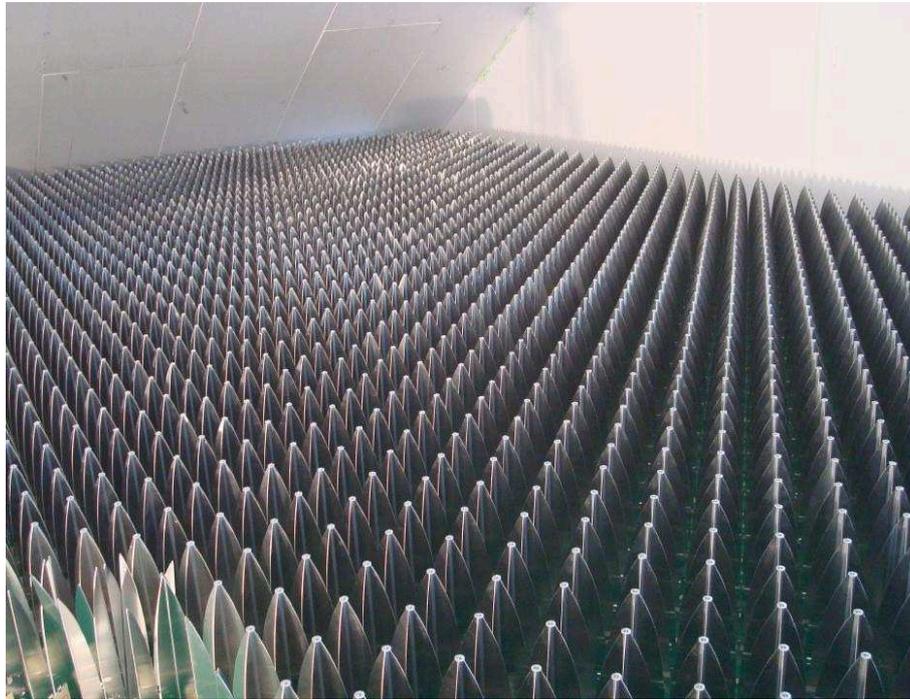


FIGURE 2.6 – Le réseau phasé à deux polarisations à la station EMBRACE de Westerbork

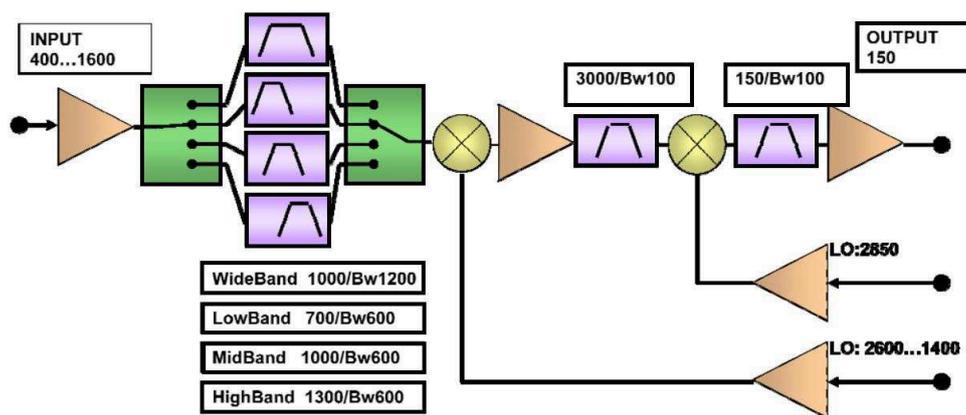


FIGURE 2.7 – Architecture du récepteur de la station EMBRACE [51]

TABLE 2.4 – Les spécifications d'EMBRACE

Spécifications	Remarques	Valeurs
Nombre de station		2
La surface effective $A_{eff}$		300 m <sup>2</sup>
Bande passante in		400-1500 MHz
Bande passante out		110-190 MHz
Oscillateurs locaux	Variable Fixe	1400-2600 MHz 2850 MHz
Température système $T_{sys}$	@ 1 GHz	≤ 100 k
Bande passante instantanée		80 MHz
Polarisation		1
Dynamique du signal	@ la sortie d'ADC	60 dB
Bande passante analogique		≥ 40 MHz
Ouverture efficace		≥ 8
Caractéristiques de convertisseur A/D	Référence Résolution Jitter Fréquence $F_s$ SNR @ 210 Msps	AD9430 12 bits 0,5 ps 200 MHz 65 dB @ 70 MHz
Horloge et synchronisation	Idem à celle de LOFAR	

### 2.1.3 GMRT

Le GMRT (Giant Meter-wave Radio Telescope) [52, 53] à Khodad en Inde est un instrument mondial d'étude des phénomènes basses fréquences en astrophysique (40-1450 MHz). Le GMRT est conçu en Inde comme étant un projet national en 1992. Le réseau d'antennes est constitué de 30 antennes de 45 m de diamètre, il est considéré comme étant le plus grand instrument dans le monde dans la bande de fréquence 40-1450 MHz. Les antennes sont réparties dans une portée de 25 km de diamètre sous forme de Y (Figure. 2.8). La longueur de base la plus grande est de 26 km, et 100 m la plus petite.

Le numériseur de l'instrument GMRT est particulièrement intéressant. Il est dédié d'un corrélateur FX (F pour la transformée de Fourier et X pour la multiplication) [55] sur 6 bits [56] développé par l'équipe du GMRT. Il est constitué d'un convertisseur A/D sur 6 bits et d'une FFT (Fast Fourier Transform) matérielle. Le signal de base est numérisé sur 8 bits. Il est muni d'un système de compensation des retards de propagation (retards aériens) manuel comme celui du radiohéliographe de Nançay. Le nombre des canaux de FFT est de 512 avec une résolution par canal variable entre 128 kHz et 500 kHz avec une bande passante totale de 16 MHz-32 MHz. La figure 2.9 montre le schéma bloc du numériseur. Le tableau 2.9 résume les caractéristiques techniques de l'instrument.

La transformée de Fourier sur 512 points du corrélateur FX du GMRT utilise quatre horloges précises. La fréquence d'échantillonnage est de 32 MHz avec une horloge de calcul de 32,25 MHz [58]. La gigue des deux horloges pour une référence de 105 MHz est égale à 500 ps

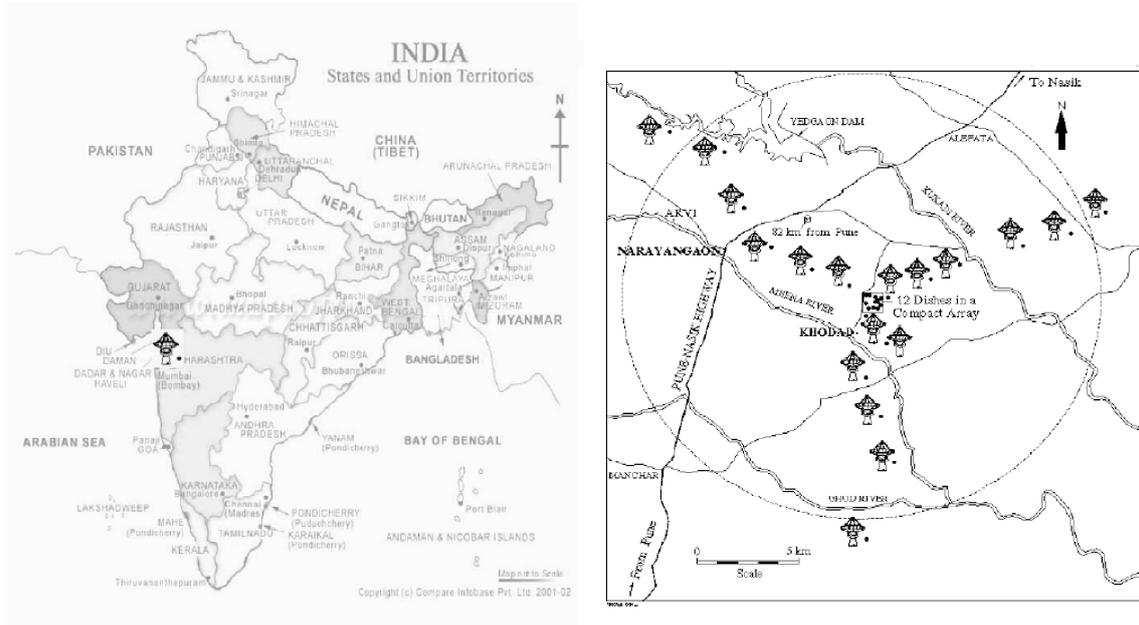


FIGURE 2.8 – Location et architecture de l’instrument GMRT [54]

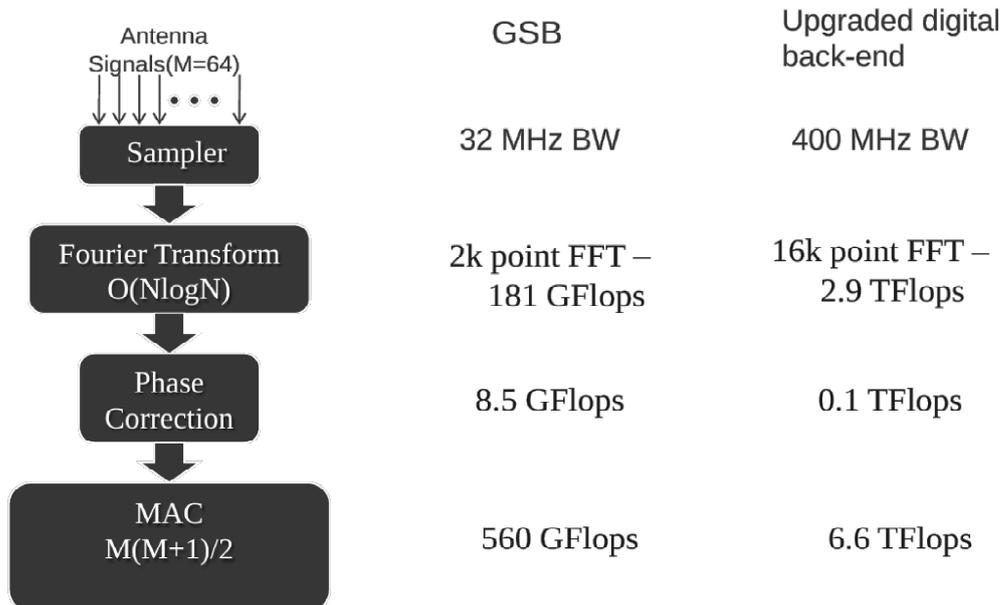


FIGURE 2.9 – Architecture basique du numériseur du GMRT avec 32 MHz de bande [57]

TABLE 2.5 – Les spécifications de GMRT

Spécifications	Remarques	Valeurs
Nombre d'antenne		32
La surface effective collectée $A_{eff}$		$300 m^2$
Bande passante in		130-1450 MHz
Bande passante out		32 MHz / 16 MHz
Oscillateurs locaux	Variable	30-1700 MHz
Transmission Antenne / Num	Fibre optique	
Bande passante instantanée		32 MHz / 16 MHz
Polarisation		2
Corrélateur	+ Beam Former FPGA( +ADC )	FX FPGA
Switch		Myricom 10 GbE
Carte GPU		Tesla C2050 and K20
Bande passante analogique		16 MHz
Convertisseur A/D	Référence Résolution Jitter Fréquence $F_s$ Retard de sortie	AD9058 8 bits (6 bits effectifs) 10 ps 50 MHz 8 ns
Horloge	GPS + Rb	

avec une précision de synchronisation de 9,5 ns pour une bande numérisée de 32 MHz (période de 31,25 ns). La précision de synchronisation effective entre les deux horloges, dépend aussi du "skew" de sortie entre les deux DDS (Direct Digital Synthesizer) (voir la figure 1 dans [58] et [59]).

## 2.2 Comparaison globale entre les instruments existants

### 2.2.1 La bande de base (bande numérisée)

Dans cette section, on va introduire les principales caractéristiques techniques des imageurs radios existants (ALMA [60–62], EVLA [63, 64], etc.) (voir ci-dessous) qui fonctionnent dans des bandes de fréquence d'observations allant jusqu'à 950 GHz (ALMA dans la bande 10). L'étude comparative se focalise sur la bande numérisée et la précision requise de synchronisation des horloges.

Le tableau 2.6 illustre la bande numérisée ainsi que la fréquence d'échantillonnage  $F_s(MHz)$  du convertisseur A/D pour les divers instruments. La bande numérisée pour la plupart des imageurs radio est inférieure à 100 MHz (Tab. 2.6). Elle est 2 GHz pour les deux instruments ALMA (Atacama Large Millimeter/submillimeter Array) et EVLA (Expanded Very Large Array) avec une bande d'observation allant de 31,3 GHz à 950 GHz. Ce choix est dû à la complexité technique du système de numérisation et au système d'acquisition des données (numérisation large bande, traitement rapide de données, synchronisation des horloges, etc.).

TABLE 2.6 – La bande de base des instruments et la fréquence  $F_s$

Instrument	Bande (MHz)	$F_s$ (MHz)
NRH	0,7	42,8
LOFAR	80	200
EMBRACE	80	200
GMRT	16	50
ATA [65,66]	100	250
EVLA	2000	4000
ALMA	2000	4000
LWA [67,68]	80	200
UTR-2 [69–71]	8-32	66

### 2.2.2 La résolution des convertisseurs A/D

Le tableau 2.7 montre le rapport  $\frac{F_s}{B}$  et la résolution binaire du convertisseur A/D pour les différents instruments radio. La résolution est inférieure ou égale à 8 bits pour les instruments ayant une bande numérisée au delà de 100 MHz, en revanche elle est supérieure à 8 bits pour les autres. La résolution augmente avec la diminution de la bande numérisée, elle est due à la simplicité de la mise en œuvre du circuit de numérisation, puisque la complexité du numériseur augmente avec l'augmentation de la bande numérisée.

TABLE 2.7 – La résolution des convertisseurs A/D

Instrument	Rapport $F_s$ /Bande	Résolution ADC
NRH	<b>61.14</b>	<b>1</b>
GMRT)	3.125	8
LOFAR	2.5	12
EMBRACE	2.5	12
ATA	2.5	8
LWA	2.5	12
UTR-2	2.06	2,8,12
WSRT	2	12
EVLA	2	8
ALMA	2	3

On distingue deux catégories d'instruments radio en fonction du rapport  $F_s$ /Bande (fréquence de Nyquist) : Les 7/10 premiers instruments dans le tableau 2.7 ont un rapport  $>2$ , trois (WSRT, EVLA et ALMA) ont un rapport égal à 2. En effet, tous les instruments respectent la fréquence de Nyquist. On constate que l'instrument NRH dispose d'une résolution binaire d'un bit avec un rapport de 61,14. Le récepteur du NRH contient deux parties indépendantes mais synchrones : Le corrélateur est sur un bit entre chaque couple d'antennes, et le détecteur de puissance sur 16 bits. Dans notre choix du rapport  $F_s$ /Bande on va utiliser une valeur supérieure ou égale à deux pour une bande analogique de 300-450 MHz. On verra dans le chapitre 4 les

TABLE 2.8 – La précision à l’application de l’horloge (Offset)

Instrument	Précision (ps)	Précision (°)
RH	23364	6
LOFAR	78	6
EMBRACE	78	6
EVLA	15	11
ALMA	15	11
GMRT	8500	99

conséquences de ce choix sur la qualité requise de l’horloge de l’échantillonneur du convertisseur A/D.

### 2.2.3 Précision de synchronisation des horloges

La précision en degré d’ajustement d’horloge du convertisseur A/D en fonction de la période d’échantillonnage  $T_{DelayClock}$  et la bande analogique  $B$  est donnée par l’équation 2.1. L’écart minimal d’ajustement des horloges du numériseur pour les différents instruments est illustré dans le tableau 2.8. On ne prend pas en considération les retards introduits par les composants (accumulation des temps de propagation dans le chemin des horloges). On constate que la précision augmente avec l’augmentation de la bande numérisée : LOFAR (78 ps pour une bande de 80 MHz), ALMA (15,62 ps, 2 GHz). D’après le tableau, la précision requise par le système de synchronisation décroît rapidement avec la bande numérisée. On conclut qu’une numérisation large bande constituée une contrainte majeure de la mise en œuvre du système de distribution des horloges synchrones des convertisseurs A/D.

$$P(^{\circ}) = 360 \times T_{DelayClock} (s) \times B (Hz) \quad (2.1)$$

$P$  : la précision de l’horloge des ADC en degré par rapport à la bande de base

$B$  (Hz) : la bande de base en Hz

$T_{delayClock}(s)$  : le retard fin appliqué à l’horloge en seconde

## 2.3 Conclusion

La précision minimale de l'application de l'horloge est de l'ordre de 15,62 ps (11,3 °) (Tableau 2.8) et 78 ps pour LOFAR et EMBRACE. Elle correspond à la bande de base de 2 GHz des instruments ALMA et EVLA. La précision est obtenue grâce à un circuit de mesure des retards logiciels fins en se basant sur un ensemble de paramètres (longueurs des câbles, localisation des antennes, ...). L'analyse comparative montre que la plupart des instruments utilisent une bande de base faible afin d'alléger la contrainte de précision au niveau de la synchronisation des horloges des circuits de numérisation (tableau 2.6).

En résumé, la précision équivalente de l'horloge du numériseur de l'instrument NRH en comparaison avec les instruments ALMA ou EVLA pour une bande analogique de 500 MHz, en respectant le rapport (fréquence d'échantillonnage/bande) est de l'ordre de quatre fois la précision de l'instrument ALMA (2 GHz de bande pour ALMA) égale à 62,48 ps. L'approche de notre étude est de réaliser une numérisation au pied des antennes. Dans l'état de l'art actuel, tous les instruments utilisent une numérisation centralisée. On va aborder, dans le chapitre suivant, l'étude de spécification pour le nouvel instrument dédié à une numérisation large bande pour un réseau d'antennes multi-reparties.



# Instrumentation et synchronisation

## 3.1 Introduction

La synchronisation est un service important dans les systèmes de télécommunication et dans les réseaux distribués, incluant l'interférométrie radio. Il existe des systèmes qui nécessitent la synchronisation du temps [72–74], d'autres de fréquence [75]. On distingue trois types de synchronisation d'une source de référence : la synchronisation du temps, de la fréquence et d'horloge (temps et fréquence). Dans ce chapitre, on va aborder la notion de synchronisation, la modélisation et la compensation des erreurs. On terminera avec la description des protocoles importants de synchronisation dans les réseaux distribués.

### 3.1.1 Les notions de synchronisation

La synchronisation dans un réseau reparté regroupe deux composantes essentielles : le temps et la fréquence. Ces deux paramètres sont quantifiables. La fréquence est proposée avec une précision à long terme mesurée en ppm (parties par million ou  $10^{-6}$ ), ppb (parties par milliard ou  $10^{-9}$ ). Le temps est caractérisé par le dispositif de mesure, par sa précision (l'intervalle de temps minimal qu'on peut mesurer) et le bruit de mesure (gigue induite par la mesure).

Pour synchroniser les horloges dans un réseau, les différents équipements doivent être ajustés sur un même rythme. Ce rythme est délivré par une horloge de référence dénommée Primary Reference Clock (PRC) [76] ou Primary Reference Source (PRS) dont la précision est supérieure à  $10^{-15}$  pour les horloges atomiques [77, 78]. La précision du système de synchronisation est définie par l'application visée, elle peut varier de la milliseconde à la femtoseconde et dépend de l'étendue du réseau [79, 80].

### 3.1.2 Les objectifs de la synchronisation d'horloges

Une horloge synchronisée permet de fournir une référence de temps et de fréquence globale et identique pour toutes les unités constituant le réseau. Un système de synchronisation est basé sur un protocole qui assure la fonction de synchronisation malgré les défauts de fonctionnement du réseau, en particulier :

- Les dérives des horloges physiques (ou logiques) des unités

- Les défaillances des sites ou de leurs horloges
- Les défaillances du réseau de communication entre les nœuds du réseau

## 3.2 Synchronisation et modélisation d'une horloge

### 3.2.1 Introduction

Le problème de la synchronisation entre deux horloges est généralement résolu par la sélection d'une horloge de référence et par la synchronisation de toutes les autres horloges avec l'horloge de référence. La synchronisation entre deux horloges (ou plus) consiste à faire tourner l'ensemble simultanément à la même vitesse (fréquence) en même temps. Dans cette section on va aborder le modèle mathématique d'une horloge, ses paramètres et les techniques existantes pour la compensation de ces dernières dans un réseau distribué. On va étudier en particulier comment compenser l'offset et le "skew" d'une horloge [81–83].

### 3.2.2 Notion des horloges logique et physique

#### 3.2.2.1 L'horloge logique

Pour introduire la notion d'horloge logique, on va se baser sur l'horloge de l'ordinateur. Il dispose d'un circuit qui permet de suivre l'évolution du temps. Ce ne sont pas des horloges au sens courant du terme mais plutôt des temporisateurs. Un temporisateur est principalement logique, il est constitué d'une source d'horloge physique de type quartz ou atomique. Deux registres sont associés à chaque source physique, un compteur et un registre de sauvegarde. Chaque oscillation du cristal décrémente le compteur d'une unité. Quand le compteur arrive à zéro, on génère une interruption et on recharge le compteur à partir du registre de sauvegarde. On peut ainsi programmer un temporisateur pour générer une interruption par seconde (1-PPS) ou à n'importe quelle autre fréquence. Chaque interruption s'appelle une impulsion d'horloge.

Dès qu'on introduit plusieurs nœuds dans un réseau, chacun d'entre eux a sa propre horloge. Même si la fréquence à laquelle le quartz fonctionne est souvent relativement stable, il est impossible de garantir que les horloges logiques de différentes machines oscilleront rigoureusement à la même fréquence. En pratique, si un système comprend  $N$  nœuds, les  $N$  horloges fonctionnent à des vitesses différentes, ce qui dégrade petit à petit la synchronisation des horloges (logiques), qui donnent donc différentes valeurs lorsqu'elles sont lues. Ce phénomène est appelé la dérive d'horloge ou le skew.

#### 3.2.2.2 Horloge physique

Bien que l'algorithme de Lamport [84] donne un ordre non aléatoire d'événements, les valeurs temporelles associées ne correspondent pas forcément à la date réelle à laquelle elles se produisent. Dans certains systèmes (par exemple, les systèmes de contrôle de procédé), le temps "réel" est important. Il est alors nécessaire d'avoir une horloge physique externe. Pour des raisons d'efficacité et de redondance, la présence de plusieurs horloges physiques est en général souhaitable, ce qui pose deux problèmes :

1. Comment allons-nous les synchroniser avec les horloges du monde réel ?

2. Comment allons-nous synchroniser les horloges entre elles ?

Grâce à l'invention de l'horloge atomique, en 1948, il est devenu possible de mesurer le temps beaucoup plus précisément et indépendamment des mouvements de la Terre, en comptant simplement les transitions de l'atome de césium 133 [85, 86]. Les physiciens ont défini la seconde comme étant le temps que prend l'atome de césium 133 pour faire exactement 9192631770 transitions ou éventuellement le Rubidium 87 ( $^{87}\text{Rb}$ ) à 6834682611 transitions (6.83 GHz). Aujourd'hui, des milliers de laboratoires à travers le monde ont des horloges atomiques en particulier le Césium 133 [87] :

Régulièrement, chaque laboratoire indique au Bureau International de l'Heure (BIH) de Paris de combien d'impulsions leur horloge a avancé. La moyenne effectuée au BIH correspond au Temps Atomique International, abrégé en TAI. Ainsi, le TAI est le nombre moyen d'impulsions des horloges au césium 133 depuis le 1er janvier 1958 (temps zéro) divisé par 9192631770. Ci-dessus la liste non exhaustive des domaines d'applications.

1. Numérisation des signaux radios issus des antennes,
2. Systèmes multiprocessus pour le calcul scientifique,
3. Systèmes de commande en temps réel des processus physiques,
4. Réseaux de télécommunications,
5. Synchronisation de flux multimédia dans la téléconférence,
6. Simulateurs embarqués.

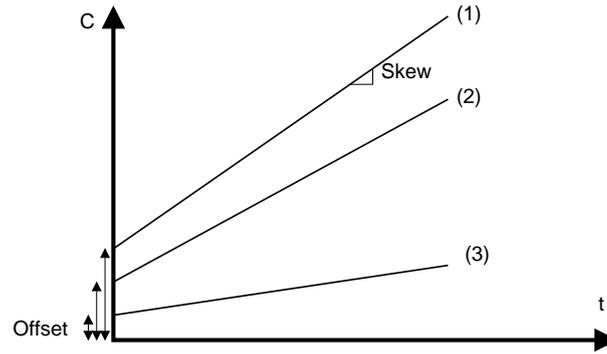
### 3.2.3 Modèle d'une horloge

Une horloge peut être modélisée par deux paramètres dans un système de synchronisation : l'offset et le skew (ou la dérive) [88]. La première est une erreur additive, définie comme étant le décalage temporel ou le retard entre l'horloge de référence et l'horloge en question. Le "skew" est de type multiplicatif. Il est illustré par le changement de la pente de l'horloge  $C_i(t)$  dans le temps illustrée dans la figure 3.1. Le "skew" est une erreur avec un effet sur la vitesse et donc sur le cadencement de l'horloge. Une horloge de référence est caractérisée par un offset nul et un "skew" égal à 1 dans le cas idéal. On distingue trois configurations d'une horloge par rapport à l'horloge de référence paramétrées par  $\alpha_i$  (skew) et  $\beta_i$  (offset) :

1. Une horloge parfaite :  $\alpha_i = 1$  &  $\beta_i = 0$  ( $C_i(t) = t$ )
2. Une horloge rapide :  $\alpha_i > 1$  &  $\beta_i \neq 0$  ( $C_i(t) = \alpha_i \cdot t + \beta_i$ )
3. Une horloge lente :  $\alpha_i < 1$  &  $\beta_i \neq 0$  ( $C_i(t) = \alpha_i \cdot t + \beta_i$ )

La figure 3.1 illustre trois horloges différentes  $C_1$ ,  $C_2$  et  $C_3$  en fonction du temps. Chacune de ses dernières est caractérisée par l'offset  $\beta_i$  et le skew  $\alpha_i$ . L'horloge  $C_i$  est définie par l'équation (3.1).

$$C_i = \alpha_i \cdot t + \beta_i \tag{3.1}$$


 FIGURE 3.1 – Horloges  $C_i$  en fonction du temps  $t$  avec divers offsets et skews

Si on prend comme référence l'horloge  $C_2$ , l'horloge  $C_3$  est plus lente et l'horloge  $C_1$  plus rapide (voir la figure 3.1). Ci-dessous les sources probables d'erreurs ("skew" et "offset") [89–92] dans un réseau distribué :

- La température de l'horloge de référence,
- Temps de transfert  $T_s$  : Temps nécessaire pour la lecture du message, l'activation de l'horloge au niveau émetteur,
- Le temps d'accès  $T_a$  : Temps d'accès au canal de transmission,
- Le temps de transmission  $T_t$  : Temps nécessaire pour effectuer la transmission du message,
- Le temps de propagation  $T_p$  : Le temps entre la transmission et la réception du message, il dépend uniquement de la distance et du type du canal de transmission,
- Le temps de réception  $T_{rp}$  : Idem que le temps de transmission  $T_{rp} = T_t$ ,
- Le temps d'acquisition  $T_{rv}$  : Temps nécessaire au récepteur pour reconstituer le message reçu,
- Les défauts de fabrication, les conditions ambiantes.

Le temps total des retards entre l'émetteur et le récepteur, d'un maître à un esclave (i)  $d_{MS}^i$  (Master to Slave Delay) est défini par l'équation (3.2) est égal à la somme des retards dans le chemin aller entre le maître et l'esclave dans le cas d'une liaison symétrique ( $d_{SM}^i = d_{MS}^i$ ). En pratique, il existe des erreurs de l'asymétrie dues au comportement aléatoire des retards entre l'émetteur et le récepteur. Dans la suite de la section, on va étudier les techniques de compensation des erreurs de l'offset et du "skew" entre le maître et l'esclave.

$$d_{MS} = T_s + T_a + T_p + T_{rp} + T_{rv} \quad (3.2)$$

### 3.2.4 Compensation de l'offset et de skew

#### 3.2.4.1 Introduction

Pour synchroniser entre les horloges dans un réseau distribué, il faut s'assurer que toutes les horloges  $C_i$  tournent à la même vitesse ( $\alpha_i = \alpha_j, i \neq j$ ) et ont également commencé en même

temps ( $\beta_i = \beta_j, i \neq j$ ). Dans un algorithme de synchronisation en temps réel, il est indispensable de définir une période de synchronisation  $T_{sync}$  pour laquelle le système resynchronise les différents nœuds du réseau. On considère deux horloges  $C_1$  et  $C_2$  ayant une erreur d'offset et de "skew" et  $C_1'$  l'horloge après compensation. On prend  $C_2$  comme étant l'horloge de référence. La figure 3.2 montre les l'effet de compensation de l'offset (A) et du skew (B) d'une horloge par rapport à une référence.

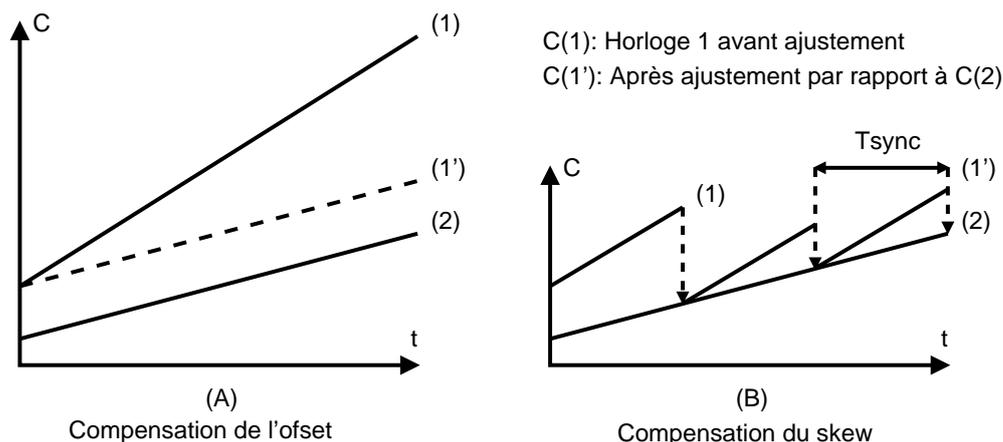


FIGURE 3.2 – Compensation de l'erreur du "skew" (A) de l'offset (B) d'une horloge dans un système de synchronisation. (1) : Horloge avant compensation et (1') après compensation de l'horloge (1) par rapport à l'horloge de référence (2).  $T_{sync}$  la période de resynchronisation du système.

La compensation du "skew" (A) dans la figure 3.2 de l'horloge  $C_1(t)$  par rapport à l'horloge  $C_2(t)$ , consiste à compenser uniquement la partie dynamique de l'horloge par rapport à l'horloge de référence pour qu'elle puisse fonctionner à la même vitesse, tandis que le décalage temporel entre les deux horloges reste le même avant et après la compensation ( $\alpha_1 = \alpha_2 = \alpha_{1'}$ ). La compensation de l'offset (B dans la figure 3.2) permet de compenser l'erreur statique de l'horloge qui peut dépendre uniquement des écarts du retard entre les deux horloges. La période de synchronisation (période de mise à jour) définit la précision de synchronisation du système, lorsque la période est petite, le dispositif a tendance à avoir une erreur minimale. Réciproquement, la complexité du système augmente avec la diminution de la période de synchronisation. Le choix de la période dépend également du temps nécessaire pour le calcul des écarts de compensation (estimation des erreurs) et le temps d'application des ajustements d'horloges (offsets et "skews").

En pratique, l'offset entre les deux horloges (Maître et esclave)  $|\beta_M - \beta_S|$  présente un retard  $d_{MS}$  entre l'horloge  $C_M$  (horloge au niveau du maître) et  $C_S$  (horloge au niveau de l'esclave). On verra dans la suite du chapitre les techniques d'estimation et de mesures des retards avec une grande précision.

### 3.2.4.2 Compensation de l'offset et du skew d'une horloge

Le temps du retard est imprédictible entre les nœuds dans un réseau distribué (comportement pseudo-aléatoire des retards et des temps de propagation), ce processus va dégrader les performances du système de synchronisation. Plusieurs stratégies existent pour compenser le problème de l'offset du retard entre deux horloges (retard entre deux nœud(i,j)). La méthode la plus simple est basée sur une technique de transmission unidirectionnelle. En effet, le nœud "i" lit l'horloge  $C_i$  à l'instant  $t_1$  et l'envoie au nœud "j" qui lit sa propre horloge  $C_j$  à l'instant  $t_2$  puis calcule l'offset  $\beta_{ij} = C_j(t_2) - C_i(t_1)$ . Cette approche suppose la symétrie des retards entre les deux nœuds, qui n'est pas toujours assurée dans le cas pratique [93,94]. De plus, le temps du retard  $T_d = d_{ij} = t_2 - t_1$  ne peut pas être calculé correctement à cause du comportement aléatoire de  $T_d$ . Néanmoins il existe des circuits de mesure du temps à haute précision de l'ordre de la picoseconde.

Il existe plusieurs approches dites bidirectionnelles, qui permettent après une communication entre les deux nœuds de synchroniser les horloges de ces derniers. Le principe de base de cette technique consiste en la retransmission au nœud "i", de l'horloge  $C_j(t_2)$  dès la réception en supposant que le retard  $T_d$  est symétrique et constant entre le nœud "i" et "j" et vice versa. L'erreur de l'offset réel après la calibration de la chaîne (suppression de l'effet des retards accumulés dans la chaîne  $T_d$  induits par le chemin des horloges entre les deux nœuds) est estimée suivant l'équation (3.3). Cette approche est utilisée dans le protocole LTS (Lightweight Time Synchronization) [95,96].

$$\beta_{ji} = C_j(t_2) - C_i(t_1) - T_d \approx C_j(t_2) - C_i(t_1) \quad (3.3)$$

On considère deux horloges  $C_i$  et  $C_j$  définies par l'équation (3.4).

$$\begin{cases} C_i = \alpha_i \cdot t + \beta_i \\ C_j = \alpha_j \cdot t + \beta_j \end{cases} \quad (3.4)$$

Les paramètres  $(\alpha_i, \beta_i, \alpha_j, \beta_j)$  des deux horloges  $C_i$  et  $C_j$  peuvent être déduites à partir des mesures locales des horloges "i" et "j" à partir de la valeur des horloges en fonction du temps. L'équation (3.5) montre le calcul des paramètres de compensation de l'horloge  $C_j$  en prenant comme référence l'horloge  $C_i$ .

$$\begin{aligned} C_j &= \frac{\alpha_j}{\alpha_i} \cdot C_i + (\beta_j - \frac{\alpha_j}{\alpha_i} \cdot \beta_i) \\ C_j &= \alpha_{ij} \cdot C_i + \beta_{ij} \end{aligned} \quad (3.5)$$

La phase de synchronisation consiste en la compensation des paramètres de l'horloge à synchroniser par rapport à une horloge de référence. Dans l'utilisation de deux horloges d'indices (i,j), elle consiste en la mise à jour des paramètres de l'horloge  $C_j$  en remplaçant  $\alpha_j$  et  $\beta_j$  par les valeurs  $\alpha_{ij}$  et  $\beta_{ij}$  estimées par la mesure des deux horloges. Dans notre système de synchronisation, on va se focaliser sur l'estimation et la compensation de l'erreur de l'offset entre les nœuds,

dans le réseau en supposant une vitesse des horloges identiques (même skew). Notre système de synchronisation permet de distribuer des horloges physiques ayant la même fréquence issue d'une horloge maître stable et précise.

Il existe des applications qui ont besoin d'avoir d'horloges tournant à la même vitesse (Synchronisation de fréquence), mais avec des offsets différents, d'autres qui exigent l'instantanéité des horloges (Synchronisation du temps). La plupart des techniques de synchronisation ne sont pas implémentables en temps réel du fait de leurs complexité et faisabilité technique. Uniquement les algorithmes se basant sur la régression linéaire supportent le temps réel. La plupart des méthodes de la gestion du temps utilise une PLL (Phase Locked Loop) pour la synthèse des fréquence NTP (Network Time Protocol) [97, 98] ou EPTMA (Exponential Peak Tolerant Midpoint Algorithm) [99]. Le EPTMA permet d'échanger les paquets de synchronisation permettant de produire un estimateur en temps réel de l'horloge au niveau du récepteur.

L'estimateur de skew en temps réel dans le système EPTMA a un compromis entre le temps de latence et l'inertie dans le réseau. Il dépend des paramètres utilisés. La détection de la pente est rapide mais l'algorithme génère des valeurs erronées dans les transitions dues au changement du réseau du transport (changement du nombre de points dans le réseau). Il peut être aussi moins sensible aux transitions locales du réseau, mais par contre l'algorithme nécessite du temps pour estimer le skew (quelques dizaines de secondes). L'idée de base consiste à prendre la valeur moyenne des variations du retard pour prendre une valeur lissée pour obtenir l'estimation du skew. On calcule la valeur moyenne des retards triés par ordre décroissant  $\delta_i$ , puis, on supprime la valeur maximale et la valeur minimale (réduire l'effet des erreurs de mesure). Le système prend une fenêtre de largeur  $w$  et de taille  $k$ . Le retard moyen  $d_m$  est donné par l'équation (3.6).

$$\begin{cases} d_m = \frac{1}{k} \cdot \sum_{i=\frac{w-k}{2}}^{\frac{w+k}{2}} \delta_i \\ 1 \leq k \leq w \end{cases} \quad (3.6)$$

A l'instant  $t$ , l'estimation du "skew",  $\alpha(t)$  est obtenue par le lissage du retard  $d_m$  par l'équation (3.7). Le "skew" estimé dépend de l'ancien "skew" et des retards moyens.  $\alpha_0$  présente le facteur de pondération de l'algorithme qui influence la vitesse de convergence ainsi que l'erreur globale de l'estimateur.

$$\alpha(t) = \alpha_0 \cdot d_m(t) + (1 - \alpha_0) \cdot \alpha(t - 1) \quad (3.7)$$

Dans la phase d'initialisation (quand la fenêtre  $w$  n'est pas encore remplie), l'algorithme calcule la valeur moyenne  $d_m$  du retard en se basant sur la valeur quadratique de  $r_n$ . Soit  $n$  le nombre des valeurs. La valeur de  $r_n$  est calculée par l'équation (3.8).

$$r_n = k - \frac{k}{w^2}(n - w)^2 \quad (3.8)$$

#### 3.2.5 Compensation de la gigue d'une horloge

Dans les systèmes de synchronisation d'horloges, une horloge de fréquence stable et précise est indispensable. Des horloges atomiques sont souvent utilisées dans les systèmes de communication qui exigent une grande précision et stabilité. On trouve des horloges atomiques assistées avec une source GPS (Global Positioning System) pour augmenter la stabilité à long terme.

Un réseau GPS est constitué d'une constellation de 24 satellites équipés par des horloges atomiques, qui peuvent être utilisées pour la synchronisation des horloges [100–102]. Le GPS permet de générer un signal 1-PPS (1 Pulse Per Second) dédié à la resynchronisation du système (période de synchronisation d'une seconde) avec une fréquence standard de 10 MHz. Dans la plupart des sources GPS, dans le cas extrême, la gigue du signal 1-PPS est de l'ordre de 100 ns. Essentiellement, le 1-PPS d'une source GPS est constitué d'une fréquence parfaite avec une gigue résiduelle. La gigue est de l'ordre de  $\pm 100$  ns dans les circuits du commerce. L'une des techniques utilisées pour la synthèse d'un signal 1-PPS à faible gigue à partir d'un signal 1-PPS issu d'un récepteur GPS consiste en l'utilisation d'une PLL à faible gigue avec un correcteur PI (Proportional Integral). Le correcteur PI (filtre passe bas) permet d'obtenir un PPS de quelques dizaines de ns [103]. En résumé, une réduction d'un facteur de 10 de la gigue du signal PPS est possible après correction par rapport au PPS à la sortie de la source GPS.

## 3.3 Instrumentation du temps/fréquence

### 3.3.1 Introduction

Pour capter la notion de temps réel (notion abstraite), il faut des équipements de mesure qui permettent de passer du temps réel au temps physique ou mesuré. Les équipements de mesure ont évolué dans le temps ainsi que leurs précisions (horloge mécanique (pendule), horloge à quartz et horloge atomique). La limite minimale de mesure du temps est le *Chronon* égal à  $5.4 \cdot 10^{-44}$  secondes (limite de la mécanique quantique). La limite maximale de mesure du temps est la vie de l'univers d'environ  $10^{17}$  secondes. La meilleure approximation du temps est donnée par des horloges atomiques à base de césium avec une précision de  $10^{-14}$  (dérive d'une microseconde tous les 3 ans environ,  $3 \text{ ans} = 0.94 \cdot 10^8 \text{ s}$ ;  $3 \text{ ans} \cdot 10^{-14} = 0.94 \mu\text{s}$ ). Les mesures du temps et de la fréquence aboutissent aux méthodes utilisées en métrologie. Le dispositif de mesure est généralement constitué d'un compteur à haute précision, de la fréquence de l'horloge à mesurer nommée DUT (Device Under Test) (Figure 3.3). Les mesures sont comparées entre le DUT et le standard (la fréquence de référence), le rapport entre le standard et la valeur mesurée est nommé TUR (Test Uncertainty Ratio). Il indique l'écart entre le DUT et la référence. La section est consacrée à l'instrumentation du temps et de la fréquence et aux diverses techniques existantes pour la mesure précise du temps.

### 3.3.2 La précision

La précision est le degré de conformité de la mesure à la valeur de référence, c'est l'offset entre la valeur mesurée et la valeur réelle (valeur de référence). Le temps d'offset d'une impulsion est la différence entre la valeur mesurée et la durée parfaite de l'impulsion, cette dernière coïncide avec l'UTC (Temps Universel Coordonné). L'offset de fréquence est la différence entre la fréquence

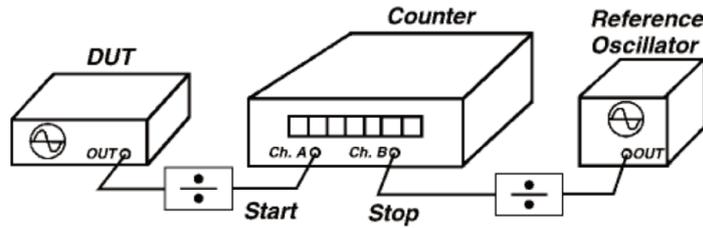


FIGURE 3.3 – Schéma basique de mesure de temps

mesurée et sa valeur réelle avec une incertitude nulle, aussi appelée la fréquence de référence. La fréquence peut être mesurée dans le domaine temporel ou fréquentiel. Pour un compteur, la valeur de la fréquence vaut la valeur du compteur avec une résolution d'un intervalle de temps, définie par l'oscillateur interne du compteur. L'offset de fréquence en pourcentage est défini par l'équation (3.9).

$$f_{Offset} = 100 \cdot \frac{f_{mesure} - f_{ref}}{f_{ref}} \quad (3.9)$$

La mesure d'un temps court (signaux rapides) n'est pas pratique avec un oscilloscope, la phase entre les signaux change souvent et la résolution de l'oscilloscope est limitée pour des fortes précisions. Pour obtenir une meilleure précision, on peut utiliser un TIC (Time Interval Counter) [104,105]. Si les deux signaux ont la même fréquence, l'intervalle de temps restera fixe, sinon l'intervalle de temps change et la vitesse de changement est définie comme étant l'offset de fréquence entre la fréquence mesurée et la référence. Par exemple, un TIC de résolution de 100 ns (10 MHz de fréquence) peut détecter un changement de fréquence de  $10^{-7}$ . Il existe actuellement des circuits de mesure d'intervalle de temps qui atteignent des résolutions temporelles inférieures à 20 ps. Autrement dit, une variation de fréquence de  $2 \cdot 10^{-11}$  peut être détectée en une seconde. À long terme, pour un grand intervalle de temps, la résolution est inférieure à 1 ps. L'offset de fréquence en valeur absolue est défini par l'équation (3.10).

$$f_{Offset} = \frac{d\phi}{T} \quad (3.10)$$

*T étant la période du signal,  $d\phi$  la déviation de la phase.*

Par exemple, pour une déviation de phase d'une microseconde pendant une période de 24 heures, l'offset en fréquence vaut  $\frac{d\phi}{T} = 1,16 \cdot 10^{-11}$ .

L'accumulation d'une déviation de phase de  $1 \mu s$  induit un offset relatif de la fréquence de  $1,16 \cdot 10^{-11}$  par rapport à la référence. L'exemple nécessite la mesure de deux temps d'intervalle et la mesure de la déviation de la phase entre les deux mesures précédentes. L'offset peut être converti en fréquence, en connaissant la valeur de la fréquence de référence en Hertz. Si on considère un oscillateur avec une fréquence de référence de 5 MHz et un offset de  $+1,16 \cdot 10^{-11}$ , l'offset de fréquence en Hertz vaut  $58 \mu Hz$ , soit une fréquence d'horloge de  $5 Hz + 58 \mu Hz$ . L'offset en fréquence d'un oscillateur à cristal peut être inférieur à la picoseconde dans une bande inférieure à 20 MHz. Il dépend de la technologie utilisée [106].

### 3.3.3 La stabilité

La stabilité est définie comme étant l'estimation statistique de la fluctuation de la fréquence ou du temps dans un intervalle de temps. Ces fluctuations sont mesurées en respectant la fréquence de référence. On distingue deux types de stabilité :

1. Intervalle de temps « 100 s : court terme
2. Intervalle de temps » 100 s : long terme

Dans le cas d'un oscillateur, elle montre comment produire le même offset de temps ou de fréquence pour une période donnée quelle que soit la valeur de la fréquence ou du temps (correcte ou non). La stabilité peut être mesurée dans le domaine temporel ou fréquentiel, en se basant sur l'offset de fréquence et sur l'intervalle de temps. L'offset est défini comme étant la déviation de la mesure par rapport à la valeur de référence en supposant que les mesures soient stationnaires (indépendantes du temps, bruit blanc). Par exemple, un oscillateur n'est pas stationnaire, les mesures changent en fonction du temps, ceci implique que la moyenne ou la déviation changent en permanence. Donc, la variance n'est pas une estimation correcte de la stabilité, d'où l'intérêt d'utiliser un autre estimateur dans le domaine du temps. Il s'agit de la variance d'Allan [107, 108]. Elle est définie par l'équation (3.11).

$$\sigma_y(\tau) = \sqrt{\frac{1}{2(M-2)} \cdot \sum_{i=1}^{M-1} (y_{i+1} - y_i)^2} \quad (3.11)$$

$$\sigma_y(\tau) = \sqrt{\frac{1}{2(M-2) \cdot \tau^2} \cdot \sum_{i=1}^{N-2} (x_{i+2} - 2 \cdot x_{i+1} + x_i)^2} \quad (3.12)$$

$y_i$  sont les mesures de l'offset de fréquence,

$M$  le nombre de mesures espacées d'une période de temps identique  $\tau$ .

$x_i$  sont les mesures de phase,  $N$  le nombre de valeurs de  $x_i$ , espacées par une période longue  $\tau$ .

Cet estimateur enlève le bruit dépendant de l'offset de fréquence (Eq. (3.12)). Les graphes dans les figures (Figure 4.5 et Figure 4.6) [109] montrent l'évolution de la variance d'Allan en fonction du temps et les types de bruit, elles illustrent la stabilité de l'horloge. Cette dernière augmente lorsque la période augmente. La stabilité est minimale pour une source à Césium d'une fréquence de 10 MHz à partir d'un point appelé "*floor*". La variance au point d'Allen est d'environ  $2 \cdot 10^{-16}$  pour  $\tau = 45$  s (voir la figure 4.5 dans [109]).

Le coefficient de qualité  $Q$  d'un rubidium est d'environ  $10^7$ . Le décalage de fréquence de résonance est dû aux collisions des atomes de *Rb* (Rubidium) et des molécules de gaz qui limitent la stabilité au long terme. L'offset de fréquence de l'oscillateur de *Rb* est d'environ  $5 \cdot 10^{-10}$  à  $5 \cdot 10^{-12}$  en une journée. Le facteur de qualité  $Q$  d'une horloge à base de Césium est d'environ  $10^8$ . La stabilité pour une période  $\tau$  d'une seconde est d'environ  $5 \cdot 10^{-12}$ , atteint  $10^{-14}$  pour une journée. Elle est extensible à plusieurs semaines ou mois. L'offset de fréquence

TABLE 3.1 – Évolution des standards du temps et de la fréquence

Standard	Date	Imprécision temporelle(s) ( $\tau = 24H$ )	Imprécision fréquentielle ( $\tau = 24H$ )
Pendule	1656	10	$10^{-4}$
Quartz	1927	$10^{-5}$	$10^{-10}$
Rubidium	1958	$10^{-7}$	$10^{-12}$
Césium beam $^{133}Cs$	1952	$10^{-9}$	$10^{-14}$
Fontaine de Césium $^{133}Cs$	1991	$10^{-10}$	$10^{-15}$

est d'environ  $10^{-12}$  après la période d'échauffement de 30 minutes. Le tableau 3.1 résume l'évolution des standards du temps et de fréquences [110–112].

Le tableau 3.1 montre l'évolution de la précision temporelle et fréquentielle d'une horloge pour  $\tau = 24H$ . Pour une source à Césium, l'imprécision temporelle est de l'ordre de 100 ps et de 1 fs pour la fréquence pour une durée d'une journée. On constate clairement un rapport de  $10^5$  entre l'imprécision temps/fréquence pour les horloges atomiques. Ce rapport influence directement la précision de synchronisation d'horloges dans un réseau distribué. On verra dans la suite du chapitre les techniques de la mesure du temps et de la précision globale des protocoles de synchronisation existants.

### 3.3.4 Mesure du temps d'intervalle

#### 3.3.4.1 Introduction

La mesure de l'intervalle de temps  $T_i$  [113, 114] d'un ou plusieurs événements est souvent requis dans les systèmes d'instrumentation du temps ou de la fréquence et dans les systèmes de synchronisation. Le temps d'intervalle consiste en la mesure du temps entre un signal de début d'événement ("Start") et de fin de cet événement ("Stop"). Ces deux signaux peuvent être induits par un système électronique de précision ou un détecteur de phénomène physique, le principe de mesure est illustré dans la figure 3.4. Dans cette section, on va illustrer les principales techniques de base des circuits de la mesure du temps TDC (Time-to-Digital Converter), leurs performances et leurs limitations.

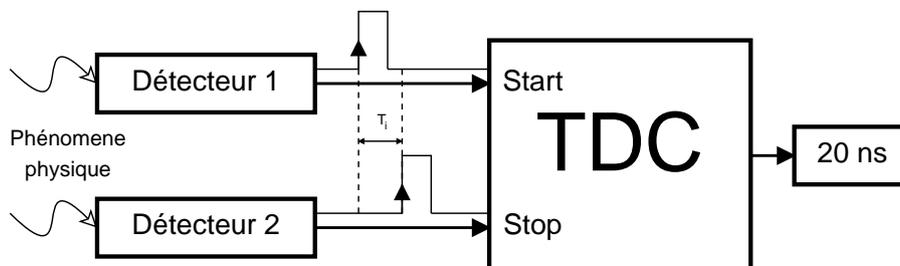


FIGURE 3.4 – Principe de circuit de la mesure du temps

#### 3.3.4.2 Caractéristiques d'un circuit de mesure du temps

Les caractéristiques principales d'un circuit de mesure du temps sont les suivantes :

- La dynamique de mesure : le plus grand intervalle de temps qui peut être mesuré.
- L'incertitude de mesure : l'écart type de la distribution des mesures autour de la valeur moyenne pour une série de mesures d'un intervalle de temps constant. Elle détermine l'erreur de reproductibilité de la mesure par le circuit TDC.
- La résolution : l'intervalle de temps minimal que le circuit peut mesurer.
- La vitesse de lecture et de production des résultats de mesures. Le paramètre est important lorsque les mesures sont effectuées de manière continue pour la mesure des événements rapides
- Non-linéarité : L'erreur intégrale, exprimant l'écart par rapport au comportement idéal du circuit TDC.

Dans ce qui suit, une brève description des techniques de mesure du temps les plus pertinentes est introduite. On s'intéresse en particulier aux techniques suivantes :

- Méthode de Nutt,
- Utilisation d'un compteur,
- Conversion temps-tension,
- Méthode de Vernier,
- Méthode des lignes à retard numérique.

Les techniques de compteur offrent une très large dynamique, mais la résolution de la mesure est limitée par la fréquence maximale du signal d'horloge. Afin d'améliorer la résolution de temps du système de mesure, le processus d'acquisition du temps doit être basé sur la méthode "Nutt" obtenue avec une mesure en deux étapes. Les techniques utilisées pour réaliser une mesure fine du temps peuvent être généralement classées comme analogiques (conversion temps-tension/courant) et numérique (techniques de ligne à retard et Vernier [115,116]). Les techniques basées sur des composants analogiques ont un temps de conversion plus long et elles sont plus sensibles aux paramètres physiques (alimentation, température, bruit). De plus, elles sont difficiles à intégrer. Les méthodes dites numériques sont plus faciles à mettre en œuvre, elles offrent une bonne stabilité, en revanche les dynamiques de mesures sont relativement faibles.

#### 3.3.4.3 La méthode de Nutt

La méthode "Nutt" [117] permet de mesurer le temps d'intervalle entre deux événements. Elle est appropriée pour les grandes dynamiques de mesure avec une grande précision. La figure 3.5 illustre le principe de fonctionnement de cette méthode. Le cycle de mesure est composé de trois phases : D'abord, l'intervalle de temps  $T_1$  entre les fronts montants du signal "Start" et le signal de l'horloge synchrone "Clock". Cet intervalle est mesuré à l'aide d'un TDC précis (mesure fine). La même procédure est exploitée pour mesurer l'intervalle de temps  $T_3$  entre les fronts montants du signal "Stop" et le signal de l'horloge. Le temps  $T_2$  présente la valeur du compteur du circuit TDC (Mesure lente). L'intervalle global  $T_i$  présente le temps entre le signal "Start" et celui du "Stop", il est donné par l'équation (3.13).

$$T_i = T_1 + T_2 - T_3 \quad (3.13)$$

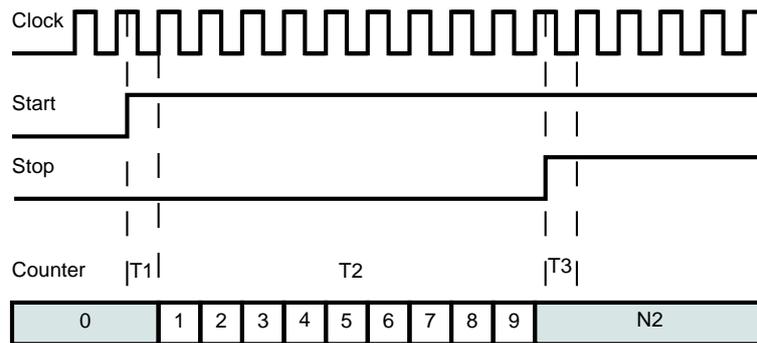


FIGURE 3.5 – Chronogrammes de la mesure du temps d'intervalle suivant la méthode Nutt

On distingue deux types de mesures : lente et fine. La mesure lente a une résolution généralement donnée par la période de l'horloge. La mesure fine utilise un TDC avec une résolution d'une période d'horloge. Les dynamiques de conversion fine  $T_1$  et  $T_3$  sont limitées à un seul cycle d'horloge de référence avec une résolution allant de 10 à 100 ns.

#### 3.3.4.4 La méthode du compteur

La mesure du temps en utilisant un compteur est une technique très simple. Elle consiste à échantillonner le temps avec une période d'horloge  $T_{clk}$  stable et précise. La précision de la mesure dépend de la qualité de l'horloge (faible gigue) et de la fréquence de l'horloge (faible dérive). Le compteur est caractérisé par un signal de début "Start" et de fin de cycle "Stop". L'intervalle de temps à mesurer est égal à la largeur de l'impulsion entre les deux signaux (voir figure 3.6). La valeur de ce dernier est  $T_i = N \cdot T_{clk}$ . La technique du compteur est limitée par la fréquence maximale du compteur. Elle est aussi limitée par la technologie utilisée ainsi que par la dynamique de mesure. En effet, la complexité du compteur augmente avec la dynamique de la mesure. Les compteurs parallèles peuvent atteindre des fréquences de l'ordre de quelques GHz avec certaines technologies CMOS et BiCMOS [118–120] et des centaines de MHz avec la synthèse FPGA. La méthode du compteur est largement utilisée dans les applications non exigeantes au niveau résolution, elle est plutôt adaptée aux mesures des événements lents avec une précision modérée (de l'ordre de la microseconde). Les chronogrammes illustrés dans la figure 3.6 montrent que le compteur est synchronisé par l'horloge de mesure, autrement dit le compteur ne peut pas détecter des événements qui se trouvent à l'intérieur de la période d'horloge même si les signaux de début et de fin de cycle sont asynchrones. Par conséquent, l'erreur de mesure (1 LSB) est de l'ordre d'une période d'horloge, elle diminue avec l'augmentation de la fréquence de cette dernière.

#### 3.3.4.5 Conversion temps-tension

Le convertisseur de temps en tension est une méthode analogique [121, 122]. Elle est basée sur la charge d'une capacité par un courant continu contrôlé par un interrupteur. Cette dernière est activée par une impulsion, la largeur de l'impulsion est égale à l'intervalle de temps que l'on veut mesurer. La tension aux bornes de la capacité est proportionnelle à l'intervalle de temps. Un convertisseur analogique-numérique (A/D) est utilisé en amont pour récupérer la valeur numérique de la tension et par conséquent, la valeur de l'intervalle de temps (voir la

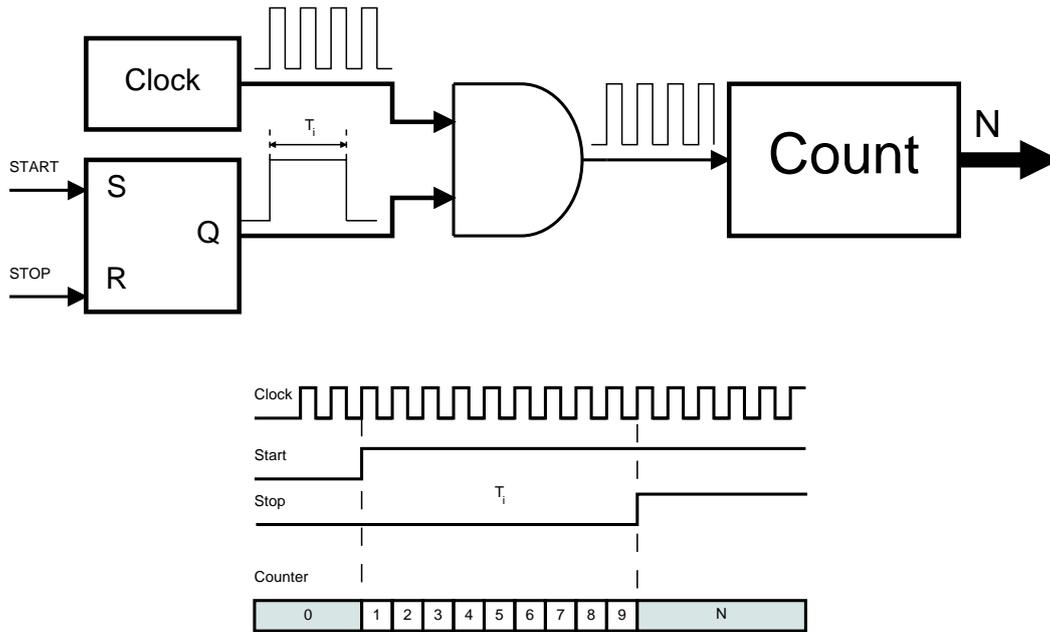


FIGURE 3.6 – Schéma bloc du principe du compteur et les chronogrammes des signaux de mesure du temps

figure 3.7). La tension aux bornes de la capacité est donnée par l'équation (3.7). Le cycle de mesure débute par une décharge de la capacité par l'intermédiaire d'un signal d'initialisation (voir la méthode de Vernier) synchronisé par le front montant de l'impulsion du contrôle de l'interrupteur. L'acquisition de la valeur par le convertisseur A/D est synchronisée par le front descendant de l'impulsion de la commande.

$$\Delta V = \frac{I}{C} \cdot T_i \quad (3.14)$$

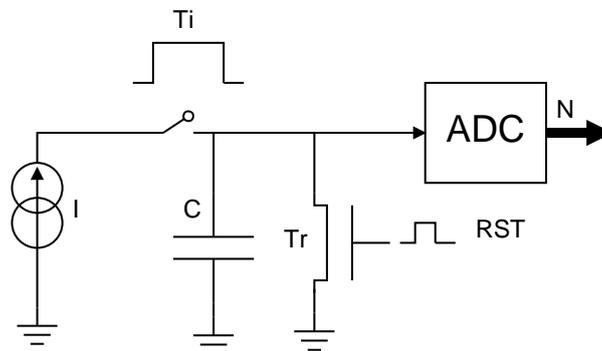


FIGURE 3.7 – Schéma de principe de la conversion temps-tension

La technique peut atteindre une précision de l'ordre de la picoseconde. Elle est limitée par le temps d'intégration du courant et par la dynamique de mesure, mais elle est également sensible

à la tension maximale de la décharge de la capacité (tension d'alimentation). Exemple, pour une tension de 5 volts avec une source de courant de 1 mA et une capacité de 1 nF, l'intervalle de temps maximal est de 5  $\mu$ s. La méthode est sensible aux bruits, à la qualité du signal de commande et à la stabilité en température de la source d'alimentation.

### 3.3.4.6 Méthode de Vernier

La méthode analogique de Vernier (Pierre Vernier, 1980) est la version améliorée du compteur. Elle consiste à élargir l'intervalle de temps  $T_i$  avant d'effectuer l'opération du comptage. La figure 3.8 montre le schéma bloc de la technique de Vernier. Le dispositif est constitué de deux horloges de fréquences  $f_1$  et  $f_2$ , avec  $f_1$  la fréquence basse et  $f_2$  la fréquence haute. Les deux sources d'horloges sont activées (ou désactivées) par la détection de coïncidence entre les deux fronts montants des deux horloges. La première horloge est déclenchée sur le front montant du signal "Start" et par conséquent démarre le premier compteur (Count 1), la seconde par le front montant du signal "Stop" qui déclenche en même temps le deuxième compteur (Count 2). Les deux horloges Clock 1 et Clock 2 ont des fréquences très proches. En effet, l'écart en fréquence est semblable à un décalage temporel (phase) entre les deux horloges, les fronts montants des deux horloges coïncident après une période de temps. La période de coïncidence dépend du décalage entre les deux fréquences, en pratique la différence peut être inférieure à 1 % ( $f_1 = f_2 + \frac{f_1}{100}$ ).

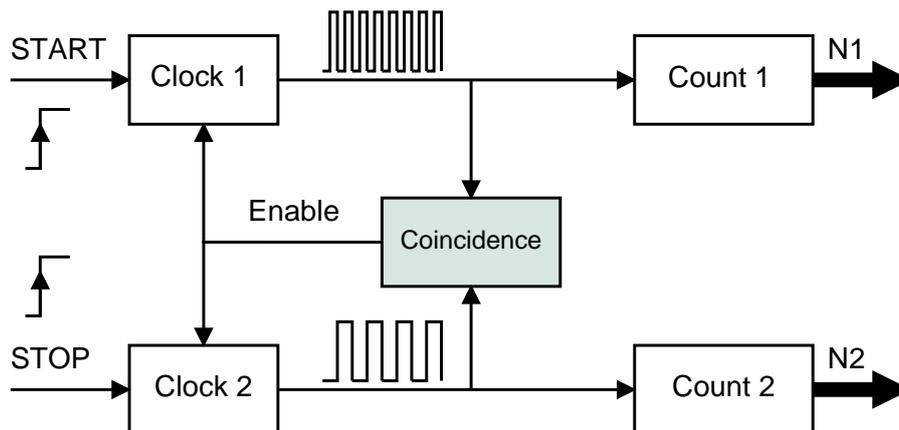


FIGURE 3.8 – Schéma de principe de la technique de mesure du temps de Vernier

L'intervalle de temps de la méthode est donnée par l'équation (3.15). La méthode peut atteindre une résolution de l'ordre de la picoseconde. La résolution  $r$  de la méthode est égale à la différence des deux périodes des horloges ( $r = T_1 - T_2$ ). La dynamique maximale de mesure vaut  $\frac{T_1 \cdot T_2}{r}$ .

$$T_i = (N_1 - 1)T_1 - (N_2 - 1)T_2 \quad (3.15)$$

La période  $T_1 = N_1 \cdot T_1$  avec  $T_1 = \frac{1}{f_1}$ .

La période  $T_2 = N_2 \cdot T_2$  avec  $T_2 = \frac{1}{f_2}$ .

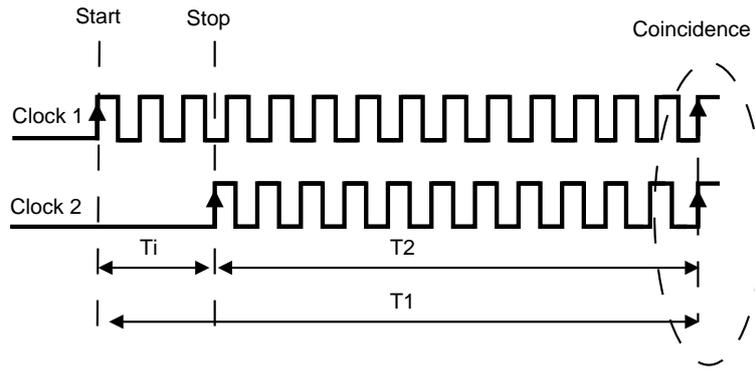


FIGURE 3.9 – Chronogrammes de la technique de mesure de l'intervalle de temps  $T_i$  de Vernier

### 3.3.4.7 Méthode à lignes à retard numérique

Le temps de retard numérique permet de mesurer des intervalles de temps courts. Le principe de base est simple, il consiste à utiliser des temps de propagation d'une cellule logique élémentaire (voir la figure 3.10) [123, 124]. Les cellules (B) de la figure 3.10 sont conçues afin d'avoir des temps de propagation identiques. Le temps d'intervalle vaut la valeur logique des cellules actives entre le signal de début et celui de la fin de l'impulsion. Pendant le début du cycle, la logique de sortie est nulle, lorsque le signal "Start" passe à 1, les sorties s'activent d'une manière successive avec la propagation du signal à travers les cellules. La valeur numérique du temps est égale à la valeur finale de la logique des sorties. Cette technique est utilisée pour la mesure fine du temps avec une faible dynamique. La dynamique du circuit de mesure dépend du nombre de cellules en cascade ( $T_i^{max} = N \cdot T_B$ ,  $T_B$  le temps de propagation d'une cellule). La précision de la technique peut atteindre quelques picosecondes. La mesure de la ligne à retard est influençable par la variation de la tension d'alimentation de la cellule et la température dans les technologies intégrées ASIC ou FPGA, la résolution peut atteindre la picoseconde.

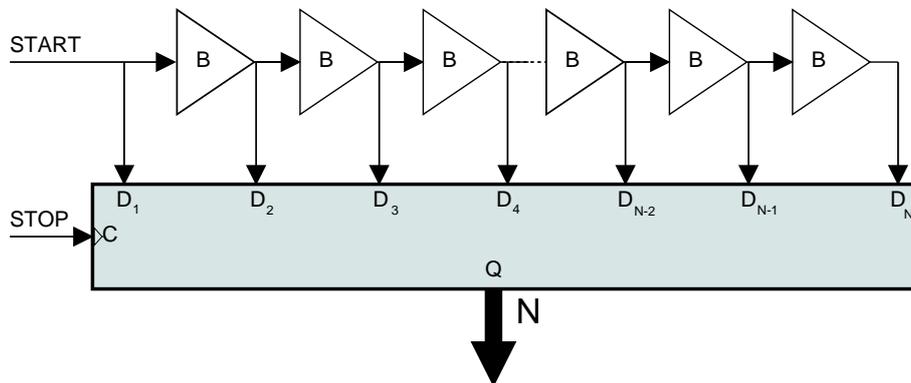


FIGURE 3.10 – Schéma bloc de la méthode lignes à retard numérique

## 3.4 Techniques de synchronisation

### 3.4.1 Algorithme de Cristian

L'algorithme de Cristian est adapté aux architectures centralisées de type Maître/Esclaves (Figure 3.11). Le principe de base consiste en l'ajustement des horloges des esclaves par rapport au maître (horloge de référence). Dans l'algorithme de Cristian, le maître est muni d'un récepteur appelé serveur de temps. Cet algorithme est développé par Cristian en 1989 [125, 126]. Périodiquement, chaque machine esclave envoie au serveur de temps (le maître) un message lui demandant l'heure courante. Ce dernier répond le plus rapidement possible par un message contenant son Heure Courante (HC), comme illustré à la figure 3.12. En première approximation, quand l'émetteur obtient la réponse, il peut se contenter de mettre son horloge à la valeur de HC. Néanmoins, cet algorithme dispose de deux limitations. La première est liée au fait que le temps est positif (les horloges ne peuvent revenir en arrière). Si l'horloge de l'émetteur est plus rapide, l'HC sera plus petit que la valeur courante, l'émetteur se contente donc de prendre la valeur de l'HC qui pourrait désynchroniser et réduire la précision du système global.

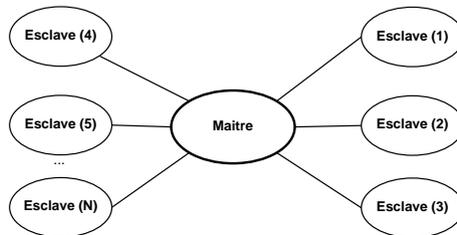


FIGURE 3.11 – Architecture basique d'un système Maître/Esclaves

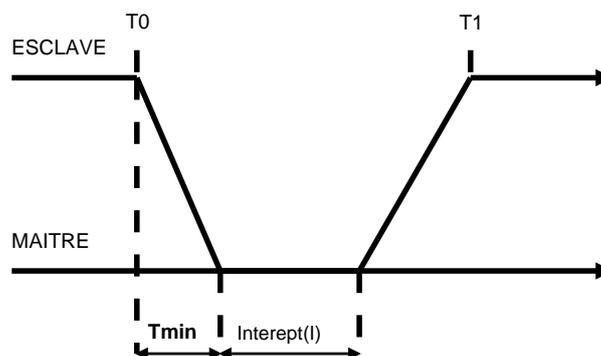


FIGURE 3.12 – Requête Maître/ Esclave de l'algorithme de Cristian

Le second problème provient du fait que la réponse du maître prend un certain temps (temps de transmission, propagation et réception) pour revenir à l'émetteur. De plus, ce délai est aléatoire et varie en fonction de la charge du réseau. La solution de Cristian pour résoudre ce problème est de le mesurer. Il est assez simple pour l'émetteur d'enregistrer précisément l'intervalle entre l'émission de la requête au serveur de temps et l'arrivée de la réponse. La date de départ,  $T_0$  et la date de fin,  $T_1$  (figure 3.12), sont mesurées par la même horloge, l'intervalle

sera relativement précis avec une précision variant de la milliseconde à la microseconde.

En l'absence de toute autre information, la meilleure estimation du temps de propagation du message est  $\frac{T_1 - T_0}{2}$ . Quand la réponse revient, on peut incrémenter de cette quantité la valeur contenue dans le message et obtenir ainsi une estimation de la valeur du temps courant du serveur. Si on connaît la valeur théorique minimum du délai de propagation, on peut en déduire d'autres propriétés de cette estimation (3.16).

$$T_{Slave} = T_{Master} + \frac{T_1 - T_0}{2} \quad (3.16)$$

On peut améliorer l'estimation avec une mesure correcte du temps en utilisant le serveur de temps pour gérer l'interruption et traiter la demande. Soit "I" le temps de gestion de l'interruption (Fig. 3.12). Alors la partie de l'intervalle de  $T_0$  à  $T_1$  qui concerne la propagation du message est  $T_1 - T_0 - I$ . La meilleure estimation du temps d'un aller simple est la moitié de cette expression en supposant que la liaison est symétrique. Pour améliorer encore la précision, Cristian a suggéré de réaliser une série de mesures. Toutes les mesures dans lesquelles  $T_1 - T_0$  excède une valeur de seuil sont éliminées en raison d'une probable congestion du réseau. Pour obtenir un meilleur résultat, on peut soit calculer la moyenne des estimations des mesures précédentes, soit considérer que la réponse rapide (temps minimal) de l'esclave est plus précise. La précision de l'algorithme de Cristian est donnée par l'équation (3.17).

$$Accuracy = \pm \frac{T_1 - T_0}{2} - T_{min} \quad (3.17)$$

### 3.4.2 Algorithme de Berkeley

Dans l'algorithme de Cristian, le serveur de temps est passif. Les autres machines lui demandent l'heure périodiquement. Il se contente de répondre à leurs demandes. Dans l'algorithme de Berkeley, c'est exactement l'approche inverse qui est prise (Gusella et Zatti, 1989). Ici, le serveur de temps est actif et scrute périodiquement chaque machine pour lui demander l'heure. À partir des réponses, il calcule l'écart moyen du temps et demande à toutes les autres machines d'avancer leurs horloges ou de les retarder. Dans cette technique, l'heure de la référence temporelle doit être régulièrement réajustée manuellement, ceci est le point faible de l'algorithme. Ci-dessous les étapes d'exécution de l'algorithme :

1. Le Serveur (la référence) demande à toutes les autres machines (les esclaves) leurs valeurs d'horloges
2. Les machines répondent avec une trame qui contient la valeur de l'horloge  $C_i$  et l'instant de mesure  $t_i$  ( $C_i(t_i)$ )(récupération des horloges)
3. Le serveur dit à chacun comment ajuster son horloge (envoi des écarts d'ajustements)

Le maître demande la valeur de l'horloge  $C_i$  pour chaque machine (étape (1)). Durant l'étape (2), les machines clients répondent au serveur en indiquant de combien de temps elles sont en avance ou en retard par rapport à lui. A partir de ces valeurs, le maître calcule le temps moyen et indique à chaque machine de combien de temps la valeur de son horloge doit être modifiée.

### 3.4.3 Protocole FTSP

Le protocole FTSP (Flooding Time Synchronization Protocol) [127, 128] est un protocole de synchronisation du temps dans un réseau distribué. Il permet d'obtenir une précision de l'ordre de la microseconde. Il est adapté à une architecture maître/esclaves, tous les esclaves sont synchronisés par rapport au maître. Le maître envoie le message de synchronisation à tous les nœuds du réseau. Le destinataire mémorise son horloge locale durant la réception du message. Ayant à la fois le temps de transmission et l'envoi du message, le récepteur peut ensuite estimer le décalage de l'horloge. Le FTSP utilise une estimation à régression linéaire des offsets. L'erreur d'offset du protocole augmente d'une façon quadratique en fonction de la distance d'un nœud avec le maître. En revanche, le protocole "PulseSync" [129] permet de maintenir une erreur relativement constante en fonction de la distance avec le maître, il présente aussi une bonne précision par rapport au FTSP de l'ordre de quelques dizaines de picosecondes.

### 3.4.4 Synchronisation par Ethernet PTP

PTP (Precision Time Protocol) est un protocole de synchronisation de temps dans un réseau distribué avec une précision de l'ordre d'une dizaine de nanosecondes (IEEE 1588) [130–132]. Il utilise la technique TWTT (Two-Way Time Transfer) : Communication bidirectionnelle entre les points du réseau durant la phase de synchronisation et l'estimation des écarts d'ajustements des horloges. La figure 3.13 montre l'échange des paquets TWTT entre l'horloge maître et un esclave. Le maître ordonne à l'esclave de déterminer l'offset relatif et d'effectuer les corrections. On suppose  $C_M(t)$  l'horloge du maître,  $C_S(t)$  celle de l'esclave et  $\theta$  l'offset entre le maître et l'esclave, alors :

$$C_S(t) = C_M(t) + \theta \quad (3.18)$$

Le principe de la technique est illustré dans la figure 3.13. Elle montre le principe de base du protocole de synchronisation PTP. On considère  $d_1$  et  $d_2$ ,  $d_1$  l'intervalle de temps entre l'instant où le paquet a quitté le maître et l'instant d'arrivée du paquet à l'esclave.  $d_2$  l'intervalle de temps entre l'instant où le paquet a quitté l'esclave et l'instant d'arrivée du paquet au maître. Chaque mesure du temps ( $t_i$ ) reflète l'intervalle de temps de départ ou d'arrivée des paquets de synchronisation. La synchronisation du temps par le protocole PTP peut atteindre des précisions de l'ordre de la nanoseconde.

Le maître envoie le paquet "PTP Synch" au temps  $t_1$ , il arrive à l'esclave au temps  $t_2$ . En utilisant le "timestamp"  $t_2 - t_1$ , l'esclave peut être synchronisé au maître. En revanche cette différence introduit aussi les temps de propagation entre le maître et l'esclave  $d_1$ . Pour résoudre ce problème l'esclave envoie la trame "Delay-Req" à l'instant  $t_3$  qui arrive au maître à l'instant  $t_4$ , après le temps de propagation  $d_2$ . Puis, le maître retourne le "timestamp"  $t_4$  en utilisant la trame "Delay-Resp". L'offset entre l'horloge du maître et l'esclave est donné par l'équation (3.19).

$$\theta = \frac{(t_2 - t_1) - (t_4 - t_3) + d_2 - d_1}{2} \quad (3.19)$$

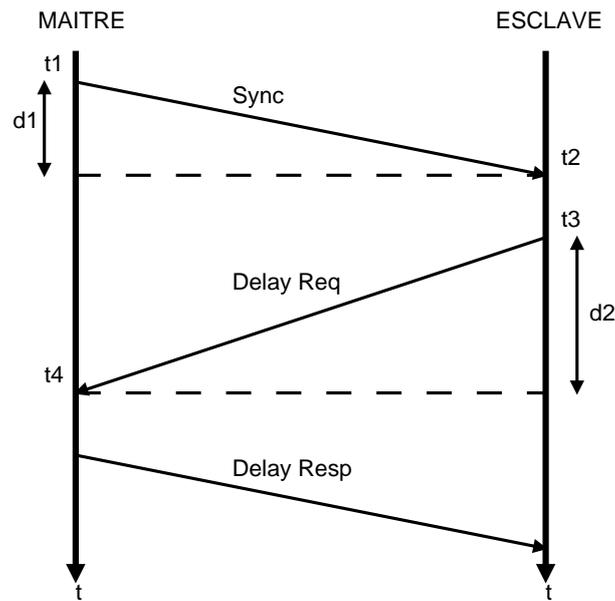


FIGURE 3.13 – Architecture du protocole de synchronisation PTP

Le protocole PTP suppose la symétrie entre le Maître et l'esclave ( $d_1 \simeq d_2$ ) [133] qui n'est pas assurée dans le cas pratique [134]. Autrement dit, le temps de propagation entre le maître et l'esclave est identique à celui entre l'esclave et le maître, alors  $d_1 = d_2$  et l'offset  $\theta$  est égale à :

$$\theta = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \quad (3.20)$$

On considère comme exemple une horloge maître  $C_1$  à l'instant  $t_1 = 10 : 00$ , et l'horloge  $C_2$  de l'esclave à l'instant  $t_2$  égal à  $10 : 30$  (Offset =  $\theta = C_2(t_2) - C_1(t_1) = 0 : 30$ ). L'objectif du protocole est de trouver la valeur de l'offset permettant de synchroniser les deux horloges. On suppose que le délai de transmission est égal au délai de la réception ( $d_1 = d_2 = d = 3s$ ). Le temps entre la réception du maître et la transmission de l'esclave est fixé à  $5s$  ( $t_3 - t_2 = d_{23} = 5s$ ). Ci-dessous la procédure de calcul des écarts et l'ajustement des horloges.

- Le maître envoie la valeur du temps ( $t_1$ ) à l'esclave à l'instant  $t_1 = 10 : 00$  (Synch)
- L'esclave reçoit le paquet à l'instant  $t_2 = t_1 + d = 10 : 30 + 00 : 03 = 10 : 33$
- L'esclave envoie la trame "Delay-Req" à l'instant  $t_3 = t_2 + d_{23} = 10 : 33 + 00 : 05 = 10 : 38$
- Le maître reçoit la trame à l'instant  $t_4 = t_3 + d = 10 : 38 + 00 : 03 = 10 : 41$ , puis le maître calcul l'offset en appliquant la formule (3.21)
- On constate bien que l'esclave est en avance de  $30s$  ( $C_2(t_1) - C_2(t_2)$ ) par rapport au maître. Le maître envoie la valeur de l'offset pour ajuster l'horloge de l'esclave.

$$\theta = \frac{(t_2 - t_1) - (t_4 - t_3) + d_2 - d_1}{2} \quad (3.21)$$

$$\theta = \frac{(10 : 33 - 10 : 00) - (10 : 11 - 10 : 38)}{2} = \frac{00 : 60}{2} = 00 : 30 \quad (3.22)$$

Le tableau 3.2 illustre la précision du protocole PTP en fonction du type de liaison physique (type du câble et de l'équipement réseau utilisé). La précision minimale de synchronisation du protocole PTP est de l'ordre de 9 ns dans le cas de l'utilisation d'un câble croisé, elle est le double dans le cas d'utilisation d'un "switch" dans un réseau chargé. La précision de synchronisation ne dépend pas uniquement du protocole, mais également de la charge du réseau. Ceci est dû à l'imprécision d'estimation des écarts d'ajustements des horloges dans le réseau. De plus la précision diminue avec l'augmentation de la période de synchronisation. Pour un câble croisé, elle passe de 146 ns en valeur maximale pour  $T_{sync} = 1 s$ , à 215 ns pour  $T_{sync} = 2 s$ . L'erreur est minimale lorsqu'on utilise le signal physique PPS d'une durée d'une seconde pour aider à la synchronisation du protocole PPT.

TABLE 3.2 – Erreur d'offset  $(\mu, \sigma)$ (ns) du protocole PTP/PPS en fonction du type de lien de communication entre les nœuds et la durée de la période de synchronisation

Type	$T_{sync}(s)$	$\mu(ns)$	$\sigma(ns)$	$Max(ns)$
PPT, câble croisé	1	13	39	146
PPT, câble croisé	2	9	52	215
PPT, switch	2	11	91	344
PPT, switch, chargé	2	18	97	348
Signal PPS	1	23	18	89

### 3.4.5 Synchronisation par Ethernet WR

Le projet White Rabbit (WR) est un projet multi-laboratoire et multi-entreprise pour obtenir une meilleure précision dans le domaine du transfert du temps. Le projet est l'extension du protocole PTP (voir la section précédente) qui permet d'obtenir une précision inférieure à la nanoseconde avec une portée maximale de 10 km [135, 136].

#### 3.4.5.1 Principe de synchronisation du WR

Le WR est l'extension de la méthode de synchronisation par Ethernet (PTP) pour le calcul et l'ajustement des horloges, il consiste à extraire une horloge des données en utilisant une PLL numérique accompagnée d'un circuit TDC de calcul de phase à haute précision pour la mesure fine d'un retard inférieur à une période d'horloge. La technique de mesure du temps est le mélange entre la technique du compteur et l'élargissement de l'intervalle de temps. La calibration et la compensation de l'offset sont obtenues grâce au protocole PTP. L'utilisation des deux techniques existantes permet au WR d'obtenir une meilleure précision. Les limitations du protocole PTP se situent au niveau de la résolution de la mesure de l'offset entre le maître et l'esclave. Dans le protocole WR, la fréquence du compteur est de 125 MHz équivalent à une période de 8 ns permettant de synchroniser le système avec une précision de quelques dizaines

de picosecondes. L'erreur de synchronisation du protocole augmente avec la distance, comme dans le cas du projet LHAASO (Large High Altitude Air Shower Observatory) [137,138]. Cette erreur due à l'asymétrie du lien entre le maître et les esclaves s'accroît avec l'augmentation de la distance.

### 3.4.5.2 Modèle du retard dans le réseau WR

Pour estimer l'offset entre deux nœuds, on doit connaître avec précision le retard entre le maître et l'esclave  $d_{MS}$  et entre l'esclave et le maître  $d_{SM}$ , le retard aller-retour  $d_{MM}$  (Master to Master Delay) est la somme des deux. Dans un système de synchronisation, incluant le protocole PTP, la liaison est supposée symétrique entre le maître et l'esclave, par conséquent on peut déduire que le retard aller est la moitié du retard  $d_{MM}$  ( $d_{MS} = d_{SM} = \frac{d_{MM}}{2}$ ). En réalité, cette supposition réduit la précision de synchronisation, mais la plupart des systèmes de synchronisation minimise l'erreur de l'asymétrie de la liaison en remettant en phase les horloges d'une façon permanente. Dans un réseau WR, les liens entre le maître et un nœud du réseau sont de la même longueur pour minimiser l'effet de l'asymétrie. Dans le cas d'utilisation des fibres optiques avec deux longueurs d'ondes différentes (une pour l'aller, l'autre pour le signal du retour dans le même câble), un coefficient de correction est utilisé pour réduire l'effet de l'asymétrie. Le modèle utilisé pour le calcul du retard entre le maître et l'esclave est présenté par l'équation (3.23). En général, le retard  $d_{MS}$  est exprimé comme étant la somme du retard fixe du circuit de transmission et du maître  $\Delta_{txm}$ . Le retard du canal de transmission  $\delta_{ms}$  et du retard fixe du circuit de réception de l'esclave  $\Delta_{rxs}$  sont décrits dans les équations (3.23) et (3.24).

$$d_{MS} = \Delta_{txm} + \delta_{MS} + \Delta_{rxs} \quad (3.23)$$

Le retard entre l'esclave  $d_{SM}$  et le maître peut être exprimé en analogie avec  $d_{MS}$ .

$$d_{SM} = \Delta_{txs} + \delta_{SM} + \Delta_{rxm} \quad (3.24)$$

Le WR utilise actuellement une fibre optique de type WDM (Wavelength Multiplexed) pour les deux directions de communication, avec une distance identique dans les deux sens. La différence des retards est causée par la vitesse de propagation dans les deux fibres. Le réseau WR utilise en particulier les longueurs d'ondes suivantes :  $\lambda_{tx} = 1550$  nm et  $\lambda_{rx} = 1310$  nm. Le coefficient du retard relatif introduit est défini par l'équation :

$$\alpha = \frac{\delta_{MS}}{\delta_{SM}} - 1 = \frac{n_{1550}}{n_{1310}} \quad (3.25)$$

$n_{1550}, n_{1310}$  sont les indices de réfraction.

Pour les applications qui demandent une grande précision en utilisant la fibre,  $\alpha$  peut être déterminé en laboratoire à partir des mesures du retard  $d_{MM}$ . Ayant  $\alpha$ , le retard précis est donné par la formule suivante :

$$\begin{aligned}\Delta &= \Delta_{txm} + \Delta_{txs} + \Delta_{rxm} \\ d_{MM} &= \Delta + \delta_{MS} + \delta_{SM} \\ d_{MS} &= \frac{1 + \alpha}{2 + \alpha}(d_{MM} - \Delta)\end{aligned}\tag{3.26}$$

Après avoir calculé le retard entre le maître et l'esclave, on peut en déduire l'offset d'horloge  $\beta_{MS}$  entre le maître et l'esclave dans l'équation (3.27).

$$\beta_{MS} = t_2 - t_1 + d_{MS}\tag{3.27}$$

### 3.4.5.3 Méthode de mesure de phase dans le WR

Le circuit de synchronisation de WR utilise un circuit numérique de la mesure de la phase avec une grande précision inférieure à la nanoseconde. Elle est basée sur le circuit DDMTD (Digital Dual Mixer Time Différence) permettant de mesurer des intervalles de temps avec une résolution inférieure à une période d'horloge. La technique existe en circuit analogique ou numérique. La figure 3.14 illustre le schéma bloc de mesure de phase. L'objectif de la technique est de convertir un intervalle de temps long en un intervalle de temps lent pour ensuite utiliser un compteur avec une fréquence modérée pour la mesure du temps. Cette technique permet de mesurer des intervalles de temps inférieurs à une période d'horloge (la phase entre le signal "Start" et "Stop").

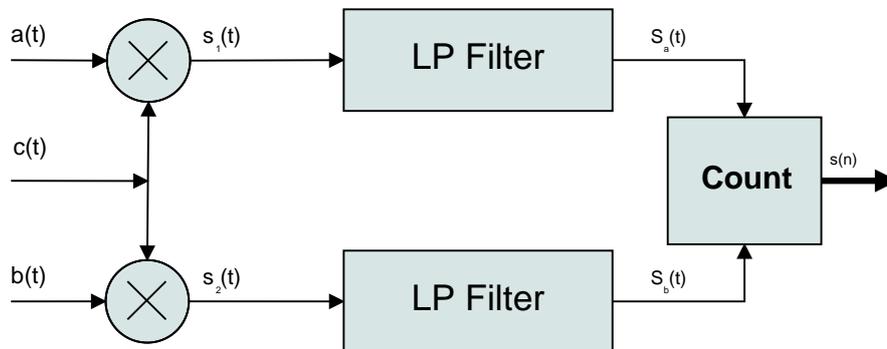


FIGURE 3.14 – DDMTD analogique

La mesure de phase (un intervalle de temps inférieur à la période du signal) entre deux signaux rapides  $a(t)$  et  $b(t)$  n'est pas toujours faisable dans le cas pratique. Le principe le plus utilisé dans l'instrumentation de fréquence est de mesurer la différence entre deux horloges basées sur un compteur (voir les techniques de mesures des intervalles du temps illustrées dans la section précédente). L'intérêt de la méthode illustrée dans la figure 3.14 est de gagner en précision en ralentissant la vitesse des signaux rapides. Un exemple d'un DDMTD analogique est illustré dans la figure 3.14. On considère deux signaux d'événements  $a(t)$  et  $b(t)$  ("Start" et "Stop") avec la même fréquence ( $f_a = f_b = f_{clk}$ ) mais avec des phases différentes  $\phi_a$  et  $\phi_b$  successivement. On génère une horloge  $c(t)$  à partir du signal  $a(t)$  avec une fréquence  $f_{offset}$  proche de la fréquence de référence. Le choix de l'écart entre les  $f_{clk}$  et  $f_{offset}$  fixe la précision de la mesure. En pratique, l'écart est inférieur à 1%.

$$\begin{cases} a(t) = \cos(2\pi f_{clk}t + \phi_a) \\ b(t) = \cos(2\pi f_{clk}t + \phi_b) \\ c(t) = \cos(2\pi f_{offset}t + \phi_{offset}) \end{cases} \quad (3.28)$$

Les signaux  $s_1(t)$  et  $s_2(t)$  sont données par les équations (3.29) et (3.30) :

$$\begin{aligned} s_1(t) &= \cos(2\pi f_{clk}t + \phi_a)\cos(2\pi f_{offset}t + \phi_{offset}) \\ &= \frac{1}{2}\cos(2\pi t(f_{clk} + f_{offset}) + \phi_a + \phi_{offset}) + \\ &\quad \frac{1}{2}\cos(2\pi t(f_{clk} - f_{offset}) + \phi_a - \phi_{offset}) \end{aligned} \quad (3.29)$$

$$\begin{aligned} s_2(t) &= \cos(2\pi f_{clk}t + \phi_b)\cos(2\pi f_{offset}t + \phi_{offset}) \\ &= \frac{1}{2}\cos(2\pi t(f_{clk} + f_{offset}) + \phi_b + \phi_{offset}) + \\ &\quad \frac{1}{2}\cos(2\pi t(f_{clk} - f_{offset}) + \phi_b - \phi_{offset}) \end{aligned} \quad (3.30)$$

Les signaux à la sortie des deux multiplieurs  $s_a(t)$  et  $s_b(t)$  ont deux composantes : une composante basse fréquence ( $f_{clk} - f_{offset}$ ) et l'autre haute fréquence ( $f_{clk} + f_{offset}$ ). Le filtre passe-bas permet de supprimer la composante HF (Haute fréquence). Le signal comprend alors la composante basse fréquence avec l'information sur le déphasage entre les deux signaux à l'entrée  $a(t)$  et  $b(t)$  décrit dans l'équation (3.31).

$$\begin{cases} S_a(t) \propto \frac{1}{2} \cdot \cos(2 \cdot \pi \cdot t \cdot (f_{clk} - f_{offset}) + \phi_a - \phi_{offset}) \\ S_b(t) \propto \frac{1}{2} \cdot \cos(2 \cdot \pi \cdot t \cdot (f_{clk} - f_{offset}) + \phi_b - \phi_{offset}) \end{cases} \quad (3.31)$$

On prend comme exemple une fréquence d'horloge de deux signaux  $a(t)$  et  $b(t)$  ( $f_{clk} = 125$  MHz) et une fréquence proche  $f_{offset} = 124,99$  MHz, les signaux  $S_a(t)$  et  $S_b(t)$  auront une fréquence de 10 kHz après multiplication et filtrage avec une phase  $\phi_a - \phi_{offset}$  et  $\phi_b - \phi_{offset}$  respectivement. Après cette transformation, il est facile de mesurer la phase entre les deux horloges lentes de fréquence 10 kHz avec une grande précision en utilisant un simple compteur. La figure 3.15 illustre les sorties  $s_a(t)$  et  $s_b(t)$  pour des entrées  $a(t)$  et  $b(t)$  avec une fréquence 125 MHz et un déphasage de  $\pi/4$  avec un filtre passe-bas d'ordre 8 de type "butterworth". La fréquence d'offset est définie par l'équation (3.32). Il est clair dans la figure 3.15 que la période des signaux des sorties augmentent lorsque "k" diminue et par conséquent l'intervalle de temps entre les maximums des deux signaux (la phase). Pour  $k = 0,1$  %, la période des signaux  $S_a(t)$  et  $S_b(t)$  est de l'ordre de 8  $\mu s$ , en sachant que la période des signaux à l'entrée est de 8 ns (fréquence de 125 MHz), soit un rapport de 1000 sur la précision de la technique par rapport à la méthode du compteur de base.

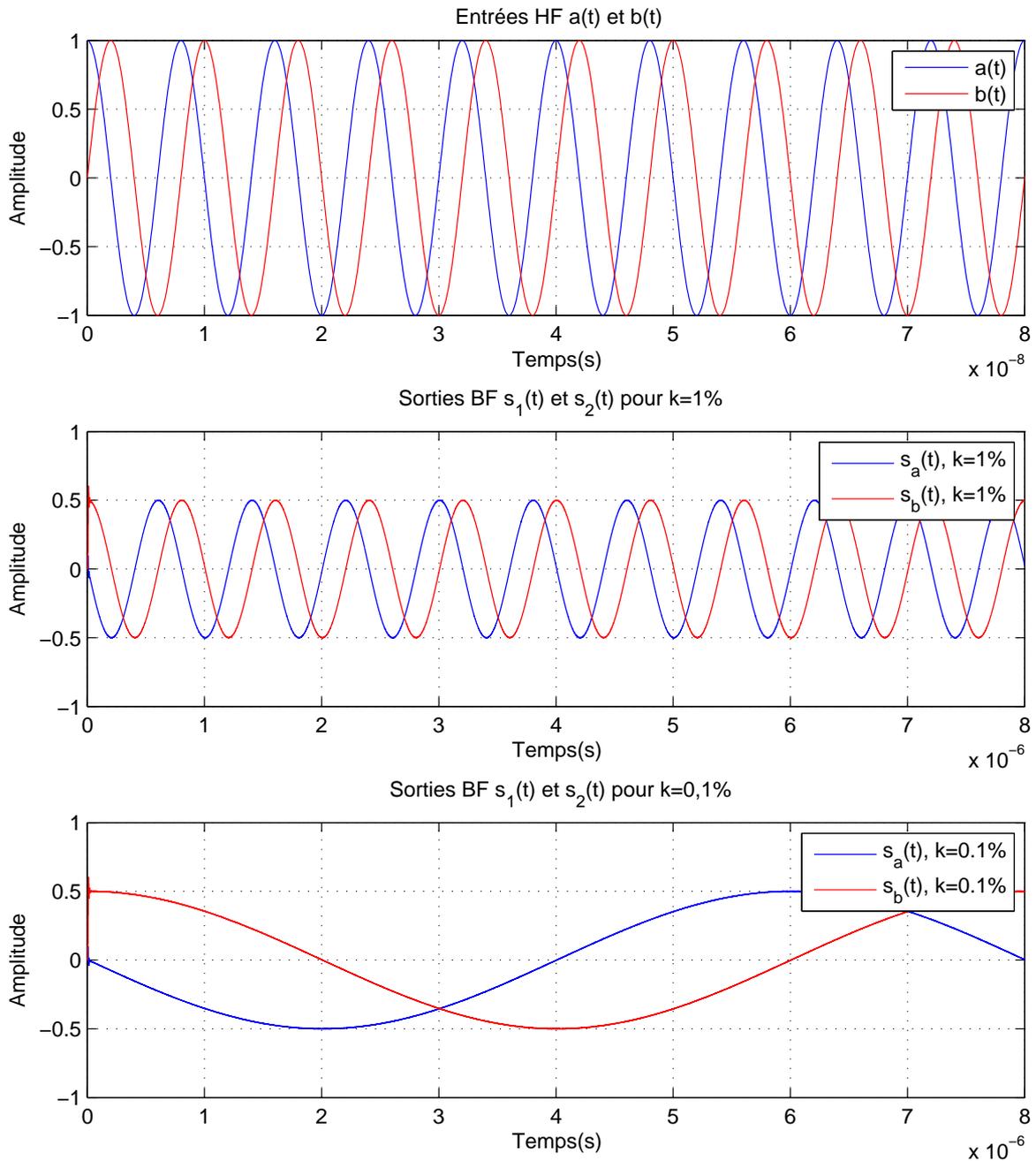


FIGURE 3.15 – Signaux d’entrées et de sorties du circuit de mesure de phase analogique pour divers rapport de fréquence

$$f_{offset} = f_{clk} \cdot (1 - k), k \ll 1 \quad (3.32)$$

On constate dans la figure 3.16 le régime transitoire des signaux à la sortie des filtres passe-bas et la présence d'un retard constant identique pour les deux filtres. Il augmente en fonction de l'ordre du filtre. Ce dernier est compensé automatiquement dans le cas où les deux filtres sont parfaitement identiques, car le compteur mesure l'intervalle de temps entre les signaux à la sortie des filtres. Dans le cas pratique, cette hypothèse n'est pas valide, car les deux filtres sont conçus à partir de composants discrets analogiques. On peut envisager dans ces conditions une procédure de calibration du compteur avec des signaux connus aux entrées. Dans la version numérique, la précision est meilleure (absence du filtre).

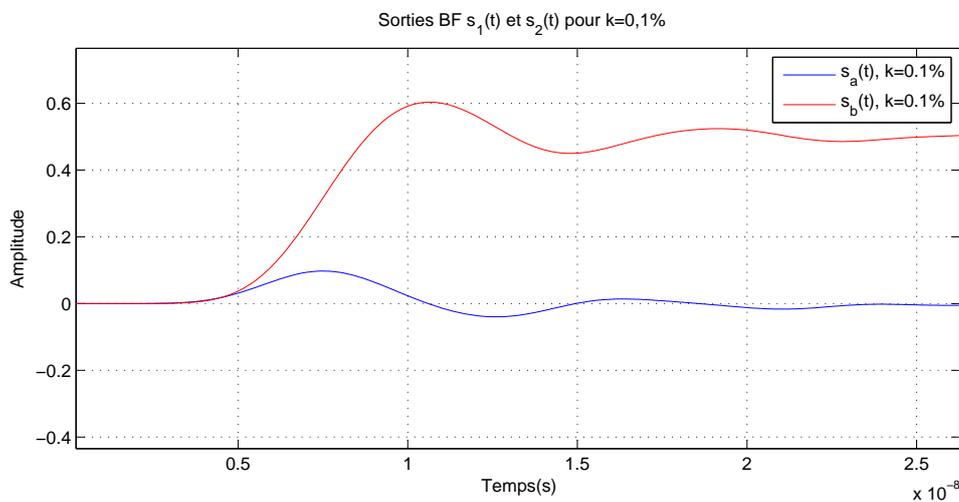


FIGURE 3.16 – Retard du filtre passe bas

Le principe de la DMTD numérique utilisé dans le protocole WR sur 1 bit reste identique à la méthode analogique, il est illustré dans la figure 3.14. Les multiplieurs peuvent être remplacés par des simples bascules avec une entrée "D" du signal  $a(t)$  et une horloge  $c(t)$ . Le bloc de filtrage est remplacé par un simple circuit anti-rebond numérique. L'architecture [135] est entièrement implémentable sur FPGA ou un ASIC numérique. La précision du protocole WR est meilleure que 100 ps. La figure 3.17 montre le signal PPS entre le maître et l'esclave avec une moyenne  $\mu = 30,4$  ps et une variance  $\sigma = 92,54$ .

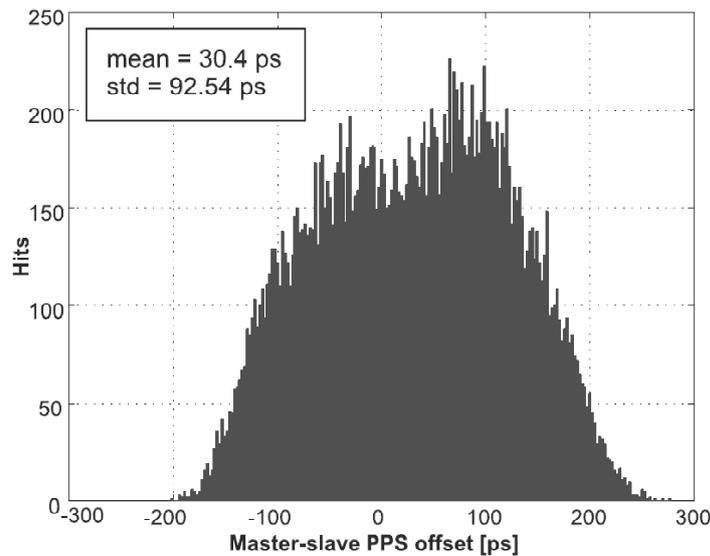


FIGURE 3.17 – Histogramme du PPS entre le maître et l'esclave

### 3.5 Conclusion

Nous avons abordé dans le chapitre l'instrumentation de précision du temps et de la fréquence ainsi que les types, les divers protocoles de synchronisation et leurs précisions. On distingue deux types importantes de synchronisation : La synchronisation du temps ou de la fréquence. On a vu que la précision de synchronisation de la fréquence est plus importante (inférieure à la picoseconde pour les horloges atomiques avec un rapport de  $10^5$  de précision par rapport à l'imprécision temporelle), en revanche la précision ultime du protocole actuel de synchronisation du temps est de l'ordre d'une centaine de picosecondes dans un réseau distribué (Protocole PTP amélioré ou WR). De plus, pour la synchronisation du temps qui nous intéresse en particulier pour notre système, la précision est limitée par l'erreur aléatoire induite par l'asymétrie du canal (chemin aller-retour) qui croît avec la distance entre la référence et les autres machines (dans une topologie en étoile). Nous avons également illustré plusieurs techniques pour obtenir la mesure du temps avec une précision allant de la microseconde à la nanoseconde (compteur, Nutt). Cependant, peu de techniques de mesure peuvent monter en précision soit en dessous de la nanoseconde (Ligne à retard, Vernier et Conversion temps-tension). De plus, la précision d'un système de synchronisation est déterminée par la précision de la mesure des retards dans le réseau et la stratégie utilisée pour la compensation des horloges (offsets et skews). Dans la suite du rapport, on va se consacrer à l'étude de conception et l'implémentation matérielle et logicielle du système de synchronisation appliquée à un réseau d'antennes multi-reparties, dans une architecture distribuée en étoile. On utilisera un circuit TDC avec une précision inférieure à 20 ps avec une dynamique de l'ordre de  $\pm 7$  secondes (14 s) sur 40 bits, il sera décrit en détails dans les chapitres suivants, ainsi que l'architecture globale du système de synchronisation.



# Étude et conception du système de synchronisation appliqué au Radiohéliographe de Nançay

## 4.1 Introduction

Dans cette partie on va étudier les exigences nécessaires à appliquer au convertisseur A/D pour une utilisation au Radiohéliographe de Nançay (NRH). La bande de fréquence d'analyse est comprise entre  $f_{low}=150$  MHz et  $f_{high}=450$  MHz, soit une bande passante de 300 MHz et une fréquence maximale de conversion de 450 MHz. Le but est de déterminer la résolution binaire du convertisseur A/D pour le nouveau système NRH pour une conversion large bande. L'étude est effectuée en se basant sur les mesures de puissance de l'instrument NRH actuel et le spectre de puissance du bruit du ciel. La deuxième partie du chapitre sera consacrée au calcul de la précision temporelle du système de synchronisation avec le principe de l'interférométrie.

## 4.2 Les caractéristiques du convertisseur A/D pour le radiohéliographe de Nançay

### 4.2.1 Calcul de la puissance du bruit du ciel

On va calculer la puissance du bruit du ciel dans la bande utilisée par l'instrument pour en déduire la puissance minimum que l'on doit détecter par le numériseur, soit la résolution du convertisseur A/D. La figure 4.1 montre l'évolution de la température du ciel  $T_{sky}(K)$  en fonction de la fréquence en Hz (voir la figure 3 dans [139]). Dans la bande de fréquence utilisée par le Radiohéliographe de Nançay, la température décroît linéairement (en échelle logarithmique), elle est proportionnelle à  $\lambda^{2.55} = \left(\frac{c}{f}\right)^{2.55}$  (voir l'étude de spécifications du convertisseur A/D de l'instrument LOFAR [140] y compris la majorité des équations utilisées durant notre étude). L'équation (4.2) décrit la puissance du bruit du ciel en fonction de la  $f_{low}$  et  $f_{high}$  après l'approximation dans la bande du NRH.

L'équation (4.1) illustre la puissance de ce dernier ( $P_{sky}$ ) dans la bande utile du NRH en

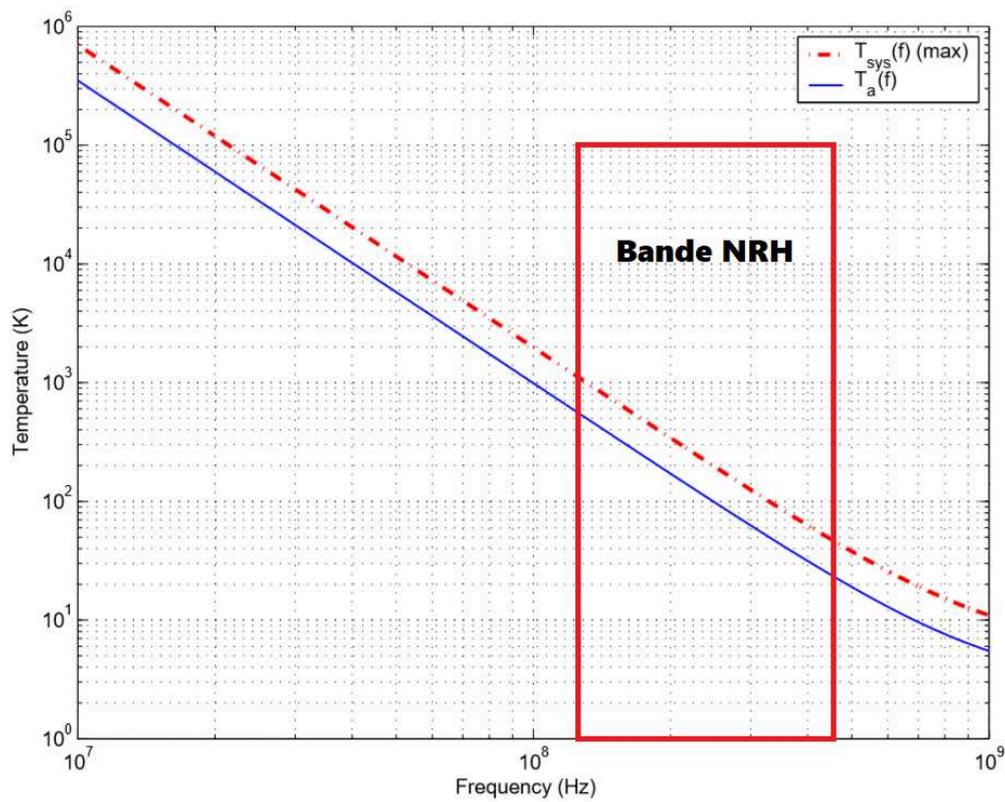


FIGURE 4.1 –  $T_{sky}(K)$  en fonction de la fréquence (Hz) (voir la figure 3 dans [139]). La température  $T_{sky}(K)$  est proportionnelle à  $\lambda^{2.55}$  dans la bande NRH (entre 150 MHz et 450 MHz environ).

fonction de sa température ( $T_{sky}$ ) [139, 141].

$$P_{sky} \simeq \int_{f_{low}}^{f_{high}} k \cdot T_{sky} df \quad (4.1)$$

$$P_{sky} \simeq \frac{60 \cdot k \cdot c^{2.55}}{1.55} \cdot [f_{low}^{-1.55} - f_{high}^{-1.55}] \quad (4.2)$$

avec :  $k$  la constante de Boltzmann ( $1.38 \cdot 10^{-23} \text{ J/K}$ ),  
 $f_{low}$  la fréquence basse et  $f_{high}$  la fréquence haute de la bande,  
la bande passante  $B$  est définie par  $f_{high} - f_{low}$ .

La figure 4.2 montre la décroissance de la puissance du ciel en dBm en fonction de la fréquence basse de la bande ( $f_{low}$ ). La courbe est équivalente à celle étudiée dans le dimensionnement du numériseur de l'instrument LOFAR [140]. Pour  $f_{low}=150 \text{ MHz}$  et  $B=400 \text{ MHz}$  ( $f_{high} = 550 \text{ MHz}$ ), la puissance équivalente du ciel  $P_{sky}$  est égale à  $-93.91 \text{ dBm}$ . Pour  $f_{low}=150 \text{ MHz}$  et  $B=300 \text{ MHz}$  ( $f_{high} = 450 \text{ MHz}$ ), la puissance équivalente du ciel  $P_{sky}$  est égale à  $-94 \text{ dBm}$ . Ces deux valeurs extrêmes (limite maximale et minimale de la puissance en fonction de la fréquence maximale et minimale) permettront de calculer la résolution du convertisseur A/D. On constate un effet négligeable de la bande passante  $B$  sur la puissance  $P_{sky}$  et une décroissance à  $1/f$  pour une fréquence  $f_{low} \leq 200 \text{ MHz}$ , cette tendance est due à notre approximation dans la bande NRH illustrée dans l'équation (4.2).

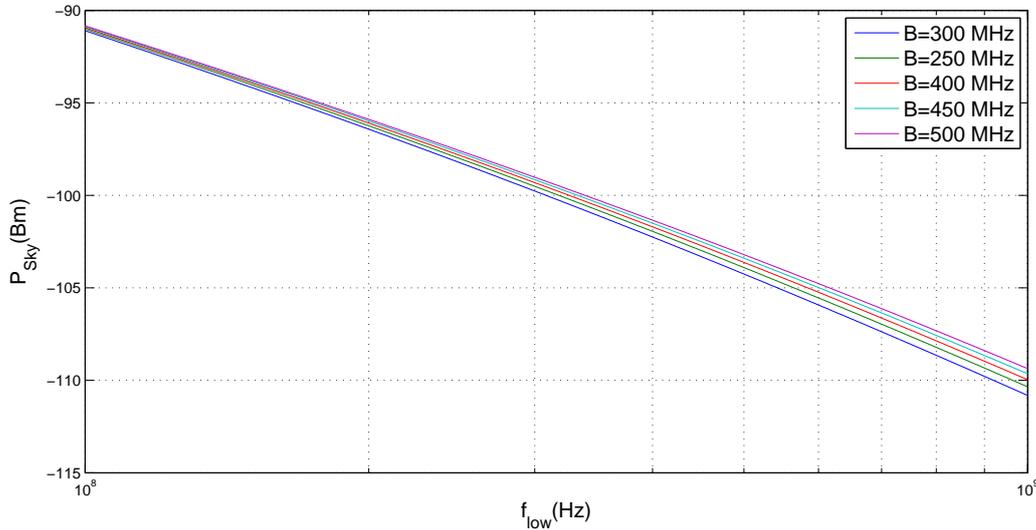


FIGURE 4.2 –  $P_{sky}(W)$  en fonction de la fréquence  $f_{low}(Hz)$  pour différentes bandes de fréquence ( $300 \text{ MHz} \leq B \leq 500 \text{ MHz}$ )

### 4.2.2 Calcul de la résolution binaire du convertisseur A/D

Pour calculer le nombre de bits requis par le convertisseur A/D ( $n_r$ ) en fonction de la fréquence basse du NRH (ou la bande B), on va effectuer dans un premier temps un ensemble de calculs en se basant sur l'activité du soleil calme et actif dans la bande NRH en prenant en compte la puissance  $P_{sky}$  calculée dans la section précédente. L'objectif est de calculer la puissance du soleil reçue au pied des antennes avec des mesures de puissance réelles pour ensuite en déduire le nombre de bits  $n_r$  à partir des équations 4.3, 4.4 et 4.5. On déduit le nombre de bits requis ( $n_r$ ) en fonction de la puissance du ciel  $P_{sky}$  et le  $SNR$  (Signal-to-Noise Ratio).  $n_s$  étant le nombre de bits nécessaire pour compenser le bruit des interférences et les pertes intrinsèques du convertisseur A/D (en pratique, on prend  $n_s$  supérieur ou égal à 1), on compense ainsi les pertes représentées par le nombre de bits effectifs (ENOB) [142–144].

$$D = P_{RFI} - P_{sky} \quad (4.3)$$

$$SNR = 6,02 \cdot n + 1,76(dB) \quad (4.4)$$

$$n_r = \frac{D - 1,76}{6,02} + n_s \quad (4.5)$$

Le flux solaire est donné en Jansky ( $Jy$ ), ainsi qu'en dB ( $W/m^2/Hz$ ), ou au format logarithmique, 0 dB ( $Jy$ ) = -260 dB( $W/m^2/Hz$ ). Le Watt par mètre carré par Hertz ( $W/m^2/Hz$ ) est équivalent au Joule par mètre carré ( $J/m^2$ ). Le SFU (Solar Flux Unit) (voir le tableau 2.1, page 44 [145]) ( $1 SFU = 10^{-22}W/m^2/Hz$ ,  $1 Jy = 10^{-26}W/m^2/Hz$ ).

La puissance en sortie d'une antenne dépend de plusieurs facteurs. Tout d'abord, de la source observée (soleil, radiogalaxie CygA ou autres sources), de son activité (actif/calme) et de la fréquence (voir la figure 1 qui montre les données d'observation du flux solaire du 2 juin 2002 par le télescope Learmonth IPS/US Air Force à la fréquence 245 MHz dans [146]). Puis, elle dépend aussi de la surface collectrice équivalente des antennes utilisées avec l'interféromètre. La surface effective  $A_{eff}$  est donnée par l'équation (4.6) en connaissant le gain d'antenne [147].

$$A_{eff} = g_a \cdot \frac{\lambda^2}{4\pi} (m^2) \quad (4.6)$$

$g_a$  : Gain de l'antenne,  $\lambda$  : Longueur d'onde observée

Pour une antenne parabolique la formule de la surface effective est donnée par l'équation (4.7), les résultats pour les antennes du NRH sont donnés dans le tableau 4.1 [148].

$$A_{eff} = \Gamma \cdot \pi \cdot \frac{D^2}{4} (m^2) \quad (4.7)$$

$\Gamma$  : le rendement de l'antenne,  $D$  : le diamètre de l'antenne

Le tableau 4.1 illustre les résultats de mesures effectués antérieurement par l'équipe NRH. Il montre la surface effective, les diamètres  $D$  des paraboles et le rendement mesuré associé à chaque antenne. On distingue trois types de parabole : Les deux paraboles Ext0 et Ext2 qui ont été ajoutées pour étendre le réseau et augmenter le nombre de points de l'interféromètre (augmenter la résolution spatiale de l'image radio), et celles du réseau NS (Nord-Sud). Le calcul de puissance prendra en considération les types de parabole.

TABLE 4.1 – La surface effective  $A_{eff}$  ( $m^2$ ) pour les antennes du NRH

Antennes	$A_{eff}(m^2)$	$D$	$\Gamma$
Ext2	43.20	10	0.55
Ext0	21.17	7	0.55
NS	10.80	5	0.55

Pour caractériser l'activité du soleil et quantifier l'énergie radio reçue, on mesure une densité de flux en fonction de la fréquence observée. Le flux du Soleil calme mesuré au NRH s'élève à 10 SFU à 164 MHz et à 30 SFU à 450 MHz. On constate un rapport de 3 du flux  $F$  (SFU) entre les deux fréquences dans le cas du soleil calme.

La température du signal  $T_s$  à la sortie d'antenne est indiquée dans l'équation (4.8). Elle dépend du flux et de la surface effective  $A_{eff}$ . Le tableau 4.2 résume la température  $T_s(k)$  du signal en fonction des limites de la bande du NRH pour divers types d'antennes constituant le réseau interféromètre.

$$T_s(k) = A_{eff} \cdot F \cdot 10^{-22} \cdot \frac{1}{k} \quad (4.8)$$

$T_s$  en degré Kelvin ( $K$ ),  $F$  = densité de flux (SFU)

TABLE 4.2 – La température  $T_s(k)$  du signal en sortie des antennes RH

	Ext2	Ext2	Ext0	Ext0	NS	NS
$A_{eff}(m^2)$	43.2	43.2	21.17	21.17	10.8	10.8
Freq(MHz)	164	450	164	450	164	450
$T_s(k)$	3130.4	9391.3	1534.1	4602.2	782.6	2347.8

La puissance du signal en sortie d'une antenne dans une bande  $B=300$  MHz ( $f_{low}=150$  MHz,  $f_{high}=450$  MHz) est donnée par l'équation (4.9). Dans la suite on va calculer la puissance  $P_s$  en fonction de l'activité du soleil. On va se focaliser sur les cas du soleil actif et du soleil calme. Dans le cas du soleil calme, le flux moyen est de l'ordre de 20 SFU. Il est de l'ordre de  $10^5$  SFU d'une très forte activité solaire. Les tableaux 4.3 et 4.4 résument la valeur de la puissance  $P_s(dBm)$  dans le cas du soleil calme et actif successivement pour diverses fréquences et types d'antennes. Le flux indiqué dans le cas du soleil calme dans les tableaux est le flux mesuré pour chaque antenne par l'équipe du NRH.

$$P_s(W) = k \cdot B \cdot T_s \quad (4.9)$$

TABLE 4.3 – La puissance  $P_s(dBm)$  du signal en sortie des antennes RH dans une bande de B=300 MHz cas du soleil calme

Soleil Calme	Ext2	Ext2	Ext0	Ext0	NS	NS
Freq(MHz)	164	450	164	450	164	450
$A_{eff}(m^2)$	43.2	43.2	21.17	21.17	10.8	10.8
F (SFU)	10	30	10	30	10	30
$T_s(k)$	3130.4	9391.3	1534.1	4602.2	782.6	2347.8
$P_s(dBm)$	-78.8739	-74.1027	-81.971	-77.200	-84.89	-80.1233

TABLE 4.4 – La puissance  $P_s(dBm)$  du signal en sortie des antennes RH dans une bande de B=300 MHz cas du soleil actif

Soleil Actif	Ext2	Ext2	Ext0	Ext0	NS	NS
Freq(MHz)	164	450	164	450	164	450
$A_{eff}(m^2)$	43.2	43.2	21.17	21.17	10.8	10.8
F (SFU)	10000	100000	10000	10000	10000	10000
$T_s(k)$	3130.4	9391.3	1534.1	4602.2	782.6	2347.8
$P_s(dBm)$	-48.8739	-48.8739	-51.9716	-51.9716	-54.89	-54.8945

On constate un écart de puissance de l'ordre de 26 dBm entre le soleil actif et calme illustré dans les tableaux 4.3 et 4.4. On constate également une baisse de 4 dBm de la puissance entre la fréquence basse (150 MHz) et la fréquence (450 MHz), cette perte en revanche ne dépend pas du type d'antennes. En résumé, la puissance moyenne émise par le soleil dans la bande utile (bande utilisée pour l'observation de la couronne solaire) dans le cas du soleil calme est égale à -79.52 dBm et -52 dBm dans le cas du soleil actif. La figure 4.3 montre la variation du nombre des bits requis ( $n_r$ ) par le convertisseur A/D en fonction de la fréquence basse ( $f_{low}(Hz)$ ) dans le cas du soleil calme et le soleil actif. La courbe est obtenue en se basant sur les équations (4.2), (4.3), (4.4), (4.5) et les mesures obtenues des puissances en fonction de l'activité du soleil. L'augmentation du nombre de bits avec la fréquence est due à la contrainte sur la qualité de la fréquence d'horloge d'échantillonnage. On verra dans la suite de la section comment calculer la précision requise sur la fréquence d'échantillonnage du convertisseur A/D.

Le nombre de bits nécessaire pour une fréquence basse  $f_{low}$  de 150 MHz (B = 300 MHz,  $f_{high} = 450$  MHz) est égal à 4,1 dans le cas du soleil calme et 8,7 pour le soleil actif (voir la figure 4.3). On constate également que l'écart en nombre de bits entre le soleil calme et le soleil actif est indépendant de la fréquence basse. Cet écart correspond à la différence de puissance d'environ 26 dBm. La figure montre également que le nombre de bits nécessaire augmente lorsque la fréquence basse augmente pour une bande B fixe, d'une autre façon le convertisseur A/D

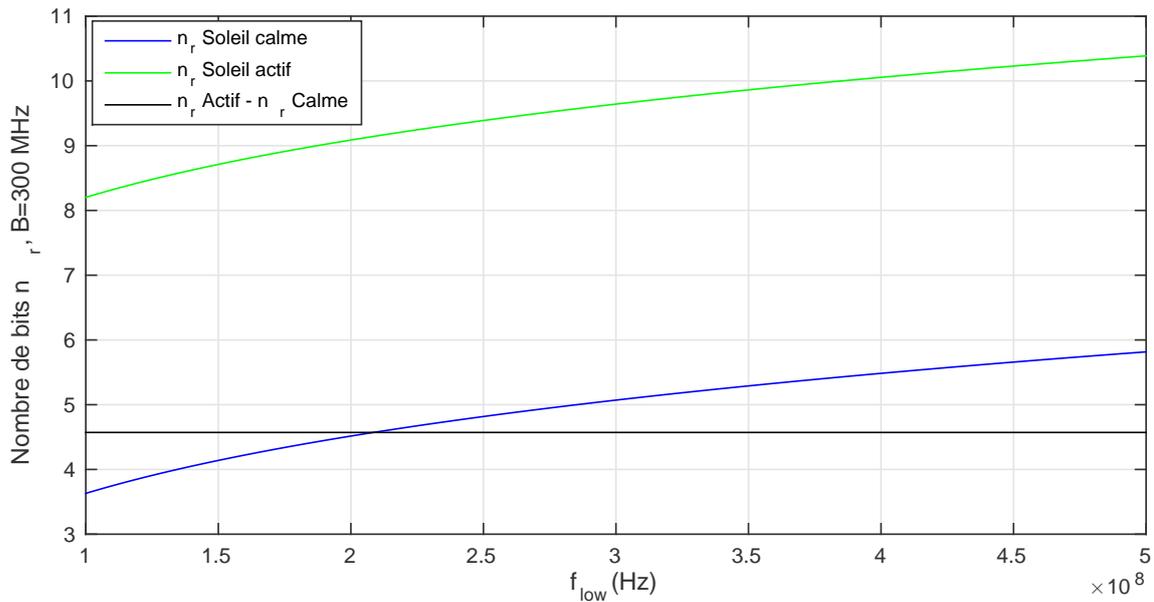


FIGURE 4.3 – Le nombre de bits requis  $n_r$  par le convertisseur A/D dans les cas du soleil actif (courbe verte) et calme (courbe bleue) en fonction de la fréquence basse  $f_{low}$  (Hz). Soleil calme (courbe bleue).

nécessite plusieurs bits supplémentaires lorsqu'on travaille dans les fréquences hautes pour la même bande. Pour un décalage de bande de 400 MHz (transition de  $f_{low}$  à  $f_{low}+400$  MHz), il correspond à une augmentation de 3 bits environ, cette augmentation en revanche ne dépend pas de l'activité solaire. Elle est fixe dans les deux cas (soleil actif ou calme).

Pour conclure, le nombre de bits  $n_r$  du convertisseur A/D du nouveau système numérique du NRH doit être supérieur à 9 bits dans les pires des cas (soleil actif) dans une bande passante de 300 MHz ( $f_{low}=150$  MHz,  $f_{high}=450$  MHz). Le nombre de bits  $n_r$  augmente linéairement avec le nombre de bits  $n_s$  consacré aux interférences (voir l'équation (4.5)), dans le cas pratique égal à 1 ou 2. Le nombre de bits dans le cas du NRH est inférieur à celui de LOFAR. La différence de bits est due à la bande utile par l'instrument (environ 30 MHz -240 MHz), dans cette bande la puissance du bruit du ciel est importante (voir la figure 4.1) et par conséquent, le nombre de bits pour couvrir la bande est important. En effet, la différence des fréquences maximum des deux instruments LOFAR et NRH est égale à 200 MHz, ce qui correspond à environ 2 bits de différence respectivement 12 et 10 bits.

### 4.2.3 Calcul de la précision de la fréquence de l'échantillonneur du convertisseur A/D

Après avoir calculé la résolution binaire du convertisseur A/D du système numérique du NRH, on va passer au calcul de la précision de l'horloge du convertisseur. Pour cela on se base sur la fréquence maximale de la bande analogique à l'entrée du convertisseur et le rapport signal

## 4.2. LES CARACTÉRISTIQUES DU CONVERTISSEUR A/D POUR LE RADIOHÉLIOGRAPHE DE NANÇAY

sur bruit SNR. L'équation (4.11) montre le SNR en fonction de la fréquence d'entrée  $f_{in}$  d'un convertisseur A/D et de la gigue  $t_j(s)$  de la fréquence d'échantillonnage [149, 150]. La figure 4.4 trace le SNR (dB) en fonction de la fréquence d'entrée  $f_{in}(Hz)$  pour différentes valeurs de la gigue  $t_j(s)$ . On constate dans un premier temps, pour une fréquence  $f_{in}(Hz)$  fixe, que le rapport SNR diminue avec l'augmentation de la gigue de l'horloge. Pour une fréquence de 500 MHz, le SNR passe de 46 dB (pour 1,6 ps) à 70 dB (pour 0,1 ps).

$$n_r = \frac{20 \cdot \log\left(\frac{1}{2 \cdot \pi \cdot f_{in} \cdot t_j}\right) - 1.76}{6.02} + n_s \quad (4.10)$$

$$SNR_{dB} = 20 \cdot \log\left(\frac{1}{2 \cdot \pi \cdot f_{in} \cdot t_j}\right) \quad (4.11)$$

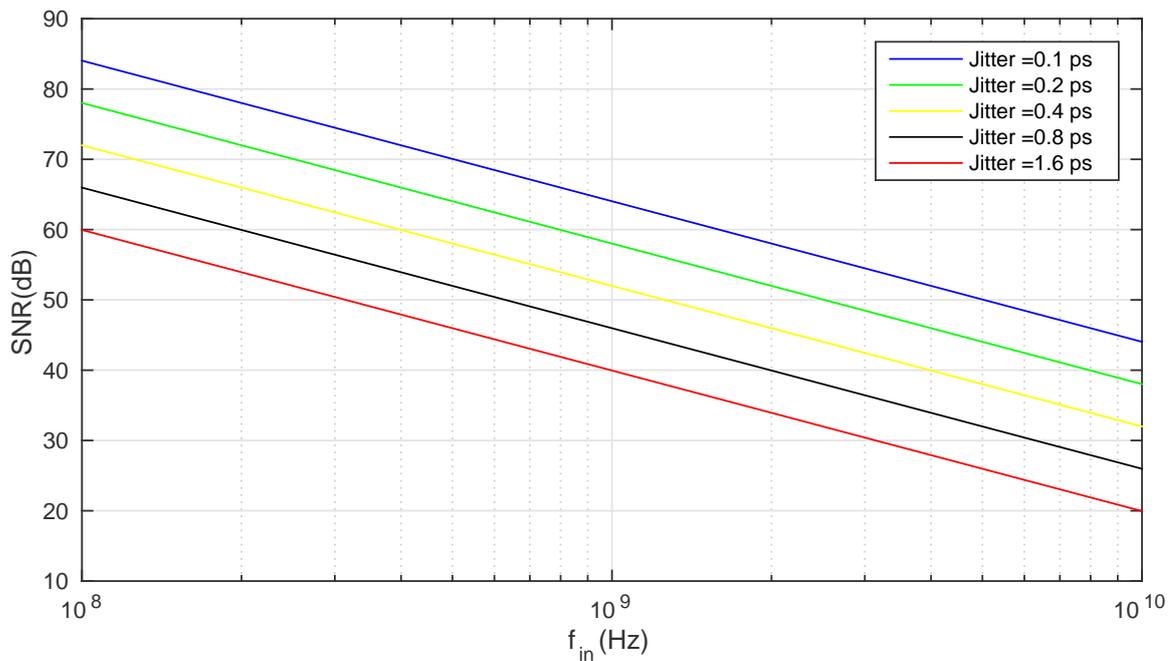


FIGURE 4.4 – Le SNR (dB) en fonction de la fréquence  $f_{in}$  du convertisseur A/D pour chaque valeur du jitter (ps)

La figure 4.5 montre le temps de la gigue  $t_j(ps)$  en fonction du nombre de bits  $n_s$  pour différents nombres de bits requis  $n_{rs}$  par le convertisseur A/D pour une fréquence maximale  $f_{in}$  égale à 450 MHz. On constate que l'exigence sur la qualité de l'horloge diminue avec la diminution du nombre de bits  $n_r$  pour la même fréquence  $f_{in}$  avec un nombre  $n_s$  fixe, le temps de la gigue  $t_j$  est égal à 4,5 ps pour  $n_s=2$  bits,  $n_r=8$  bits et  $f_{in}=450$  MHz et inférieur à 1 ps pour  $n_s=2$  bits,  $n_r=11$  bits et  $f_{in}=450$  MHz. On déduit, que pour avoir un nombre de bits élevé, il faut en contre-partie une réduction du jitter. Le SNR (ou le nombre de bits) décroît

avec la croissance de la fréquence d'entrée du convertisseur A/D. Dans la suite de la section on va aborder une autre méthode théorique pour le calcul de la précision de l'horloge en fonction de la fréquence maximale et du nombre de bits requis pour le convertisseur A/D.

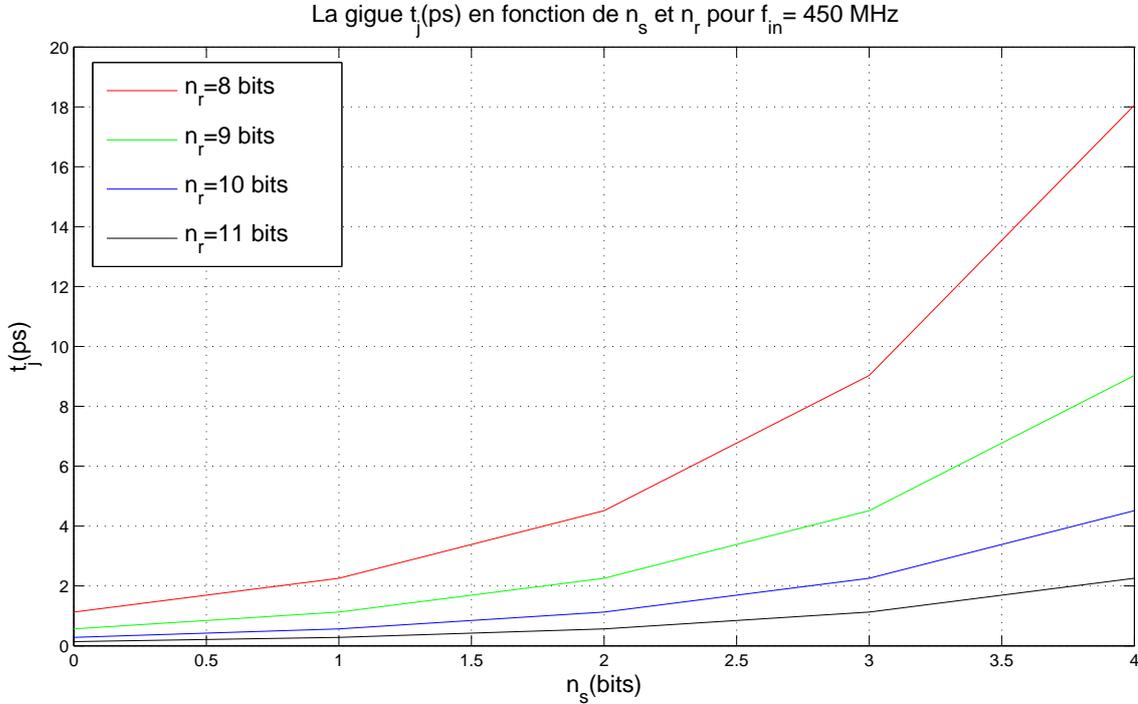


FIGURE 4.5 – Le temps de la gigue  $t_j(ps)$  en fonction du nombre de bits  $n_s$  pour différents nombres de bits requis  $n_r$  par le convertisseur A/D pour une fréquence maximale  $f_{in}$  de 450 MHz

Shinagawa, Akazawa and Wakimoto (1990) [151] avaient proposé un modèle pratique des différentes contributions des giges. Le modèle est constitué de 3 giges d'échantillonnage :  $t_{aj}$ ,  $t_{sj}$  et  $t_{ij}$ . L'erreur  $t_{aj}$  est induite par le bruit d'alimentation.  $t_{sj}$  est la gigue de la fréquence d'échantillonnage (fluctuation de la période d'échantillonnage par rapport à la fréquence de référence) du convertisseur A/D. En effet, la fréquence d'échantillonnage peut être différente de celle du convertisseur en fonction de l'architecture utilisée. Cette erreur est toujours présente dans le signal d'horloge. Elle est due au bruit de phase de l'oscillateur (l'horloge utilisé introduit un bruit de phase dans la bande utilisée qui peut être converti en gigue dans le domaine temporel).  $t_{ij}$  (Input jitter) est induit par l'erreur sur le signal d'entrée au moment du prélèvement de l'échantillon. Le temps de la gigue totale  $t_j$  est la somme quadratique des trois contributeurs. Si l'on suppose que chaque composante est indépendante de l'autre, on aura :

$$t_j = \sqrt{t_{aj}^2 + t_{sj}^2 + t_{ij}^2} \quad (4.12)$$

Lors de la modélisation des fluctuations de la tension du signal de l'horloge d'échantillonnage, chaque contribution temporelle ( $t_x$ ) de la gigue induit une erreur  $\epsilon_j$ . Elle est définie comme la somme des différentes incertitudes des tensions (voir l'équation (4.13)).

$$\epsilon_j = \epsilon_{aj} + \epsilon_{sj} + \epsilon_{ij} \quad (4.13)$$

La gigue totale est proportionnelle à la pente du signal d'entrée du convertisseur A/D. L'effet de la tension d'erreur totale  $\epsilon_j$  définit la limite maximale de la fréquence d'entrée qui correspond au maximum de variation du signal à l'échantillonneur (Slew Rate). On suppose un signal sinusoïdal suivant :

$$x(t) = X_m \cdot \sin(2\pi \cdot f \cdot t) \quad (4.14)$$

Le maximum de "Slew Rate" est égal à :

$$\frac{dx(t)}{dt} = X_m \cdot 2\pi \cdot f \cdot \cos(2\pi \cdot f \cdot t) \quad (4.15)$$

$$\left(\frac{dx(t)}{dt}\right)_{max} = X_m \cdot 2\pi \cdot f \quad (4.16)$$

Pour que l'erreur  $\epsilon_j$  n'affecte pas la précision du convertisseur A/D, elle doit être inférieure à 1/2 LSB au maximum du "slew rate". Pour un convertisseur de N bits, on a :

$$\epsilon_j = t_j \cdot \frac{dx}{dt} = \frac{1}{2} LSB = \frac{2 \cdot X_m}{2^{N+1}} \quad (4.17)$$

$\epsilon_j$ , le RMS de la tension totale de la gigue  $t_j$

$$\frac{2 \cdot X_m}{2^{N+1}} = 2 \cdot \pi \cdot f \cdot X_m \cdot t_j \quad (4.18)$$

Alors la fréquence maximale pour un jitter est donnée par :

$$f_{max} = \frac{1}{t_j \cdot \pi \cdot 2^{N+1}} \quad (4.19)$$

La figure 4.6 montre l'évolution de la gigue de l'horloge du convertisseur A/D en fonction du nombre de bits N. On constate une décroissance rapide du temps de la gigue  $t_j(ps)$  en fonction du nombre de bits N ( $n_r$ ) requis par le convertisseur A/D (voir la figure 4.6). Elle est de l'ordre de 1,4 ps pour N=8 bits et inférieure à 200 fs lorsque N est inférieur à 11 bits. D'après notre étude précédente concernant le calcul du nombre de bits pour le numériseur du NRH, le temps de la gigue  $t_j$  doit être inférieur à 0,69 ps pour N=9 bits et  $f_{max} = 450$  MHz d'après la deuxième

méthode (voir la figure 4.6).

Cependant, les résultats illustrés dans la figure 4.5 de la première méthode de calcul montrent que la valeur  $t_j$  ne dépend pas uniquement du nombre de bits requis, mais aussi du nombre de bits  $n_s$  consacré au codage des parasites. La valeur de  $n_s$  dans le cas pratique, en radio-astronomie, peut prendre une valeur supérieure ou égale à 1. En résumé, la valeur de  $t_j$  est supérieure à 1 ps lorsque  $n_s$  est supérieur ou égal à 1 bits pour  $n_r=9$  bits (courbe verte dans la figure 4.5). Néanmoins dans la même figure, lorsque  $n_s = 0$ , le  $t_j$  devient inférieur à 1 ps et par conséquent, on obtient des résultats plus proches (uniquement lorsque  $n_s = 0$  qui n'est pas envisageable dans le cas pratique). Dans notre étude on va se baser sur les résultats de la figure 4.5 qui prennent en considération la valeur de  $n_s$ .

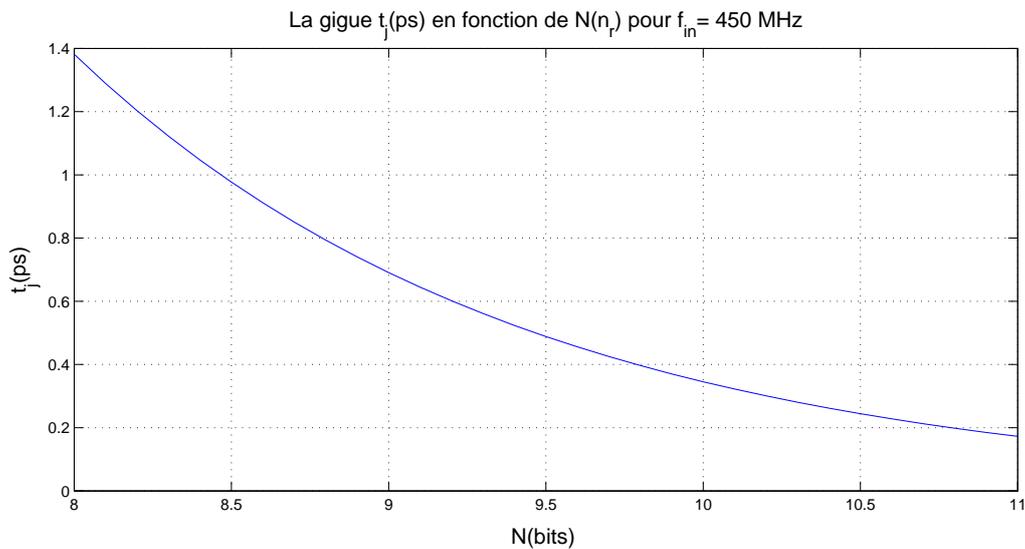


FIGURE 4.6 – La gigue maximum en fonction du nombre de bits pour une fréquence maximale de 450 MHz

### 4.3 Précision du système de synchronisation

La synchronisation est une étape cruciale dans un réseau distribué. La distribution des horloges à faible gigue pour les convertisseurs A/D intégrés aux pieds des antennes nécessite une synchronisation globale du système. Le problème de synchronisation est induit par la différence de longueur des connexions entre les antennes et le système central (le maître). Dans le cas du NRH, les distances varient entre 10 m et 3200 m. Pour étudier l'effet de la désynchronisation dans un réseau d'antennes dédiées à l'interférométrie radio [152] pour l'imagerie de la couronne solaire, on va étudier l'effet de l'imprécision de synchronisation sur le taux de corrélation pour ensuite en déduire la précision minimale à appliquer aux horloges du système. La figure 4.7 montre le schéma de base d'un interféromètre radio. On considère les paramètres suivants :

- $D$  : la longueur de la ligne de base
- $\theta$  : l'angle de pointage à partir de la direction du zénith
- $\tau_g$  : le retard géométrique  $= \frac{D}{c} \cdot \sin(\theta)$

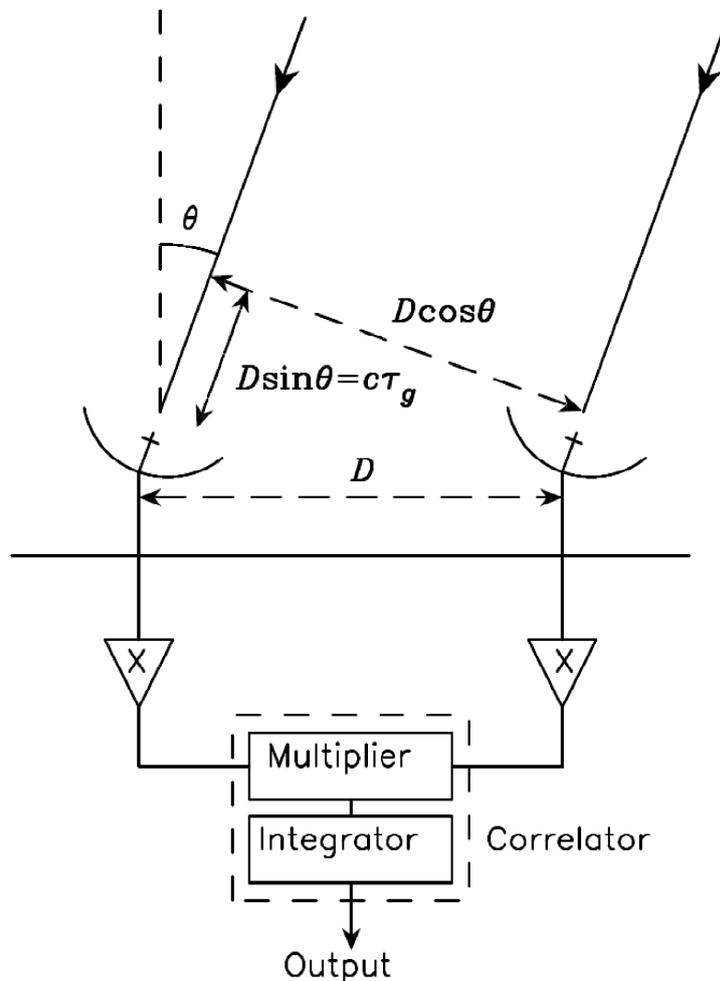


FIGURE 4.7 – Schéma de principe d'un interféromètre radio

L'onde plane qui vient de la source atteint l'antenne à droite avec un retard de  $\tau_g$ . On prend deux signaux  $x_1(t)$  et  $x_2(t)$  avec :

$$x_1(t) = E_1 \cdot \cos(2\pi \cdot f \cdot t) \quad (4.20)$$

$$x_2(t) = E_2 \cdot \cos(2\pi \cdot f \cdot (t - \tau_g)) \quad (4.21)$$

La sortie du multiplicateur  $F(t)$  :

$$F(t) = x_1(t) \cdot x_2(t) = E_1 \cdot \cos(2\pi \cdot f \cdot t) \cdot E_2 \cdot \cos(2\pi \cdot f \cdot (t - \tau_g)) \quad (4.22)$$

$$F(t) = \frac{E_1 \cdot E_2}{2} [\cos(2\pi \cdot f \cdot \tau_g) + \cos(4\pi \cdot f \cdot t - 2\pi \cdot f \cdot \tau_g)] \quad (4.23)$$

L'intégration dans une période  $T$ , très grande devant l'inverse de la fréquence observée, permet de récupérer la composante basse fréquence du signal ( $\omega\tau_g$ ,  $\omega = 2\pi f$ ) et filtrer la composante haute fréquence ( $2\omega t$ ). La sortie de l'intégrateur  $r(\tau_g)$  est proportionnelle à l'intensité du signal et dépend de la position de la source dans le ciel :

$$r(\tau_g) \propto \int_0^T F(t) \cdot dt = \frac{E_1 \cdot E_2}{2} [\cos(2\pi \cdot f \cdot \tau_g)] \quad (4.24)$$

Le radiohéliographe actuel de Nançay utilise deux cartes indépendantes, une pour la mesure de puissance et l'autre pour effectuer la corrélation 1 bit entre deux voies dans une période d'intégration de 5 ms. Chaque pixel de l'image est constitué des deux paramètres (puissance et corrélation). On considère  $\tau_1$  et  $\tau_2$ , deux erreurs induites par la compensation des retards instrumentaux entre les deux antennes. Si les deux erreurs sont identiques et égales à  $\tau$  ( $\tau_1 = \tau_2 = \tau$ ), l'équation (4.24) implique que le taux de corrélation ne change pas. Dans la suite, on va étudier l'effet d'une erreur différente pour chaque antenne ( $\tau_1 \neq \tau_2$ ). La sortie du multiplicateur  $F(t)$  est donnée par :

$$\begin{aligned} F(t) &= x_1(t - \tau_1) \cdot x_2(t - \tau_2) = E_1 \cdot \cos(2\pi \cdot f \cdot (t - \tau_1)) \cdot E_2 \cdot \cos(2\pi \cdot f \cdot (t - \tau_2 - \tau_g)) \\ &\Rightarrow F(t) = E_1 \cdot \cos(2\pi \cdot f \cdot (t - \tau_1)) \cdot E_2 \cdot \cos(2\pi \cdot f \cdot ((t - \tau_1) + \tau_1 - \tau_2 - \tau_g)) \end{aligned} \quad (4.25)$$

La comparaison entre l'équation (4.25) et (4.23) montre qu'on remplace  $t$  par  $t - \tau_1$  et  $\tau_g$  par  $\tau_g - d\tau_g$ . Après insertion de cette dernière substitution dans l'équation (4.24) et (4.27).

$$r(\tau_g) \propto \int_0^T F(t) \cdot dt = \frac{E_1 \cdot E_2}{2} \left[ \cos(2\pi \cdot f \cdot (\tau_g - d\tau_g)) \right] \quad (4.26)$$

$$r(\tau_g) \propto \int_0^T F(t).dt = \frac{E_1.E_1}{2} \left[ \cos(2\pi.f.\tau_g(1 - \frac{d\tau_g}{\tau_g})) \right] \quad (4.27)$$

L'équation 4.27 indique l'effet de l'erreur du retard sur le taux de corrélation  $r(\tau_g)$ . Le tau ne dépend que des puissances des deux signaux et de l'erreur différentielle du retard  $d\tau_g$ . Cette erreur n'a pas d'influence sur l'intensité du signal (amplitude), mais sur le taux de corrélation, il change avec la différence de l'erreur entre les deux antennes. Dans le nouveau système de synchronisation, la différence  $d\tau_g$  reste constante pendant la période de synchronisation. Afin de simuler l'effet de l'erreur sur le taux de corrélation, on va calculer  $d\tau_g$  en fonction du tau de corrélation  $\tau_g$  dans (4.28). A partir de l'équation (4.27), le retard différentiel  $d\tau_g$  dépend du taux de corrélation selon :

$$d\tau_g = \tau_g \cdot \left[ 1 - \frac{a \cos(\frac{r(\tau_g)}{E_0})}{2.\pi.f.\tau_g} \right], \quad \text{avec } E_0 = \frac{E_1.E_1}{2} \quad (4.28)$$

Les courbes de la figure 4.8 illustrent l'évolution de l'erreur différentielle  $d\tau_g$  en fonction du taux de corrélation pour différentes fréquences. Le taux de corrélation  $r(\tau_g)$ (%) est normalisé par rapport à la puissance. Le taux de corrélation diminue avec l'augmentation de l'erreur différentielle. Par contre, l'erreur diminue avec l'augmentation de la fréquence. La précision du système de synchronisation (décalage entre les horloges) à 5% de perte de taux de corrélation (95%) est égale à 112 ps, elle est à 50 ps à 1 % (figure 4.8) pour une fréquence maximale de 450 MHz.

## 4.4 Conclusion

Nous avons effectué dans ce chapitre une étude des spécifications du nouveau système numérique NRH basée sur les calculs théoriques et les mesures de puissance et les caractéristiques de l'instrument actuel. Nous avons défini la résolution binaire du convertisseur A/D dans le cas du soleil calme ou actif. Nous avons calculé les paramètres requis pour l'horloge du convertisseur A/D. La suite du chapitre sera focalisée sur le calcul de la précision temporelle du système de synchronisation autorisée sans dégrader le taux de corrélation (facteur de dégradation) qui peut induire la perte de la source visée dans le ciel. Pour conclure, la précision théorique requise par le système de synchronisation du NRH est de l'ordre de 50 ps et de 50,74 ps en comparaison avec les autres instruments radio pour une bande de 300 MHz avec une erreur de 1%. Le tableau 4.4 résume les spécifications du nouveau système numérique NRH.

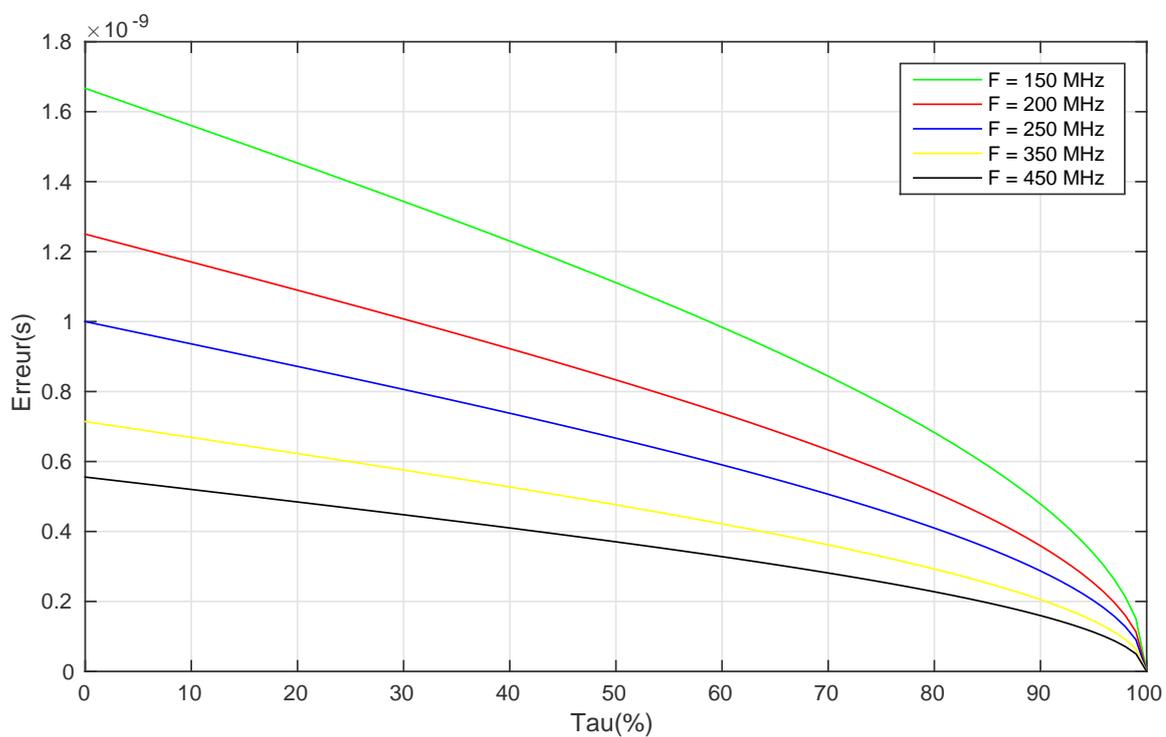


FIGURE 4.8 – L'erreur différentielle  $d\tau_g(s)$  en fonction du taux de corrélation exprimé en pour cent

TABLE 4.5 – Les spécifications du nouveau système numérique du Radiohéliographe de Nançay

Bande passante	300 MHz
Fréquence maximale	450 MHz
Fréquence minimale	150 MHz
Fréquence d'échantillonnage du convertisseur A/D	1 GHz
Résolution binaire du convertisseur A/D @ 450 MHz (soleil actif @ $10^5$ SFU)	$\geq 9$ bits
Résolution binaire du convertisseur A/D @ 450 MHz (soleil calme @ 20 SFU)	$\geq 4,5$ bits
Gigue d'horloge @450 MHz	$\leq 1$ ps
Précision de synchronisation calculée @ 5%	$\sim 112$ ps
Précision de synchronisation calculée @ 1%	$\sim 50$ ps
Précision de synchronisation @ 500 MHz par rapport à ALMA/EVLA	$\sim 15,62 \cdot 500/2000 = 62,48$ ps
Précision de synchronisation @ 500 MHz par rapport à LOFAR/EMBRACE	$\sim 78 \cdot 250/500 = 39$ ps
Précision de synchronisation @ 500 MHz moyenne	$\sim 50,74$ ps

# Algorithme de synchronisation et calibration de la chaîne

## 5.1 Introduction

De nombreux algorithmes de synchronisation du temps ont été développés, mais peu d'entre eux peuvent atteindre la précision des sous-ns dans les réseaux distribués [20, 153–155]. En outre, plusieurs sources d'erreurs de système limitent la précision, tels que les retards asymétriques, les retards de propagation, le nombre de nœuds et d'autres sources de retards induites par la méthode de mesure. Les principales limitations du NRH actuel sont les erreurs de retards d'offsets introduites par l'étalonnage manuel entre chaque antenne et le récepteur. En outre, le temps de commutation entre chaque fréquence analysée introduit une latence dans les images solaires, qui réduit également le rapport signal sur bruit. Le système de synchronisation proposé du nouveau NRH numérique nécessite une fréquence d'horloge du convertisseur A/D de 1 GHz au minimum pour chaque antenne (échantillonnage pour une bande complète). Ce nouveau système permettra d'acquérir simultanément des images radio à différentes fréquences avec une résolution configurable. Les principaux problèmes de ce système sont : la distribution de l'horloge haute fréquence requise pour chaque antenne, les différentes longueurs de câbles entre les antennes individuelles et le récepteur, qui nécessitent une synchronisation de temps globale avec une précision de l'ordre de la sous-ns.

L'objectif de ce chapitre est de parvenir au développement du premier prototype du système de synchronisation du temps du NRH de Nançay. Le Chapitre est constitué des étapes suivantes : Nous proposons, dans un premier temps, la théorie de l'algorithme basée sur le calcul et la compensation des retards à la synchronisation d'un réseau d'antennes multi-reparties. L'algorithme est appliqué à une architecture Maître/Esclaves dans le cas de liens symétrique ou asymétrique. Ensuite, nous testerons l'algorithme en nous basant sur les paramètres de travaux antérieurs dans un réseau distribué avec des longueurs différentes de câbles pour illustrer les performances et les limites de l'algorithme. La deuxième partie est dédiée à la description de la procédure de calibration des retards aller/retour entre le maître et les esclaves en utilisant l'estimateur gaussien.

## 5.2 Algorithme de calcul et compensation des offsets

### 5.2.1 Introduction

Dans les réseaux filaires et sans fils, la synchronisation est un service important pour divers gamme d'applications dans les systèmes distribués. Cela comprend l'interférométrie radio. Cependant, les recherches dans la littérature montrent que de nombreux protocoles distribués ne peuvent pas satisfaire la précision de la synchronisation du temps en dessous de la nanoseconde en raison de l'asymétrie entre le trajet aller et le retour, des jitters accumulés et éventuellement du protocole utilisé pour ajuster les décalages. Cette section décrit une stratégie de synchronisation du temps dans les réseaux distribués. La technique proposée nous permet de minimiser les retards asymétriques avec une précision inférieure à une nanoseconde afin d'assurer une synchronisation de temps globale. On va commencer par une étude théorique de l'algorithme qui permet de calculer les écarts d'ajustement des horloges et la minimisation de l'erreur due à l'asymétrie dans le canal, suivi des simulations des performances et limitations de l'algorithme. On terminera par une conclusion.

### 5.2.2 Ajustement des offsets symétriques

Dans cette section, nous décrivons l'algorithme de synchronisation basé sur les recherches précédentes dans les réseaux distribués. Considérons un réseau d'interféromètre avec  $N$  antennes ( $i = 1, 2, \dots, N$ ) et que chaque antenne (c.à.d. chaque esclave) aie un retard aller-retour  $d_{MM}^i$  (figure 5.1).

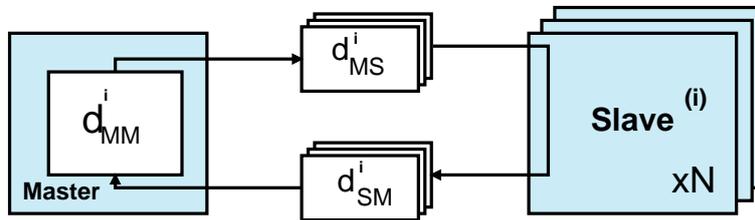


FIGURE 5.1 – Modèle du retard entre le maître et l'esclave (Antenne).

$$d_{MM}^i = d_{MS}^i + d_{SM}^i \quad (5.1)$$

L'équation (5.1) présente le modèle de base des retards entre le maître et les esclaves. Le retard aller  $d_{MS}^i$  pour l'antenne ( $i$ ) inclut le retard de transmission (le retard que prend le maître pour envoyer le signal), la somme du temps de propagation et du délai nécessaire par la réception du signal est représentée par  $d_{SM}^i$ . Considérons deux esclaves d'indices :  $i, i + 1$ . Ici, nous faisons des hypothèses, nous prenons en considération les retards asymétriques entre le maître et l'esclave ( $d_{MS}^i \neq d_{SM}^i$ ). De plus, les délais aller-retour ( $d_{MM}^i$ ) sont différents pour différents esclaves ( $d_{MM}^i \neq d_{MM}^{i+1}$ ). On considère  $T_i$  le retard aller/retour entre le maître et l'esclave avant la phase d'ajustement et  $d_0^i$  l'erreur due à l'asymétrie du canal ( $i$ ).

On considère  $T'_i$ , le retard entre le maître et l'esclave après la phase d'ajustement,  $d_0^i$  l'offset du retard pour l'antenne ( $i$ ). La différence entre chaque paire d'antennes (esclaves)  $T'_i$  et  $T'_{i+1}$  est donnée par l'équation (5.2).

$$\Delta T'_{i,i+1} = 0 \Rightarrow 2d_{MS}^1 + d_0^1 = 2d_{MS}^2 + d_0^2 \quad (5.2)$$

Pour réaliser la synchronisation globale du temps entre les deux horloges, il est nécessaire de forcer à zéro l'écart différentiel (5.2) entre chaque paire d'esclaves, la condition de la synchronisation est la suivante :  $\Delta T'_{i,i+1} = 0$ . On injecte deux retards inconnus  $d_x^1$  et  $d_x^2$  entre le maître et l'esclave (1,2) pour compenser les offsets. L'équation (5.3) résume le concept de synchronisation d'une paire de deux antennes ( $i=1,2$ )

$$d_{MS}^1 + \frac{1}{2}[d_0^1 + dx_1] = d_{MS}^2 + \frac{1}{2}[d_0^2 + dx_2] \Rightarrow dx_i = \Delta T_{i,i+1} + dx_{i+1} \quad (5.3)$$

A partir de l'équation (5.3), on peut noter le vecteur d'ajustement des offsets pour N antennes et illustrer la formule (5.4) [156].

$$\begin{bmatrix} dx_1 \\ dx_2 \\ \dots \\ dx_{N-1} \end{bmatrix} = \begin{bmatrix} \Delta T_{1,2} \\ \Delta T_{2,3} \\ \dots \\ \Delta T_{N-1,N} \end{bmatrix} + \begin{bmatrix} dx_2 \\ dx_3 \\ \dots \\ dx_N \end{bmatrix} \quad (5.4)$$

La phase d'ajustement des offsets en pratique consiste à injecter un délai propre à chaque antenne dans le chemin d'horloge, obtenu par la résolution du vecteur d'offsets (5.4). Ce retard est égal à  $\frac{dx_i}{2}$ . Selon l'hypothèse (Retards symétriques), toutes les horloges arrivent simultanément à chaque antenne, dans le cas idéal avec une précision qui dépend du système de mesure et de la synchronisation ( $d_{xi} + d_{MM}^i = d_{x(i+1)} + d_{MM}^{i+1}, 1 \leq i \leq N$ )(Figure : 5.5). On distingue quatre configurations d'horloges A-1, A-2, B-1 et B-2 :

- A-1 : Chronogrammes des horloges  $Clk_i$  ( $t = 0$ ) avant l'ajustement au niveau du maître. Les horloges sont générées à partir d'une horloge maître  $Clk_M$  stable et précise. Dans le cas idéal, le skew des horloges  $Clk_i$  est nul. En pratique, dans le cas d'utilisation d'un diviseur de précision, le skew est de l'ordre de quelques picosecondes. Dans la figure, on ignore l'effet du skew, par conséquent les horloges sont parfaitement synchronisées au niveau du maître
- A-2 : Chronogrammes des horloges  $Clk_i$  avant l'ajustement au niveau des esclaves. Chaque horloge est arrivée après un retard  $d_{MS}^i$  différent. La différence des retards induit une désynchronisation des horloges aux pieds des antennes (au niveau des esclaves).

- B-1 : Chronogrammes des horloges après l’ajustement au niveau du maître. L’ajustement consiste en l’utilisation d’un circuit de retard programmable DL (Delay Chip) ou DTC (Digital-to-Time Converter) (Figure 5.5) à haute résolution dans le chemin de chaque horloge d’une valeur égale à  $\frac{dx_i}{2}$ . En effet, les horloges sont asynchrones au niveau du maître grâce à l’intégration de l’écart d’ajustement dans le chemin aller (du maître à l’esclave)  $Clk_i(t = \frac{dx_i}{2})$  (voir la figure 5.5).
- B-2 : Chronogrammes des horloges après l’ajustement au niveau de l’esclave. On constate une parfaite synchronisation des horloges aux pieds des antennes. Les horloges  $Clk_i$  arrivent en même temps aux pieds des antennes après un délai égal à  $\frac{dx_i}{2} + d_{MS}^i = \frac{dx_i}{2} + \frac{d_{MM}^i}{2}$  identique pour tous les esclaves.

### 5.2.3 Ajustement des offsets asymétriques

Dans cette section, on présente l’algorithme itératif pour calculer le vecteur des offsets d’ajustement  $[d_{x1}, d_{x2}, ..d_{xN}]$  des horloges. Cet algorithme permet d’atteindre une synchronisation du temps global dans un réseau distribué. L’algorithme permet également de minimiser l’erreur due à l’asymétrie entre le chemin aller pour divers retards aller/retour  $d_{MM}^i$  qui est le cas en pratique (voir l’équation (5.5)). L’algorithme (voir la figure 5.3) dispose de plusieurs paramètres essentiels : On définit le paramètre *Min* comme étant la précision de la synchronisation à atteindre par l’algorithme et  $\mu$  le pas de calcul. Les valeurs des deux paramètres influencent directement la vitesse de convergence. "l" est le nombre d’itérations de l’algorithme. La période de synchronisation  $T_{sync}$  est le nombre d’itérations nécessaire pour atteindre la précision requise *Min*.  $d_{MS}^A$  et  $d_{MM}^A$ , définissent le délai du maître à l’esclave et le délai d’aller-retour après l’ajustement de l’offset.  $d_{MM}^s$  représente le vecteur des offsets triés par ordre croissant afin d’obtenir des délais différentiels positifs entre chaque couple d’antennes  $(i, i + 1)$  ( $\Delta T_{i,i+1} > 0$ ) (Équation (5.6)). L’objectif de cette opération est de trier et d’obtenir des écarts  $dx_i$  positifs à la fin du calcul. Dans la phase d’implémentation, on peut ajuster uniquement des délais positifs en utilisant des composants de retard programmable avec une résolution inférieure à 100 ps. La figure 5.3 décrit le pseudo-code de l’algorithme proposé de calcul, ajustement et minimisation de l’erreur asymétrique

$$\begin{cases} d_{MS}^i \neq d_{SM}^i \\ d_{MM}^i \neq d_{MM}^{i+1} \end{cases} \quad (5.5)$$

$$\Delta T_{i+1} = d_{MM}^s(i) - d_{MM}^s(i + 1) \quad (5.6)$$

Dans la phase d’ajustement des offsets, la dernière antenne est considérée comme étant la référence de calcul (Antenne ayant un délai maximal d’aller-retour). Par conséquent, l’écart d’ajustement pour l’antenne de référence est nul ( $d_x^N = 0$ ). On obtient les valeurs suivantes par la méthode de dichotomie suivant l’équation (5.7). L’ajustement de l’offset est assuré en intégrant les écarts  $d_x^i$  dans le chemin aller de chaque antenne (Équation (5.8)).

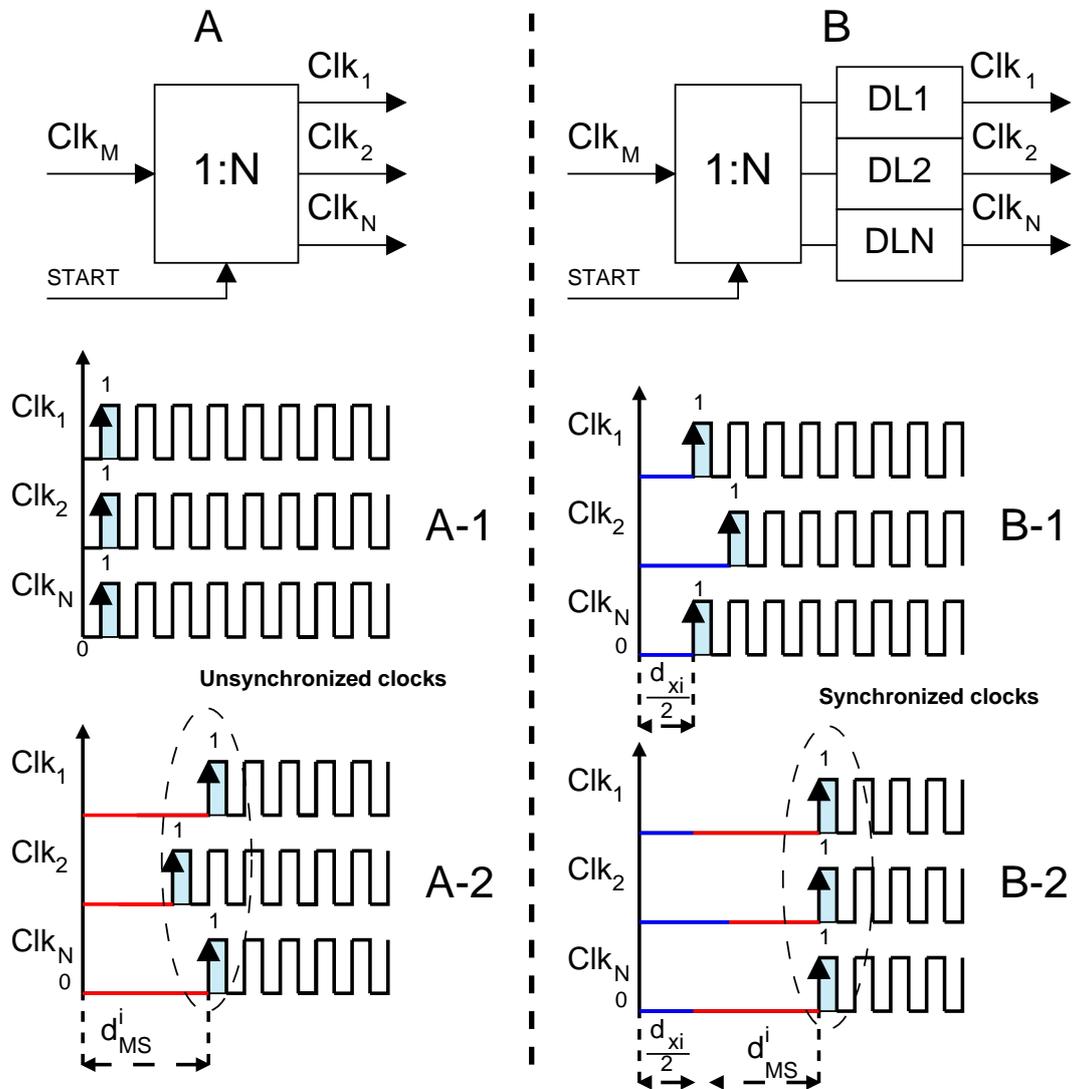


FIGURE 5.2 – Les chronogrammes des horloges  $Clk_i$  dérivées de l’horloge Maître  $Clk_M$  avant (A) et après (B) le processus d’ajustement. A-1, A-2 : Horloges au niveau du maître et des esclaves avant le processus d’ajustement. B-1, B-2 : horloges au niveau du Maître vu du côté des esclaves après l’ajustement en utilisant le circuit du retard programmable DL avec une valeur égale à  $\frac{d_{xi}^i}{2}$  pour chaque horloge.

```

variable  $\mu$ ,  $Min$ ,  $T_{sync}$ ,  $l$  ;
Get  $d_{MM}$  for each antennas;
Repeat
  /* Offset delays computation */
   $d_{MM}^s \leftarrow sort(d_{MM})$ ;
  for  $i=1$  to  $N-1$  do
     $\Delta T(i) \leftarrow d_{MM}^s(i+1) - d_{MM}^s(i)$ ;
  end for;
  /* Offsets ajustement */
   $d_x(N) \leftarrow 0$ ;
  for  $i=N-1$  to  $1$  do
     $d_x(i) \leftarrow \Delta T(i) + d_x(i+1)$ ;
  end for;
   $d_{MS}^A \leftarrow \frac{1}{2}[d_{MM}^s + d_x]$ ;
  /* Error estimation */
   $e \leftarrow d_{MS}^A - d_{MS}$ ;
   $\mu_e \leftarrow Mean(e)$ ;
  /* Delays updating */
  if not ( $|\mu_e| < Min$ )
     $d_{MM} \leftarrow d_{MM} - sign(\mu_e) * \mu * d_{MS}^A$ ;
     $T_{sync} \leftarrow T_{sync} + 1$ ;
     $l \leftarrow l + 1$ ;
  end if;
Until ( $|\mu_e| < Min$ )

```

FIGURE 5.3 – Pseudo-code de l’algorithme de synchronisation du temps d’un réseau d’antennes multi-reparties avec la prise en compte de l’asymétrie du retard aller-retour

$$d_x(i+1) = \Delta T(i+1) + d_x(i) \quad (5.7)$$

*Équation de calcul des écarts d’ajustement*

$$d_{MS}^A = \frac{1}{2}[d_{MM}^s + d_x] \quad (5.8)$$

$d_{MS}^A$  : retard aller après ajustement  
 $d_{MM}^s$  : Retard aller/retour trié

Pour chaque itération, on estime l’erreur moyenne globale pour toutes les antennes après l’ajustement des offsets ( $\mu_e$ ) (Équation (5.9)). Cette erreur est due aux retards asymétriques entre le maître et les antennes. Quand la précision requise n’est pas atteinte, l’algorithme met à jour l’erreur estimée et le retard  $d_{MM}$  selon l’équation (5.10). L’utilisation de la fonction *sign* dans l’équation (5.10) joue un rôle important sur le sens de convergence de l’erreur  $\mu_e$ . L’absence de cette dernière, peut induire une divergence de l’algorithme (lorsque  $sign(\mu_e) = -1$ ).

$$\mu_e = \frac{1}{N} \sum_{i=1}^N (d_{MS}^{A(i)} - d_{MS}^i) \quad (5.9)$$

*Estimation de l’erreur moyenne due à l’asymétrie du canal*

$$d_{MM}^{i+1} = d_{MM}^i - \text{sign}(\mu_e) \cdot \mu \cdot d_{MS}^A \quad (5.10)$$

*Mise à jour des retards,  $\mu$  est le pas de convergence*

## 5.2.4 Performances et limitations

Dans les sections précédentes, nous avons proposé un algorithme de synchronisation du temps itératif permettant le calcul et l'ajustement des offsets des horloges en prenant en compte l'asymétrie dans le canal. Maintenant, on va aborder la phase de simulation et l'analyse des performances et les limitations de l'algorithme appliqué à un réseau d'interféromètres radio. Les paramètres de simulation sont basés sur l'algorithme [157].

### 5.2.4.1 Paramètres de simulation

Pour simuler l'algorithme étudié précédemment, on va modéliser l'erreur induite par l'asymétrie et le retard entre le maître et l'esclave en fonction de la longueur  $L$  du câble. Dans la référence [157], la longueur du câble de la fibre optique aller est égale à la longueur du câble du retour avec deux longueurs d'onde différentes. On considère un réseau de  $N$  antennes multi-reparties suivant la topologie en étoile (Maître/Esclaves) illustrée dans la figure 5.4. On considère des liens optiques aller-retour ayant des longueurs d'ondes successivement  $1550 \text{ nm}$  et  $1390 \text{ nm}$ . On suppose que  $L_i$  est la longueur en mètre du lien de la fibre entre le maître et l'esclave ( $i$ ). L'erreur du retard asymétrique décroît lentement entre le maître et l'esclave en fonction de la longueur de fibre d'après les mesures illustrées dans tableau 2 de [157]. L'approximation linéaire du retard  $Average^i(ns)$  en fonction de la longueur du câble  $L$  entre le maître et l'esclave ( $i$ ) est donnée par l'équation (5.11). La valeur 16 ns est l'offset du retard pour  $L = 0$  m ( $L=16.05$  ns pour  $L= 30$  cm dans le tableau 2, ligne intitulée "Average" [157]).

$$Average^i(ns) \simeq -7.8 \cdot 10^{-5} \cdot L^i + 16 \quad (5.11)$$

$L^i(m)$  : la longueur du câble entre le maître et l'esclave ( $i$ )  
 $Average^i(ns)$  : le retard moyen entre le maître et l'esclave ( $i$ ).

On considère  $e_{Ad}^i$  comme étant l'erreur moyenne du retard asymétrique entre le maître et l'esclave ( $i$ ) ( $e_{Ad}^i = d_{MS}^i - d_{SM}^i$ ) déduit de l'équation (5.11). Cette erreur ne dépend que de la longueur du câble entre le maître et l'esclave, voir l'équation (5.12).

$$e_{Ad}^i(ns) = -7.8 \cdot 10^{-5} \cdot L^i \quad (5.12)$$

Considérons une gigue aléatoire  $e_{jit}(ps)$  modélisée par une distribution normale ( $\mathcal{N}$ ) due à la répétabilité de la mesure du retard du maître à l'esclave avec une moyenne de  $50 \text{ ps}$  et un écart-type de  $30 \text{ ps}$  (Équation (5.13)) (tiré du tableau 2 de la référence [157]). Cette erreur est supposée être l'accumulation des giges des différents composants dans le chemin d'horloge.

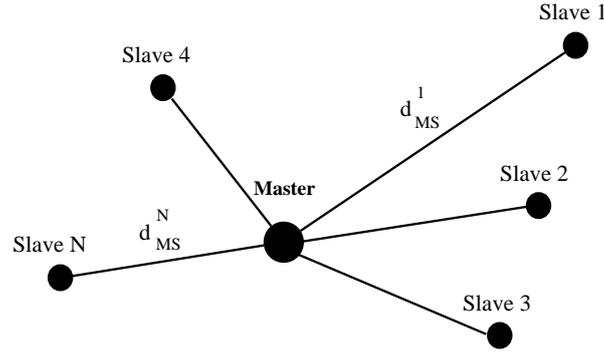


FIGURE 5.4 – Topologie en étoile du réseau d’antennes du Radiohéliographe de Nançay avec une distance différente entre le maître et chaque esclave ( $d_{MS}^i \neq d_{MS}^{i+1}$ ).

$$e_{jit} \sim \mathcal{N}(\mu_{jit} = 50ps, \sigma_{jit} = 10ps) \quad (5.13)$$

Pendant les simulations, on va prendre en considération uniquement les deux sources l’erreurs (erreurs asymétrique et l’erreur de l’imperfection de la mesure). D’après le tableau 1 dans la référence [157], le retard entre le maître et l’esclave suit une loi linéaire :

$$d_{MS}^i(ns) \simeq 9.8 \cdot L^i + 470 \quad (5.14)$$

#### 5.2.4.2 Résultats

Afin de simuler les performances globales de l’algorithme de synchronisation illustrées dans la figure 5.3 et ses limites dans un réseau d’antennes radio distribuées, nous considérons la topologie en étoile indiquée à la figure 5.4. Les retards et les erreurs sont générés selon les équations (5.12), (5.13) et (5.14). Nous fixons le paramètre de l’algorithme  $Min$  à  $10^{-12}$ . Le nombre d’antennes  $N$  varie de 10 à  $10^4$ . Le paramètre  $\mu$  prend successivement les valeurs  $10^{-6}$ ,  $10^{-7}$  ou  $10^{-8}$  pour différentes simulations. La figure 5.5 illustre les résultats de simulation.

La figure 5.5 illustre la période de synchronisation ( $T_{Sync}$ ) en fonction du nombre d’antennes ( $N$ ) pour différentes valeurs de  $\mu$ . La période de synchronisation définit le nombre d’itérations nécessaire pour que la valeur absolue de  $\mu_e$  devienne inférieure à  $Min$ . Autrement dit, l’algorithme atteint la précision désirée ( $\mu_e \leq Min$ ). On constate une décroissance rapide de  $T_{Sync}$  lorsque  $N$  augmente pour  $N \leq 1000$ . En revanche, elle reste quasi-constante pour des valeurs de  $N$  supérieures à 2000 pour chaque valeur de  $\mu$  ( $T_{Sync}(\mu = 10^{-7}, N = 10) = 2397, T_{Sync}(\mu = 10^{-7}, N = 10^4) = 191$ ). Il est clair que la période de synchronisation augmente significativement quand  $\mu$  diminue ( $T_{Sync}(\mu = 10^{-7}, N = 10) = 2397, T_{Sync}(\mu = 10^{-6}, N = 10) = 209$ ).

On fixe la valeur de  $N$  à 100,  $\mu$  à  $10^{-5}$  et  $Min$  à 0 pour créer une boucle infinie (voir la figure 5.3). La figure 5.6 illustre la précision globale de l’algorithme  $\mu_e(ps)$  après ajustement entre le maître et l’esclave en fonction du nombre d’itérations ( $l$ ) (Figure 5.3). On constate une décroissance linéaire de l’erreur en fonction de ( $l$ ) dans la phase transitoire : La phase de compensation avant l’obtention de la précision désirée ( $Min$ ). Le nombre d’itérations entre le début et la fin du calcul définit la période de synchronisation  $T_{Sync}$ . Par la suite, l’erreur oscille

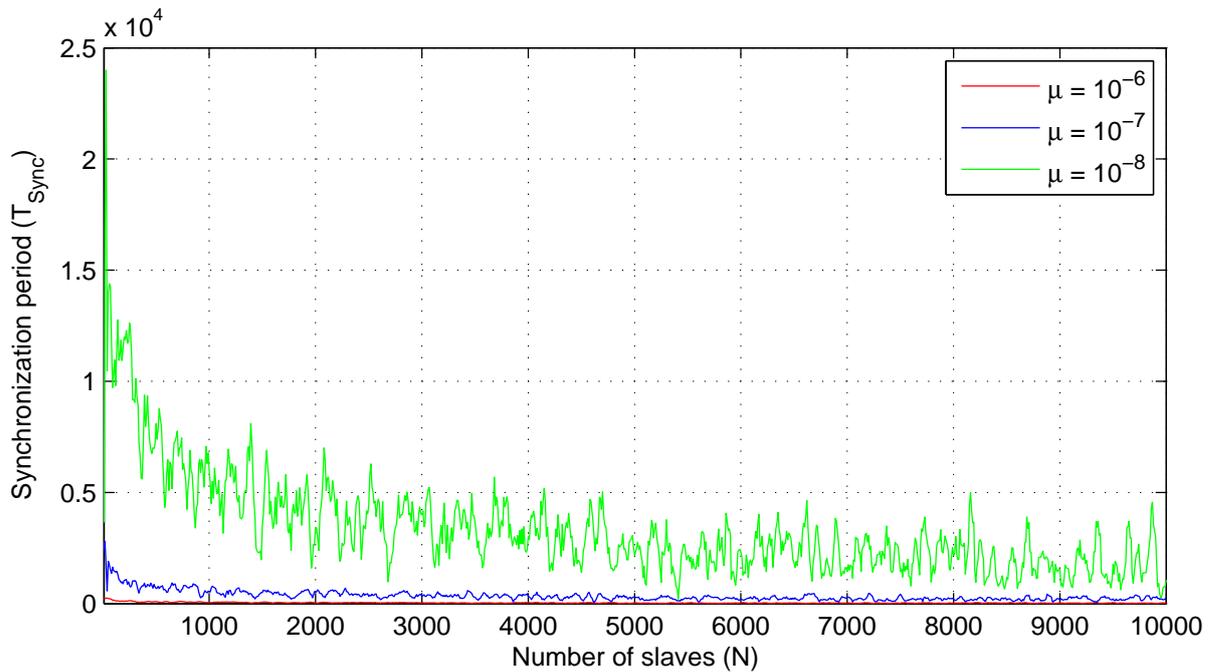


FIGURE 5.5 – Période de synchronisation ( $T_{Sync}$ ) en nombre d'itérations en fonction du nombre d'antennes ( $N$ ) pour différentes valeurs de ( $\mu$ ).  $\mu = 10^{-6}$  (courbe rouge),  $\mu = 10^{-7}$  (courbe blue) et  $\mu = 10^{-8}$  (courbe verte).

autour de la précision globale de l'algorithme  $\mu_0$  avec une gigue crête-à-crête  $e_{pp}$  (Figure 5.6). Après l'analyse des simulations, l'erreur moyenne  $\mu_e$  peut être décrite dans l'équation (5.15) dans la zone d'oscillation :

$$\mu_e = \mu_0 \pm \frac{e_{pp}}{2}, e_{pp} \sim \mu_0 \quad (5.15)$$

L'erreur globale  $\mu_e$  de l'algorithme ne dépend que de la valeur crête dans le régime permanent décrit dans l'équation (5.16).

$$\mu_e \sim \mu_0 \pm \frac{\mu_0}{2} = e_{pp} \pm \frac{e_{pp}}{2} \quad (5.16)$$

La figure 5.7 montre les retards  $d_{MS}^i$ ,  $d_{SM}^i$  et  $d_{MM}^i$  en fonction de l'asymétrie du canal et leurs distributions dans le temps. Les retards sont synthétisés avec une distribution gaussienne autour d'une valeur moyenne de  $0,568 \mu s$  avec une gigue due au bruit de mesure. On constate également dans la figure que le retard aller est légèrement supérieur au retard retour ( $d_{MS}^i > d_{SM}^i$ ) à cause de l'erreur de l'asymétrie induite par la longueur du câble  $L$  (voir les équations (5.11) et (5.12)). La figure 5.8 montre les valeurs des retards (figure en haut) et l'erreur (figure en bas) après l'ajustement des offsets en utilisant l'algorithme ( $\frac{d_{MM}^i + d_{xi}}{2}$ ) et l'ajustement réel ( $d_{MS}^i + \frac{d_{xi}}{2}$ ) pour une liaison asymétrique ( $d_{MS}^i \sim \mathcal{N}(\mu = 568,02ns, \sigma =$

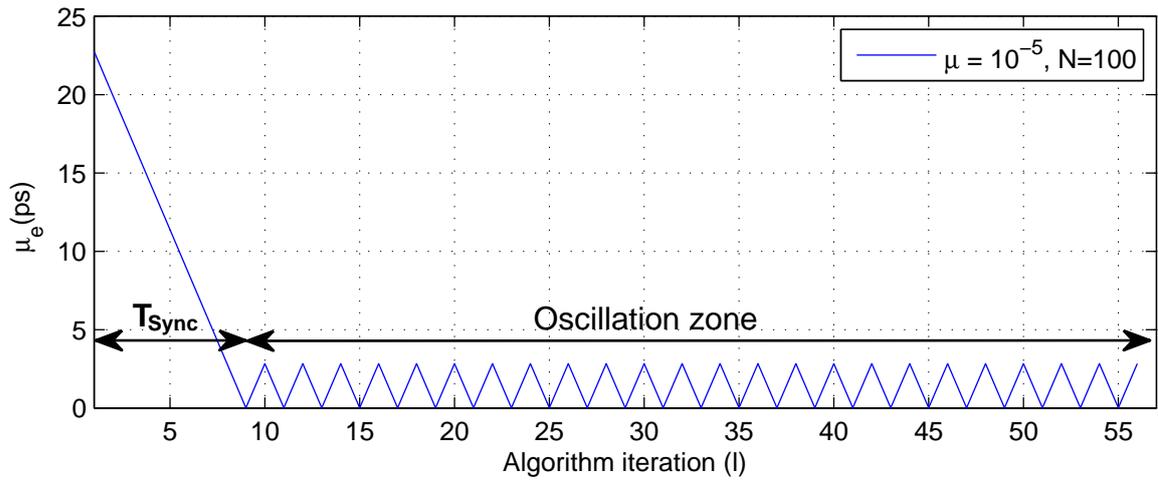


FIGURE 5.6 – Allure de l’erreur de l’algorithme  $\mu_e(ps)$  après le calcul et l’ajustement des offsets en fonction du nombre d’itérations ( $l$ ) pour  $N=100$  et  $\mu = 10^{-5}$ .

$18,65ps) < d_{SM}^i \sim \mathcal{N}(\mu = 568,04ns, \sigma = 15,56ps) \neq \frac{d_{MM}^i}{2}$ ) au début de la minimisation de l’erreur (première itération) et au  $10^4$  itérations pour  $N = 10^3$ ,  $\mu = 10^{-5}$ .

On considère  $I_{ter}$  le nombre d’itérations de l’algorithme. On constate clairement dans la figure 5.8 que les retards aux pieds des antennes  $\frac{d_{MM}^i + d_{xi}^i}{2}$  après ajustement maintiennent une valeur constante pour  $I_{ter} = 1$  (courbe rouge) et pour  $I_{ter} = 10^4$  (courbe verte). Durant l’itération  $10^4$ , on constate une translation des retards pour s’approcher de la valeur médiane du retard aller/retour  $d_{MM}^i$  (correction parfaite). En revanche, dans le cas d’un canal asymétrique on constate l’apparition d’une gigue au début et à la fin de la minimisation (régime permanent) de l’erreur. L’erreur  $e(i)$  entre la compensation par l’algorithme et la compensation réelle décroît en fonction du nombre d’itérations, elle stagne autour de zéro en régime final ( $I_{ter} = 10^4$ ) (voir la figure 5.8) avec une gigue due à l’asymétrie du canal. La figure 5.9 montre les résultats des mêmes retards pour une liaison symétrique ( $d_{MS}^i = d_{SM}^i = \frac{d_{MM}^i}{2} \sim \mathcal{N}(\mu = 568,04ns, \sigma = 14,64ps)$ ).

Les retards aux bouts des esclaves sont parfaitement identiques pour les deux ajustements (réel ou par algorithme) illustrés dans la figure 5.9. Les valeurs des retards sont égales à  $5.681 \cdot 10^{-7}s$  pour tous les  $i$ . En effet, l’erreur est nulle pour les deux ajustements indiquant une parfaite adéquation entre l’ajustement par algorithme ou l’ajustement réel dans le cas d’un lien symétrique. Dans le cas pratique le retard médian aller/retour  $\frac{d_{MM}^i}{2}$  est différent du retard aller  $d_{MS}^i$  ou du retour  $d_{SM}^i$  (Fig. 5.8) à cause du comportement aléatoire des retards aller et retour.

La figure 5.10 montre l’évolution de la valeur moyenne de l’erreur avant et après ajustements pour les diverses valeurs de  $\mu$  ( $10^{-6}$ ,  $10^{-7}$  ou  $10^{-8}$ ) en fonction du nombre d’itérations de l’algorithme. Le paramètre  $Min$  est mis à zéro (performances maximales) pour pouvoir observer le comportement de l’erreur dans le régime permanent. Le nombre d’esclaves  $N$  est fixé à 1000. La valeur moyenne de l’erreur  $\mu_e$  décroît linéairement avec une itération pour chaque valeur de  $\mu$

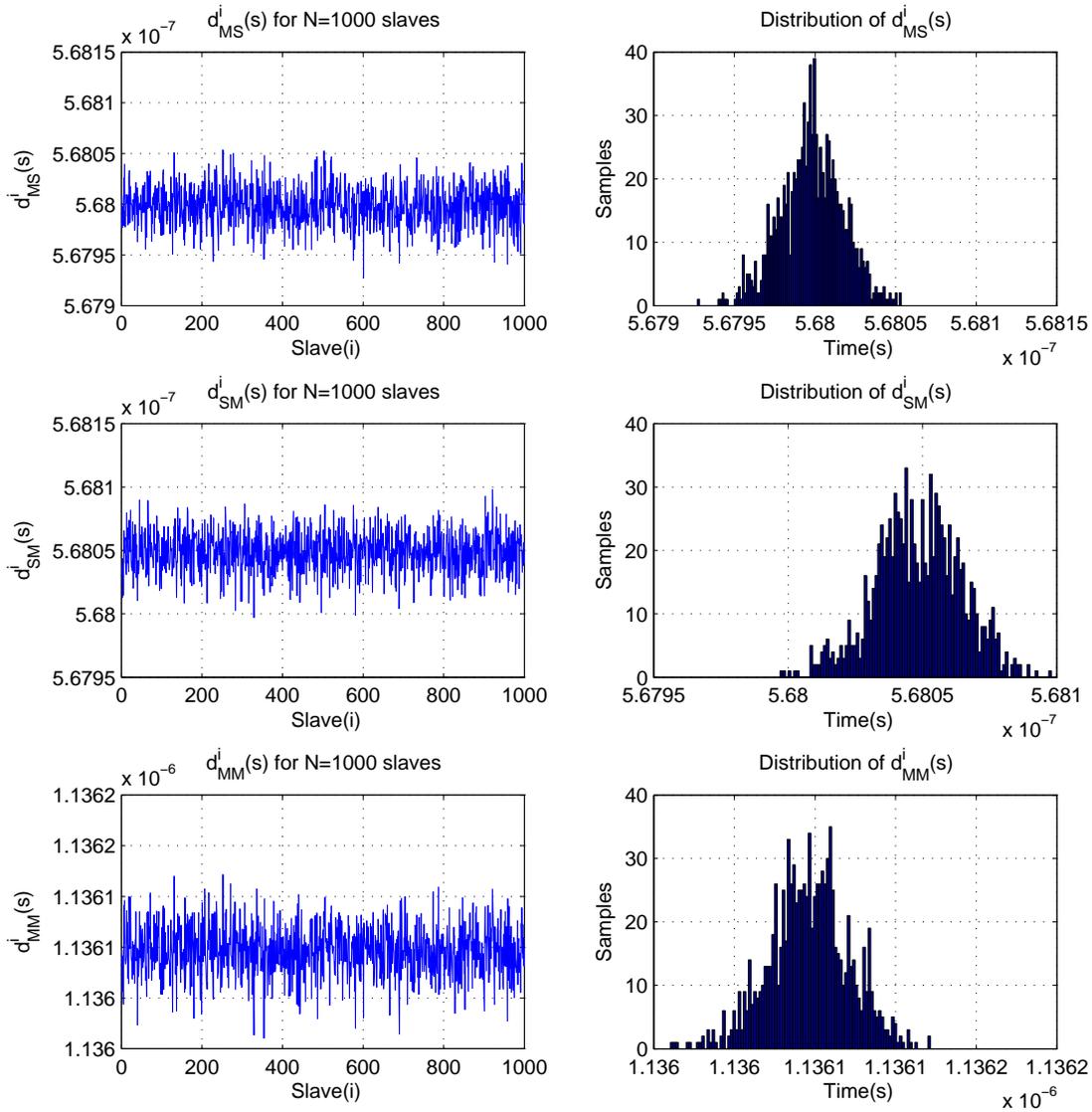


FIGURE 5.7 – Les retards  $d_{MS}^i$ ,  $d_{SM}^i$  et  $d_{MM}^i$  en fonction de  $(i)$  ( $i = 1, 2, \dots, 1000$ ) et leurs distributions dans le temps

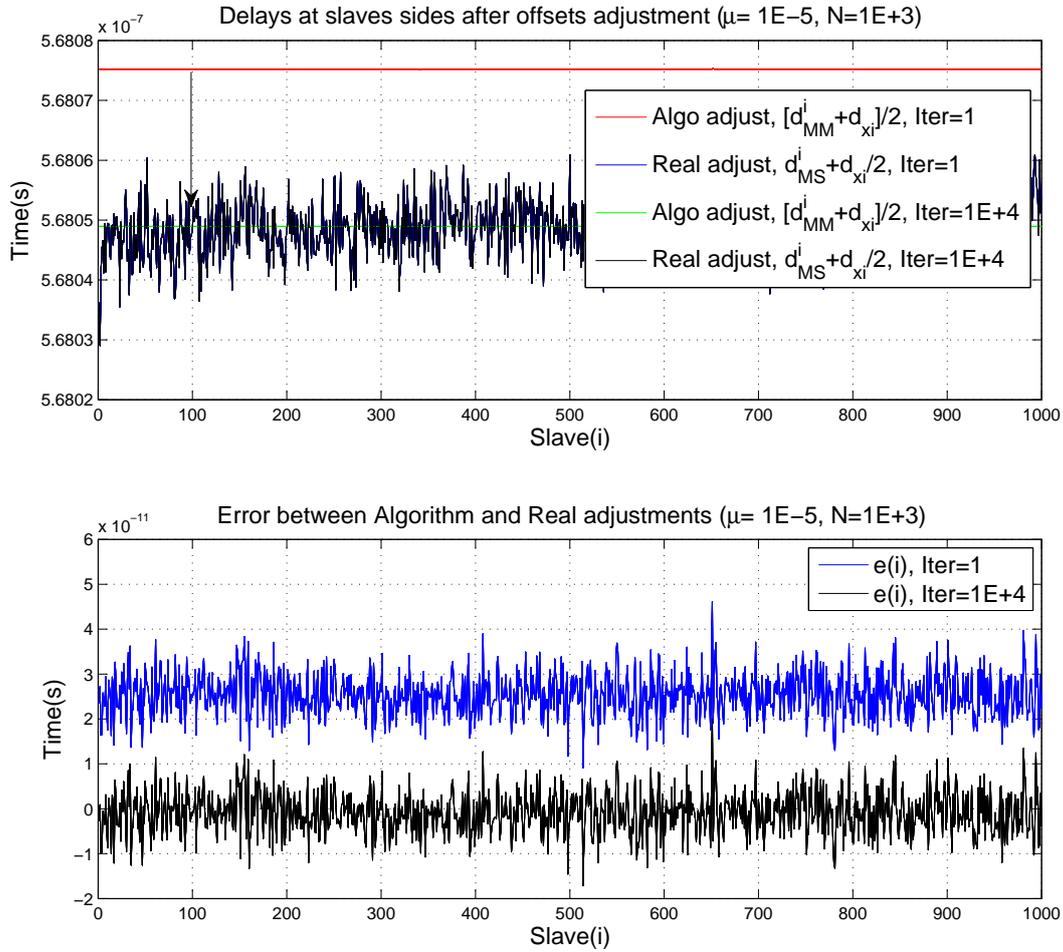


FIGURE 5.8 – Les retards au niveau des esclaves après l’ajustement des offsets en utilisant l’algorithme ( $\frac{d_{MM}^i + d_{xi}^i}{2}$ ) et l’ajustement réel ( $d_{MS}^i + \frac{d_{xi}^i}{2}$ ) pour une liaison asymétrique ( $d_{MS}^i \sim \mathcal{N}(\mu = 568,02ns, \sigma = 18,65ps) < d_{SM}^i \sim \mathcal{N}(\mu = 568,04ns, \sigma = 15,56ps) \neq \frac{d_{MM}^i}{2}$ ) au début de la minimisation de l’erreur (première itération) et avec  $10^4$  itérations pour  $N = 10^3, \mu = 10^{-5}$ .

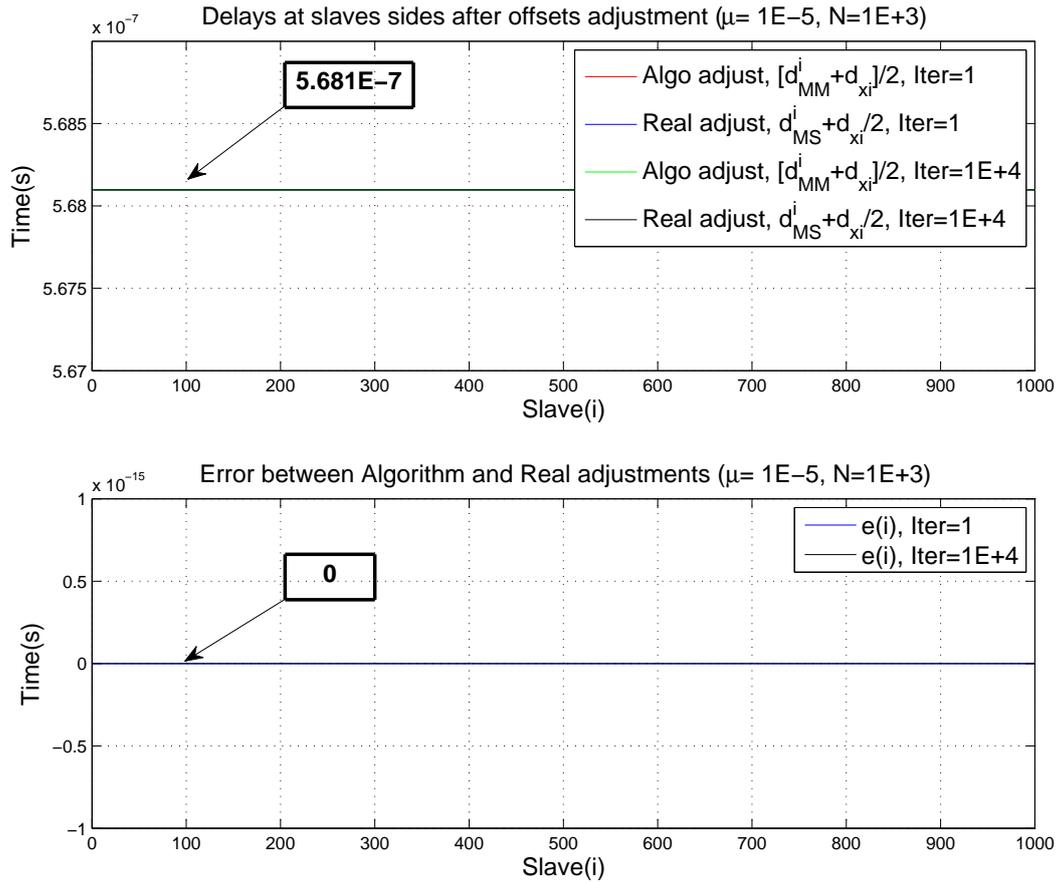


FIGURE 5.9 – Mêmes paramètres de simulation de la figure 5.8 pour une liaison parfaitement symétrique ( $d_{MS}^i \sim \mathcal{N}(\mu = 568.02ns, \sigma = 18.65ps) < d_{SM}^i \sim \mathcal{N}(\mu = 568.04ns, \sigma = 15.56ps) \neq \frac{d_{MM}^i}{2}$ ).

pour ensuite atteindre le régime permanent. Elle passe de 25 ps à une erreur inférieure à 200 fs pour  $\mu < 10^{-6}$ . La durée du régime transitoire de l'erreur est définie comme étant la période de synchronisation  $T_{sync}$  de l'algorithme (voir la figure 5.6), cette période indique également le temps de convergence de l'algorithme. Cette dernière augmente avec la diminution de  $\mu$ , en revanche l'amplitude des oscillations (précision de l'algorithme) en régime permanent décroît avec la diminution de  $\mu$ . La variance maintient une valeur quasi-constante (12,80 ps) quelle que soit la valeur de  $\mu$ .

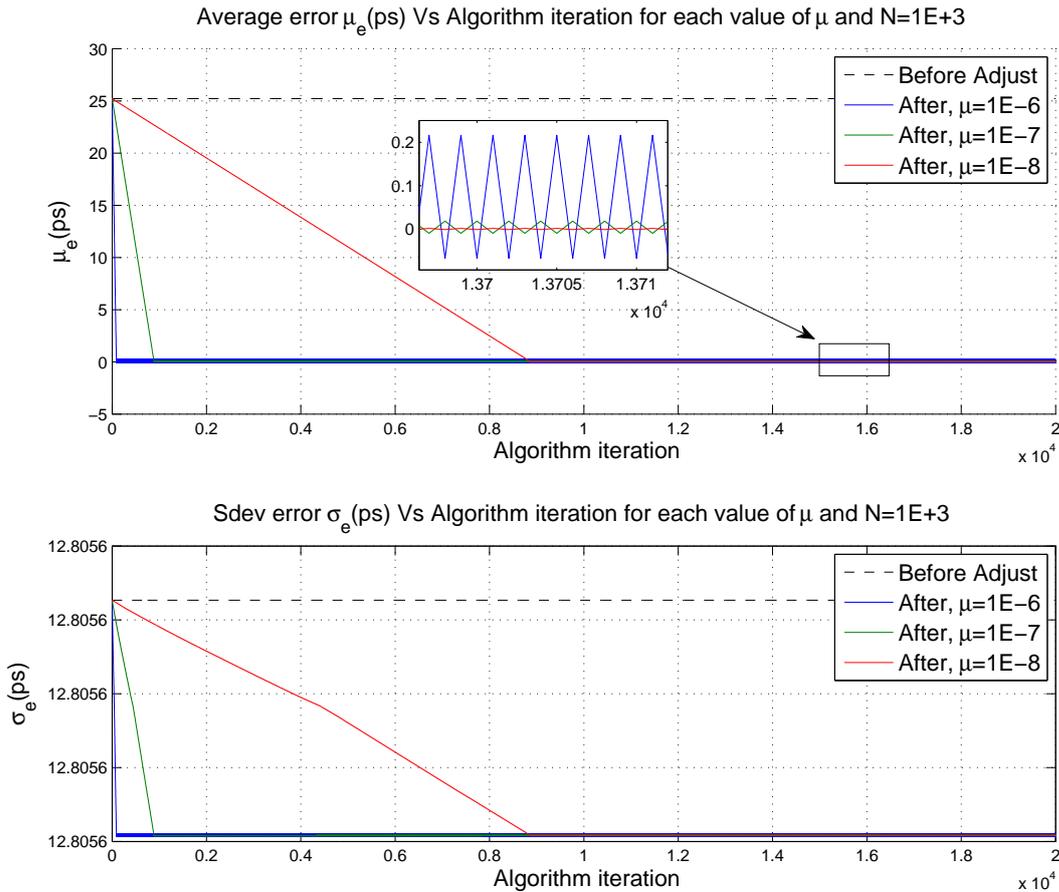


FIGURE 5.10 –  $\mu_e(ps)$  et l'écart-type  $\sigma_e(ps)$  avant et après ajustement en fonction du nombre d'itération (l) et du paramètre d'algorithme ( $\mu$ ) pour  $N=1000$ .

Pour évaluer la précision de l'algorithme et établir une comparaison avec la précision des algorithmes illustrés dans la littérature [157–159], on procède à l'analyse du comportement de l'erreur globale  $\mu_e$  en fonction de  $N$  (10, 100 ou 1000) et  $\mu$ . La précision désirée  $Min$  est fixée à 0. Le tableau 5.1 montre le paramètre  $\mu_0$  indiquant la précision globale du système de synchronisation en fonction de  $\mu$ .

On constate que l'algorithme a maintenu une précision inférieure à 100 ps pour les différentes valeurs de  $\mu$  et de  $N$  (Tab. 5.1). On remarque que  $\mu_0$  diminue rapidement lorsque  $\mu$  décroît.

TABLE 5.1 – La précision globale de l’algorithme  $\mu_0(ps)$  après l’ajustement des offsets en fonction de  $\mu$  et  $N$

$\mu$	$\mu_0(ps), N = 10$	$\mu_0(ps), N = 100$	$\mu_0(ps), N = 1000$
$10^{-3}$	14.11	79.74	98.64
$10^{-4}$	-7.42	-9.40	-10.23
$10^{-5}$	1.17	1.15	0.65
$10^{-6}$	0.10	$4.54 \cdot 10^{-2}$	$4.32 \cdot 10^{-2}$
$10^{-7}$	$1.112 \cdot 10^{-2}$	$1.19 \cdot 10^{-2}$	$1.01 \cdot 10^{-3}$
$10^{-8}$	$-7.73 \cdot 10^{-4}$	$1.13 \cdot 10^{-3}$	$6.97 \cdot 10^{-4}$

Il passe de 14,11  $ps$  pour ( $N = 10, \mu = 10^{-3}$ ) à  $7,73 \cdot 10^{-4}ps$  pour ( $N = 10, \mu = 10^{-8}$ ) en valeur absolue. D’autre part, on peut constater que la précision est meilleure lorsque  $N \geq 100$  et  $\mu \leq 10^{-5}$ . Finalement, la précision est supérieure lorsque  $N = 10$  et  $\mu \geq 10^{-4}$  par rapport aux autres valeurs de  $N(100, 1000)$  (Tab. 5.1).

### 5.2.5 Discussions et conclusion

Avec l’algorithme proposé, la période de synchronisation ( $T_{sync}$ ) diminue rapidement avec l’augmentation de la taille du réseau  $N$  (Figure 5.5). Cette diminution est induite par l’augmentation de la taille du vecteur d’estimation d’erreurs ( $N$ ) (figure 5.3). L’estimation de l’erreur moyenne devient efficace lorsque  $N$  augmente. Par conséquent, la période de synchronisation  $T_{sync}$  dans un réseau étendu diminue significativement ainsi que le temps de convergence de l’algorithme. L’algorithme proposé peut atteindre la précision désirée ( $Min$ ). Cela dépend du paramètre ( $\mu$ ) et du nombre d’esclaves ( $N$ ) (Tab. 5.1). Cependant, afin d’obtenir un algorithme convergent,  $Min$  doit être choisi supérieur à  $|\mu_e|$  (Équation (5.15)), car la précision dans le cas idéal de l’algorithme est limitée par  $\mu_e$ . Idéalement,  $Min$  doit être choisi très proche de  $|\mu_e|$ . En outre, à partir des résultats présentés dans la figure 5.5, on peut constater que la période de synchronisation augmente rapidement avec  $\mu$ . Selon ces résultats, nous concluons que nous pouvons contrôler la précision de l’algorithme en ajustant les paramètres  $Min$  et  $\mu$ . Nous avons illustré également que le paramètre  $\mu$  influence la vitesse de convergence de l’algorithme.

En résumé, l’algorithme permet de supprimer l’offset entre les canaux en réduisant la valeur moyenne de l’erreur en régime permanent après le calcul et l’ajustement. Par contre, la variance  $\sigma_e$  de l’erreur moyenne  $\mu_e$  est quasiment constante (voir la figure 5.10). L’évaluation ci-dessus montre que la synchronisation du temps global peut être réalisée de manière effective avec une précision en dessous de la nanoseconde dans les réseaux distribués. L’algorithme proposé peut compenser les retards asymétriques avec une précision réglable inférieure à  $10 ps$ . (Tab. 5.1). Les simulations théoriques de l’algorithme ont prouvé que la précision des résultats est meilleure que celle des algorithmes cités précédemment (PTP, PPS, WR). En pratique, la précision peut diminuer pendant l’implémentation de l’algorithme dans une plateforme embarquée à causes des imprécisions dues au circuit de mesure, au circuit d’ajustement et à la limitation en résolution binaire de l’estimateur. L’algorithme développé est compatible avec une architecture centralisée en topologie étoile (Maître/Esclave) qui est le cas du Radiohéliographe de Nançay.

## 5.3 Calibration des retards

### 5.3.1 Introduction

Dans cette section, on va introduire brièvement le principe de calibration des retards induits pour les diverses composantes de la chaîne de mesure. Le but étant de démontrer par l'analyse théorique et les simulations, la possibilité de corriger la chaîne en utilisant un estimateur de type Gaussien. Le processus de calibration sera décrit par la suite, ainsi que l'introduction à l'implémentation simplifiée de la technique sur FPGA. La phase de test sur les mesures réelles sera entamée dans la section tests et résultats.

### 5.3.2 Processus de calibration de la chaîne de mesure

L'objectif principal de la calibration est la détermination et la compensation des erreurs statiques (retards) et/ou dynamiques (gigue, évolution du comportement des composants avec la température, vieillissement, etc.) accumulées dans la chaîne de mesure. La phase de calibration consiste à injecter des intervalles de temps connus par l'utilisateur dans la chaîne et récupérer les mesures. En effet, la stratégie de mesure de cet écart pour chaque canal détermine la précision globale du système de synchronisation. L'approche la plus simple pour déterminer l'offset de la mesure est de soustraire à la valeur injectée la valeur moyenne des échantillons précédents. Dans ces conditions, la phase de calibration consiste à la soustraction de l'offset obtenu pour chaque mesure obtenue. Cette approche ne peut compenser que les erreurs statiques, elle est insensible aux erreurs dynamiques. Dans la suite, on va se focaliser sur une autre approche qui permet de minimiser l'effet des erreurs dynamiques dans la chaîne de mesure en se basant sur l'estimateur Gaussien.

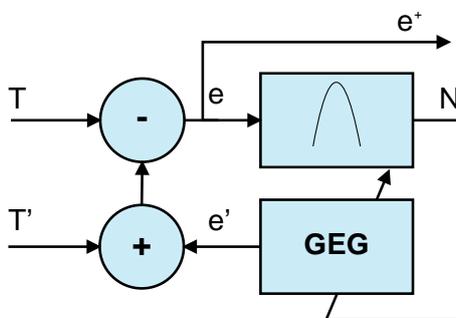


FIGURE 5.11 – Schéma bloc du principe de calibration

La figure 5.11 montre le schéma bloc du principe de calibration des mesures. Il est constitué essentiellement d'un estimateur Gaussien de l'erreur et d'un générateur d'échantillons aléatoires GEG (Gaussian Error Generator) ayant des paramètres issus de l'estimateur Gaussien. Le modèle est alimenté par les mesures des intervalles du temps avant et après la calibration ( $T$  et  $T'$ ) et génère pour chaque échantillon une valeur approchée de correction. En effet, le modèle permet de corriger en continu les échantillons de mesures en se basant sur l'estimation Gaussienne. Pendant la phase d'initialisation du processus, l'erreur  $e' = 0$ . On considère  $T$  la mesure issue du circuit TDC (voir l'implémentation) et  $T'$  l'intervalle de temps théorique à mesurer (Figure 5.11).

$$\epsilon = T - (T' + \epsilon') = T - T', \epsilon' = 0 \quad (5.17)$$

L'équation (5.17) révèle l'erreur du retard avant la calibration.

$$\epsilon^+ = T - (T' + \epsilon') \quad (5.18)$$

$T$  définit la valeur mesurée du retard aller-retour entre le maître et l'esclave, elle présente la valeur réelle de retard et la somme des erreurs des retards de propagation et les giges accumulées dans la chaîne de mesure (horloge, circuit de distribution de l'horloge, générateur des intervalles du temps, ...). Cette dernière est caractérisée par une distribution normale, elle est illustrée dans l'équation (5.19).

$$T = T' + \epsilon'' \quad (5.19)$$

On considère  $\epsilon'$  l'erreur estimée avec une distribution gaussienne ( $\mathcal{N}$ ) ayant une moyenne  $\mu'$  et une variance  $\sigma'$ .  $\epsilon''$  est aussi supposée être représentée par une loi normale avec une moyenne  $\mu''$  et un écart-type  $\sigma''$  (Équation (5.19))

$$\epsilon' \sim \mathcal{N}(\mu', \sigma'), \epsilon'' \sim \mathcal{N}(\mu'', \sigma'') \quad (5.20)$$

Alors, l'erreur  $\epsilon^+$  est définie par l'équation (5.20).

$$\epsilon^+ = \epsilon'' - \epsilon' \quad (5.21)$$

L'erreur  $\epsilon^+$  aboutit à une loi normale caractérisée par sa moyenne  $\mu^+$  (Équation (5.21)).

$$\mu^+ = \mu'' - \mu' \quad (5.22)$$

Et la variance  $\sigma^+$  (Equation (5.22))

$$\sigma^+ = (\sigma^2)'' + (\sigma^2)' \quad (5.23)$$

Les équations (5.22) et (5.23) montrent par définition que la différence statistique de deux distributions gaussiennes est gaussienne. La valeur moyenne de la distribution est la différence entre les deux valeurs moyennes, la variance est la somme des deux variances. On conclut qu'une simple soustraction peut réduire significativement l'offset de l'erreur induit par la chaîne (la valeur moyenne) ou l'erreur statique après calibration. Réciproquement l'erreur dynamique (la variance) augmente légèrement par rapport à l'erreur à l'entrée du processus (erreur avant la calibration).

$$\epsilon' \sim \mathcal{N}(\mu, \sigma/\alpha) \quad (5.24)$$

$\mu' = \mu$  : la valeur moyenne des mesures,  
 $\alpha$  : coefficient de correction,  
 $\sigma' = \alpha \cdot \sigma$  : la variance de la mesure.

La mise à jour des paramètres du modèle d'estimation est illustrée dans l'équation (5.25). L'équation montre la minimisation de l'erreur dans le temps,  $\alpha$  est le coefficient qui permet d'influencer la vitesse de convergence de l'estimateur.

$$\begin{cases} \mu'(i+1) = \mu'(i) - \frac{1}{N} \sum_{k=1}^N \mu'(i-k) \\ \sigma'(i+1) = \sigma'(i) \\ \alpha \geq 1 = cst \end{cases} \quad (5.25)$$

Dans la suite, on va aborder les simulations du modèle en comparant les résultats de l'estimateur Gaussien avec et sans minimisation de l'erreur. Le modèle de base de l'estimateur est décrit dans l'équation (5.26).

$$\begin{cases} \mu'(i+1) = \mu'(i) \\ \sigma'(i+1) = \sigma'(i) \\ \alpha \geq 1 = cst \end{cases} \quad (5.26)$$

### 5.3.3 Simulation

Le but ici est de simuler le comportement du modèle vis à vis d'une trame des mesures qui varient autour d'une valeur moyenne. On va varier le coefficient  $\alpha$  et observer l'allure de l'erreur avant et après la calibration. On considère un vecteur  $T$  des mesures aléatoires régies par la loi gaussienne de taille  $10^4$  :  $T \sim \mathcal{N}(\mu = 80ps, \sigma = 20ps)$ . Le délai  $T$  est constant ( $T' = cst = 80ps$ ),  $\alpha$  varie entre 1 et 100. La figure 5.12 montre les paramètres de l'erreur ( $\mu(ps)$ ,  $\sigma(ps)$ ) après calibration pour l'estimateur Gaussien avec et sans minimisation en fonction de  $\alpha$ .  $\mu$  et  $\sigma$  décroissent rapidement quand  $\alpha$  augmente dans le cas de la minimisation de l'erreur (courbes en rouge dans la figure 5.12). Toutefois, la variance de l'erreur stagne dans le cas de l'estimateur Gaussien basique (courbes en vert), elle reste constante autour de la variance de la mesure avant calibration (environ 20 ps). Ci-dessous les résultats de simulation pour  $\alpha = 100$  en comparant la calibration avec minimisation (équation (5.25)) et sans minimisation (équation (5.26)) :

- Gaussien avec minimisation :  $\mu(100) = -1,74 \cdot 10^{-5} ps$ ,  $\sigma(100) = 1,38 ps$
- Gaussien sans minimisation :  $\mu(100) = 5,98 ps$ ,  $\sigma(100) = 20,09 ps$

La minimisation de l'erreur avec l'approximation gaussienne permet de réduire considérablement les erreurs statique (offset) et dynamique (la gigue ou la variance). Dans notre cas, en utilisant l'estimateur gaussien on réduira considérablement le bruit de mesures dû aux diverses perturbations, en particulier la qualité de l'horloge utilisée par le circuit de mesure (TDC), précision des alimentations et autres perturbations externes.

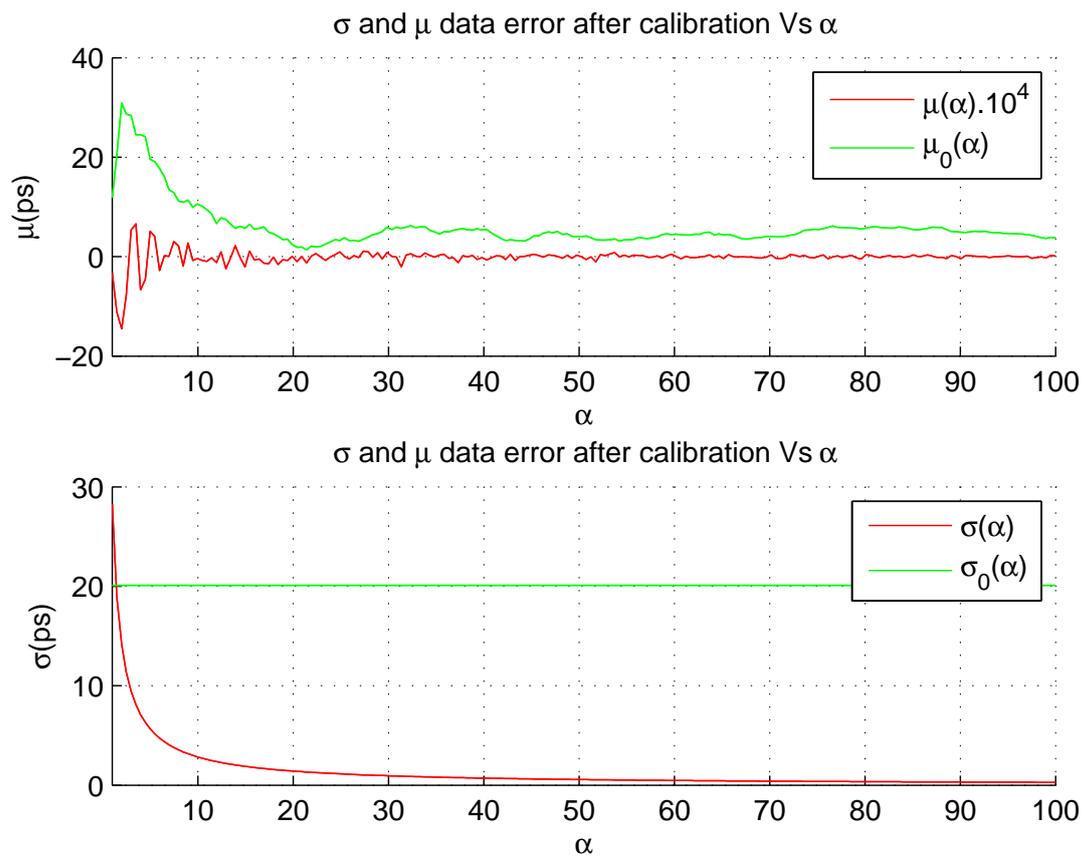


FIGURE 5.12 –  $\mu(ps)$  et  $\sigma(ps)$  de l'erreur de mesure après calibration en fonction de  $\alpha$ .

### 5.3.4 Conclusion

La section est dédiée à une approche simplifiée pour calibrer les erreurs statiques et minimiser l'effet de la gigue (fluctuation de la mesure) dans le cas d'une approximation Gaussienne des mesures. On se base sur les résultats de simulation, le rapport de correction (précision de la calibration) entre les deux méthodes de calibration (Gaussiennes avec et sans minimisation) égal à 14,53 pour l'estimation de la variance et  $3,43 \cdot 10^5$  pour l'estimation de la moyenne. Il s'avère que l'estimateur réduit considérablement l'offset de l'erreur après la phase de calibration. Réciproquement, l'implémentation en temps réel du processus peut être une contrainte durant la phase de mise en œuvre de l'algorithme dans une cible embarquée. Dans la phase de l'implémentation, on va se limiter à la première approche qui consiste au calcul et à la compensation de l'erreur statique avec l'approximation gaussienne de l'erreur. On verra dans le chapitre suivant sur l'implémentation une approche simplifiée pour l'implémentation matérielle du processus de calibration consacrée à la correction de l'erreur statique de la mesure.

# Implémentation du système de synchronisation

## 6.1 Introduction

Dans ce chapitre on va illustrer la conception et l'implémentation logicielle sur FPGA et décrire la partie matérielle du système de mesure de l'ajustement des offsets pour 4 canaux indépendants (4 antennes/esclaves) en temps réel. Le système complet de calcul et de l'ajustement est illustré dans la figure 6.1.

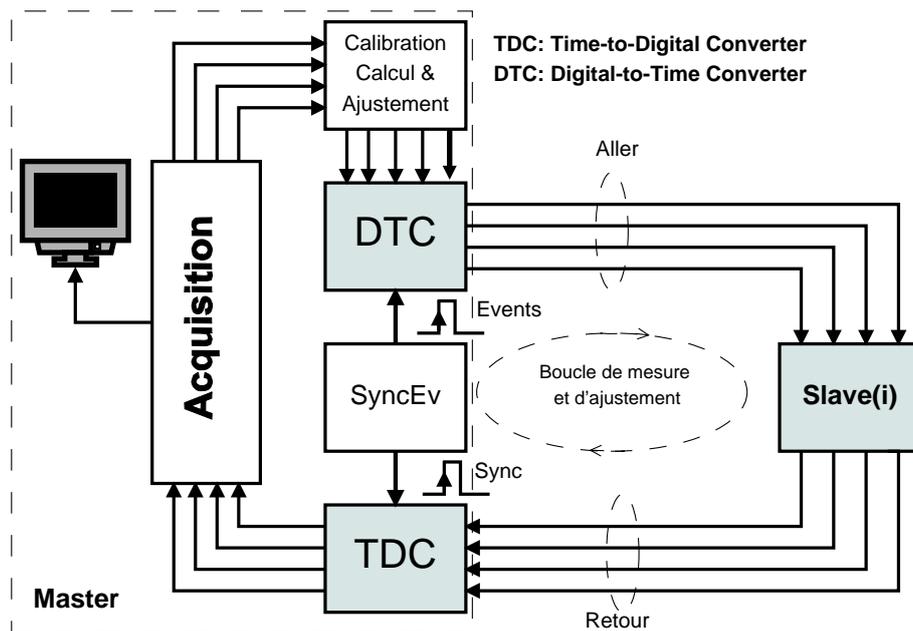


FIGURE 6.1 – Schéma bloc du système de mesure et d'ajustement des offsets en temps réel

L'objectif du système est la mesure et l'estimation en temps réel des écarts d'ajustement puis l'application de ces derniers pour chaque horloge en se basant sur la théorie du système décrite précédemment. La boucle de mesure et d'ajustement est constituée des éléments suivants :

- un TDC : Un circuit de mesure du retard aller/retour  $d_{MM}^i$  pour chaque canal.

- un DTC : Plusieurs circuits d'ajustement des offsets des horloges de haute précision (Digital-to-Time Converter).
- un bloc d'acquisition : Une Unité d'acquisition des retards mesurés  $d_{MM}^i$  pour chaque canal et le transfert des mesures à l'ordinateur pour l'analyse et le traitement différés (validation du fonctionnement du circuit TDC). Au même moment, l'interface d'acquisition est utilisée pour la mesure en temps réel des retards.
- un bloc SyncEv : Générateur des intervalles de temps programmable et du signal "Start" (Sync) durant la phase de mesure des  $d_{MM}^i$ .
- un bloc Calibration Calcul & Ajustement : Unité de calibration des retards mesurés, des calculs des écarts d'ajustement  $d_{xi}$  et du transfert des écarts au circuit à retard programmable (DTC)
- Slave : Nœud dans le réseau modélisant une antenne.

La première partie du chapitre traitera de la description de l'architecture Maître/Esclaves du système de synchronisation. Puis, on abordera la description détaillée des diverses unités constituant le système d'acquisition : le calcul, l'ajustement et la calibration des offsets des horloges, le contrôle des circuits TDC et DTC ainsi que leurs fonctionnements et implémentations en VHDL sur FPGA. La deuxième partie sera consacrée à la description de l'architecture matérielle utilisée. On terminera par les résultats de test et les mesures de la procédure de calibration. Les performances et limitations du système global seront définies.

## 6.2 Contrôle et acquisition des retards

Le circuit de mesure des retards aller/retour  $d_{MM}^i$  est essentiellement constitué du composant TDC et du FPGA (Fig. 6.2). Le premier permet de mesurer un intervalle de temps avec une résolution inférieure à 100 ps par canal. La manipulation de plusieurs canaux parallèles de données rapides ( $4 \times 300 Mbps$ ) avec des précisions inférieures à la nanoseconde impose l'utilisation de circuits avec des entrées et des sorties rapides.

Le circuit TDC utilisé [160] comprend une unité de mesure du temps avec quatre canaux (Events) et un canal de synchronisation (Sync). Il intègre une architecture de conversion Temps-Numérique ou TDC (Time-to-Digital Converter) pour des mesures rapides et précises. Le TMU (Time Measurement Unit) constituant le cœur du circuit peut atteindre une résolution temporelle de 13 ps (LSB) avec un écart-type de 8 ps (1 sigma). L'unité de mesure du TMU est synchronisée avec une horloge externe de 200 MHz. Le composant assure la résolution de mesure avec une horloge ayant une gigue maximale de 3 ps. Le composant intègre une interface série rapide (50 Mbps) pour la configuration des registres avec divers modes d'utilisation pour chaque canal (résolution binaire des résultats, fréquence interne des résultats, dynamique de mesure, etc.). La figure 6.3 illustre les interfaces de contrôle implémentées sur FPGA pour communiquer avec le circuit TDC.

L'objectif du processus est la synthèse des intervalles de temps fixes. Puis de mesurer ces derniers avec le circuit TDC. Les largeurs d'intervalles sont contrôlées et ajustées par le circuit FPGA. L'erreur globale du processus de mesure est l'écart statistique entre la valeur théorique de l'intervalle et la valeur mesurée, la valeur théorique des intervalles de temps  $T_i$  est imposée

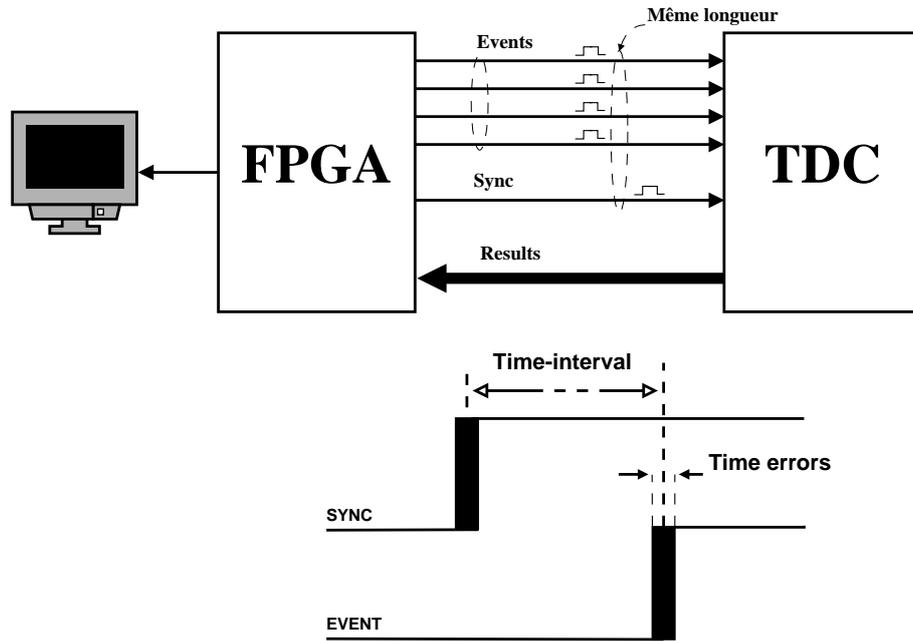


FIGURE 6.2 – Schéma de principe du système de mesure des intervalles de temps.

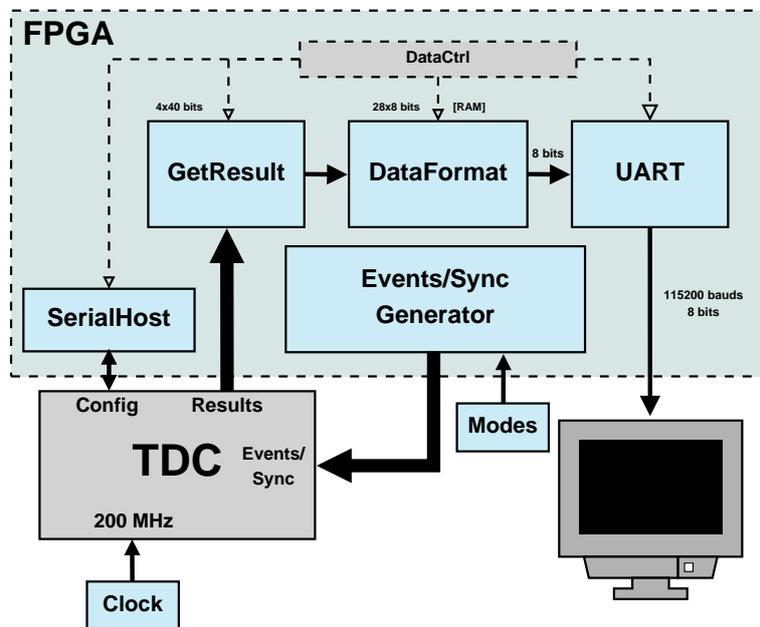


FIGURE 6.3 – Schéma bloc détaillé du système de mesure et d'acquisitions des données rapides issues du circuit TDC

par le générateur (voir la figure 6.1). On va utiliser Matlab pour comparer et caractériser les résultats de mesure pour chaque intervalle de données. Le circuit "Events/Sync Generator" permet de définir manuellement la durée d'intervalle à mesurer pour les 4 canaux. Le circuit "SerialHost" est dédié à la configuration des registres (états et contrôle) du circuit TDC. Le circuit "GetResult" récupère les mesures du circuit TDC. Ensuite, les données sont formatées par "DataFormat" et transmises à l'ordinateur via la liaison série pour l'analyse et la comparaison. Dans la suite de ce chapitre, on va étudier l'implémentation en VHDL sur FPGA de chaque partie constituant le processus de mesure. Les interfaces seront abordées en détail.

## 6.2.1 Générateur des intervalles de temps $T_i$

### 6.2.1.1 Analyse de fonctionnement

Le générateur des impulsions permet de synthétiser des intervalles de temps précis. On peut estimer et calibrer les retards induits par chaque canal (aller/retour) en l'absence et en présence des câbles. En résumé, cette procédure de calibration réduit l'effet des temps de propagation dans le chemin d'horloge et les "skew" entre les composants. Le générateur d'impulsions est un composant permettant de synthétiser des intervalles de temps ajustables avec une grande précision. Il est capable de générer 4 signaux d'événements (Events) et un signal de synchronisation (Sync) au format différentiel. Ces derniers sont directement liés avec le composant TDC (voir figure 6.4). Tous les signaux sont synchronisés avec une seule horloge (CLK) externe de 200 MHz. Le générateur est constitué des trois sous-blocs ci-dessous :

- *Count* comprend un compteur binaire synchrone ayant une horloge à une fréquence de 200 MHz (période de 5 ns). Le compteur a une résolution binaire de 16 bits, chaque sortie du compteur est une division par 2 de la fréquence de référence ( $2^1$  à  $2^N$ ). D'autre part la fréquence de sortie peut être ajustée entre 100 MHz (période de 10 ns) et 3,05 kHz (période de 327,68  $\mu s$ ). L'équation (6.1) montre la relation entre la fréquence du générateur  $F_{Clk_M}$  et la fréquence de référence  $F_{Clk}$  et le mode utilisé.

$$F_{Clk_M} = F_{Clk} \times 2^{-N(Modes)} \quad (6.1)$$

- *ModeSelect* permet la sélection d'une sortie parmi 16 du compteur binaire. La sortie sélectionnée sera liée avec l'horloge  $Clk_M$  du circuit de synthèse des signaux "Sync" et "Events". Pour des raisons de synchronisation, une sortie du compteur parmi les 8 sera utilisée par l'intermédiaire de trois entrées externes commandées avec des boutons poussoirs grâce à un kit de développement (Modes). Le premier mode est 0, le dernier est 7. Le passage d'un mode à un autre est une multiplication par deux de la durée de l'intervalle de temps à mesurer (division par deux de la fréquence  $Clk_M$ ). Les modes vont nous permettre de générer diverses valeurs d'intervalles de temps  $T_i$  (temps entre le front montant du signal Sync et le front montant du signal Events) qui seront utilisées durant les phases de mesure, calibration et ajustement des offsets des retards.
- *Sync+Events* : La figure 6.5 illustre l'ordonnancement des différents signaux de calibration et leurs durées en nombre de coups d'horloge. Le signal de synchronisation (Sync) dure deux coups d'horloge, il est activé par le front descendant. Le mode d'activation par front (montant ou descendant) est ajusté dans le registre de contrôle du circuit TDC.

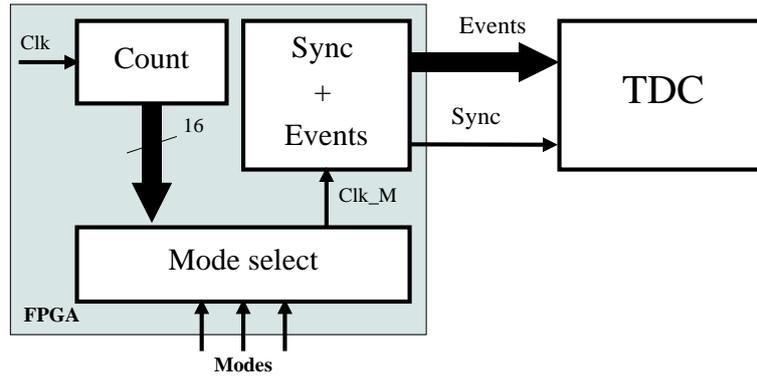


FIGURE 6.4 – Schéma bloc du générateur des intervalles de temps programmables

Pendant la phase de test, les signaux du générateur (Sync et Events) sont activés par le front montant. La durée  $T_i$  indique l'intervalle de temps entre le signal de synchronisation (Sync) et l'événement (Event) pour chaque canal. Il présente le temps entre les deux fronts montants des deux signaux. Le tableau 6.1 résume la durée en nombre de coups d'horloge pour chaque intervalle de temps pour les différents canaux.

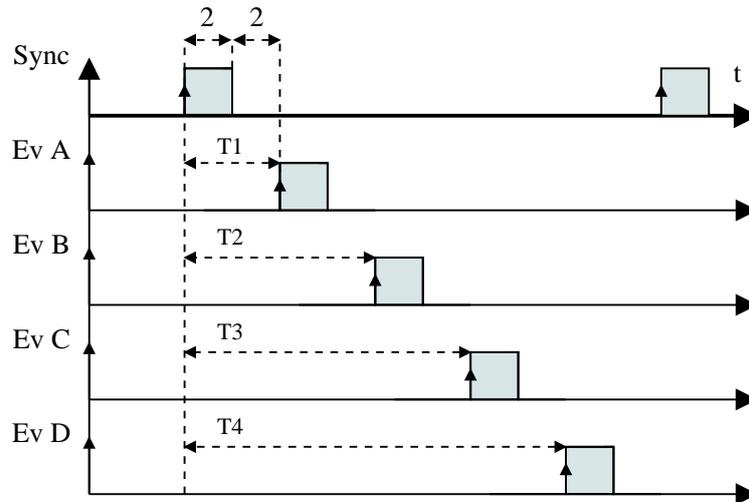


FIGURE 6.5 – Schéma bloc du générateur des intervalles de temps programmables

TABLE 6.1 – Le nombre de coups d'horloge durant l'intervalle de temps pour différents canaux

Événement (Canal)	Coups d'horloge
A	4
B	8
C	12
D	16

Le circuit TDC dispose de deux limites de mesure de temps : La durée d'un signal (Sync ou Events) doit être supérieure à 250 ps et la durée de l'intervalle de temps doit être supérieure à une période d'horloge maître du composant (200 MHz) qui correspond à une période  $T_0$  de 5 ns. La figure 6.5 illustre l'intervalle minimal de temps ( $T_1 = 4 \times T_0 = 20 \text{ ns} > 5 \text{ ns}$ ) et la durée pour toutes les impulsions ( $2 \times T_0 = 10 \text{ ns} > 250 \text{ ps}$ ). Les deux conditions sont largement satisfaites avec la méthode utilisée (voir la figure 6.5).

La figure 6.6 décrit la machine à état du circuit *Sync + Events* du générateur. Le tableau illustre la valeur des signaux "Events" et "Sync" pour chaque état. La description en VHDL du coeur du générateur peut être effectuée en utilisant la synthèse de la machine à état ou bien tout simplement via un compteur binaire qui scrute en continu la mémoire de taille égale au nombre d'états de la machine sur 5 bits (4 signaux d'événements + 1 signal de synchronisation). Cette mémoire est initialisée par les valeurs du tableau illustré dans la figure 6.6. Pendant la synthèse, les deux approches aboutissent au même résultat en terme de ressources matérielles, cependant la mise en œuvre du compteur semble plus simple (stockage en mémoire des états des signaux Sync et Events (5 bits) et lecture en boucle de la mémoire). Pendant la synthèse en VHDL, on va utiliser la machine à état qui est caractérisée par sa lisibilité des états et l'accès direct aux valeurs des sorties. Cette approche est plus efficace dans le cas de la réutilisation et de la mise à jour du code VHDL.

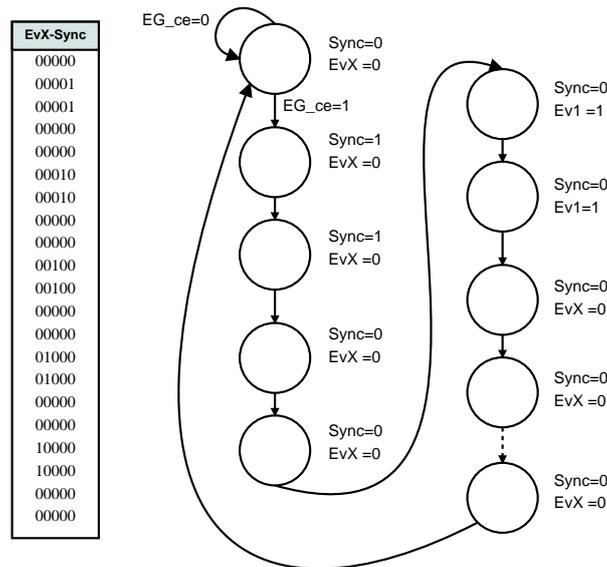


FIGURE 6.6 – Machine à état du générateur des signaux des événements et de synchronisation (Events + Sync) avec le tableau des états des sorties sous la forme  $Ev_3Ev_2Ev_1Ev_0Sync$

En résumé, une seule sortie est activée à la fois parmi les cinq pendant deux coups d'horloge dans l'ordre du chronogramme (Sync, EvA, EvB, EvC et EvD)(voir la figure 6.6). On constate la mise à zéro pendant deux coups d'horloge pour toutes les sorties. La machine à état dispose également d'un signal d'activation CE activé au niveau haut qui permet le déclenchement du générateur. Lorsque  $CE = 0$ , la machine reste en état d'attente.

6.2.1.2 Implémentation en VHDL

Entité



FIGURE 6.7 – Entité du générateur des intervalles de temps pour quatre canaux différentiels pour le circuit TDC

La figure 6.7 montre l'entité du générateur. Le circuit dispose de 4 entrées et de 2 sorties différentielles. Des "buffers Single-LVDS" seront utilisés pour le passage du mode "single" en mode différentiel. Les sorties sont directement raccordées avec les entrées différentielles du circuit TDC. Le tableau 6.2 montre la description des signaux de l'entité.

TABLE 6.2 – Description des différents signaux du générateur

Signal	Direction	Description
EG_modes	Entrée	Entrée sur 3 bits activée au niveau haut destinée à la sélection de la fréquence du générateur
EG_CE	Entrée	Entrée d'activation activée au niveau haut
EG_Clk	Entrée	Entrées d'horloge précise, externe de fréquence de 200 MHz
EG_Rst	Entrée	Entrées d'initialisation synchronisée activée au niveau haut
EG_Events_N	Sorties	Sorties négatives sur 4 bits des signaux des événements
EG_Events_P	Sorties	Sorties positives sur 4 bits des signaux des événements
EG_Sync_N	Sorties	Sortie négative du signal de synchronisation
EG_Sync_P	Sortie	Sortie positive du signal de synchronisation

Programme (voir Annexe I)

Ressources

Le tableau 6.3 montre les ressources en surface occupées par le générateur des intervalles de temps ainsi que la fréquence maximale supportée par l'architecture en fonction de la contrainte de vitesse de la puce visée. On constate que la fréquence est minimale pour la vitesse (-1) et maximale pour (-3). Les ressources en surface restent intactes pour les différentes vitesses. La fréquence 619,38 MHz (voir le tableau 6.3) est largement suffisante pour la synthèse du générateur. En effet, l'architecture doit supporter une fréquence maximale de 300 MHz pour pouvoir communiquer avec le circuit TDC. Le temps minimal requis à l'arrivée de l'horloge

(temps de setup) à l'entrée diminue avec l'augmentation de la vitesse, il passe de 1,44 ns (-1) à 1,06 ns pour la vitesse (-3).

TABLE 6.3 – La fréquence maximale d'horloge supportée par le générateur et les ressources en surface en fonction de la technologie de la puce (-1,-2 ou -3)

Vitesse	$F_{max}(MHz)$	LUTs	Registres FF	IOB	Temps horloge (ns)
-1	619.38	26	19	16	1.44
-2	727.72	26	19	16	1.23
-3	833.01	26	19	16	1.06

### 6.2.1.3 Simulations

Dans cette section on va montrer la validation du fonctionnement du générateur par simulation. Les figures 6.8 et 6.9 montrent les chronogrammes des signaux "Events" et "Sync" pour les modes 1 et 2 des sorties positives (Sortie "P"). L'allure des signaux respecte parfaitement celle définie précédemment dans la figure 6.5. La durée de chaque impulsion est le double de la période de l'horloge interne "Clk\_EvSync". En revanche, la période d'échelle pour le mode (2) vaut  $8,940\mu s - 8,930\mu s = 0,01\mu s$ . C'est le double du mode (1). On conclut que le passage du mode (1) au mode (2) a permis de doubler la durée des impulsions.

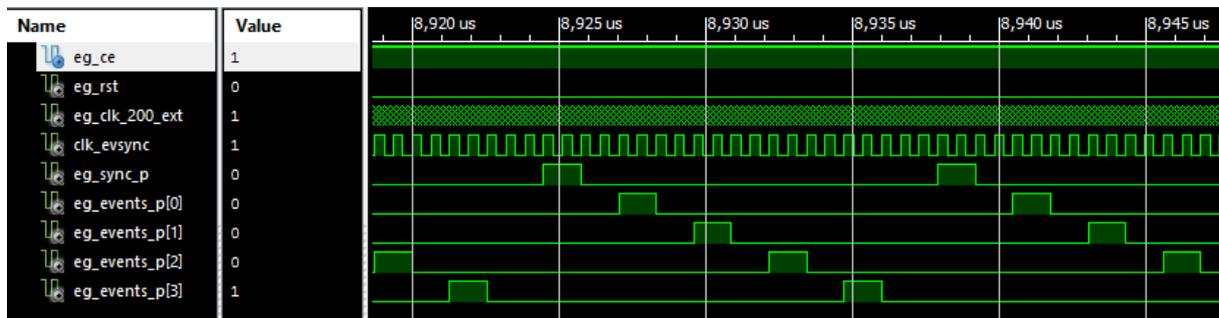


FIGURE 6.8 – Chronogramme des signaux Events et Sync pour le mode (1)

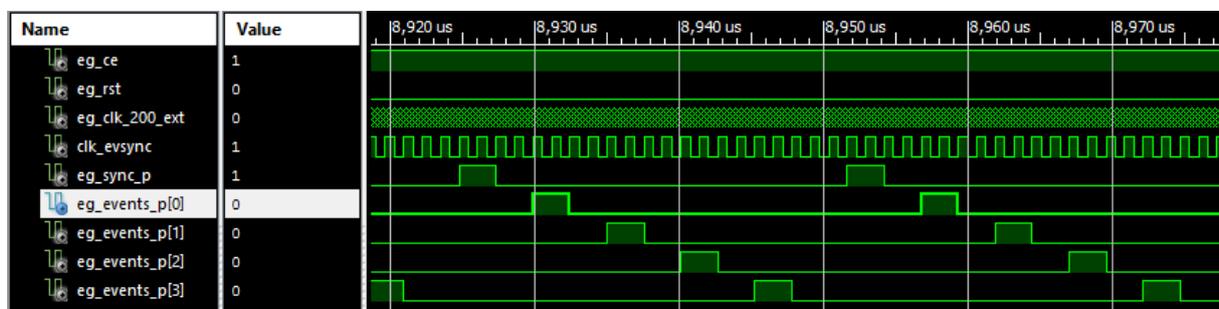


FIGURE 6.9 – Chronogramme des signaux Events et Sync pour le mode (2)

### 6.2.2 Interfaces d'acquisition et de contrôle

L'Annexe 0 contient les interfaces suivantes :

- Interface série TDC-FPGA
- Interface série TDC-PC
- Interface série FPGA-TDC

## 6.3 Calibration, calcul et ajustement des offsets

### 6.3.1 Introduction

Le noyau du système de synchronisation est basé sur le calcul des écarts de calibration à partir des mesures des retards aller-retour entre le maître et chaque esclave ( $d_{MM}^i$ ) (Figure. 7.4 - 6.1). Le but de la section est de décrire une implémentation possible en VHDL d'un circuit de calcul des écarts des ajustement. La figure 6.10 illustre le schéma bloc de base du circuit de calcul et d'ajustement des offsets. Il est constitué de cinq unités essentielles :

1. Calibration : Calibration des mesures des canaux
2. Tri des mesures : Tri des échantillons pour le circuit de calcul d'ajustement afin d'obtenir des écarts d'ajustement des offsets positifs
3. Calcul des offsets : Circuit permettant le calcul des offsets d'ajustement  $d_x^i$  pour chaque canal
4. Conversion des mesures du TDC au DTC (adaptation des résolutions binaires)
5. Ajustement : Transfert multiplexé des écarts  $\frac{d_x^i}{2}$  d'ajustement aux circuits DTC

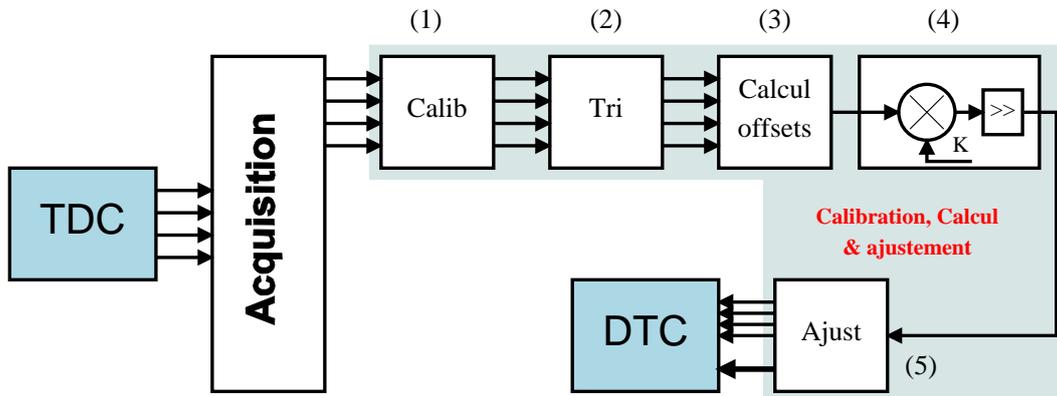


FIGURE 6.10 – Schéma bloc du circuit de calcul des écarts d'ajustement

La section suivante sera dédiée à une brève description de diverses parties constituant le circuit de calcul des écarts d'ajustement, puis on terminera avec l'approche pratique possible pour la calibration de la chaîne de mesure.

### 6.3.2 Calibration de la chaîne

#### 6.3.2.1 Introduction

L'objectif de la section consiste en la synthèse en VHDL du circuit de calibration de la mesure du retard aller-retour ( $d_{MM}$ ) (Eq. 6.2) issu du circuit TDC. Cette première phase permet de calibrer l'effet des retards accumulés dans le chemin aller-retour des horloges. L'intérêt de

la calibration est le réglage de zéro des canaux en temps réel. La figure 6.11 illustre les différents délais dans un trajet aller-retour entre le maître et l'esclave. On distingue quatre délais :

- $T_i$  : Intervalle du temps générée par FPGA. La durée de ce dernier est connue d'avance. Elle est générée par le générateur des événements,
- $d_0^i$  : La somme des retards instrumentales accumulés dans la chaîne, due généralement au temps de propagation des divers composants dans le chemin aller-retour (générateur des intervalles de temps, circuits de transmission et de réception),
- $d_1^i$  : Retard induit par la longueur du câble
- $d_{MM}^i$  : Mesure du retard global (du maître au maître) du circuit TDC et la somme des retards dans le chemin.

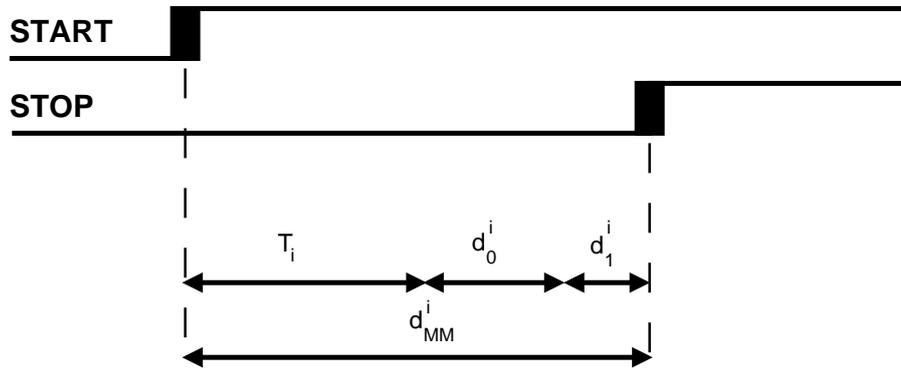


FIGURE 6.11 – Chronogrammes des retards dans un trajet aller-retour

$$d_{MM}^i = T_i + d_0^i + d_1^i \quad (6.2)$$

Durant la phase de calibration, l'effet du retard du câble  $d_1^i$  est supprimé en utilisant des câbles avec des longueurs connues et identiques conçus spécialement à cet effet dans le chemin aller-retour des signaux START et STOP. Par conséquent, la mesure du TDC  $d_{MM}^i$  (Eq. 6.3) est réduite à  $T_i + d_0^i$ , donc on peut facilement déduire la valeur  $d_0^i$  à partir de la mesure de  $d_{MM}^i$  et par la connaissance préalable du  $T_i$ .

$$d_{MM}^i \simeq T_i + d_0^i \quad (6.3)$$

### 6.3.2.2 Architecture du circuit de calibration

Le schéma bloc du circuit de calibration (1) illustré précédemment dans la figure 6.10 est décrit dans la figure 6.12. Il permet de calibrer la mesure  $d_{in}$  issue de l'interface d'acquisition. Il est constitué essentiellement de trois soustracteurs (C1, C2 et C4) et de deux filtres de lissage de type moyenneur (C3 et C5). La sortie  $d_{out}$  est filtrée et calibrée, elle sera ensuite utilisée par le bloc de calcul et d'ajustement de l'offset (voir la figure 6.10). L'entrée  $d_{in}^i$  est l'entrée de la mesure du retard aller-retour  $d_{MM}^i$  pour le canal ( $i$ ) ( $d_{in}^i = d_{MM}^i$ ).  $T_i$  est la valeur numérique de l'intervalle de temps sur 40 bits avec un LSB de 13,02 ps (les résultats du circuit TDC sont

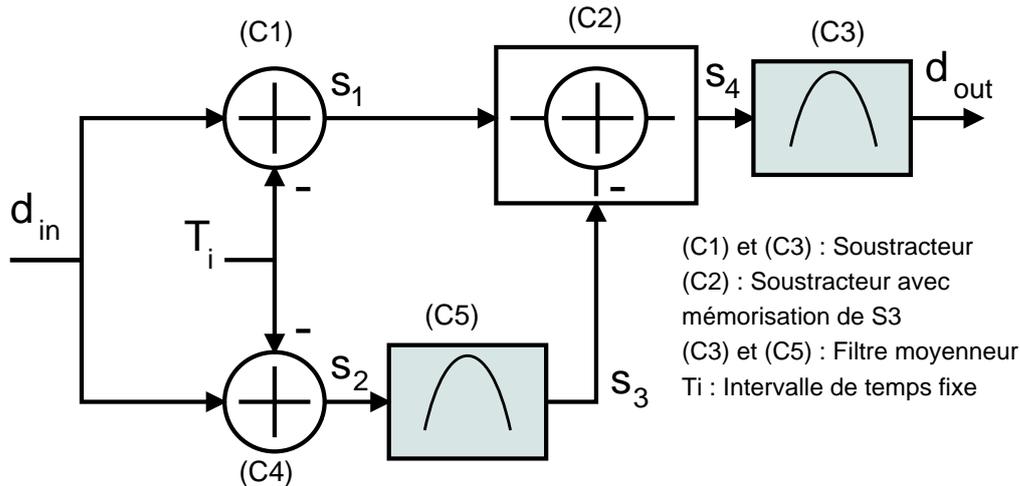


FIGURE 6.12 – Schéma bloc du circuit de calibration

sur 40 bits (10 octets) avec une résolution d'environ 13 ps). Exemple, la valeur numérique qui correspond à  $1 \mu s$  est égale à  $\frac{10^{-6}}{13 \times 10^{-12}} = 76923$  en hexadécimal sur 10 octets : `0x"0000012C7B"`. On distingue deux phases de fonctionnement du circuit :

**Phase 0 :** Cette phase permet de calculer le délai moyen dû aux différents composants dans le chemin aller-retour. Elle permet d'estimer la valeur moyenne du délai  $\hat{d}_0^i$ , le délai  $\hat{d}_1^i$  du aux câbles est nul (voir la figure 6.11). Le délai  $d_{in}^i$  est donné par l'équation (6.4)

$$d_{in}^i = d_{MM}^i = T_i + d_0^i \quad (6.4)$$

Les sorties  $S_2$  et  $S_3$  sont données par :

$$\begin{cases} S_2 = d_{MM}^i - T_i \simeq d_0^i \\ S_3 = \hat{d}_0^i \end{cases} \quad (6.5)$$

Durant la phase 0, le soustracteur (C2) est désactivée ( $S_4 = d_{out} = 0$ ). La longueur du filtre (C5) ainsi que la période de mesure de  $d_0^i$  sont configurables. A la fin de la période de "calibration", le soustracteur avec mémorisation (C2) maintient en mémoire la valeur moyenne  $S_3 = \hat{d}_0^i$  durant la phase 0. A la fin de la phase 0, la sortie  $S_3$  reste intacte quelque soit la valeur de l'entrée  $d_{in}$ .

**Phase 1 :** Phase normale de mesure des intervalles de temps avec la prise en compte de la calibration. Cette phase débute automatiquement à la fin de la période de calibration définie par un compteur (voir la suite de la section). Durant cette phase, on débranche les câbles (de référence) dédiés à la calibration, puis on branche les câbles réels (la procédure peut être automatisée en utilisant des multiplexeurs numériques). Par conséquent, le retard aller-retour à l'entrée  $d_{in}$  est donné par l'équation (6.2). Les sorties  $S_1$  et  $S_4$  sont données par l'équation (6.6).

$$\begin{cases} S_1 = d_{in}^i - T_i = d_{MM}^i - T_i = (T_i + d_0^i + d_1^i) - T_i \simeq d_0^i + d_1^i \\ S_4 = S_1 - \widehat{d}_0^i = (d_0^i - \widehat{d}_0^i) + d_1^i \simeq d_1^i \end{cases} \quad (6.6)$$

La sortie " $d_{out}$ " est la mesure après calibration, elle est donnée par l'équation (6.7). On conclut que le circuit de calibration permet de calibrer les délais dans le chemin aller-retour de l'horloge.

$$d_{out}^i \simeq \widehat{d}_1^i \quad (6.7)$$

On va entamer dans la section suivante l'implémentation en VHDL du filtre de lissage (C3 et C5) et du filtre du soustracteur (C2). On termine avec des simulations du circuit complet de calibration illustré dans la figure 6.12.

### 6.3.2.3 Filtrage des mesures

#### Introduction

Le filtre moyennneur est le plus utilisé dans le traitement numérique du signal, grâce à sa simplicité au niveau implémentation et à son efficacité dans le domaine temporel. En effet, il permet de réduire le bruit d'un signal. Cependant, il n'est pas adapté pour le filtrage fréquentiel d'un signal. L'équation (6.18) montre le modèle du filtre moyennneur. Le filtre est équivalent à un filtre à Réponse Impulsionnelle Finie (RIF) avec des coefficients identiques égaux à  $\frac{1}{M}$  avec  $M$  le nombre de coefficient. Le filtre moyennneur filtre les mesures durant la phase de calibration afin d'obtenir une meilleure précision temporelle du système de synchronisation.

$$y(i) = \frac{1}{M} \cdot \sum_{j=0}^{M-1} x(i-j) \quad (6.8)$$

*Avec  $M$  la taille du filtre,  $x$  les échantillons à l'entrée et  $y$  la sortie du filtre.*

La figure 6.13 montre la simulation du filtre appliquée aux ensembles des mesures d'un intervalle de temps de 1 ns avec un bruit du circuit de mesure (TDC) ayant une gigue de 120 ps pour différentes tailles de filtre (1, 8, 16 et 32). Lorsque la taille du filtre augmente, les amplitudes diminuent autour de la valeur désirée (mesures réelles de 1 ns). Le tableau 6.4 montre la gigue (pour 1 sigma) et la valeur moyenne du signal en fonction de  $M$ . Le bruit de la mesure diminue de 10 ps environ pour chaque multiplication par deux de la taille du filtre.

La réponse fréquentielle  $H(f)$  est décrite par l'équation (6.9). Il est clair que le filtre ne peut séparer deux bandes de fréquence. Il a une pente lente pour une fréquence de coupure donnée illustrée dans la figure 6.14. La pente diminue avec l'augmentation de la taille du filtre. L'amplitude des lobes secondaires croît avec  $M$ . En résumé, le filtre offre de bonnes performances dans le domaine temporel en dégradant celui du domaine fréquentiel. Le moyennneur d'un signal temporel peut être considéré comme un filtre passe bas de mauvaise qualité. Dans notre analyse, on s'intéresse aux mesures et à la précision de la synchronisation temporelle, d'où l'intérêt de l'utilisation du filtre moyennneur.

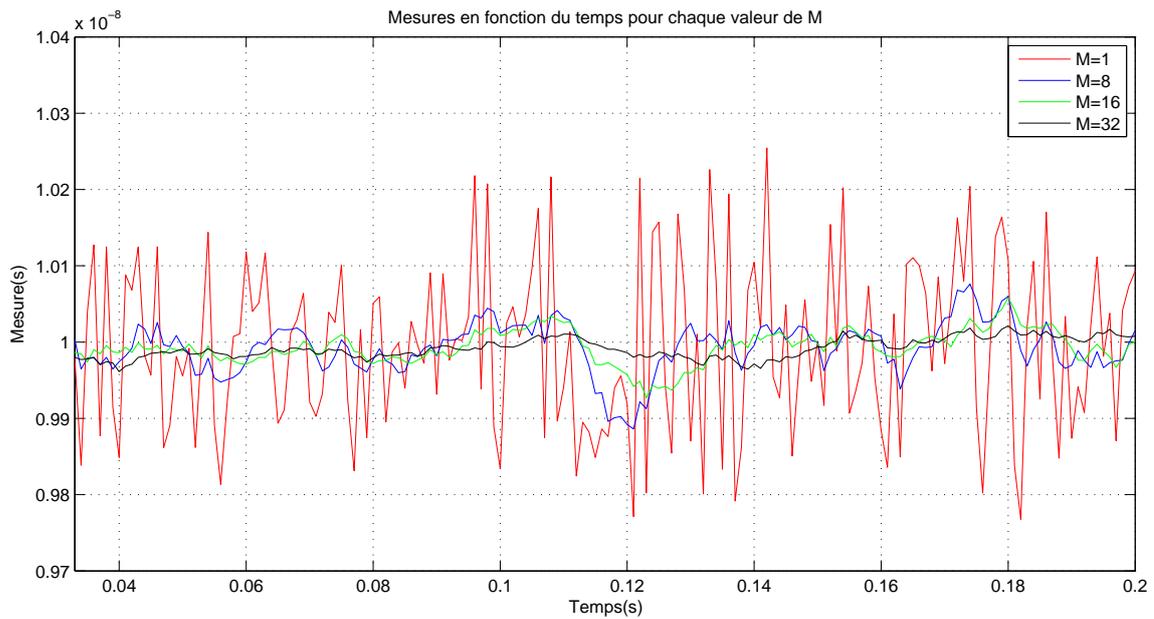


FIGURE 6.13 – Mesures d’intervalles de temps de 1 ns avec une gigue de 120 ps (1 sigma) en fonction de la taille du filtre  $M$ .

TABLE 6.4 – Effet de la taille du filtre  $M$  sur la restitution de la mesure réelle

$M$	$\mu(ns)$	$\sigma(ps)$
1 (Mesure brute)	1	120.00
8	1	38.91
16	1	28.21
32	1	19.86

$$H(f) = \frac{\sin(\pi \cdot f \cdot M)}{M \cdot \sin(\pi \cdot f)} \quad (6.9)$$

## Fonctionnement

Les données des mesures arrivent avec un grand débit du circuit TDC. Une implémentation matérielle est nécessaire pour des calculs à haut débit. Avant tout traitement de ses mesures, l’opération de filtrage est nécessaire pour réduire l’influence du bruit de mesure induit par le circuit de mesure. Il peut réduire les giges des signaux du générateur des événements et du signal de synchronisation. Dans cette section, on va expliquer le fonctionnement de deux techniques d’implémentation du filtre moyenneur. La première est par bloc, la deuxième utilise une mémoire FIFO permettant d’effectuer la moyenne pour chaque coup d’horloge. Les deux implémentations sont faisables et adaptées à notre processus de mesure et d’ajustement. Le système peut effectuer des calculs d’une façon différée (pas besoin de calcul en temps réel).

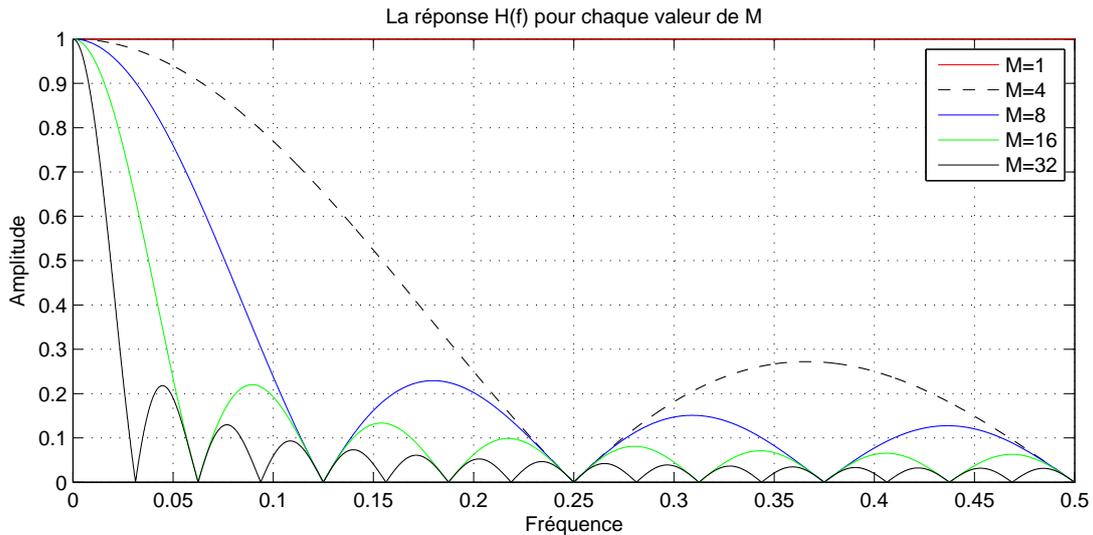


FIGURE 6.14 – La réponse fréquentielle du filtre H(f) en fonction de la fréquence pour divers tailles de filtre M

La perte des échantillons de mesure n'a pas d'importance dans notre cas. En effet, le système de mesure envoie des intervalles de temps en permanence et récupère les résultats en continu durant la phase de calibration ou d'ajustement des offsets.

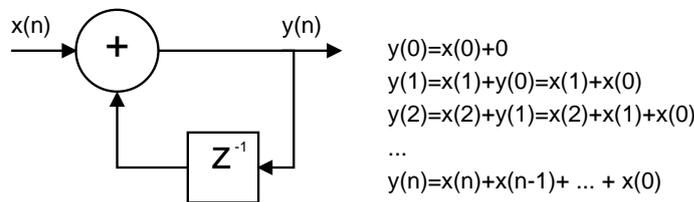


FIGURE 6.15 – Architecture basique de calcul par bloc de la moyenne

La mise en œuvre du moyennneur est relativement simple, la figure 6.15 montre le fonctionnement de base de la cellule d'accumulation des mesures. Elle est composée d'un additionneur et d'un bloc de retard d'un coup d'horloge. Le résultat est la somme de M échantillons des mesures précédentes. Pour chaque coup d'horloge, la sortie est égale à la somme des échantillons précédents et l'échantillon actuel à l'entrée. L'équation (6.19) montre la transformé en Z du filtre de taille M normalisée par rapport à M. Ce type d'architecture nécessite M coups d'horloge (multiplexage temporel du calcul) pour obtenir une valeur de la moyenne. Le temps de calcul augmente en fonction de M. Pour M=128, on obtient la valeur moyenne après 128 coups d'horloge. Pour obtenir la valeur moyenne réelle, il suffit de décaler la sortie vers la droite de log<sub>2</sub>(M) bits (7 bits pour une taille de 128). L'opération de décalage doit être effectuée pour chaque M coups d'horloge suivi d'une opération d'initialisation du "buffer" de données. Dans notre cas, on maintient l'ancienne valeur de la moyenne durant le cycle en cours de calcul, pour que le composant en amont puisse lire à tout moment le résultat de calcul (voir simulations). La figure 6.16 montre le schéma bloc du fonctionnement du filtre.

$$Y(Z) = \sum_{j=0}^{M-1} X(Z) \times Z^{-j} \quad (6.10)$$

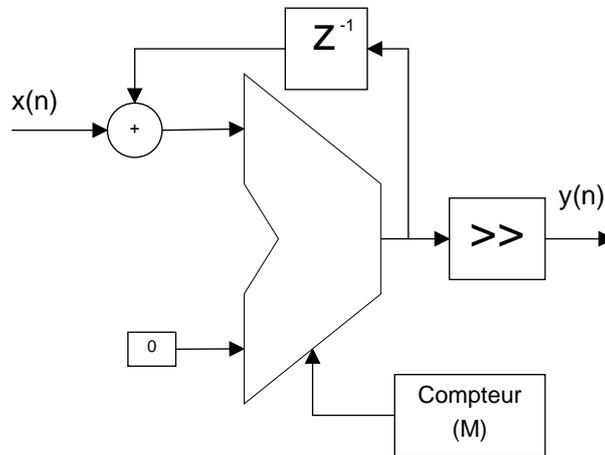


FIGURE 6.16 – Architecture globale de calcul par bloc de la moyenne

L'architecture du moyennneur basée sur la mémoire FIFO est semblable à celle du calcul par bloc. Elle diffère uniquement par la mémorisation des M échantillons (profondeur de la mémoire). L'architecture permet de générer le résultat  $y(n)$  pour chaque coup d'horloge. Elle est plus performante au niveau débit de données (un résultat par coup d'horloge) à la sortie, mais elle consomme plus de ressources lorsque la taille du filtre devient importante.

### 6.3.2.4 Implémentation (Filtrage des mesures)

#### Entité

Le tableau 6.5 illustre la description des signaux de l'entité du filtre moyennneur en figure 6.17. Ce composant contient une entrée sur 40 bits. En effet, la taille de ce dernier peut être ajustée en fonction du bus de données à l'entrée (composant en aval). Il dispose également d'une entrée d'initialisation et de validation et une entrée d'horloge synchronisée avec la donnée. Pour les sorties, la taille est définie en fonction de la taille des données en entrée, de la taille du filtre M et de la précision de calcul des résultats désirée (position de la virgule fixe). D\_outReady est activée uniquement lorsque le résultat est disponible pendant un coup d'horloge puis repasse à zéro. Le composant de tri peut récupérer le résultat par la détection du front du signal D\_outReady. Il est possible de lire le bus de résultats d'une façon asynchrone, en d'autres termes sans prendre en considération l'indicateur de résultat, car le bus de sortie maintient en mémoire la donnée précédente.

**Programme : Voir Annexe V**

#### Simulations

La figure 6.19 illustre les chronogrammes du filtre moyennneur. Le bus de données prend respectivement une série de 10 valeurs à l'entrée du filtre (2, 4, 4, 16, 5, 2, 1, 8, 8, et 8) sur

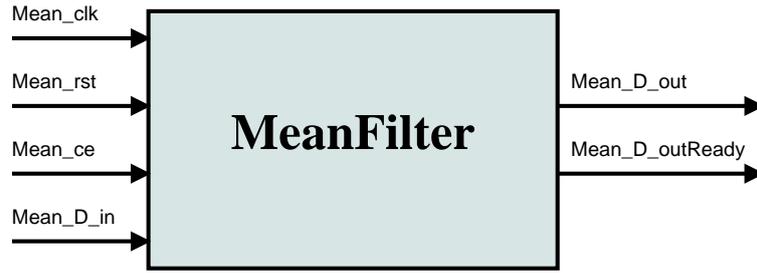


FIGURE 6.17 – Entité du filtre de lissage du bruit de mesure

TABLE 6.5 – Description des signaux du moyeneur

Signal	Direction	Description
Mean_rst	Entrée	Initialisation du composant
Mean_clk	Entrée	Horloge de données de mesure
Mean_ce	Entrée	Activation du composant
Mean_D_in	Entrée	Données de mesure sur 40 bits
Mean_D_outReady	Sortie	Indicateur de présence de résultat à la sortie
Mean_D_out	Sortie	Sortie du lisseur

40 bits. Ces derniers simulent les mesures acquises par FPGA du circuit TDC. Le nombre de bits consacré à la partie fractionnelle est nul. Par conséquent, on perd les bits après le décalage de 3 bits de l'accumulateur pour  $M = 2^3$ . La sortie Mean\_D\_out prend les valeurs 4 ou 6. On observe le passage à 1 de l'indicateur de fin de cycle après 8 coups d'horloge (filtre de taille 8). D'après les chronogrammes, la valeur 4 correspond à  $(\frac{2+1+8+8+8+2+4+4}{8} = \frac{37}{8} = 4,62)$ . Après l'opération de décalage, la sortie prend la valeur 4 (partie entière) au lieu de 4,62. L'erreur 0,62 est due à la perte des bits de décalage ; l'erreur maximale pour 3 bits est égale à  $2^{-1} + 2^{-2} + 2^{-3} = 0,87 \cdot LSB$ . Pour récupérer les bits, il suffit tout simplement de ne pas effectuer l'opération du décalage dans le cas où  $m\_virgule=0$ . Dans le cas contraire, il faut affecter une valeur non nulle au paramètre dans le programme  $m\_virgule \geq \log_2(M)$ .

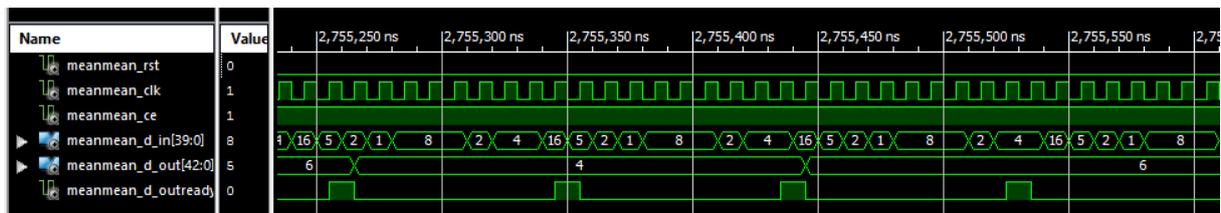


FIGURE 6.18 – Chronogrammes du filtre moyeneur d'une taille égale à 8

## Ressources

Cette section sera dédiée à l'analyse des ressources matérielles en fonction des paramètres du filtre pour un niveau de vitesse fixe de la cible (-1 pour FPGA Virtex-6). La figure 6.19 montre l'évolution des ressources en fonction de la taille du bus de données à l'entrée et la taille du filtre. Elle décrit la fréquence maximale autorisée, le nombre des registres FF et le nombre de LUTs pour différents nombres de bits.

On constate que la fréquence maximale supportée par l'architecture diminue lentement lorsque la taille du filtre augmente, elle passe de 365 MHz pour 4 bits à 354 MHz pour 64 bits. Le nombre de registres FF et de LUT augmente lentement également en fonction de  $M$ . Ces deux derniers croissent d'une manière linéaire et identique.

Dans la figure 6.19, la fréquence maximale décroît rapidement lorsque la taille du bus d'entrée augmente. Le passage de 4 à 8 bits entraîne une chute rapide d'environ 100 MHz par rapport à la fréquence maximale. Puis, elle continue à décroître en fonction de la taille du bus. Les mêmes observations pour le nombre de registres FF et le nombre de LUT, montrent une croissance rapide avec l'augmentation de la taille du bus. Le nombre de LUT passe de 24 pour 4 bits à 204 pour 64 bits, de la même façon le nombre de registres FF passe de 18 pour 4 bits à 136.

En résumé, l'effet de la taille du bus dégrade la fréquence maximale supportée par l'architecture. De plus, le nombre des ressources en surface augmente rapidement aussi avec l'augmentation de la taille du bus. L'effet induit par  $M$  reste négligeable par rapport à la taille du bus. Ce phénomène est dû au fait que le changement de la taille du filtre avec  $M$  engendre une augmentation des registres de calcul de  $\log_2(M)$ . En revanche l'augmentation de la taille du bus à l'entrée de  $M$  bits entraîne une croissance de tous les registres de  $M$  bits. Dans la phase de mesure, on fixe la taille du filtre à 64 bits avec des données sur 40 bits. Dans ces conditions, la fréquence maximale est supérieure à 350 MHz. Par conséquent, la contrainte de la fréquence maximale de l'interface d'acquisition est respectée ( $\simeq 300$  MHz).

### 6.3.2.5 Soustracteur (C2)

Le soustracteur (C2) (voir la figure 6.12) est l'élément de la chaîne de calibration qui permet de calculer et mémoriser le délai moyen induit par le chemin aller-retour  $d_0^i$  de l'horloge dans la "Phase 0" puis de compenser cet effet dans la phase de mesure "Phase 1" des retards  $d_{MM}^i$  pour chaque canal. La figure 6.20 illustre la machine à état des deux phases 0 et 1, elle est constitué de 4 états :

- État initial : Initialisation de la sortie du soustracteur (Phase 0)
- État '2' : Lecture et mémorisation de  $d_0^i$  (sortie  $S_3$ ) (Phase 0)
- État '3' et '4' : Compensation de l'effet de  $d_0^i$  dans la mesure  $d_{MM}^i$  ( $S_4 = S_1 - S_3$ )

Le cycle commence par la mise à un du signal "Start". La phase 0 dure la période de calibration en coups d'horloge. La période de lecture de  $S_3$  doit être largement supérieure à la taille du filtre C5 afin d'assurer la lecture d'une valeur correcte et stable de  $d_0^i = S_3$  et d'autre part le fonctionnement dans le régime permanent du filtre. La machine passe à la phase 1 à la fin de la période et reste en boucle infinie (fonctionnement en soustracteur normal ( $S_4 = S_1 - S_3$ )).

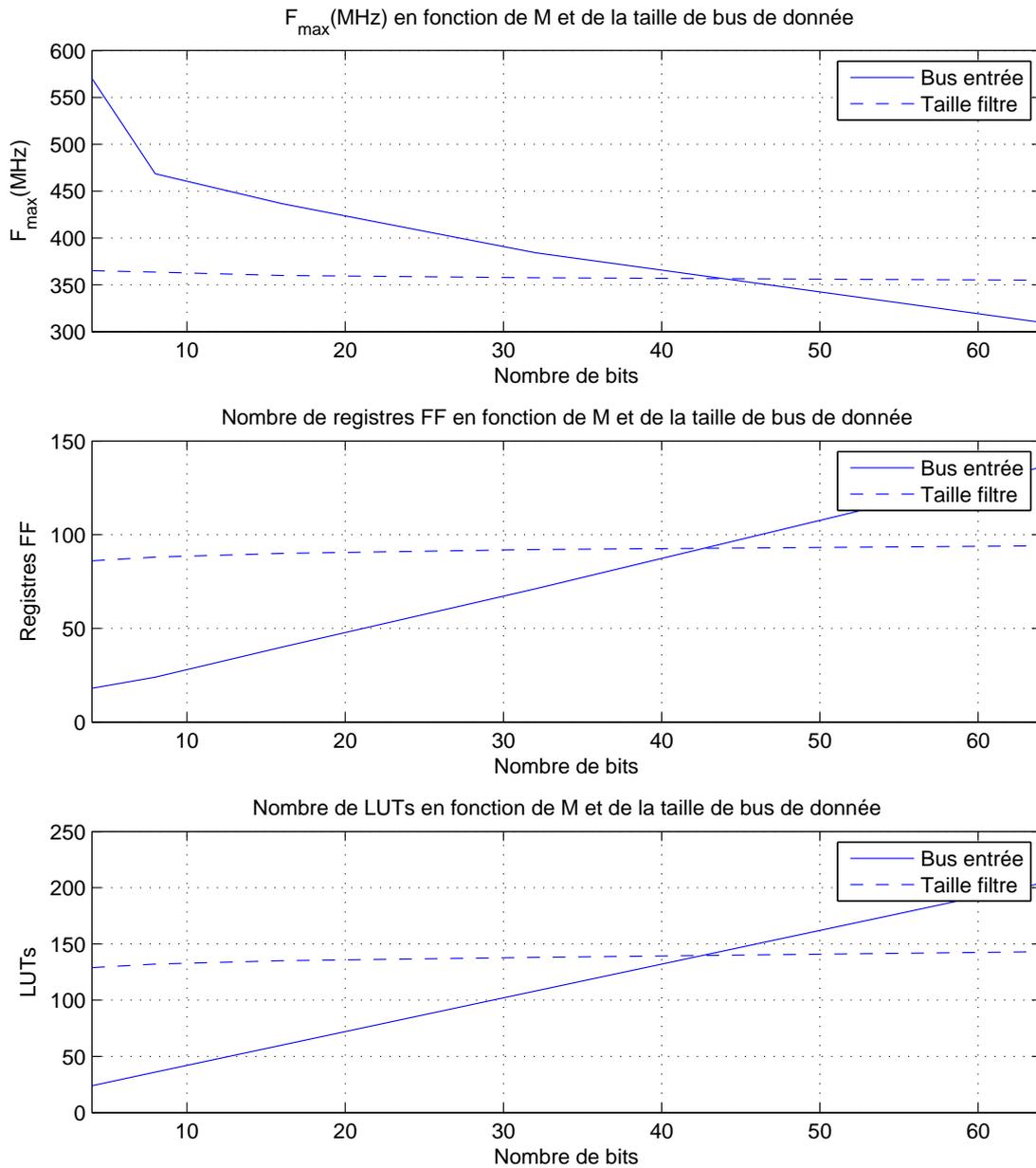


FIGURE 6.19 – Ressources normalisées en fonction de la taille du bus de données à l’entrée

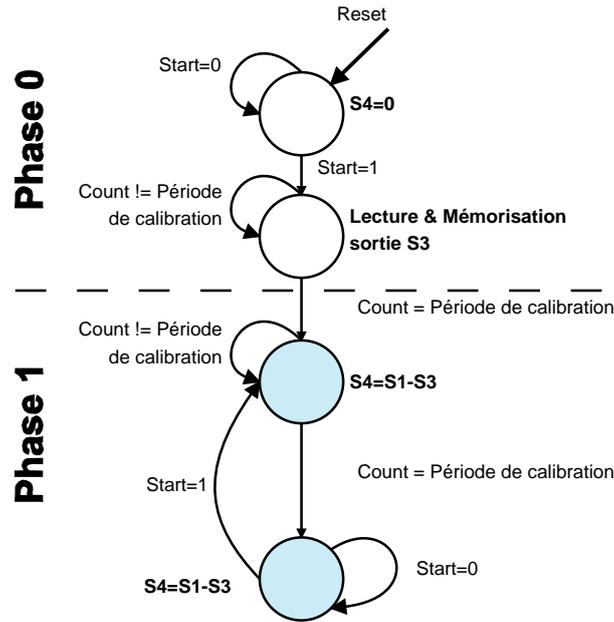


FIGURE 6.20 – Machine à état du soustracteur avec mémoire

### Entité

La figure 6.21 indique l'entité du circuit de compensation de  $d_0^i$  pour quatre canaux indépendants (A, B, C et D). Le tableau 6.6 indique la description des signaux essentiels d'E/S.

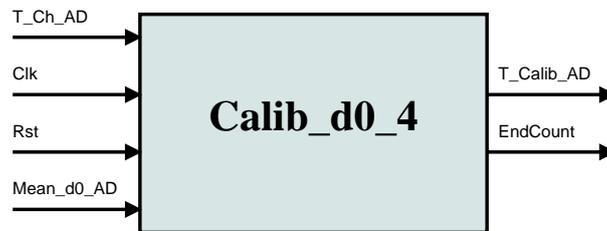


FIGURE 6.21 – Entité circuit de compensation de  $d_0^i$

TABLE 6.6 – Description des signaux du moyeneur

Signal	Direction	Description
T_Ch_AD	Entrée	4 canaux $S_1$ sur 40 bits (sorties de C1)
Mean_d0_AD	Entrée	4 canaux $S_3$ sur 40 bits (sorties de C5)
T_Calib_AD	Sortie	4 canaux $S_4$ sur 40 bits (sorties de C2)
EndCount	Sortie	Indicateur de fin du cycle de calibration

### 6.3.2.6 Simulations et résultats

Dans cette section on va valider le fonctionnement du circuit de calibration des retards dus aux composants dans le chemin des horloges pour les quatre canaux. On va comparer les résultats du circuit synthétisé en VHDL (illustré dans la figure 6.22) et le modèle matlab en utilisant les mêmes données de simulation à l'entrée des deux modèles. L'objectif des simulation des modèles est l'estimation du délai  $d_1^i$  pour chaque canal et la compensation de l'intervalle du temps  $T_i$  ainsi que le délai induit par la chaîne  $d_0^i$ .

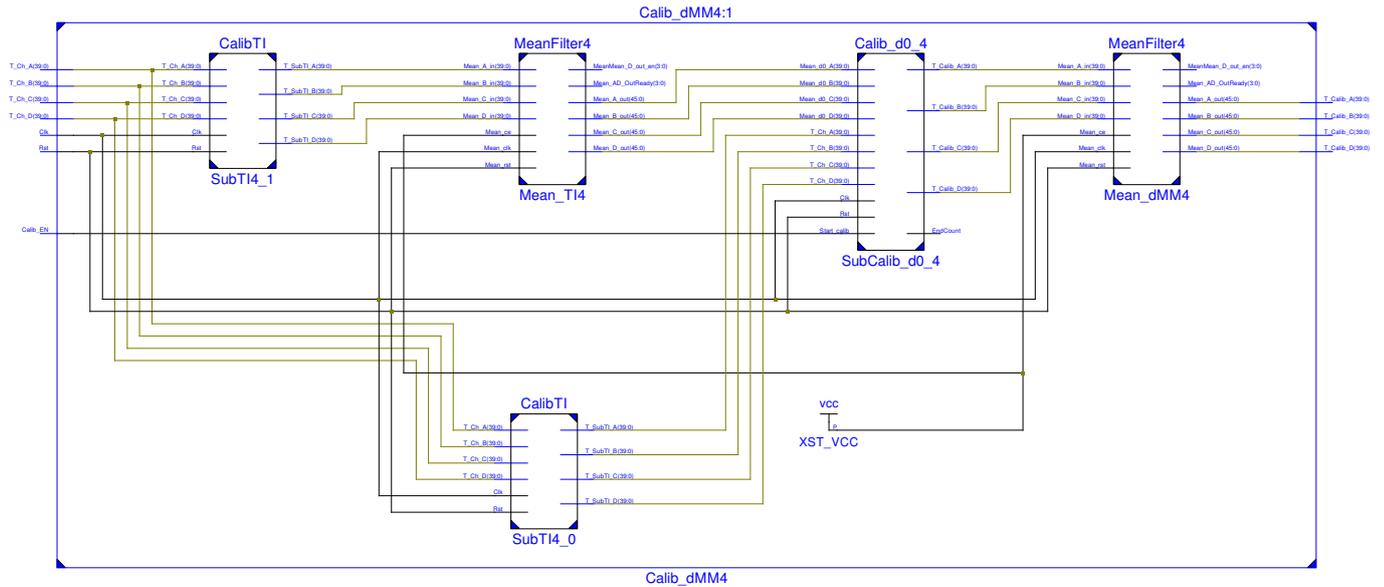


FIGURE 6.22 – Schéma bloc du modèle VHDL du circuit de calibration pour quatre canaux du modèle théorique illustré dans la figure 6.12

#### Les paramètres de simulation :

Les données à l'entrée de chaque canal sont composées de trois paramètres :  $T_i$ ,  $d_0^i$  et  $d_1^i$ . Elles sont subdivisées en deux phases (phase 0 et 1). La phase 0 est composée de  $10^5$  échantillons, la phase 1 de  $3 \cdot 10^5$ . La gigue de mesure de chaque canal est supposée égale à 120 ps (1 LSB). La taille du filtre est fixée à 64. Les données sont normalisées et arrondies par rapport à un pas de 13,02 ps (bruit du circuit de mesure). Les vecteurs des entrées des canaux A à D sont constitués de la somme de trois composantes :  $T_i$ ,  $d_0^i$  et  $d_1^i$ . Les trois composantes ont un comportement aléatoire dû aux bruits de mesure. Le retard dû au chemin d'horloge ( $d_0^i$ ) est de l'ordre de 100 ns pour les divers canaux. L'objectif des deux modèles de calibration est d'estimer le retard dû uniquement au câble  $d_1^i$  qui vaut à la fin de calibration de l'ordre de 10 ns. Ci-dessous, les paramètres des trois composantes aléatoires pour chaque canal. La figure 6.23 montre les données des délais aller-retour  $d_{MM}^i$  pour les quatre canaux A à D pour les phases 0 (utilisation des câbles de référence) et 1 (présence des câbles réels).

1. Canal A :
  - $T_i$  :  $\mu = 1,28\mu s$  ,  $\sigma = 1LSB$

- $d_0^i : \mu = 100ns , \sigma = 1LSB$
  - $d_1^i : \mu = 10ns , \sigma = 1LSB$
2. Canal B :
    - $T_i : \mu = 2,56\mu s , \sigma = 1LSB$
    - $d_0^i : \mu = 100ns , \sigma = 1LSB$
    - $d_1^i : \mu = 10ns , \sigma = 1LSB$
  3. Canal C :
    - $T_i : \mu = 3,84\mu s , \sigma = 1LSB$
    - $d_0^i : \mu = 100ns , \sigma = 1LSB$
    - $d_1^i : \mu = 10ns , \sigma = 1LSB$
  4. Canal D :
    - $T_i : \mu = 5,15\mu s , \sigma = 1LSB$
    - $d_0^i : \mu = 100ns , \sigma = 1LSB$
    - $d_1^i : \mu = 10ns , \sigma = 1LSB$

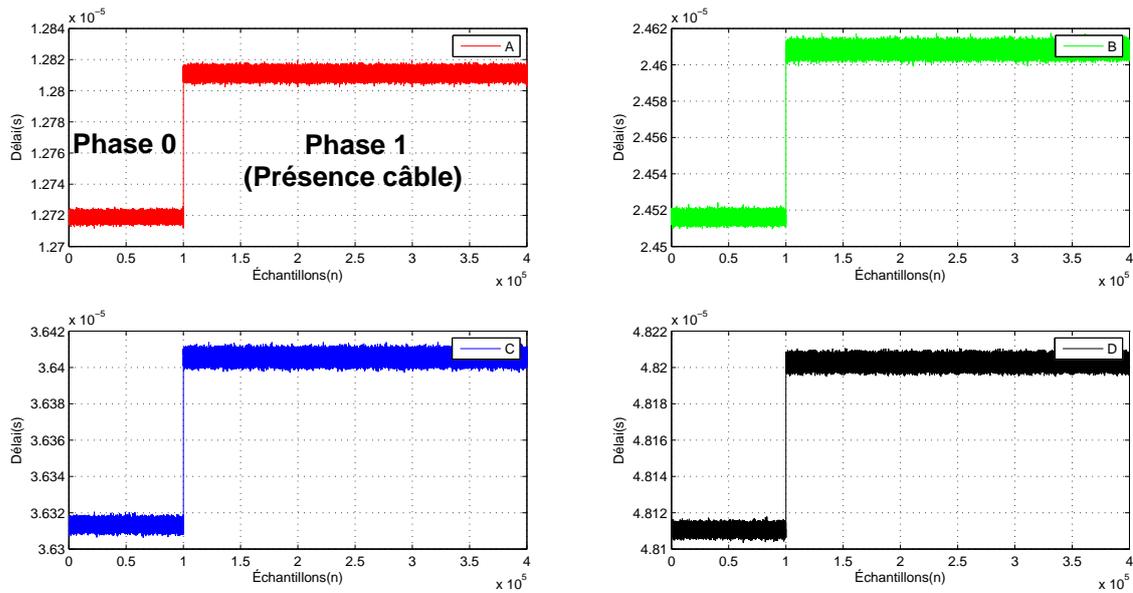


FIGURE 6.23 – Données des canaux A à D avant la calibration pour les deux phases (0 et 1).  $d_1^i = 0$  durant la phase 0 (absence du câble)

La figure 6.24 illustre les résultats à la sortie du modèle Matlab de calibration pour les quatre canaux. Le signal carré "EN" indique le début de calibration, il est initialement à zéro durant une période égale à  $8 \cdot 10^4 LSB$ . Cette période indique la durée de la phase 0. On constate durant la phase 0, pour  $n \leq 4 \cdot 10^4 LSB$  que toutes les sorties sont à 0 (mode calibration désactivé). Lorsque la calibration est activée (EN=1) pour  $4 \cdot 10^4 \leq n \leq 10^5$ , les sorties oscillent autour de zéro. Durant cette phase, l'effet des câbles est nul ( $d_1^i = 0$ ). Pendant la phase 1, les données oscillent autour de 10 ns. La figure 6.25 illustre les résultats du modèle VHDL pour

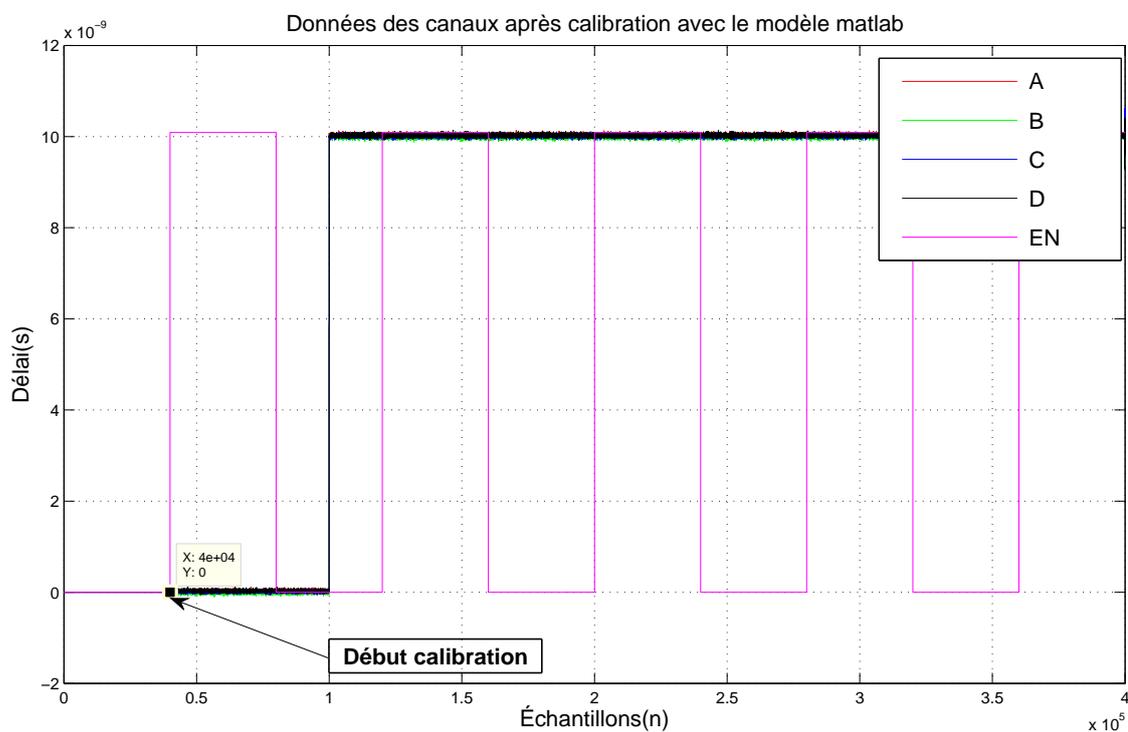


FIGURE 6.24 – Données des canaux A à D à la sortie du circuit de calibration en utilisant le modèle matlab

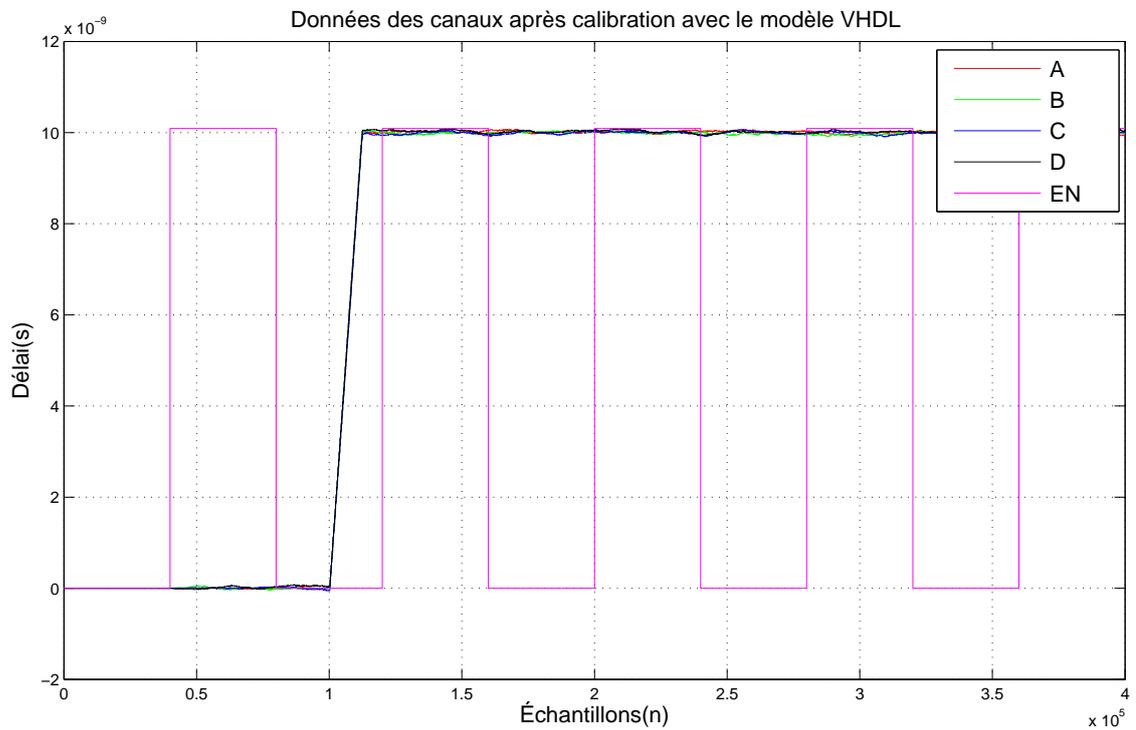


FIGURE 6.25 – Données des canaux A à D à la sortie du circuit de calibration en utilisant le modèle VHDL

les quatre canaux. On remarque une tendance des canaux vers la valeur 10 ns.

En résumé, les deux modèles permettent d'estimer correctement le délai  $\hat{d}_1^i$  et compensent les retards indésirables. La figure 6.26 montre la distribution de  $\hat{d}_1^i$  pour le canal A des deux modèles. Les deux modèles suivent une distribution gaussienne caractérisée par les paramètres suivants :

- Modèle VHDL :  $\mu = 10,18ns, \sigma = 26,97ps$
- Modèle Matlab :  $\mu = 10,03ns, \sigma = 25,93ps$

Les deux modèles permettent de converger vers l'estimation moyenne  $\hat{d}_1^i$  réelle en réduisant le bruit de mesure. La gigue de mesure passe de 120 ps à 26,97 ps pour le modèle VHDL et de 120 ps à 25,93 ps pour le modèle Matlab. Le circuit de calibration permet également de réduire le bruit de mesure d'un facteur égal à 4 par rapport à la mesure initiale.

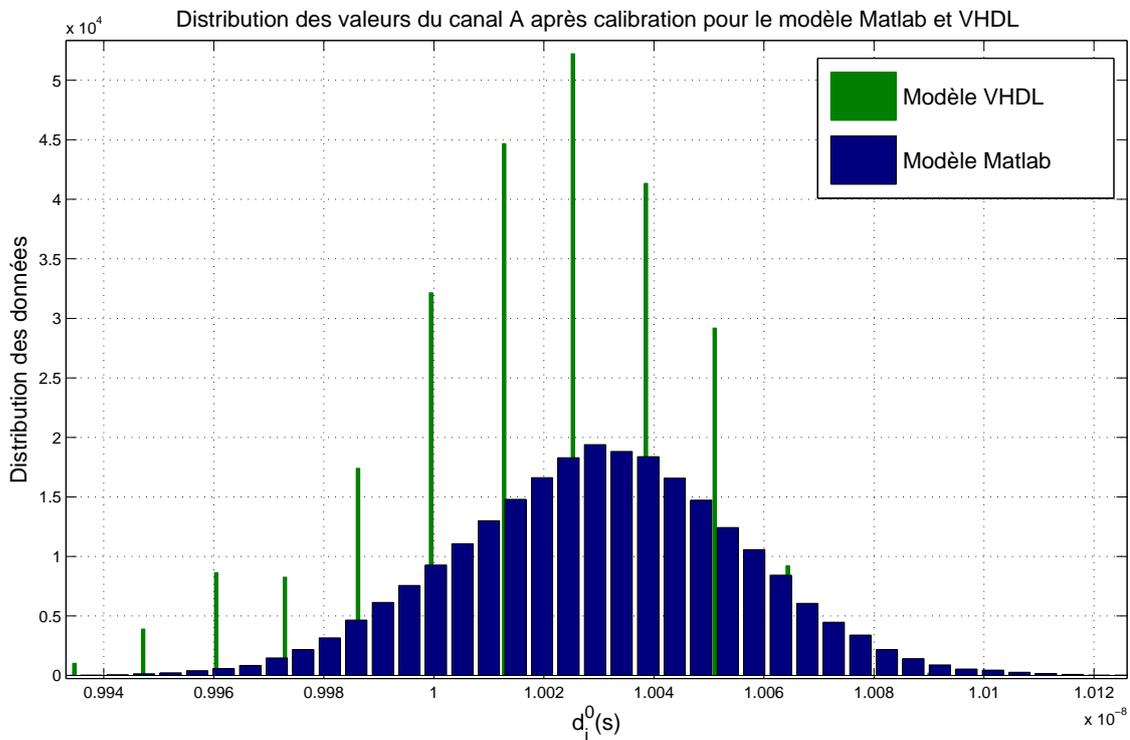


FIGURE 6.26 – Distribution des données dû canal A pour les modèles Matlab et VHDL dans le régime permanent ( $1.5 \times 10^5 \leq n \leq 4 \times 10^5$ )

### 6.3.3 Tri des canaux

#### 6.3.3.1 Introduction

Cette section abordera la deuxième étape après l'acquisition et la calibration des mesures (voir la figure 6.10). L'opération de tri est la première étape durant l'implémentation sur FPGA de l'algorithme de calcul des écarts d'ajustement  $d_{xi}$  des horloges (voir la figure 5.3). L'intérêt global est de trier les mesures de manière à obtenir des valeurs différentielles positives pour chaque couple d'antennes ( $i, i+i$ ). Par conséquent, il s'agit d'obtenir des écarts  $d_{xi}$  positifs qui seront intégrés dans un circuit à retard programmable de haute précision. Ici, on propose un composant de tri pour quatre canaux. Il est constitué d'une unité de tri pour trois canaux, plus un canal dont le retard est maximal (voir la figure 6.27). En effet, la description sera focalisée sur le fonctionnement du composant de tri de trois canaux avec leurs indices. La première valeur (valeur maximale) avec l'indice correspondant est générée par le détecteur de la valeur maximale. La figure 6.27 présente l'architecture globale du circuit de tri de 4 canaux.

#### 6.3.3.2 Tri de trois canaux

La figure 6.28 illustre l'entité du composant de tri de trois canaux avec les indices. La restitution des indices pour les canaux permet par la suite de faire correspondre l'écart calculé au canal correspondant. Le composant de tri est composé de trois entrées parallèles sur N bits, et de trois sorties parallèles triées par ordre décroissant. Le composant est muni d'une sortie indiquant la présence de données dans les bus de sortie. Ce composant est illustré dans le tableau 6.28 et la figure 6.7. Les données à l'entrée et à la sortie sont synchronisées par l'horloge Clk. Le programme VHDL se trouve dans l'annexe VI.

TABLE 6.7 – Description des signaux du composant de tri de trois canaux

Signal	Direction	Description
Rst	Entrée	Initialisation
Clk	Entrée	Horloge synchrone de données
D_in_x	Entrée	Entrée 1 à 3 sur N bits
Data_ready	Sortie	Indicateur de données disponible sur les bus de sortie
Out_x	Sortie	Sorties de tris sur 40 bits
I_x	Sortie	Indices sur deux bits des sorties

#### 6.3.3.3 Simulations

La figure 6.29 montre les chronogrammes de simulation du composant Sort\_3 pour diverses valeurs d'entrée. Les entrées 1, 2 et 3 sont initialisées respectivement par 1, 21 et 64, les sorties 1, 2 et 3 prennent les valeurs 64, 21 et 1. On constate que le composant permet effectivement de trier les entrées par ordre décroissant. Les mêmes constatations pour les valeurs des entrées (8,34, 3) engendrent les sorties (34, 8, 3). L'indicateur de présence de données à la sortie dure un coup d'horloge avec une fréquence deux fois plus faible que la fréquence d'horloge de référence Clk. On observe également que les indices évoluent avec les données.

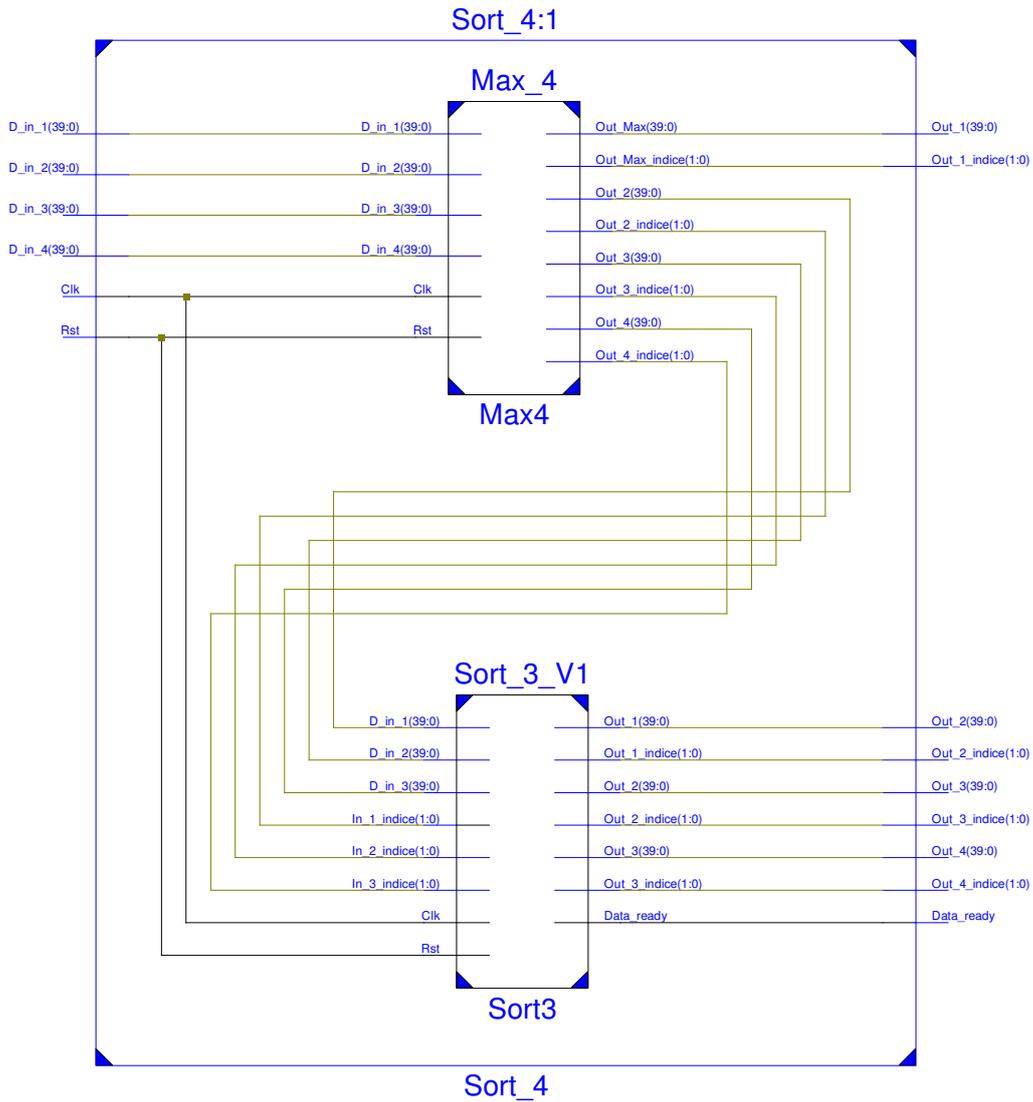


FIGURE 6.27 – Entité du composant de tri de quatre canaux

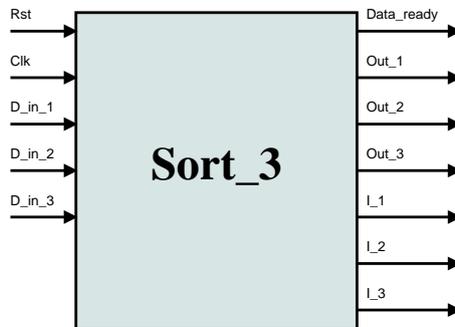


FIGURE 6.28 – Entité du composant de tri de trois canaux

En effet, le composant en amont permet de récupérer les données de mesures avec la moitié du débit de données à l'entrée (perte d'un échantillon sur deux des résultats). Ce comportement n'a pas d'effet sur le circuit de calcul des offsets. Ce dernier récupère les échantillons à la sortie et le circuit d'acquisition dispose d'un temps nécessaire permettant le calcul des écarts d'ajustement. On constate dans la figure 6.29 que les données des sorties sont synchronisées par l'horloge, et le composant génère les résultats pour chaque coup d'horloge avec une latence d'une période d'horloge. L'utilisateur peut lire les résultats avec le même débit à l'entrée synchronisé par l'horloge Clk.

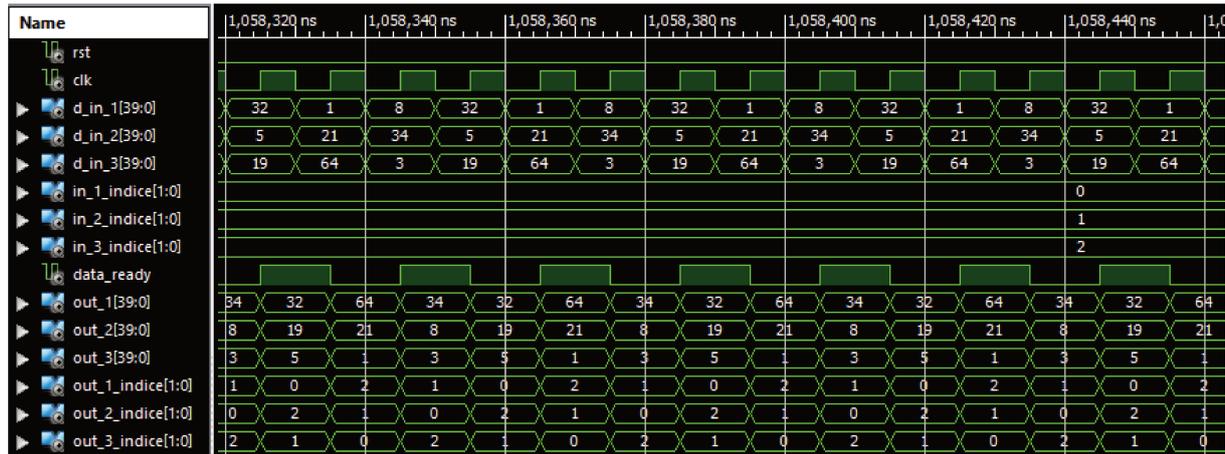


FIGURE 6.29 – Chronogrammes des retards mesurés et leurs tris pour les quatre canaux

La synthèse du composant dans une cible FPGA Virtex-6 avec un niveau de vitesse (-1) indique une fréquence maximale de 793,70 MHz, cette dernière dépasse largement la fréquence d'acquisition des données séries (300 Mbps, 40 bits). La section suivante abordera l'implémentation du circuit permettant le calcul itérative des écarts d'ajustement.

La figure 6.30 illustre les chronogrammes de la simulation de la détection de la valeur maximale pour quatre canaux. Le circuit permet de détecter la valeur maximale qui est la première valeur dans le tableau de tri, les trois autres canaux sont conservés et envoyés au circuit de tri à trois canaux illustré précédemment. Pour le vecteur à l'entrée égal à (1, 21, 64, 48), la sortie Out\_max prend la valeur 64 avec l'indice 2 donc l'entrée 3 qui contient la valeur maximale (indice commence à 0) (voir la figure 6.30). On conclut que le circuit permet effectivement de détecter la valeur maximale ainsi que l'indice de cette dernière.

## 6.3.4 Calcul des écarts d'ajustement des horloges

### 6.3.4.1 Fonctionnement

Après la phase de calibration des canaux et le tri des canaux, la phase suivante consiste à calculer des écarts des offsets de compensation des horloges permettant de mettre en phase ces dernières durant la phase de synchronisation. Cette section sera dédiée à l'implémentation du circuit permettant le calcul des écarts d'ajustement des horloges pour les divers canaux. D'après l'algorithme illustré dans la figure 5.3, après le filtrage et le tri des mesures par ordre

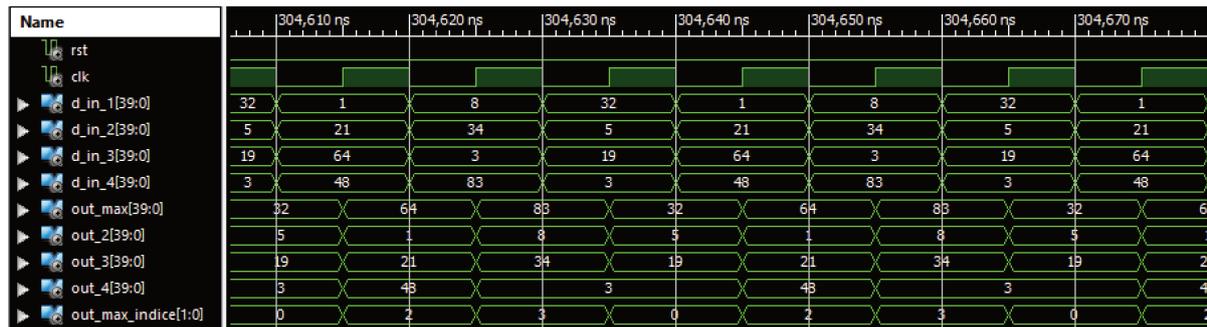


FIGURE 6.30 – Chronogrammes de simulation de la détection de la valeur maximale des quatre canaux de mesures

décroissant, le dernier canal est défini comme étant la référence de calcul et par conséquent la valeur de  $dx_x$  est mise à 0. Ci-dessous un exemple illustrant les étapes de calcul des écarts pour 4 canaux. On considère des valeurs numériques des mesures issues du circuit TDC avec une résolution de mesure de 13 ps (LSB).

1. Résultats des mesures des retards aller-retour  $T_i = d_{MM}^i$  pour les divers canaux en valeurs entières (normalisées par rapport à LSB). Exemple pour le retard entier  $T_1 = 5$  correspond à un retard temporel de  $5 \times 1 \text{ LSB} = 5 \times 13 \text{ ps} = 75 \text{ ps}$ .

- $T_1 = 5, T_2 = 8 \Rightarrow \Delta T_{1,2} = 3$
- $T_2 = 8, T_3 = 10 \Rightarrow \Delta T_{2,3} = 2$
- $T_3 = 10, T_4 = 15 \Rightarrow \Delta T_{3,4} = 5, dx_4 = 0$

2. Calcul des écarts d'ajustement  $dx_i$  :

- $dx_4 = 0$
- $dx_3 = \Delta T_{3,4} + dx_4 = 5$
- $dx_2 = \Delta T_{2,3} + dx_3 = 7$
- $dx_1 = \Delta T_{1,2} + dx_2 = 10$

3. Ajustements des offsets des horloges pour les antennes  $A_i$ ,  $T'_i$  étant le retard aller-retour après l'ajustement

- $A_1 : T'_1 = T_1 + dx_1 = 5 + 10 = 15$
- $A_2 : T'_2 = T_2 + dx_2 = 8 + 7 = 15$
- $A_3 : T'_3 = T_3 + dx_3 = 10 + 5 = 15$
- $A_4 : T'_4 = T_4 + dx_4 = 15 + 0 = 15$

La figure 6.31 illustre le processus de synchronisation des horloges en intégrant des écarts  $d_x^i/2$  sur le chemin de chaque horloge. L'écart du canal 4 est nul ( $dx_4 = 0$ ) car il est défini comme le canal de référence, tous les autres canaux sont ajustés par rapport à ce dernier. Durant la phase de synchronisation des horloges, les quatre canaux sont alignés après une activation instantanée par le signal de synchronisation "Sync" commandé par une source stable et précise

(voir l'architecture de synchronisation). En pratique, le circuit de division d'horloge 1 vers 4 dispose d'un "skew" de sortie de quelques picosecondes. Cette valeur est inférieure à 1 LSB du circuit TDC et peut être négligée. Réciproquement, l'horloge de référence doit être précise et stable dans le temps.

On constate pour chaque canal que la somme  $T'_i = T_i + dx_i$  est identique pour tous les canaux et par conséquent, tous les offsets des horloges sont ajustés et synchrones pour toutes les horloges. Le retard aller-retour après la phase d'ajustement est noté par  $T'_i$ . L'algorithme itère l'opération de calcul et d'ajustement jusqu'à l'obtention d'une valeur minimale de  $T'_i$ . Le temps de convergence de l'algorithme est noté  $T_{sync}$ . Il dépend des paramètres internes de l'algorithme et de la précision désirée (voir l'algorithme dans la figure 5.3). On va évoquer brièvement la description en VHDL de l'architecture itérative du circuit de calcul des écarts d'ajustement. L'équation 5.3 peut être écrite sous la forme suivante :

$$dx_i = dx_{i-1} - \Delta T_{i,i-1} \tag{6.11}$$

$$dx_i = dx_{i-1} - (T_i - T_{i-1}) \tag{6.12}$$

La transformée en Z de l'équation (6.12) est décrite dans l'équation (6.13).

$$d_x(Z) = Z^{-1}d_x(Z) - (T(Z) - Z^{-1}T(Z)) \tag{6.13}$$

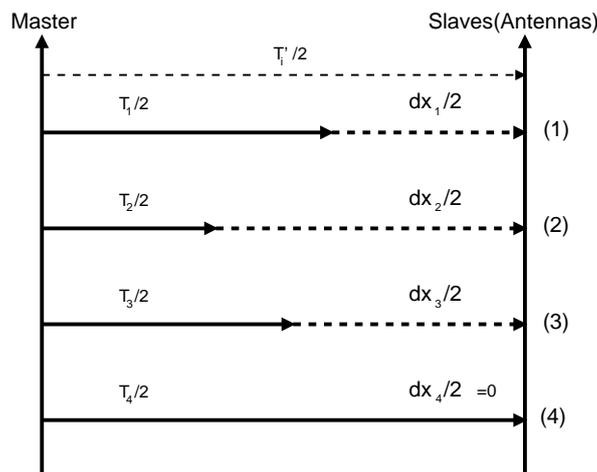


FIGURE 6.31 – Chronogrammes d'ajustement des offsets des horloges

La figure 6.32 montre le schéma bloc récursif de l'implémentation de l'équation (6.13). Il est constitué de deux soustracteurs et deux blocs de retard ( $Z^{-1}$ ). Le circuit permet de calculer l'écart  $d_x^i$  pour chaque coup d'horloge d'une façon itérative suivant l'équation (6.13).

Les mesures après calibration, filtrage et tri sont cadencées en série sur M bits (avec M la taille du bus de sortie du circuit de lissage), idem pour les sorties. Ils sont multiplexés dans le temps et synchronisés par l'horloge maître du circuit. Ci-dessous le format des vecteurs d'entrée et de sortie du circuit (voir l'équation (6.13)).

$$T = \begin{bmatrix} T_1 \\ T_2 \\ \dots \\ T_N \end{bmatrix}, d_x = \begin{bmatrix} dx_1 \\ dx_2 \\ \dots \\ dT_N \end{bmatrix} \quad (6.14)$$

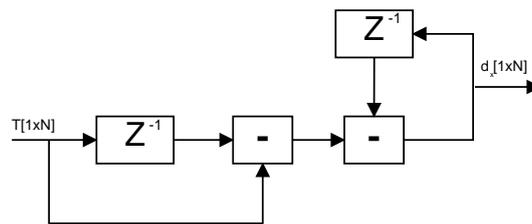


FIGURE 6.32 – Schéma bloc du circuit de calcul des écarts d'ajustement des horloges

## 6.3.4.2 Entité

FIGURE 6.33 – Entité du circuit de calcul des écarts d’ajustement des horloges  $d_{xi}$ 

Le tableau 6.8 décrit les signaux du composant 6.33. Il dispose essentiellement d’une entrée de retards filtrés et triés sur  $M$  bits (Exemple  $M=40$  bits) et une entrée sur  $N$  bits qui définit le nombre de canaux à ajuster ( $N=3$  pour 8 canaux ( $2^3$ )). Il contient également deux indicateurs de début et de fin de cycle et la sortie des écarts d’ajustement synchronisés par l’horloge  $Clk$  sur  $M$  bit.

TABLE 6.8 – Description des différents signaux de circuit de calcul des écarts d’ajustement des offsets des horloges

Signal	Direction	Description
Rst	Entrée	Initialisation
Clk	Entrée	Horloge des entrées et résultats
Ce	Entrée	Activation du composant
Num	Entrée	Entrée sur $N$ bits indiquant le nombre de canaux (Taille de $T_i$ )
T_in	Entrée	Données de mesures $T_i$ de taille $M$
StartCyc	Sortie	Indicateur de début de calcul
EndCyc	Sortie	Indicateur de fin de calcul
d_x_out	Sortie	Écarts d’ajustement de taille $M$ $d_x^i$

## 6.3.4.3 Programme : Voir annexe VII

## 6.3.4.4 Simulation et performances

La figure 6.34 montre les chronogrammes de simulation du composant qui calcule les écarts d’ajustement des offsets. La taille du bus d’E/S est fixée à 40 ( $M=40$ ) avec  $N=3$  (synchronisation entre calcul de 7 antennes (écarts)). Le bus de données à l’entrée est branché avec une mémoire d’une taille de 8 mots sur 40 bits, contenant la séquence suivante triée d’avance (1, 2, 3, 4, 8, 16, 32). Un compteur *count* (Fig. 6.34) sur  $N$  bits permet de parcourir la mémoire synchronisée par l’horloge de référence  $Clk$ . On observe également l’apparition d’une sortie de test  $Out\_test$  sur 40 bits qui contient la valeur après l’ajustement des offsets ( $T' = T + d_x$ ) qui doit être identique pour tous les canaux après la phase d’ajustement.

Le cycle de calcul débute par la mise à 1 de l'indicateur de début du cycle "StartCyc". La séquence des écarts  $d_x$  est la suivante (31, 30, 29, 28, 24, 15, 0) pour les mesures du vecteur d'entrée  $T$  (1, 2,3, 4, 8, 16 et 32). La sortie de test indique une valeur 32 pour chaque canal (canal 1 : 1+31, canal 2 : 2+30, ...), donc les valeurs calculées correspondent parfaitement aux écarts d'ajustement permettant d'aligner tous les canaux ayant des retards différents.

Les données des écarts sont cadencées par l'horloge Clk sous forme de 40 bits. Une mémoire en amont est nécessaire pour maintenir les valeurs en mémoire. Elle sera mise à jour en permanence par une période ajustable (période de synchronisation des horloges physiques). Les valeurs de la mémoire seront ensuite dupliquées dans les circuits de programmation des retards (DTC) dans la phase de synchronisation qui consiste tout simplement à transférer les écarts au circuit DTC intégré dans le chemin des divers horloges, et à déclencher le circuit du diviseur d'horloge par un signal stable et précis.

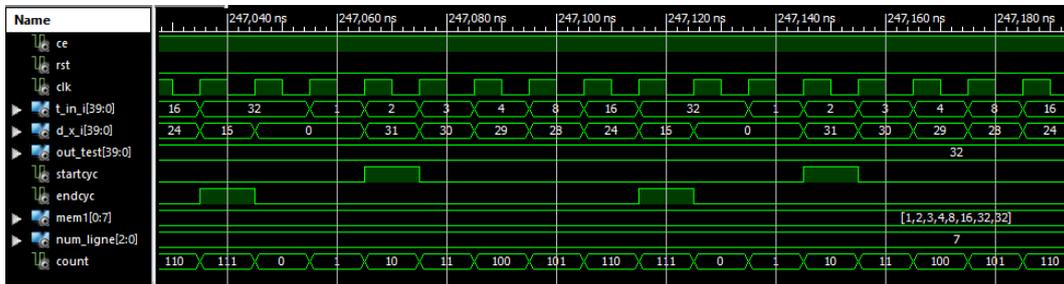


FIGURE 6.34 – Chronogrammes de simulation de l'entité OffsComp

Le tableau 6.9 montre les résultats de synthèse sur FPGA Virtex-6. Il montre les caractéristiques temporelles de base du circuit pour  $N = 3$  bits (8 canaux) et  $M = 40$  bits pour un niveau de vitesse (-1) de la puce (performances minimales). La fréquence maximale supportée par le composant est de 270,63 MHz avec un temps de "setup" de 3,42 ns (voir le tableau 6.9).

TABLE 6.9 – Caractéristiques temporelles pour  $N=3$  bits &  $M=40$  bits, Vitesse (-1)

$F_{max}$ (MHz)	270,63
Temps minimal de l'entrée avant l'horloge (ns)	3,42
Temps maximal requis par la sortie après l'application de l'horloge (ns)	0,78

La figure 6.35 montre l'évolution de la fréquence maximale en MHz en fonction de la taille du bus de données (entrée/sortie) pour différentes valeurs de  $N$  (3, 8 ou 16 bits). Le nombre de bits  $M$  varie entre 3 et 64. On constate une décroissance rapide de la fréquence en fonction de  $M$ . Elle passe de 455 MHz pour ( $M=4$ ,  $N=3$ ) à 270 MHz ( $M=40$ ,  $N=3$ ) pour un niveau de vitesse (-1). La figure montre aussi que l'effet du nombre de canaux est indépendant de la fréquence maximale de l'architecture, elle stagne pour  $N=3$ , 8 ou 16 (Fig. 6.35). Toutefois, l'augmentation du nombre de canaux ( $N$ ) fait croître le cycle de calcul des offsets. Autrement dit l'architecture consomme plus de temps que dans le calcul des écarts des offsets ( $N$  coups

d'horloge pour le cycle complet).

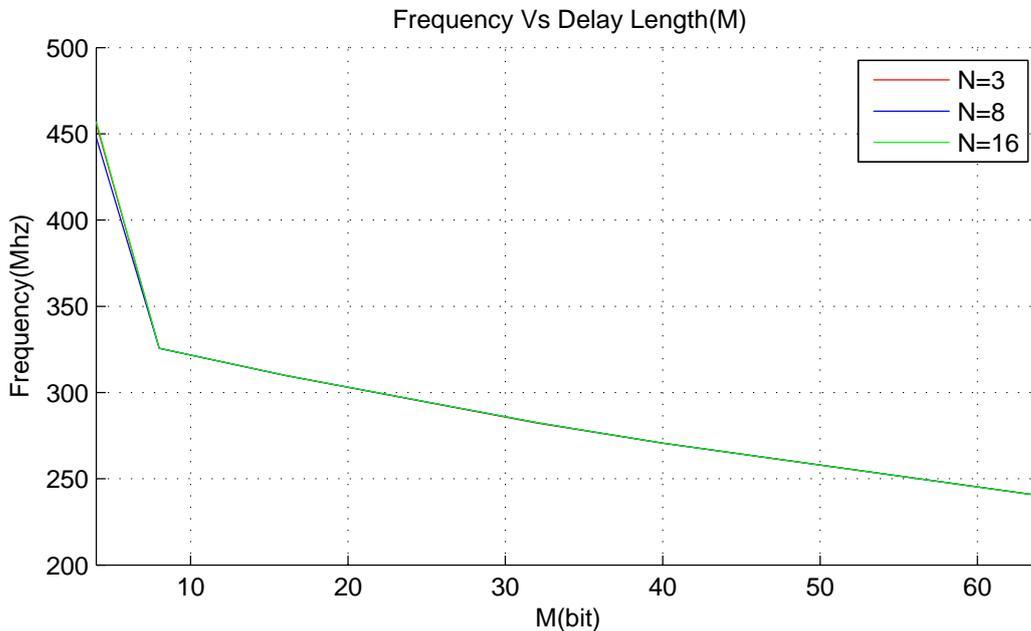


FIGURE 6.35 – La fréquence maximale de l'architecture (MHz) en fonction de la taille des bus d'E/S pour  $N=[3, 8 \text{ or } 16 \text{ bits}]$

On constate dans la figure 6.36 que le nombre des registres augmente linéairement avec l'augmentation de  $M$ , il passe de 20 pour  $(M=4, N=3)$  à 260 pour  $(M=64, N=3)$ . Réciproquement, le nombre de canaux a peu d'effet sur les ressources consommées en termes de registres. On constate clairement dans la figure 6.36, dans le cas où  $N=8$ , le nombre de registres croit par un facteur 4 pour chaque augmentation d'un bit du bus d'E/S.

Pour conclure, le composant permet de calculer d'une façon itérative les écarts d'ajustement pour chaque canal. La fréquence maximale ne dépend pas du nombre de canaux (taille du réseau). Par contre elle décroît avec l'augmentation de la taille du bus de données. L'architecture est très simple à mettre en œuvre avec un minimum de ressources. En revanche, le temps occupé par le calcul augmente linéairement avec la taille du réseau dû au multiplexage temporel de l'algorithme.

### 6.3.5 Calibrage des résolutions TDC/DTC

Le circuit de mesure des retards aller-retour entre le maître et les esclaves dispose d'une résolution  $R_{TDC}$  et  $R_{DTC}$  (Circuit à retard programmable). Dans le cas idéal où  $R_{DTC}$  est égal à  $R_{TDC}$ , on n'aura pas besoin d'un coefficient d'adaptation sur les écarts d'ajustement. Durant l'implémentation matérielle (voir le chapitre suivant), on a utilisé des circuits avec des résolutions différentes :  $R_{TDC} = 13,02 \text{ ps}$  et  $R_{DTC} = 10 \text{ ps}$  d'où l'intérêt d'utiliser un coefficient de correction  $K$  sur les écarts d'ajustement  $S_1$ . Le coefficient  $K$  est donné par l'équation (6.15).

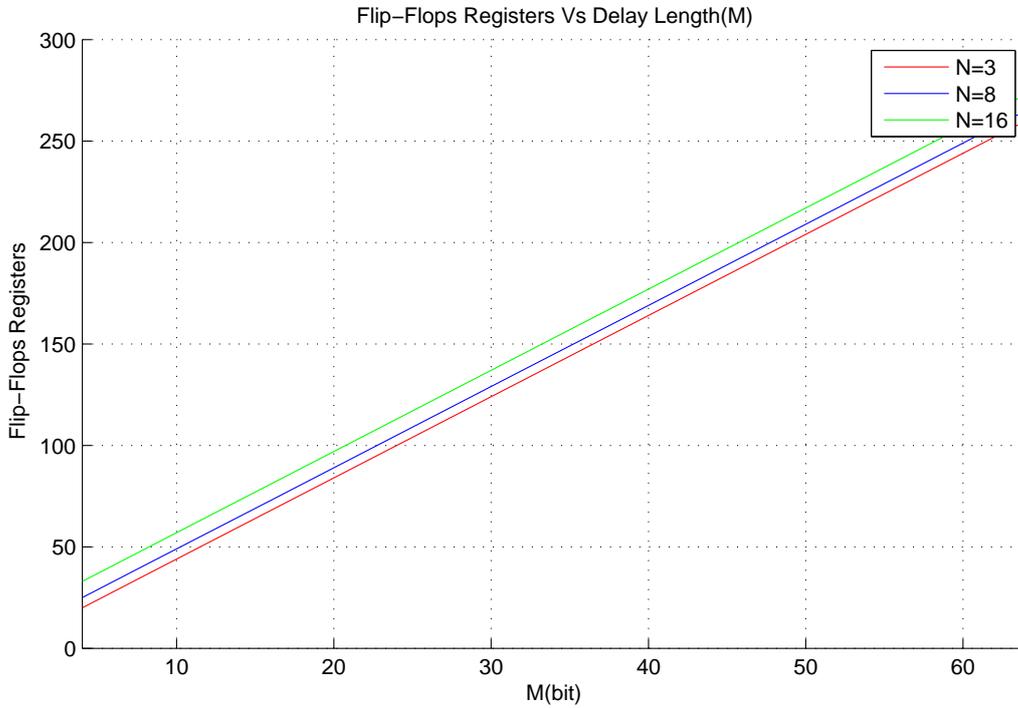


FIGURE 6.36 – Nombre de registres Flip-Flop en fonction de la taille des bus d'E/S, M pour  $N=[3, 8 \text{ ou } 16 \text{ bits}]$ .

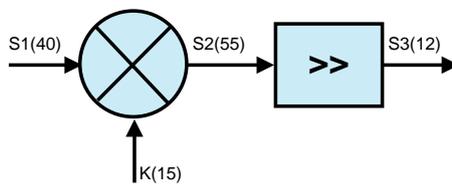


FIGURE 6.37 – Schéma bloc d'adaptation des résolutions TDC/DTC

$$K = \text{Round}\left(2^Q \times \frac{R_{TDC}}{R_{DTC}}\right) \quad (6.15)$$

$$S3 = (S_1 \times K) \gg Q = S_2 \gg Q \quad (6.16)$$

Q : La précision de conversion en virgule fixe en bit.  $K = 42664$  pour  $Q = 15$  bits et  $x'' A6A8''$  en format hexadécimal. L'équation (6.16) montre la valeur de l'écart après correction. Après multiplication par  $K$  de  $S_1$ , on récupère uniquement les bits de poids faibles du circuit matériel d'ajustement en décalant à droite par  $Q$  bits la sortie  $S_2$ . Le circuit d'ajustement matériel contient 12 bits (voir l'implémentation matérielle). La figure 6.38 illustre la sortie  $S_3$  entre le calcul en virgule flottante et en virgule fixe. On constate que l'erreur est inférieure ou égale à  $\frac{1}{2}LSB$  soit 5 ps en valeur absolue.

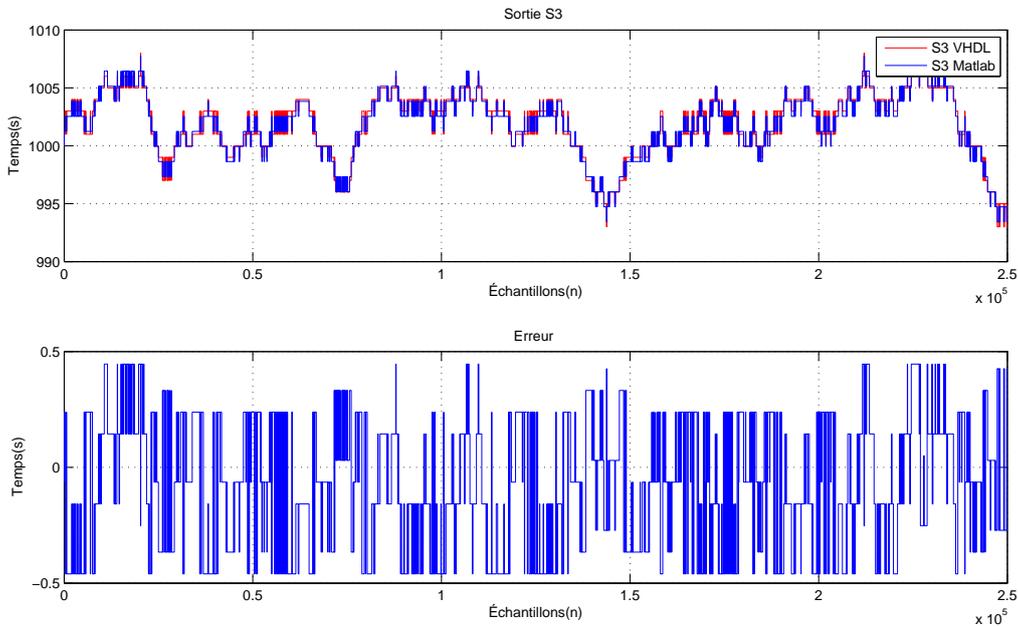


FIGURE 6.38 – Comparaison de la sortie  $S_3$  entre le calcul avec Matlab et le calcul en virgule fixe

### 6.3.6 Transfert des écarts d'ajustement aux horloges

#### 6.3.6.1 Introduction

Dans cette section on va étudier le circuit de transfert des écarts d'ajustement (voir la figure 6.10) au circuit de retard programmable (DTC). La figure 6.39 montre le schéma bloc de la partie matérielle sur carte et l'FPGA du circuit de transfert des écarts de calibration  $\frac{d_{xi}}{2}$ . La partie matérielle sur FPGA dédiée à la description de la technique utilisée pour le transfert des écarts calculés précédemment aux circuits DTC est aussi développée.

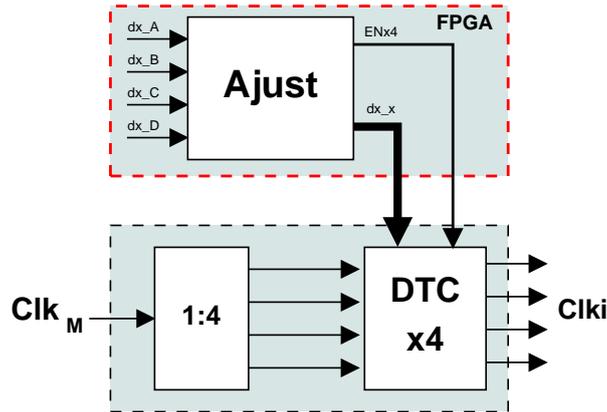


FIGURE 6.39 – Schéma bloc de la partie matérielle sur carte et FPGA du circuit de transfert des écarts de calibration  $\frac{d_{xi}}{2}$

#### 6.3.6.2 Fonctionnement

Le circuit d'ajustement récupère les données  $d_{xi}/2$  pour les quatre canaux sur 12 bits à la sortie du circuit de calibrage des résolutions. Il permet ensuite de transférer ces derniers aux canaux d'une façon séquentielle et en boucle. Chaque circuit DTC est muni d'un bus d'adresse sur 12 bits permettant d'ajuster un écart maximal de 20 ns avec une précision de 10 ps. Le bus de données est commun pour tous les canaux. L'activation du circuit est assurée par un signal EN pour chaque canal. Le début de l'ajustement est assuré par un signal de début "Start". Il se fait ensuite en boucle avec une période fixe, qui peut être ajustée par une variable externe sur 3 bits (8 périodes possibles). La figure 6.40 illustre les états du contrôleur d'ajustement des offsets pour quatre canaux.

Le contrôleur du circuit d'ajustement est constitué de cinq états :

- État 1 : État d'initialisation dont les circuits DTC sont chargés par une valeur nulle (décalage nul pour tous les canaux)
- État 2 à 4 : Transfert séquentiel de l'écart  $\frac{d_{xi}}{2}$
- État 5 : Transfert au canal D de l'écart et génération du signal indiquant la fin du cycle. Ce dernier dure un coup d'horloge et reste à zéro dans les autres états.

La figure 6.40 montre que l'ajustement (états 2 à 5) se réalise en boucle tant que le signal "Start" est à 1. Le passage d'un état à l'autre de la machine est synchronisé par une horloge du

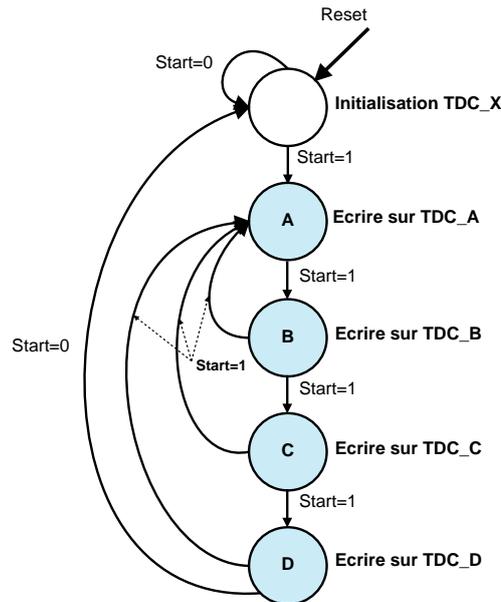


FIGURE 6.40 – Machine à état du contrôleur d'ajustement des offsets des horloges pour quatre canaux

circuit d'ajustement. La période de l'horloge dépend de la valeur du signal "PeriodSync" codé sur 3 bits. Voir le programme dans l'annexe XIII pour plus de détails.

### 6.3.6.3 Simulations

La figure 6.41 montre les chronogrammes des signaux d'ajustement des canaux. Les entrées des canaux A à D sont initialisées successivement à  $x"111"$ ,  $x"222"$ ,  $x"333"$  et  $x"444"$ . La période de transfert est le double à celle de l'horloge de référence Clk. Le paramètre de la période est initialisé à  $b"000"$  (sortie 0 du compteur cadencé par l'horloge Clk) (voir le programme). Lorsque le signal "Start" est égal à 0, la  $x"000"$  est transmise à tous les canaux ( $EN=b"1111"$ ). Lorsque le signal Start passe à 1, un canal à la fois est activé et la donnée à l'entrée correspondante est transmise au bus de sortie, l'indicateur de fin de cycle passe à 1 durant la transmission de la donnée du canal D. Pour conclure, les chronogrammes décrivent parfaitement le fonctionnement du contrôleur illustré dans la figure 6.40.

## 6.4 Description de l'architecture matérielle

### 6.4.1 Introduction

Cette section vise d'abord à aborder l'architecture globale mixte du système de synchronisation. Elle débouche sur la description des différents blocs constituant l'architecture matérielle Maître/Esclaves. La figure 6.42 indique l'architecture globale du système de synchronisation conçue à la station de radioastronomie. Elle est composée de deux parties essentielles : Le maître et les esclaves. Le maître compose le noyau du système, il permet la mesure des retards aller-retour avec le circuit TDC, la calibration de la chaîne de mesure (compensation des retards) et le calcul des retards d'ajustement avec FPGA. Enfin, la carte du maître permet la

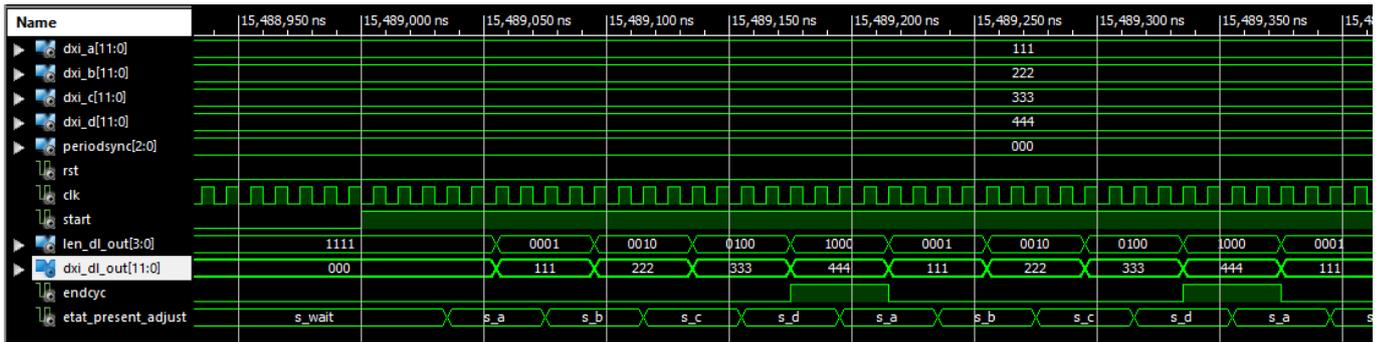


FIGURE 6.41 – Chronogrammes des signaux du circuit de transfert des écarts d’ajustement aux circuits DTC

compensation des horloges avec une grande précision et la distribution des horloges synchrones. L’esclave est constitué particulièrement d’une PLL à très faible gigue, d’une fréquence maximale d’un GHz dédiée au convertisseur A/D et un diviseur de fréquence pour le signal de retour. Ce chapitre décrit en détails la composition de l’architecture matérielle maître/esclaves.

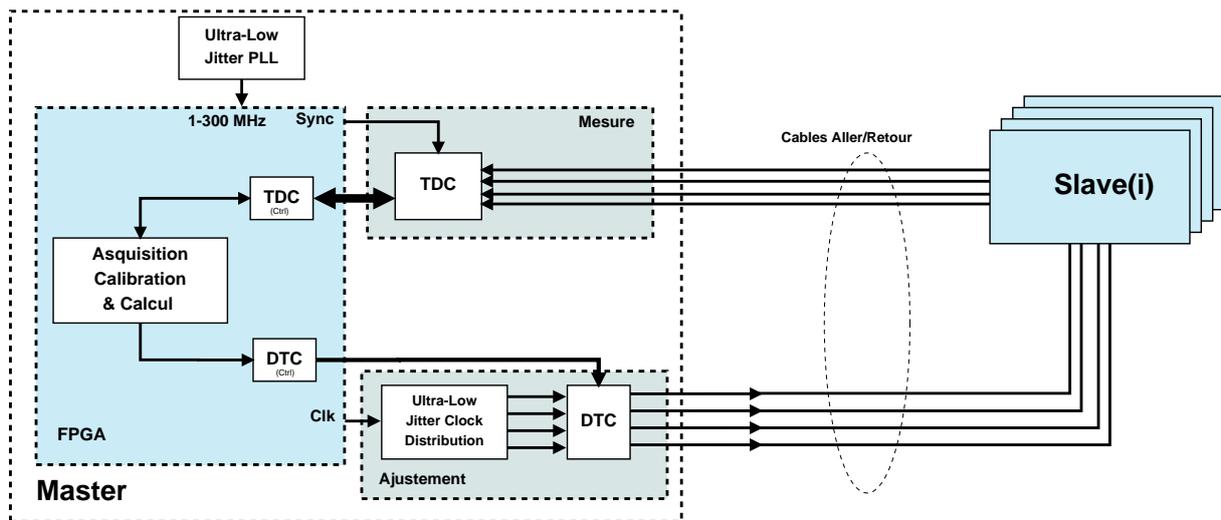


FIGURE 6.42 – Architecture globale du système de synchronisation

### 6.4.2 Circuit de mesure des intervalles de temps

La figure 6.42 montre la carte TDC. Elle est conçue pour la mesure des intervalles de temps déterminés et la calibration de la chaîne de mesure (Voir la section 6.5 dédiée aux résultats). La distance des pistes entre les connecteurs SMA et le composant pour les signaux d'Events (2-3) (voir la figure 6.42) sont identiques afin de réduire les erreurs du retard induit par le rééquilibrage des canaux. Les canaux différentiels LVDS sont munis d'une résistance externe de 100 ohms au plus proche du composant. La carte dispose de 4 sources d'alimentation de précision à haut ampérage (voir la figure 6.43). La carte est conçue de 4 couches et est protégée contre les effets mutuels des liens LVDS. Ci-dessous la signification des différentes interfaces de la carte TDC.

1. Circuit TDC
2. Signal Event A/B
3. Signal Event C/D
4. Résultats (Rdata)
5. Résultats (RstrobeAB)
6. Résultats (RstrobeCD)
7. Sortie d'horloge des résultats Rclk
8. Signal de synchronisation Sync 8
9. Horloge Maître du circuit TDC de 200 MHz fixe
10. Interface SPI de programmation (Hdata, Hstrobe, Hclk et Reset)
11. Alimentation linéaire 3.3V, 1A
12. Alimentations non linéaires : (3.3V, 3A, variable), (2.5, 3A, fixe) et (3.3v, 3A, fixe)



FIGURE 6.43 – Carte du circuit TDC de mesures des retards aller/retour pour quatre canaux

### 6.4.3 Architecture Maître/Esclaves

#### 6.4.3.1 Maître

##### Description

La carte illustrée dans la figure 6.44 décrit la composition du maître sans prendre en compte les cartes FPGA. Le tableau 6.10 montre la signification des différents composants constituant le maître. La PLL permet de synthétiser une fréquence de référence propre (1) pour le circuit de mesure (22) et pour le circuit de distribution des horloges synchronisées (15). La PLL alimente également la carte FPGA pour la synthèse des signaux "Sync" et "Events" à faible gigue. Elle intègre également 8 circuits DTC (5-8) de retards programmables avec une grande précision et une bande passante de 1,2 GHz, 2 DTC par canal mis en cascade pour doubler la dynamique d'ajustement. Ces derniers, permettant l'ajustement des offsets de quatre canaux indépendants (9-11). La dynamique des ajustement dépend de la résolution binaire du circuit du retard DTC. Le diviseur 1 vers 4 est lié avec les circuits DTC (Figure 6.42-6.44), la distance des pistes est maintenue identique entre chaque sortie du diviseur et l'entrée du TDC. Le diviseur 1 vers 4 (15) est indépendant, il sera câblé avec le circuit TDC durant la phase de test pour synthétiser les signaux "Events" synchrones à quelques picosecondes de "skew". Le tableau 6.10 montre la signification des différents composants de la carte 6.44

TABLE 6.10 – Composition de la carte illustrée dans la figure 6.42

Numéro	Description
1	PLL 1 GHz, 8 sorties à faible gigue
2	Entrée single ou différentielle de l'horloge de référence
3	Sortie différentielle 1 (4 vers 1) de la PLL
4	Sortie différentielle 2 (4 vers 1) de la PLL
5-8	Deux circuits en cascade du retard programmable (DTC) par canal pour l'ajustement des offsets des horloges
9	Sorties différentielles décalées des canaux A-B
10	Sorties différentielles décalées des canaux C
11	Sorties différentielles décalées des canaux D
12	Circuit de distribution d'horloge à faible gigue 1 vers 4 à 1,2 GHz
13	Entrée 0 du circuit DTC
14	Entrée 1 du circuit DTC
15	Circuit de distribution d'horloge à faible gigue 1 vers 4 à 1,2 GHz
16-17	Entrées de référence 0 et 1
18-21	4 Sorties différentielles du circuit de distribution d'horloge
22	Circuit TDC

##### Ajustement des offsets

Le circuit d'ajustement des offsets est basé sur le composant des retards programmables TDC [161] dédié à l'ajustement des horloges. Il contient une entrée et une sortie différentielles de type NECL/PECL compatible avec la logique LVDS25. Le principe de base consiste en la

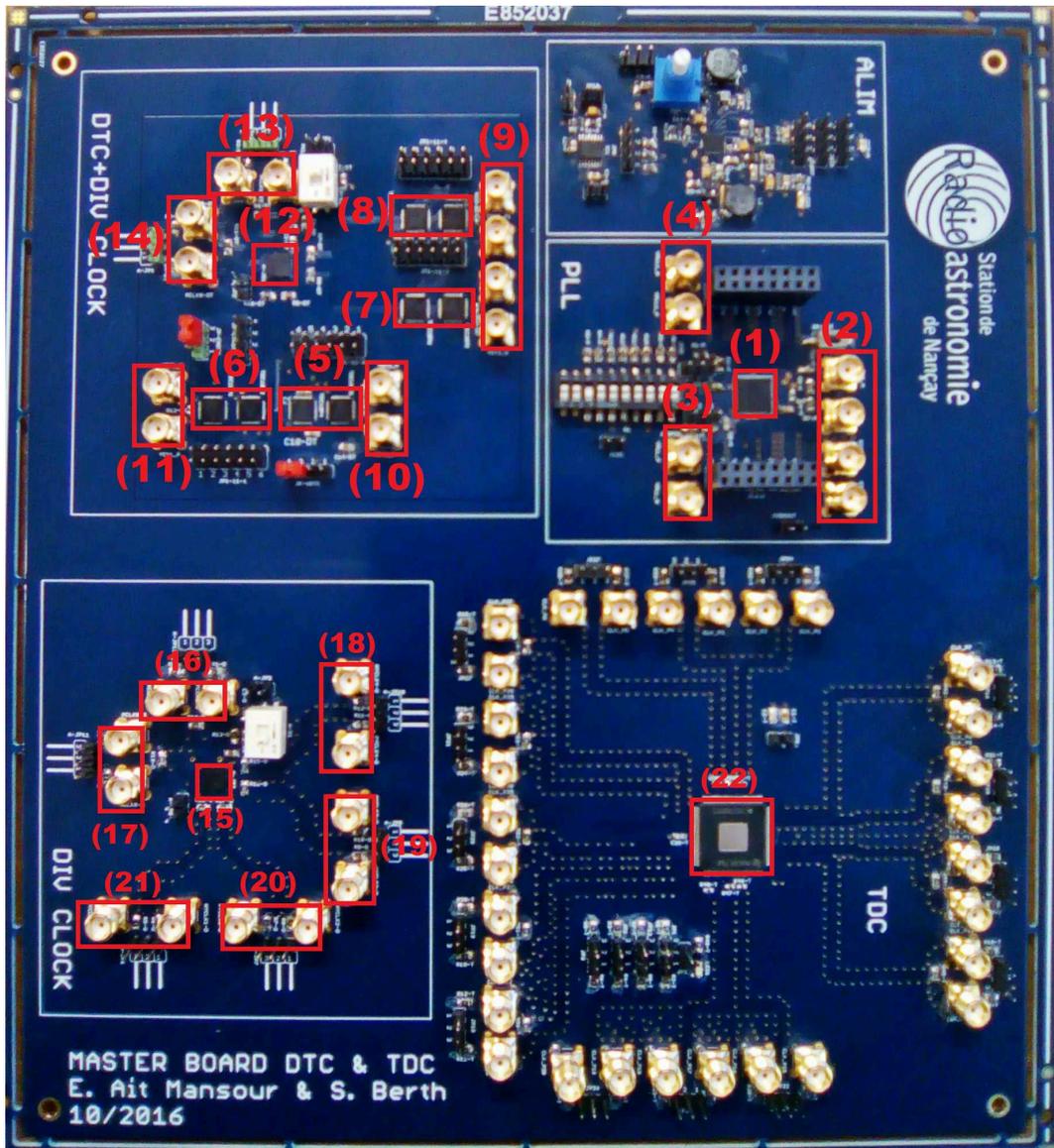


FIGURE 6.44 – Carte du Maître

programmation d'une matrice de portes et de multiplexeurs pour synthétiser le retard. Le composant est muni d'un codeur de retard sur 10 bits sous forme d'un bus de données avec une résolution de 10 ps (MSB). En effet, le retard peut être ajusté entre 0 et 10 ns. Le circuit intègre une entrée de validation "LEN" (Load Enable) active au niveau haut, le passage à 1 du signal "LEN" permet de charger la donnée. Le composant a un retard interne entre l'entrée et la sortie de 2,2 ns, le retard global et la somme du retard programmable et le retard interne (de 2,2 ns à 12,2ns) (voir le tableau 6.11 pour plus de détails). La figure 6.45 montre la mise en cascade de 2 cellules de retard, le retard global est le double de celui d'une cellule ( $2 \times 12.20ns = 24ns$ ) (voir (5)-(8) dans la figure 6.44).

Après la calibration des retards dans le chemin des horloges (le retard pour chaque canal est compensé et ramené au retard induit par le circuit de mesure (TDC), avec une gigue de 8 ps et une résolution de 12 ps), la dynamique de compensation effective des offsets des horloges est de 20 ns avec un retard fixe de 4,4 ns. Dans ces conditions, l'écart de distance  $D_{max}$  (différence de distance entre les canaux) maximal que le prototype peut compenser est égal à 6 mètres. Voir l'équation (6.17), avec  $d_{xi}^{max}$  l'offset du retard maximal.

$$D_{max} = d_{xi}^{max} \times C = 20 \times 10^{-9} \text{mtres} \times (3 \times 10^8) = 6 \text{ m} \quad (6.17)$$

TABLE 6.11 – Caractéristiques principales du circuit DTC

Paramètre	Valeur
Retard	2,2 ns
Résolution binaire	10 bits
Précision	10 ps
Dynamique	2,2-12,2 ns
Fréquence d'entrée	1,2 GHz
Logique de l'E-S	PECL - NECL
Logique de la donnée	ECL-LVCMOS-LVTTL
Gigue à 1,2 GHz	1,16 ps

Le retard entre l'entrée et la sortie du DTC varie d'une façon linéaire par rapport à la valeur numérique programmée (commande). Le retard augmente légèrement avec la température et également avec la commande de la configuration qui joue sur la valeur de précision du retard. Le coefficient vaut  $16,67 \text{ ps}/C$  (pour une commande de 900 sur 12 bits), il est de la moitié pour une commande égale à 400 (Figure 4. dans [161]). Pour une meilleure précision, il est recommandé de maintenir le composant à une température constante avant et après l'ajustement des horloges.

### Distribution d'horloge

Le circuit [162] est un buffer 1 vers 4 avec une bande passante de 2 GHz. Il supporte deux entrées d'horloge multiplexées en mode différentiel. Il est caractérisé par une très faible gigue de l'ordre de 40 fs et un "skew" de sortie maximal de 5 ps. Toutefois, le temps de propagation du composant est typiquement égal à 320 ps. Le composant supporte la logique LVDS, LVPECL

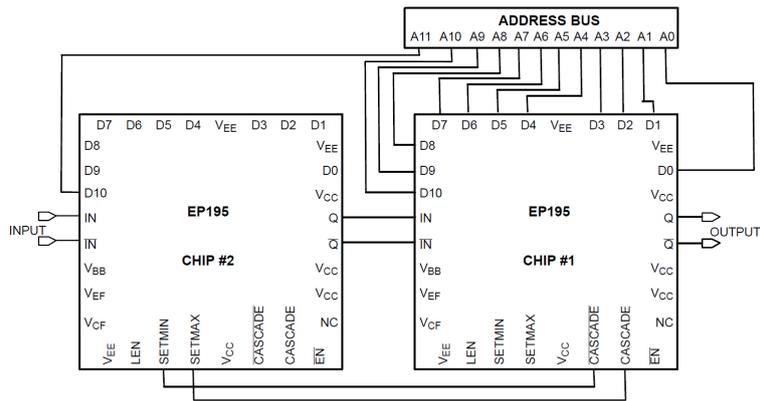


FIGURE 6.45 – Mise en cascade de 2 cellules DTC

ou CML. Il intègre une entrée LVCMOS de sélection de l'entrée d'horloge (0 ou 1) (Fig. 6.46). La figure 6.47 montre la carte de la mise en œuvre du "buffer". Elle sera utilisée durant la phase de test de l'algorithme.

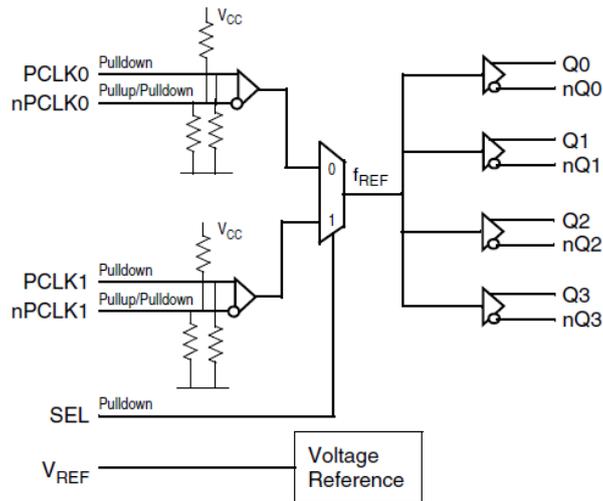


FIGURE 6.46 – Schéma interne du buffer 1 : vers 4

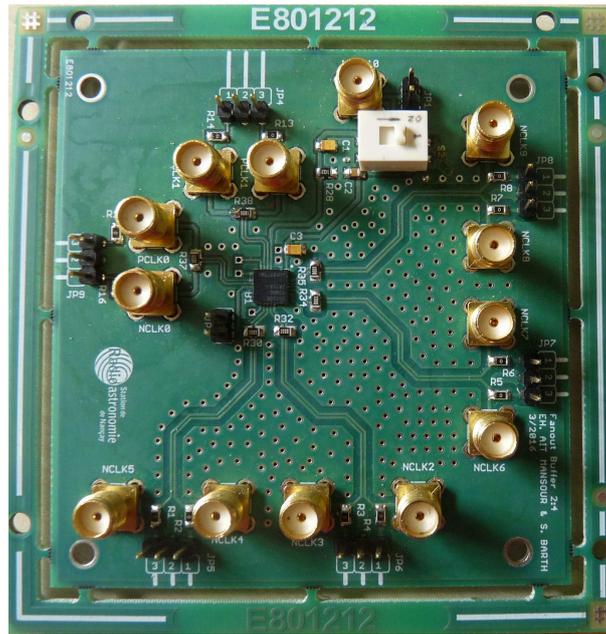


FIGURE 6.47 – Carte distributrice de l’horloge 1 vers 4

## FPGAs

L’implémentation de l’algorithme de calcul des offsets, des interfaces d’acquisition des données rapides et de contrôle des différentes cartes imposent le choix d’une plateforme FPGA à haute performance (vitesse et ressources). Durant la phase de test, on va procéder à l’utilisation de deux cibles FPGA : Spartran-3A (Fig. 6.49) et Vertex-6 (Fig. 6.48). La première sera consacrée à la programmation du circuit TDC via l’interface série (*SerialHost*). La deuxième sera utilisée pour l’acquisition des données différentielles rapides et le calcul des écarts d’ajustement des offsets. Les kits de développement indiqués dans [163] et [164] seront utilisés durant la phase de test.



FIGURE 6.48 – Kit de développement FPGA Vertex-6



FIGURE 6.49 – Kit de développement FPGA Spartran 3A [165]

### 6.4.3.2 Esclave

#### Introduction

L'unité esclave constitue un nœud dans le réseau d'antennes multi-reparties. Il permet de recevoir et retransmettre le signal dédié à la mesure du retard aller-retour entre le maître et les esclaves. Il est équipé également d'une PLL pour la synthèse de fréquence d'horloge haute fréquence pour le convertisseur A/D. Ce convertisseur est situé au même endroit que l'esclave (voir la figure 6.50). L'esclave reçoit un signal à l'entrée avec une fréquence  $F_{in}$ , cette dernière servira comme référence à la PLL au pied de l'antenne. La PLL génère un signal avec une fréquence  $F_{out} = N_{PLL} \times F_{in}$  qui servira comme entrée du diviseur de fréquence afin de retrouver la fréquence d'entrée. La sortie du diviseur de fréquence vaut  $F'_{in} = F_{out} / N_{Div}$ . Pour une mesure correcte du retard entre le maître et l'esclave ( $d_{MM}^i$ ), la fréquence  $F_{in}$  doit être égale à la fréquence  $F'_{in}$  pour une fréquence du convertisseur désirée  $F_{out}$ . Les rapports du circuit de division sont connus pour une fréquence  $F_{in}$  définie par l'équation (6.18)).

$$\begin{cases} F_{out} = N_{PLL} \times F_{in} \\ F'_{in} = \frac{F_{out}}{N_{Div}} \end{cases} \quad (6.18)$$

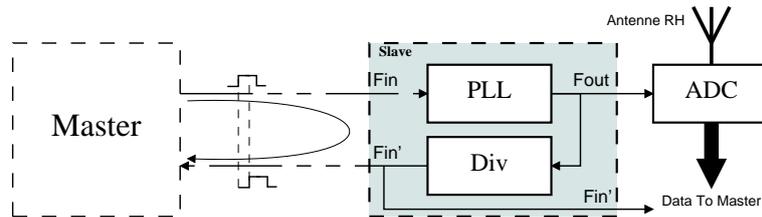


FIGURE 6.50 – Schéma bloc de l'esclave dans l'architecture globale du système de synchronisation

En respectant la condition  $F_{in} = F'_{in}$  et en remplaçant  $F_{in}$  par  $F'_{in}$  dans l'équation (6.18), on obtient (6.19). Exemple pour obtenir une fréquence  $F_{out}$  du convertisseur A/D de 1000 MHz, il faut une horloge à l'entrée  $F_{in}$  de 67.5 MHz avec un rapport de division égal à 16.

$$\begin{aligned} F_{out} &= N_{PLL} \times F'_{in} = N_{PLL} \times \frac{F_{out}}{N_{Div}} \\ \Rightarrow N_{PLL} &= N_{Div} \end{aligned} \quad (6.19)$$

#### Description

La figure 6.51 montre le circuit matériel de l'esclave. Il est constitué essentiellement d'un diviseur de fréquence programmable (1) (voir la figure) et d'une PLL (6). Les deux blocs (PLL et diviseur) ont des alimentations différentes et non connectées l'une à l'autre. L'intérêt de ce choix est de permettre l'utilisation du diviseur seul avec un rapport égal à 1 (signal aller =



FIGURE 6.51 – Carte esclave (Vue de dessus)

TABLE 6.12 – Composition de la carte esclave

Numéro	Description
1	Diviseur de fréquence (1-2-4-8-16), trois sorties à 2 GHz de bande
2-5	Sorties du diviseur (1)
6	PLL, 1 GHz, 8 sorties faible gigue
7	Alimentation 3,3 v du diviseur
8	Alimentation 3,3 v de la PLL

signal retour). Dans ce cas, la PLL est déconnectée, l'entrée de la fréquence  $F_{in}$  est liée directement avec l'entrée du diviseur. La sortie  $F'_{in}$  correspond à la sortie du diviseur.

Un diviseur 1 vers 3 programmable de hautes performances est utilisé pour la récupération de la fréquence à l'entrée de la carte esclave. Il dispose d'un "skew" à la sortie de 15 ps et d'une gigue typique de 250 fs. Il a également un retard de propagation maximal de 600 ps [166]. Le circuit dispose de trois sorties indépendantes qui peuvent être activées en même temps en fonction de la commande. Cette particularité nous sera utile pour l'observation du signal aller au bout de l'esclave pour ensuite le comparer avec le signal du retour au bout du maître. La PLL avec une très faible gigue [167] est utilisée pour la synthèse de la fréquence du convertisseur A/D. Elle permet d'augmenter la fréquence de l'horloge synchrone issue du maître à une fréquence maximale de 1 GHz pour le convertisseur A/D d'une part. Elle sert d'autre part à nettoyer la gigue de l'horloge reçue, car le convertisseur A/D tolère une gigue maximale de 670 fs pour une résolution binaire de 9 bits et une bande passante analogique de 450 MHz.

La gigue à la sortie de la PLL est de l'ordre de 100 fs. Cette valeur est largement satisfaisante au vu de la précision de l'horloge du convertisseur A/D. De plus, la PLL dispose de 8 sorties différentielles programmables via l'interface I2C. La PLL supporte une fréquence à l'entrée allant de 1 à 300 MHz au format single ou différentielle ou bien une entrée à quartz avec une fréquence variant entre 10 et 52 MHz. D'après le tableau 6.12, on déduit que les rapports de décision doivent être supérieurs ou égales à 4 ( $F_{in} < 300$  MHz) dans le cas où la fréquence du

convertisseur est de 1 GHz. La figure 6.52 montre le schéma bloc de la PLL. La programmation de la PLL peut être effectuée en deux modes : Mode manuel (ROM) ou logiciel (EEPROM). La carte 6.51 intègre des interrupteurs pour la sélection du mode et la fréquence à la sortie de la PLL via une combinaison sur 5 bits. Le mode manuel est limité par le type (Sortie single) et les valeurs possibles de la fréquence de l'horloge à l'entrée.

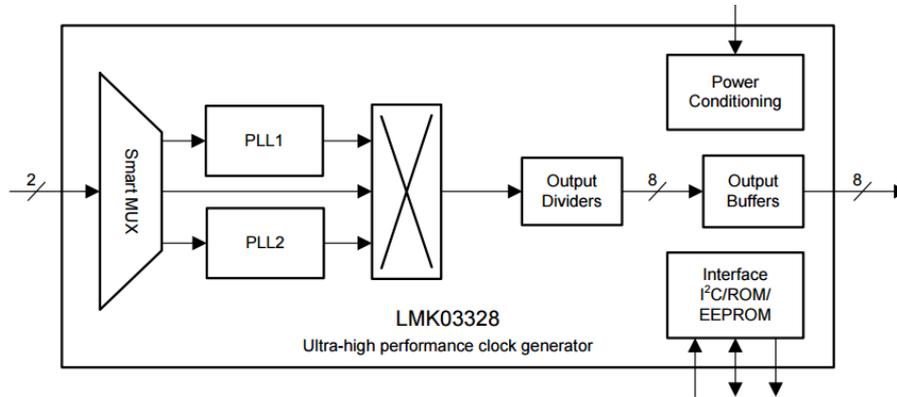


FIGURE 6.52 – Schéma bloc de la PLL

## 6.5 Résultats de test et mesures

### 6.5.1 Calibration de la chaîne Maître/Esclaves

#### 6.5.1.1 Introduction

La section décrit la mesure et la calibration du retard aller-retour ( $d_{MM}^i$ ) entre le maître et les esclaves. Les figures 6.53 et 6.54 montrent l'architecture du système de mesure et l'architecture matérielle du système global. Cette première phase permet de calibrer l'effet des retards accumulés dans le chemin aller-retour des horloges  $d_0^i$ . L'intérêt de la calibration est le réglage des zéros des canaux. Durant cette phase, les liaisons sont de même longueur entre le maître et les esclaves. La longueur du câble du signal "Start" (Sync) entre le FPGA et le TDC est égale à la longueur du câble aller+retour du signal Stop afin de réduire l'erreur due aux câbles des deux signaux. La PLL au niveau de l'esclave est désactivée, on a utilisé un rapport de division de fréquence égal à 1.

Le maître envoie des intervalles de temps  $T_i$  (l'intervalle de temps entre le front montant des signaux "Start" et le front montant du "Stop") contrôlés par la carte FPGA (générateur des  $T_i$ ) pour les quatre canaux, puis il récupère les résultats des mesures via l'interface d'acquisition. On a effectué des fenêtres de mesure sur 10 secondes pour les divers modes de mesures (différentes valeurs des  $T_i$ ). Les mesures sont ensuite traitées en temps différé sur ordinateur pour analyse et comparaison. En effet, on a utilisé la moitié des mesures pour l'estimation des retards dus à la chaîne, et le reste pour la calibration. La section illustre les mesures des retards en fonction des modes pour divers canaux avant et après la phase de calibration.

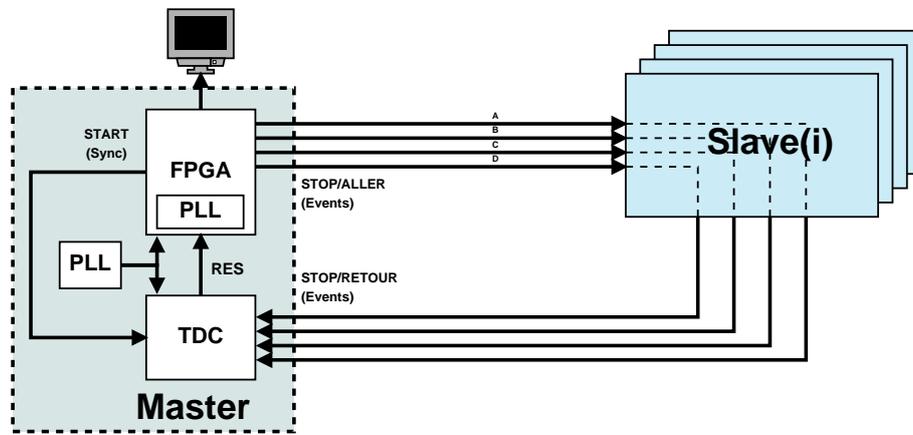


FIGURE 6.53 – Architecture du circuit de mesure et de calibration de la chaîne.

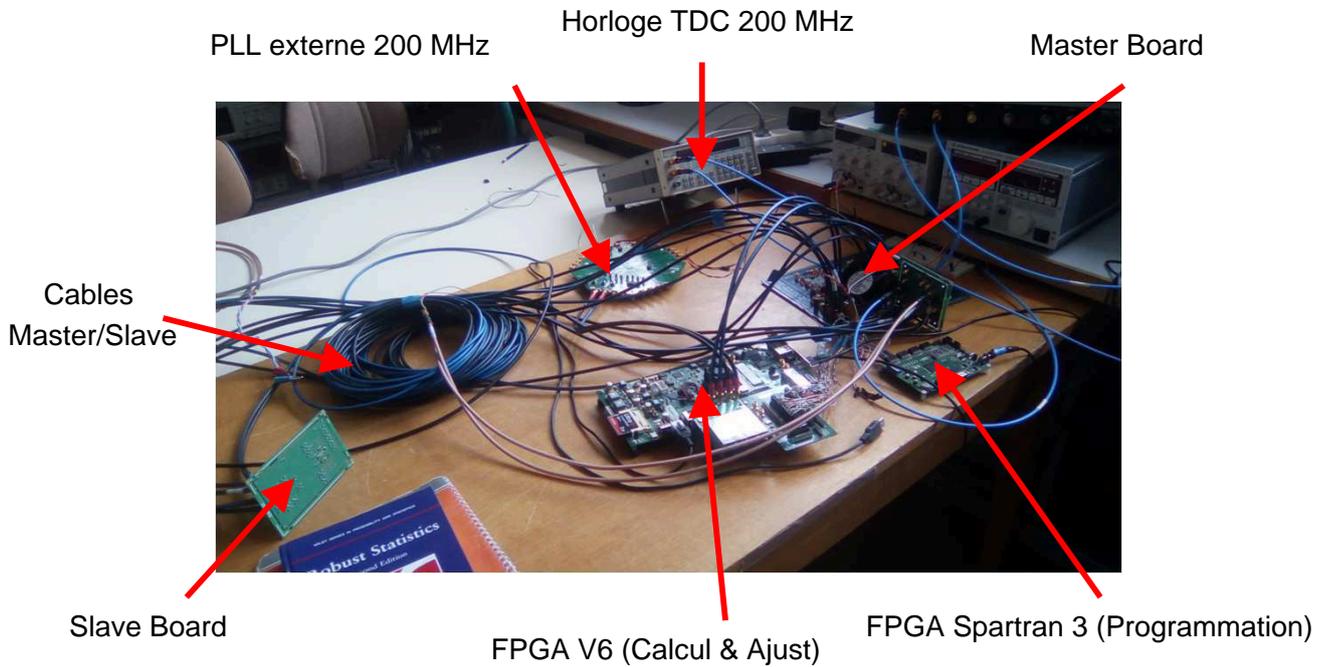


FIGURE 6.54 – Architecture matérielle du système.

Durant la phase de calibration (phase 0), l'effet du retard du câble  $d_1^i$  est supprimé en utilisant des câbles avec des longueurs communes à un millimètre près, conçues spécialement à cet effet. Elles sont dédiées au trajet aller-retour des signaux "Start" et "Stop". Par conséquent, la mesure du TDC  $d_{MM}^i$  est réduite à  $T_i + d_0^i$ , donc on peut facilement déduire la valeur  $d_0^i$  à partir de la mesure de  $d_{MM}^i$  et par la connaissance préalable du  $T_i$ . De plus, on a choisi les mêmes longueurs pour tous les canaux pour qu'on puisse faire des comparaisons entre les canaux. Dans le cas réel, on peut calibrer des canaux d'une façon indépendante en utilisant des câbles de différentes longueurs pour chaque canal.

### 6.5.1.2 Mesures en utilisant la PLL interne du FPGA

Durant le processus d'acquisition des mesures, on a utilisé la PLL interne du FPGA (IP Core) comme référence d'horloge du générateur des intervalles du temps. La PLL est alimentée avec une fréquence de 100 MHz issue de la PLL externe (PLL à faible gigue intégrée dans la carte Maître [167]). La PLL interne du FPGA est utilisée pour passer de 100 MHz à 200 MHz. Le circuit TDC utilise une sortie de 200 MHz de la même PLL externe. Les intervalles de temps  $T_i$  sont définis et synthétisés par le générateur. Le tableau 6.13 montre les valeurs de ces derniers en fonction du mode pour chaque canal en microseconde. Les intervalles varient entre  $1,28\mu s$  (mode 0, canal A) et  $655,36\mu s$  (mode 7, canal D). L'acquisition des données est effectuée sur une fenêtre de 10 s pour divers modes. Le circuit TDC est configuré en mode 40 bits avec une fréquence d'acquisition des résultats de 75 MHz. L'horloge du générateur des  $T_i$  est branchée avec l'horloge de la carte [168] à une fréquence de 200 MHz (période de 8 ns). L'horloge du circuit TDC (200 MHz) est branchée avec la sortie 200 MHz de la PLL externe.

La figure 6.55 montre la sortie différentielle de la PLL externe de 200 MHz. La courbe C1 est la sortie (+) et C2 la sortie (-). F1 est la sortie différentielle ( $C3 = C1 - C2$ ). On constate également deux distributions sous forme Gaussienne, la première (rose), indique le "skew" entre C3 et C1, la deuxième, le skew entre C2 et C3. Le "skew" de l'horloge de 200 MHz des deux distributions est quasiment identique, il est inférieur à 50 ps à 3 sigma et environ 17 ps à 1 sigma.

TABLE 6.13 – Intervalles de temps  $T_i(\mu s)$  pour les quatre canaux A à D en fonction du mode du générateur de signaux Start et Stop

Mode/Canal	A	B	C	D
0	1.28	2.56	3.84	5.12
1	2.56	5.12	7.68	<b>10.24</b>
2	5.12	<b>10.24</b>	15.36	20.48
3	<b>10.24</b>	20.48	30.72	40.96
4	20.48	40.96	61.44	81.92
5	40.96	81.92	122.88	163.84
6	81.92	163.84	245.76	327.68
7	163.84	327.68	491.52	655.36

Les figures 6.56 et 6.57 indiquent l'intervalle de temps  $T_i$  entre les signaux Start (SYNC) et Stop (Events) du canal A Mode (0) (voir le tableau 6.13) et le skew entre les deux signaux au

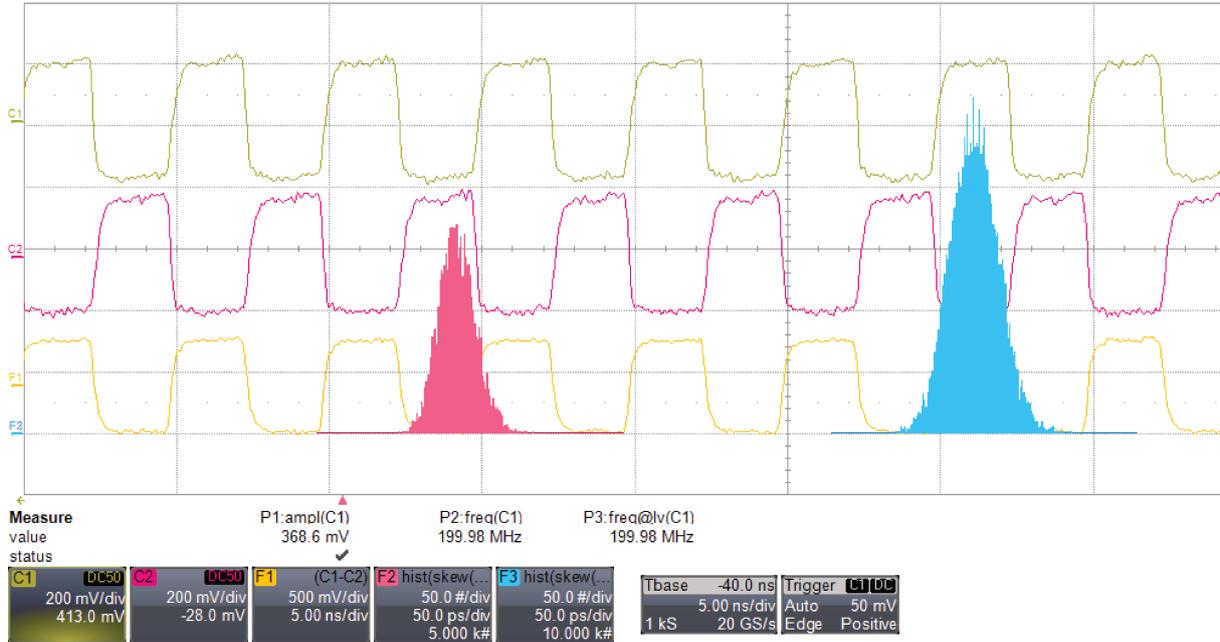


FIGURE 6.55 – Gigue de la sortie différentielle de la PLL intégrée dans la carte maître avec une fréquence de 200 MHz. La distribution en rose indique le skew entre C1 et F1 (50 ps/div, 200 mV/div). La distribution en bleu indique le skew entre C2 et F1 (50 ps/div, 500 mV/div).

niveau du Maître (Sorties du générateur). Le skew suit une distribution gaussienne avec une moyenne nulle et une variance (1 sigma) inférieure à 36 ps (voir la figure 6.56). On constate également que le skew au niveau de l'esclave du signal aller (Start) et du signal (Stop) est quasiment le même (voir la figure 6.57). Ce résultat indique que le skew induit par le trajet aller est négligeable. Les figures 6.58, 6.59, 6.60 et 6.61 montrent les résultats des mesures avant et après calibration des divers canaux (A à D) du mode 2 de mesure.

La figure 6.58 illustre les distributions des mesures du retard aller/retour  $d_{MM}^i(s)$  des canaux A à D (Mode 2) avant calibration. La reproductibilité des mesures aboutit aux lois normales autour de la valeur théorique des intervalles de temps  $T_i$  illustrés dans le tableau 6.13. Autrement dit, l'erreur induite par la chaîne est négligeable par rapport à l'intervalle de temps ( $T_i \gg d_0^i$ ). La figure 6.59 montre le retard induit par la chaîne de mesure ( $d_0^i$ ) pour divers canaux. Ce retard est constitué d'une partie fixe (de l'ordre de la nanoseconde) et d'une partie dynamique (centaines de picosecondes) autour de la valeur moyenne. La valeur moyenne est l'accumulation des temps de propagation des divers composants dans le chemin du signal aller/retour (Stop), la partie variable est due au skew entre le signal aller/retour et le signal de synchronisation (Start). La figure 6.60 montre la mesure des divers canaux après calibration (compensation de l'intervalle de temps  $T_i$  et  $d_0^i$ ) du même mode.

La figure 6.61 montre l'erreur quadratique moyenne (EQM) pour chaque canal pour les modes 2, 3 et 4 avant et après calibration. Il est évident que le circuit de calibration permet de compenser la différence des canaux quel que soit le mode avec une précision finale inférieure à 3 ps. Le skew entre les canaux après calibration est inférieur à 0,5ps. Dans la même figure,

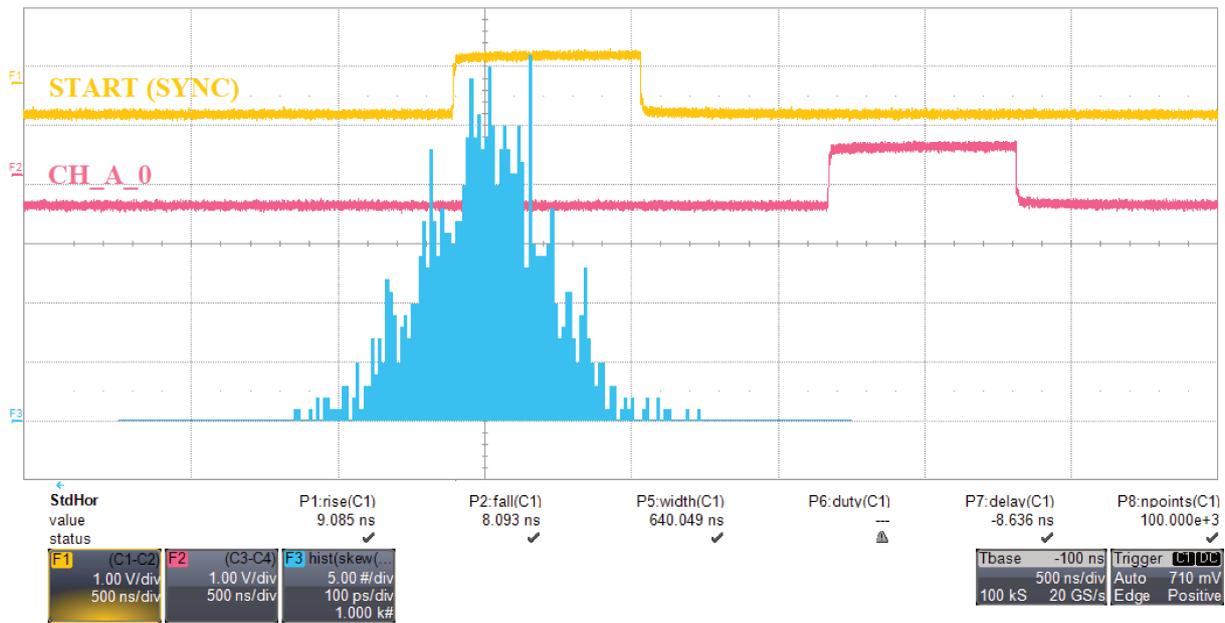


FIGURE 6.56 – Intervalle de temps  $T_i$  entre Start(SYNC) et Stop (Events) du canal A Mode (0) et le "skew" entre les deux signaux au niveau du Maître (Sorties du générateur) (100 ps/div, 1V/div)

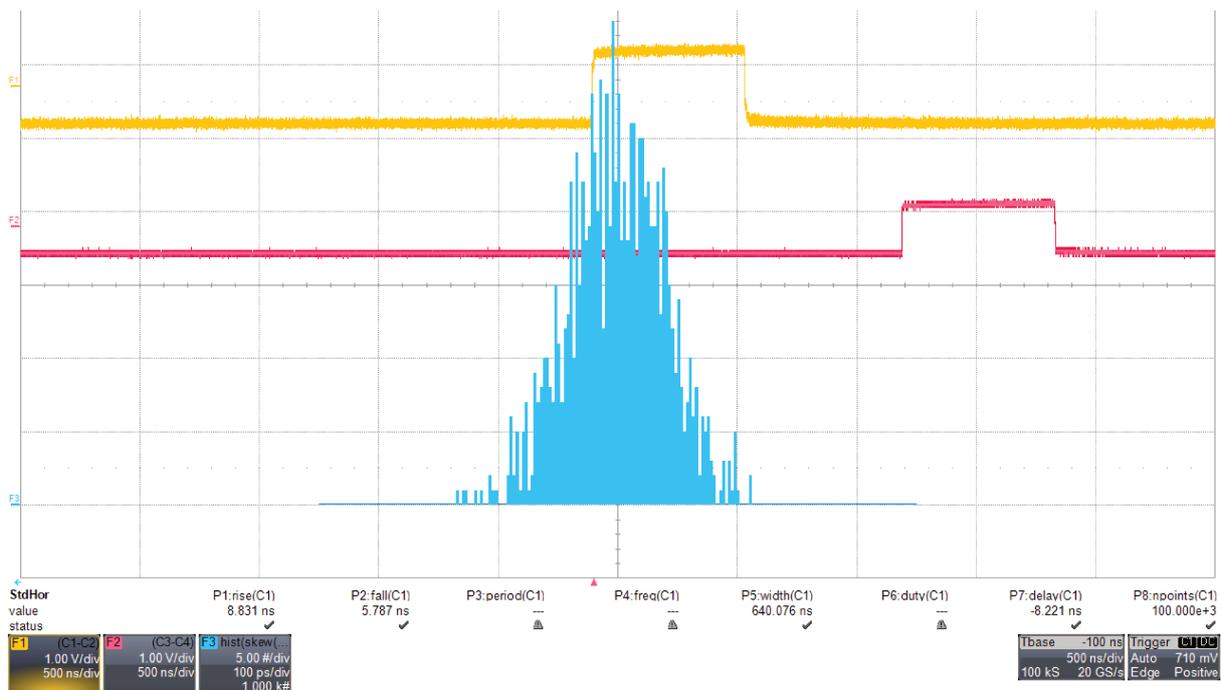


FIGURE 6.57 – Intervalle de temps  $T_i$  entre Start (SYNC) et Stop (Events) du canal A Mode (0) et le "skew" entre les deux signaux au niveau de l'esclave (100 ps/div, 1V/div)

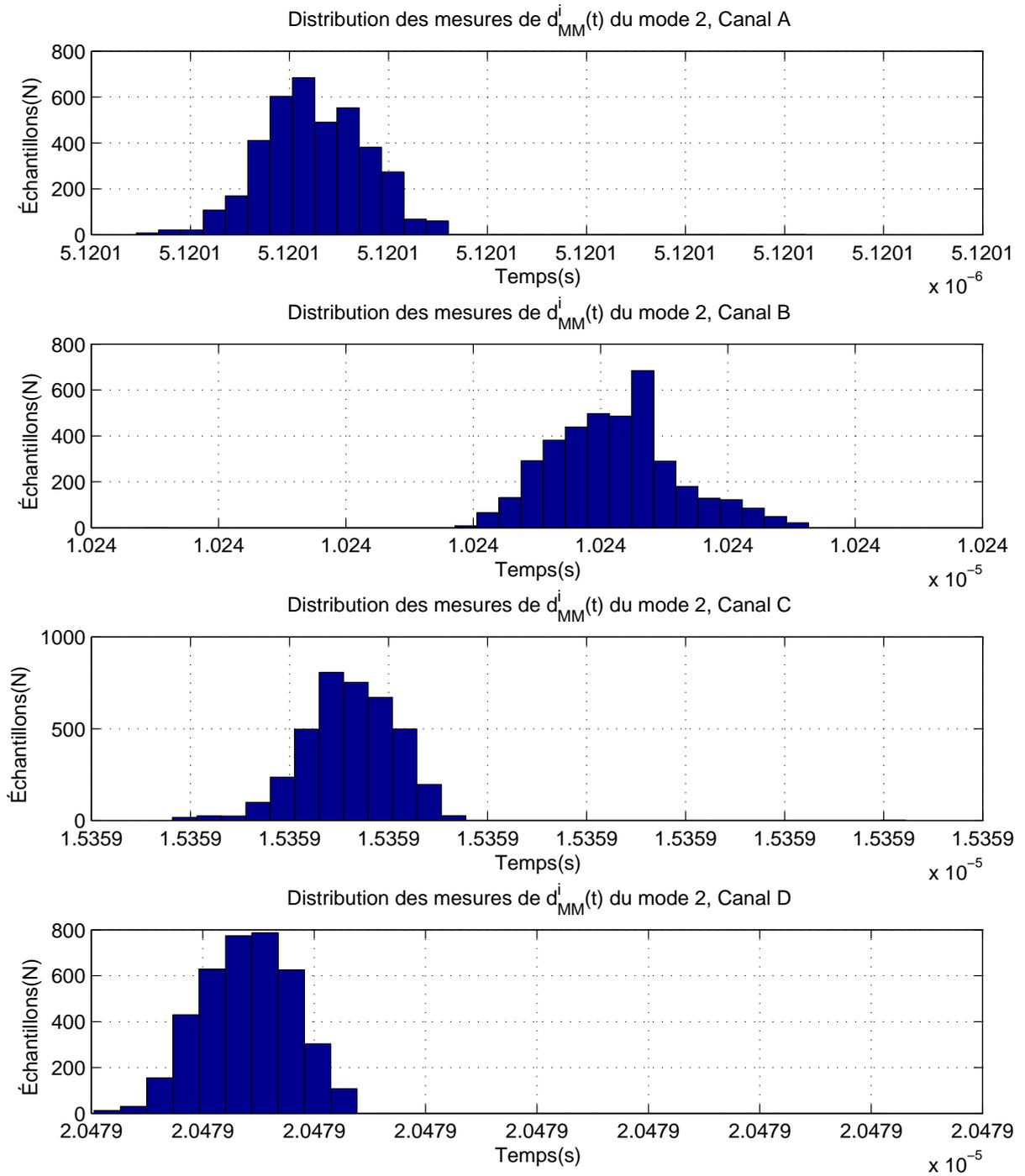


FIGURE 6.58 – Distributions des retards  $d_{MM}^i(s)$  pour les quatre canaux pour le mode 2 (avant calibration)

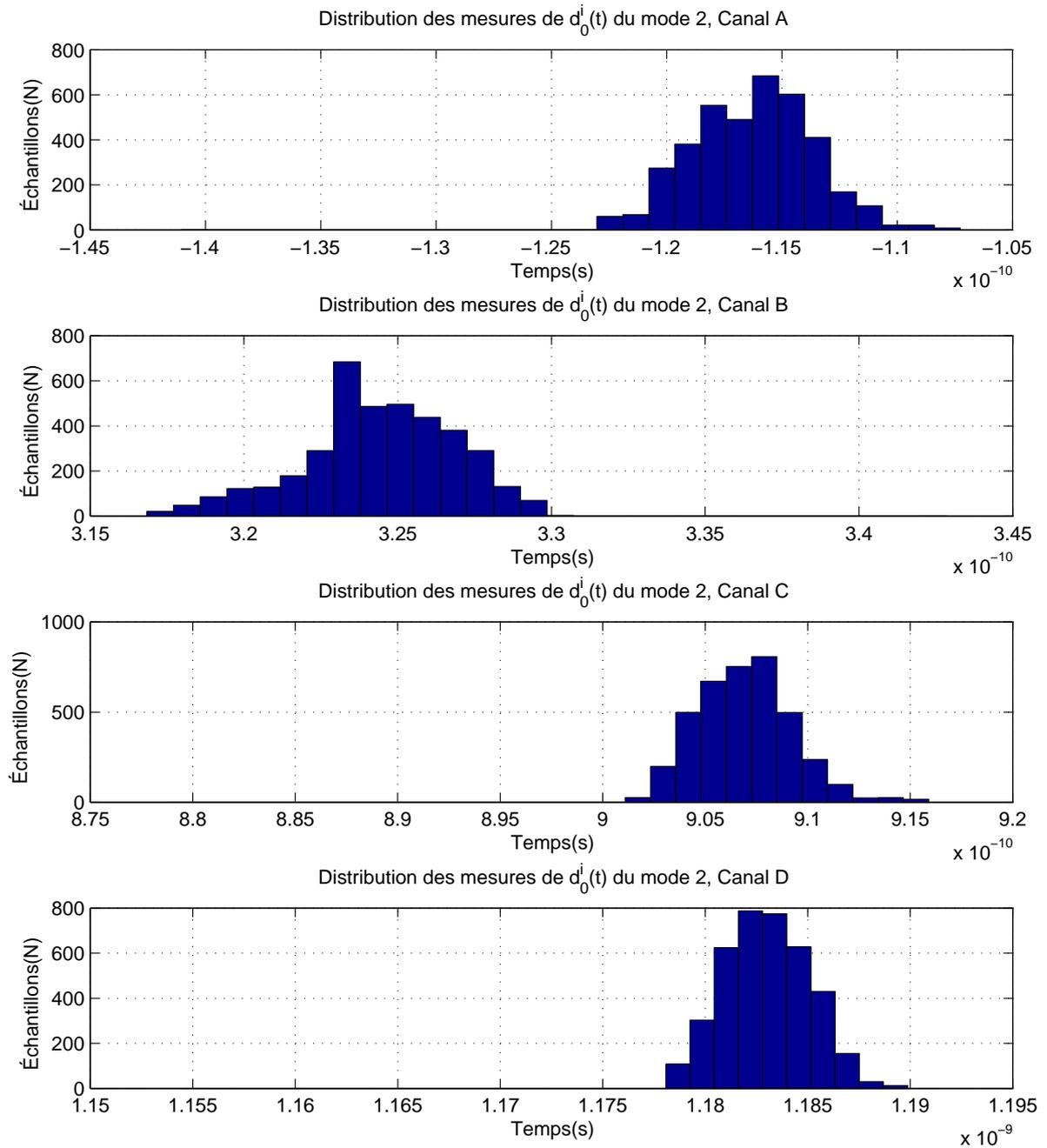


FIGURE 6.59 – Distributions des retards de la chaîne  $d_0^i(s)$  pour les quatre canaux pour le mode 2

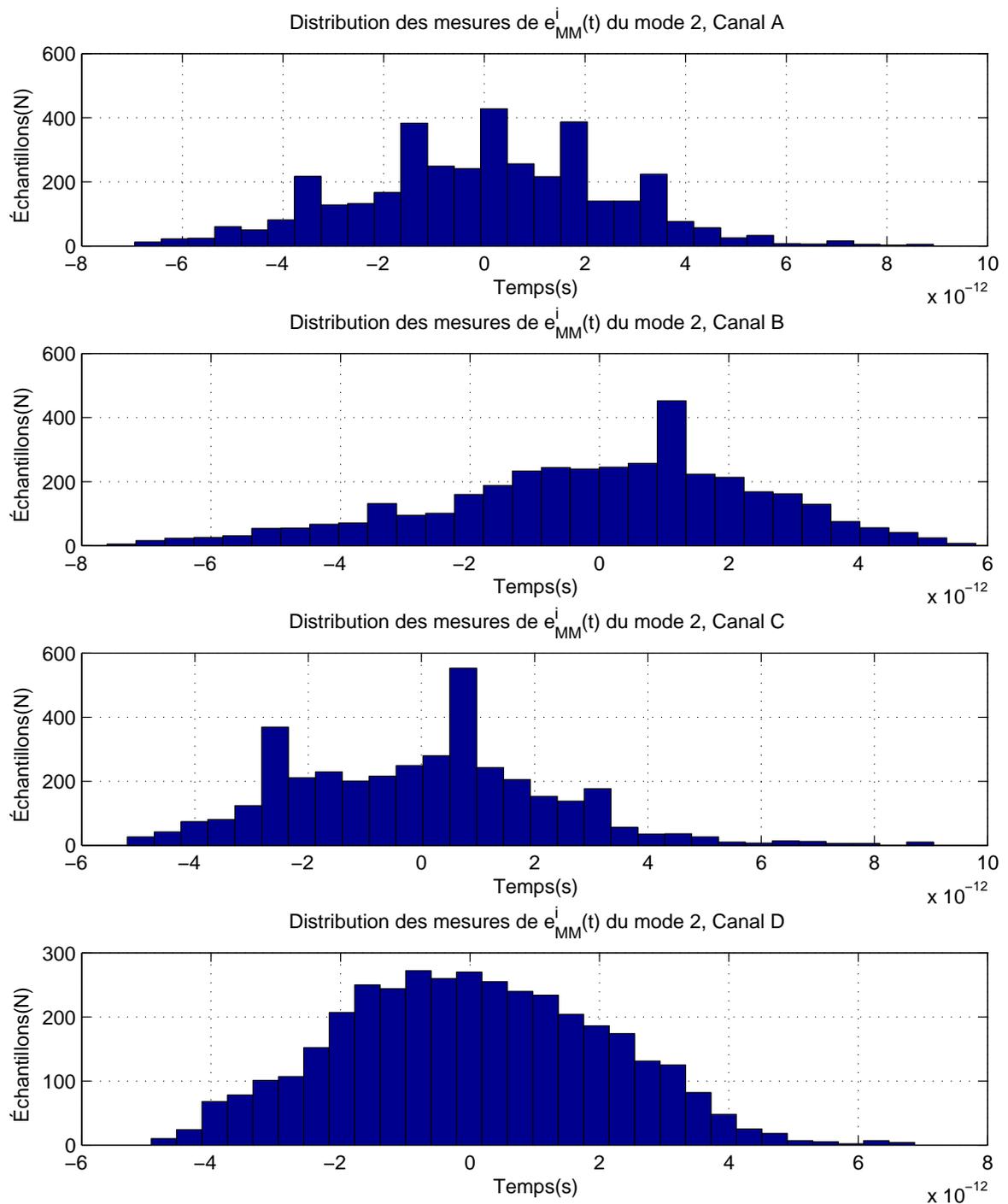


FIGURE 6.60 – Distribution de l'erreur  $e_{MM}^i(s)$  pour les quatre canaux pour le mode 2 après calibration

on constate l'apparition de deux phénomènes de croissance du retard. Le premier phénomène consiste en la multiplication par deux du maximum du retard en fonction du mode (Mode 2 : 1,2ns, Mode 3 : 2,5ns et 5 ns pour le mode 4). Cette augmentation du retard est due au temps de propagation induit par le générateur des signaux Start et Stop. En effet, le passage d'un mode (i) en mode (i+1) est effectué par la sélection de la sortie (i+1) à la place de la sortie (i) d'un compteur binaire. Autrement dit, on augmente le nombre de bascules en cascade et par conséquent, le temps de propagation entre la fréquence de référence du compteur et la sortie sélectionnée. Le deuxième phénomène consiste en la croissance du retard en fonction du canal. Exemple : le passage du Mode 2 au Mode 3 induit un retard d'environ 1 ns. Cette augmentation est due à l'augmentation de l'intervalle de temps entre les canaux A et D (voir le tableau 6.13).

Pour conclure, le circuit de calibration permet effectivement de compenser les retards induits par la chaîne de mesures, ainsi que les erreurs apportées par le rééquilibrage des canaux avec une précision inférieure à 3 ps (1 sigma).

### 6.5.1.3 Mesures en utilisant la PLL externe du FPGA

Dans ces tests, on va utiliser les mêmes paramètres d'acquisition des mesures que dans la section précédente. Cependant, on va utiliser une horloge externe de 200 MHz issue d'une PLL externe pour alimenter directement la PLL interne du FPGA pour la synthèses de la fréquence 200 MHz du générateur des  $T_i$ . Le TDC est alimenté par une horloge de 200 MHz de laboratoire à faible gigue. La figure 6.62 montre l'horloge F1=C1(+)-C2(-). F2 montre la le "skew" entre le signal C1 et F1, le F3 le "skew" entre C2 et F1. Les deux distributions montrent que la gigue de l'horloge de référence F1 du circuit de mesure TDC dispose d'une précision inférieure à 8 ps (3 sigma). Les figures 6.63, 6.64, 6.65 et 6.66 montrent les résultats des mesures avant et après la phase de calibration des retards.

La figure 6.66 montre l'estimation moyenne du retard induit par la chaîne en fonction du mode pour chaque canal. Le retard est compris entre 250 ps et 5 ns. Il est minimal pour le canal 1 (A) et maximal pour le canal 4 (D). Le retard croit en fonction du canal et du mode comme illustré précédemment. La précision maximale du système après calibration est inférieure à 2 ps et quasiment indépendante des modes avec un "skew" inter-canaux inférieur à 0,5 ps.

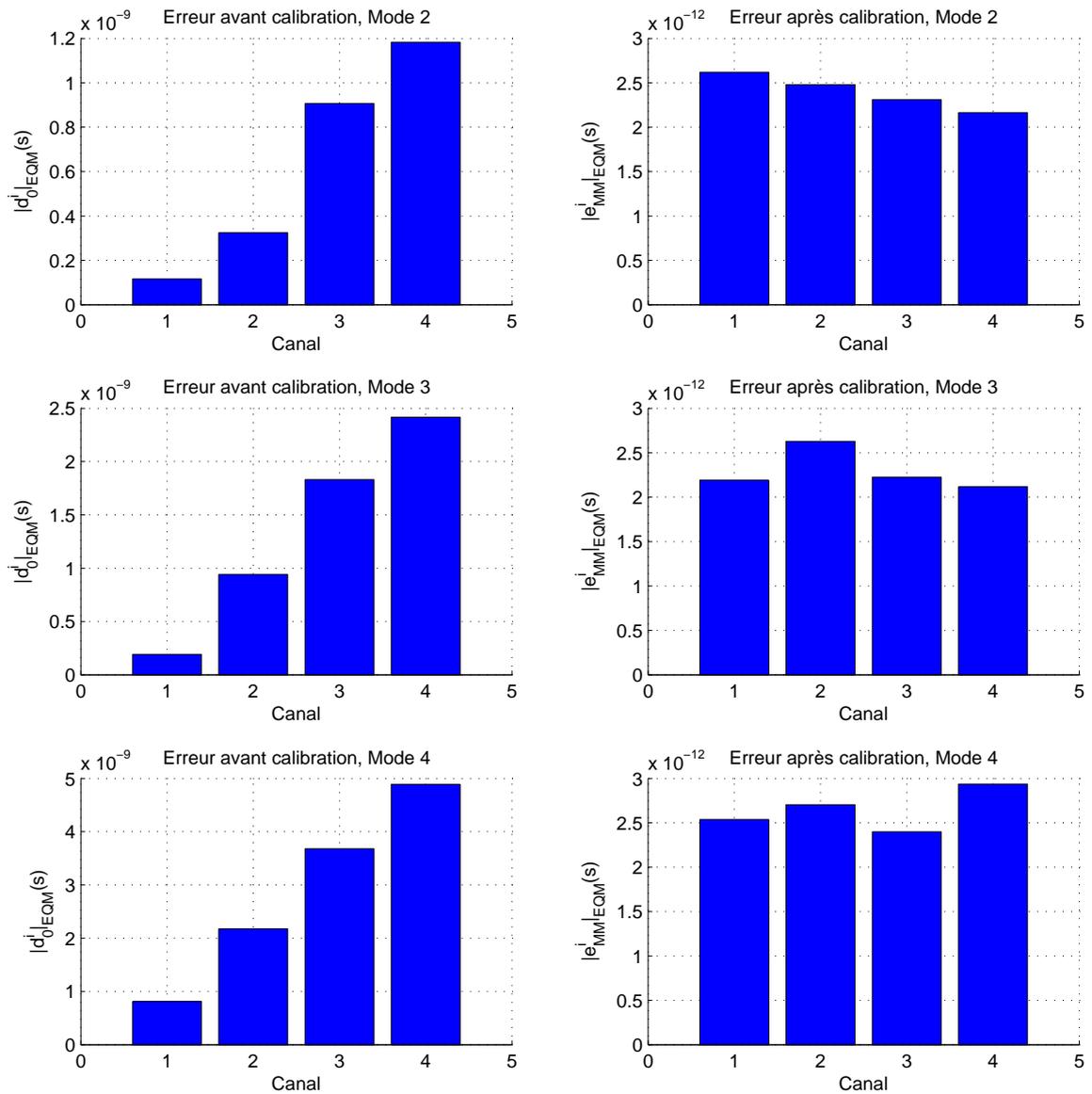


FIGURE 6.61 – Erreur quadratique moyenne(EQM) avant et après la phase de calibration pour les modes (2, 3 et 4) pour les quatre canaux

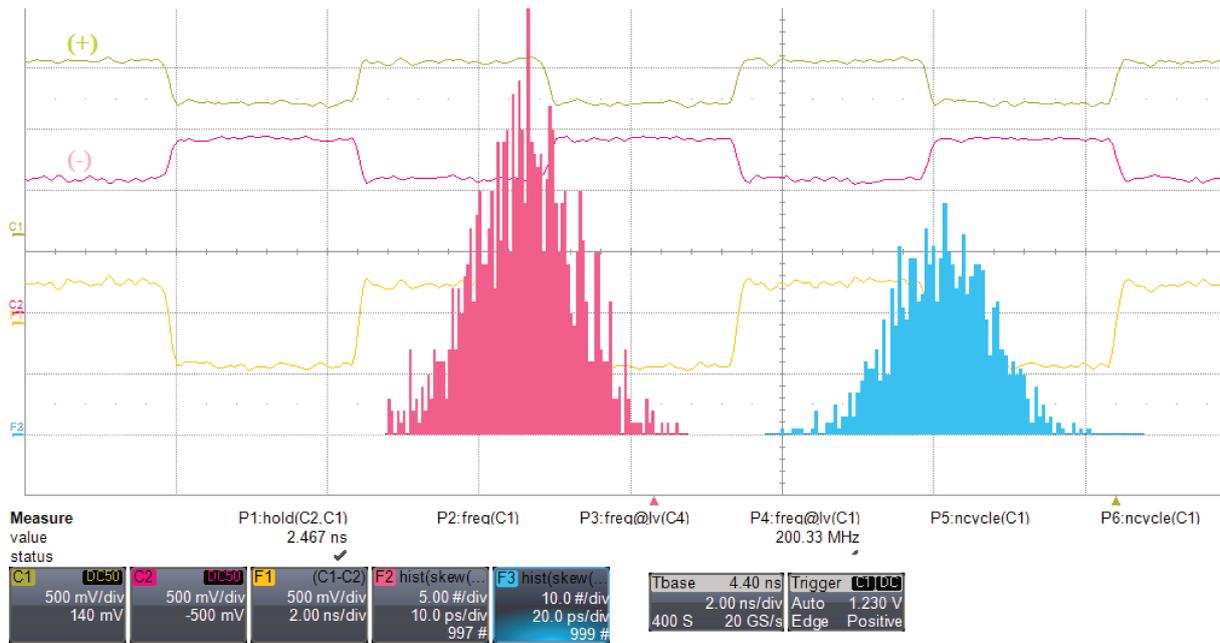


FIGURE 6.62 – Gigue de l’horloge de la fréquence de référence du circuit TDC à 200 MHz (20 ps/div, 500 mV/div)

### 6.5.1.4 Conclusions

On a illustré dans la section les résultats des mesures du retard aller/retour  $d_{MM}^i$  pour les canaux A à D en fonction de divers modes de mesures. Les résultats du système de calibration montrent la compensation des retards induits par la chaîne de mesure. On a constaté également la compensation des retards induits par le générateur des intervalles de temps pour chaque canal. La précision du système après calibration est inférieure à 3 ps dans le cas de l’utilisation d’une horloge avec une gigue de 35 ps (1 sigma) (voir la figure 6.55). Elle est de 2 ps dans le cas d’utilisation d’une horloge ayant une gigue inférieure à 5 ps (1 sigma) (voir la figure 6.62). On conclut que la précision après calibration augmente avec la qualité de l’horloge utilisée. De plus, le circuit de calibration permet d’obtenir des mesures précises malgré l’utilisation des signaux brouillants (Start et Stop avec une gigue supérieure à 200 ps (6 sigma)) et le non respect de la contrainte sur l’horloge du circuit TDC (gigue inférieure à 3 ps). On constate que la précision du système calibration est meilleure dans le cas de l’utilisation de mesures réelles que pour les données théoriques. Elle est de 26,97ps pour le modèle VHDL (voir la figure 6.26). Cette augmentation de précision est due à la compensation du skew entre les signaux Start et Stop, qui sont deux signaux synchrones synthétisés par le même générateur. Cela n’est pas vrai dans le cas théorique (signaux aléatoires et indépendants). Le tableau 6.14 résume les performances et limitations de l’approche de synchronisation proposée.

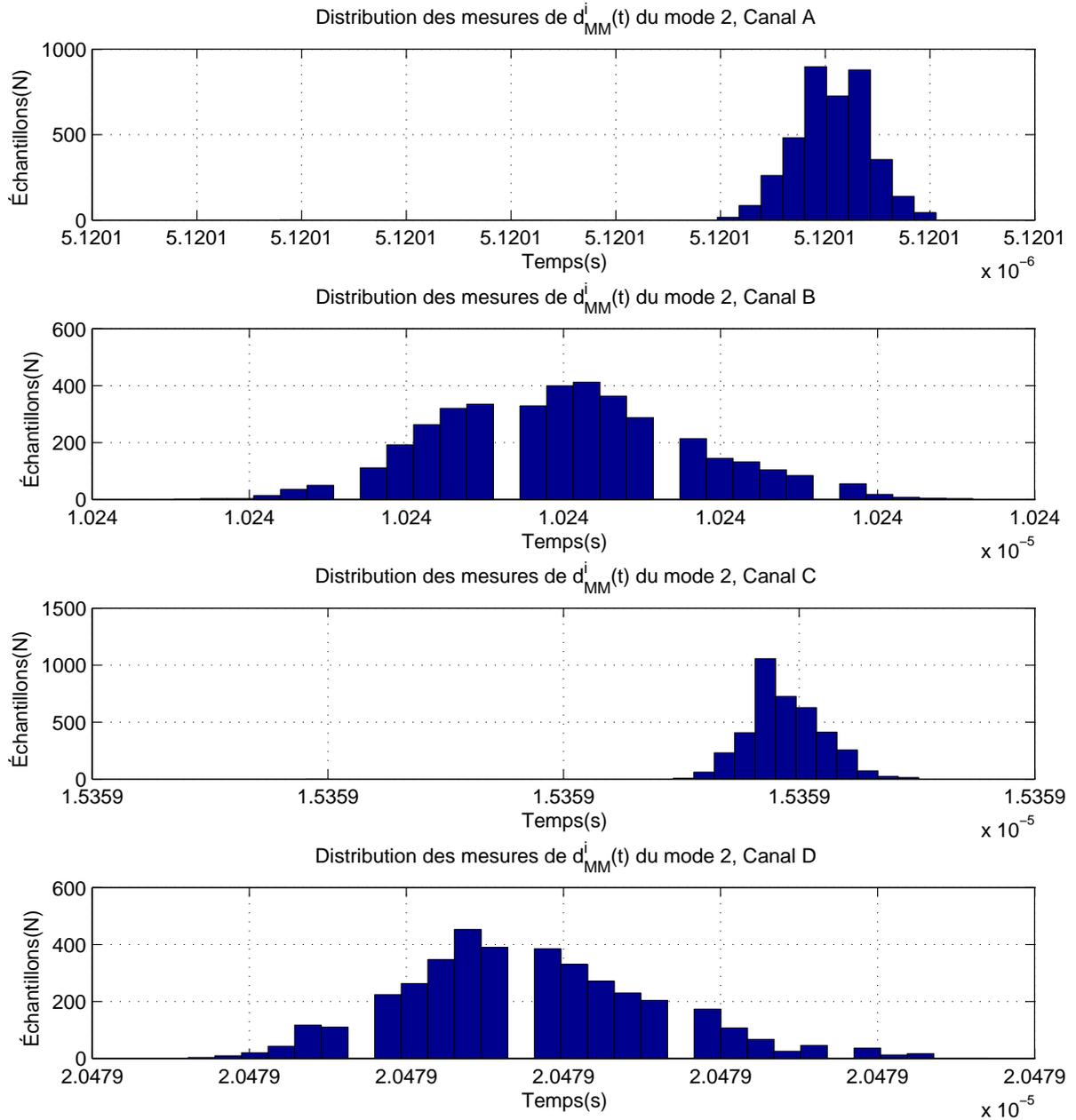


FIGURE 6.63 – Distribution des retards  $d_{MM}^i(s)$  pour les quatre canaux pour le mode 2 (avant calibration)

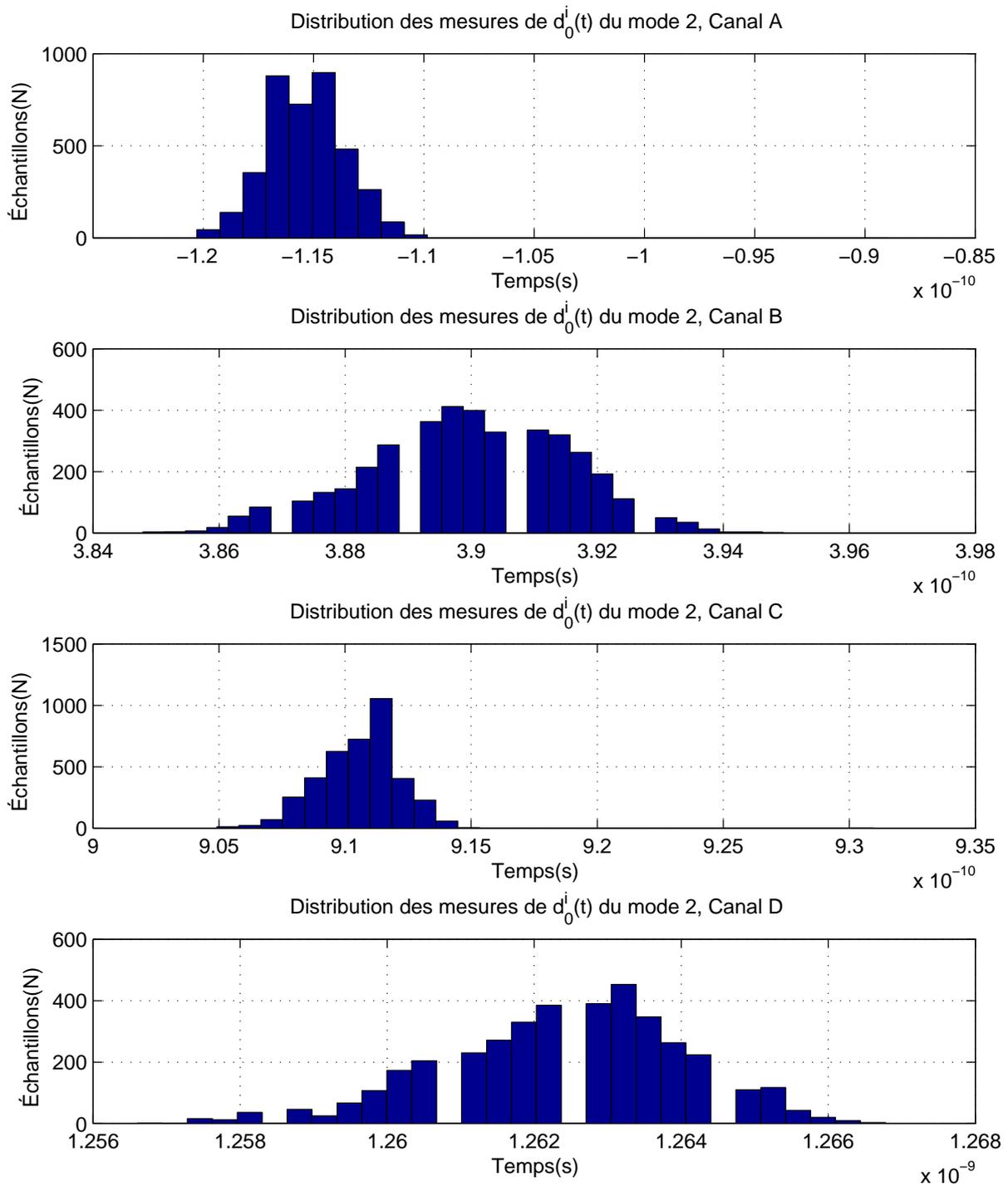


FIGURE 6.64 – Distribution des retards de la chaîne  $d_0^i(s)$  pour les quatre canaux pour le mode 2

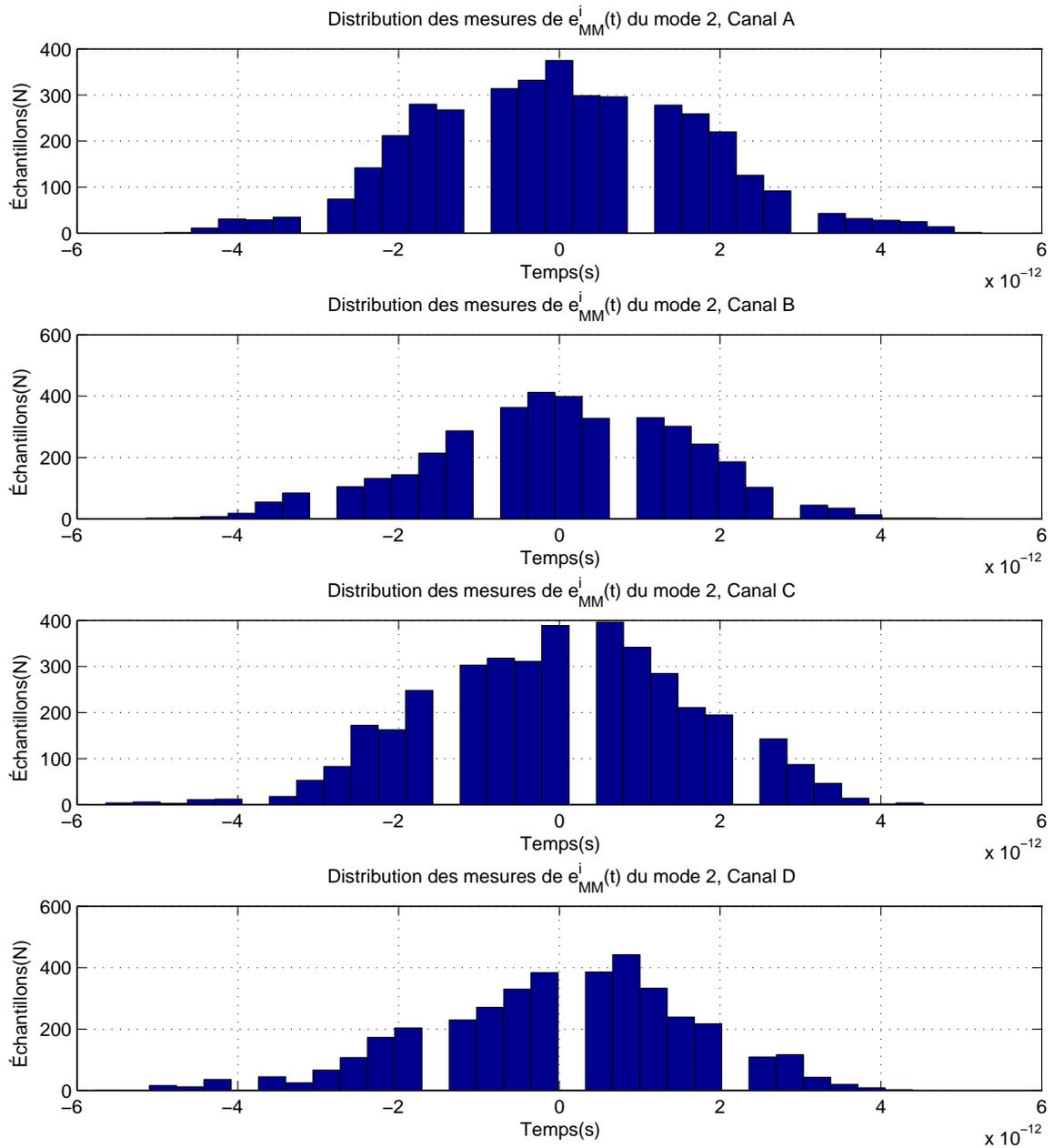


FIGURE 6.65 – Distribution de l'erreur  $e_{MM}^i(s)$  pour les quatre canaux pour le mode 2 après calibration

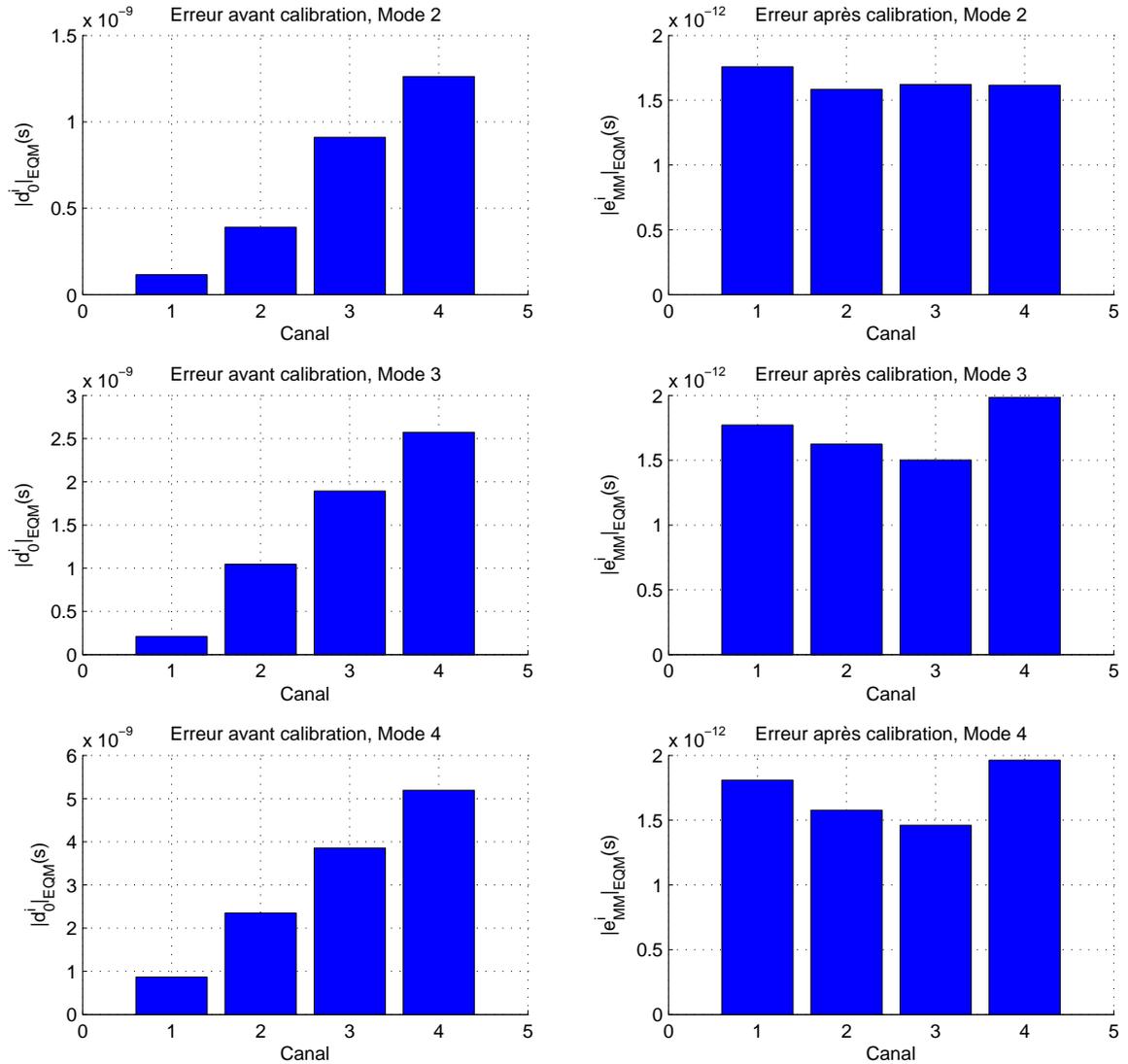


FIGURE 6.66 – Erreur quadratique moyenne(EQM) avant et après la phase de calibration pour les modes (2,3 et 4) pour les quarts canaux

TABLE 6.14 – Caractéristiques de la technique proposée. (\*) : Précision simulée dans le cas symétrique, (\*\*) : Précision simulée dans le cas asymétrique, (\*\*\*) : Précision mesurée sans minimisation de l'erreur.

	Complexité (1)	Minimisation (2)	Asymétrie (3)	Type de mesure	Adaptabilité	Précision
Technique proposée	Minimal	Ajustement périodique Avec minimisation	Réduction de l'effet de l'offset dû à l'asymétrie	Mesure unique au niveau du maître	Optimisé pour la topologie étoile des antennes	Simulée(*) : 0 ps Simulée(**) : <30 ps Mesurée(***) : < 2 ns
WR	Élevée Couches 1,2	Ajustement périodique	Supposé symétrique	Deux mesures	Générique	< 500 ps Augmente avec la longueur
PPT	Élevée Couches 1,2	Ajustement périodique	Supposé symétrique	Deux mesures	Générique	350 ns Augmente avec la longueur
PPS	Élevée Couches 1,2	Ajustement périodique	Supposé symétrique	Deux mesures	Générique	100 ns Augmente avec la longueur

# Conclusion générale

Le travail de thèse présenté ici a porté sur l'étude de la faisabilité technique et le développement d'un nouveau concept pour les radiotélescopes de futur : La numérisation large bande aux pieds des antennes et la mise en place d'un système de synchronisation dédié. Ce travail est réalisé au sein de la Station de Radioastronomie de Nançay, département micro-électronique. Pour mener à bien cette étude, nous nous sommes appuyés sur les moyens disponibles au sein du laboratoire électronique de la station.

Le travail a eu pour objectif d'étudier un nouveau concept technique propre à la radioastronomie, dédié à l'observation du soleil. Il porte sur la définition des contraintes techniques liées à la numérisation large bande aux pieds des antennes. La distribution des horloges pour les convertisseurs A/D intégrés dans les antennes, nécessite une synchronisation globale. Pour la première fois, d'après la littérature, une étude de la synchronisation a pu être réalisée pour un réseau distribué. Elle sera utile pour les prochaines générations de radiotélescopes. En effet, les techniques existantes sont basées sur des ajustements manuels et approximatifs des offsets des horloges avec une numérisation centralisée. L'étude permettra d'améliorer, dans le cas du radiohéliographe de Nançay, la qualité des observations multifréquences de la couronne solaire, de réduire les temps de latence des observations et d'affiner les estimateurs des grandeurs observables (température, densité d'électrons, etc.).

Les spécifications du nouveau système numérique NRH requièrent une résolution binaire du convertisseur A/D supérieure à 9 bits à 450 MHz dans le cas du soleil actif. Un écart de 4,5 bits est disponible entre le soleil actif et calme. L'étude montre que la gigue de l'horloge du convertisseur doit être inférieure à 700 fs à 450 MHz. La technologie actuelle des convertisseurs A/D répond suffisamment à ces deux contraintes. De plus, il existe des PLLs nettoyeurs de la gigue avec une valeur inférieure à 150 fs.

La synchronisation est un problème majeur pour les systèmes distribués qui nécessitent un temps universel commun, y compris l'interféromètre radio. Il existe des protocoles de synchronisation du temps dans la littérature, mais peu d'entre eux répondent au besoin d'une précision inférieure à la nanoseconde. Il est apparu durant l'étude que la synchronisation temporelle est plus fortement contrainte que la synchronisation fréquentielle. Un rapport de  $10^5$  est apparu entre les imprécisions des deux pour une source d'horloge commune. En effet, les horloges des convertisseurs A/D pour une bande analogique de 450 MHz nécessitent une précision sur la synchronisation temporelle de l'ordre de 50 ps pour une erreur  $\leq 1\%$  sur le taux de corrélation.

Dans un interféromètre radio constitué d'antennes multi-réparties, le taux de corrélation (image radio) dépend uniquement de l'erreur temporelle différentielle entre les voies. Ce résultat réduit la contrainte de conception du système de synchronisation. Nous avons proposé un algorithme permettant de calculer, calibrer et ajuster les horloges en temps réel. La théorie et les simulations montrent la synchronisation globale dans un réseau d'antennes. L'algorithme proposé permet d'ajuster parfaitement les canaux symétriques et de minimiser l'erreur dans le cas de l'asymétrie des canaux. La précision temporelle obtenue dans les simulations est inférieure à 100 ps, ajustable par les paramètres de l'algorithme. Les résultats de mesures sur quatre voies du système de calibration sont plus performantes que ceux des modèles théoriques (Matlab et VHDL). L'erreur différentielle inter-canaux est inférieure à 0.5 ps après la calibration. Nous avons montré, par les mesures, que la précision du système augmente avec la qualité des horloges utilisées.

Les perspectives ouvertes par ce travail sont importantes, puisqu'il est désormais possible d'optimiser la chaîne complète de mesure et d'ajustement des horloges en temps réel. En effet, les circuits à haute précision de mesure et d'ajustement des retards aller/retour (TDC et DTC) peuvent être intégrés sur FPGA. Il existe dans la littérature des modèles VHDL optimisés de ces derniers, comme celui utilisé dans le projet White Rabbit. Pour les liens à longues distances il est préférable d'utiliser les fibres optiques à la place des câbles coaxiaux pour la transmission de la référence et des intervalles de temps, caractérisés par une faible atténuation et une erreur du retard minimale due à l'asymétrie du canal (signal aller/retour dans le même câble avec des longueurs d'ondes différentes).

## Annexes

### 7.1 Annexe 0 : Interfaces de communication

#### 7.1.1 Interface série TDC-FPGA

##### 7.1.1.1 Introduction

Dans cette section on va aborder la synthèse et l'implémentation sur FPGA de l'interface de capture des résultats. Cette interface permet de récupérer les résultats des mesures des intervalles du temps des quatre canaux et les stockes en mémoire. L'interface série des résultats des mesures (voir la page 9 dans [160]) est constituée de trois signaux essentiels au format différentielle de type LVDS25 :

1. RdataX : Sortie de la donnée série du canal X (A, B, C ou D) activée au niveau haut
2. RCLK : Horloge des résultats
3. RstrobeX : Sortie du canal X indiquant la présence de la donnée dans la sortie RdataX activée au niveau bas

La section débutera par la description des chronogrammes de l'interface des résultats et les modes d'utilisation. Une description détaillée de la machine à état de l'acquisition du résultat pour les quatre canaux est abordé par la suite. On terminera avec la synthèse en VHDL de l'interface accompagnée des simulations et le code commenté se trouve en annexe.

##### 7.1.1.2 Analyse de fonctionnement

L'unité de mesure du temps (TMU) récupère et transmet les résultats à l'interface série des résultats via une liaison différentielle. L'interface dispose de trois signaux pour chaque canal et un signal d'horloge commun. L'interface des résultats est configurable par la liaison série illustrée précédemment. En effet, la taille en bits des résultats est configurable entre 8 et 40 en fonction de la dynamique de l'intervalle du temps à mesurer par l'intermédiaire des bits Rlength[0..2] du registre de contrôle (voir le tableau 1 dans [160]). La fréquence de l'horloge HCLK de synchronisation des données "Rdata" et "Rstrobe" est également programmable, il peut être ajustée entre 37,5 MHz et 300 MHz. Durant la phase de test, on va utiliser le mode 40 bits afin d'exploiter la dynamique maximales des mesures.

7.1.1.3 Interface des résultats

La figure 7.3 et 7.4 illustrent les signaux de l'interface en mode 40 bits et 16 bits successivement du circuit TDC. Les signaux "Rstrobe" et "Rdata" sont synchronisés par l'horloge "Rclk". Le début du cycle est indiqué par le passage à 0 du signal "Rstrobe". Le premier bit (TAG) reçu du signal de données "Rdata" indique la validité de la donnée (TAG=0 : Donnée valide) suivi par les 40 ou 16 bits des résultats en commençant par le poids faible LSB (R0), le dernier bit MSB (R39) indique le signe des résultats dans le cas où le mode signé est activé dans les registres de contrôle. Le bit R39 est suivi par deux bits à 0 indiquant la fin du cycle. Le signal "Rstrobe" remonte à 1 dès que la l'envoi du bit R39 (Figure. 7.3). Le cycle complet d'acquisition nécessite  $1+40+2=43$  coups d'horloge pour les résultats sur 40 bits. Pour une horloge de 150 MHz, le temps d'acquisition d'une donnée est de  $43 \times \frac{1}{150 \times 10^6} = 286,66ns$ .

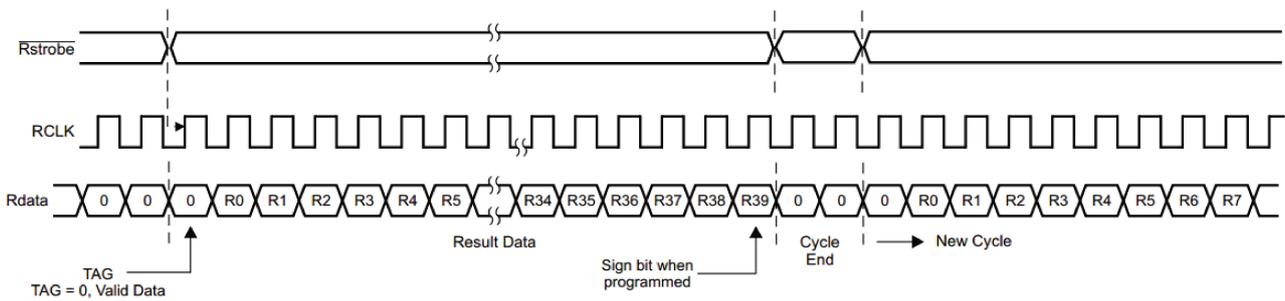


FIGURE 7.1 – Chronogrammes des signaux de l'interface des résultats en mode 40 bits

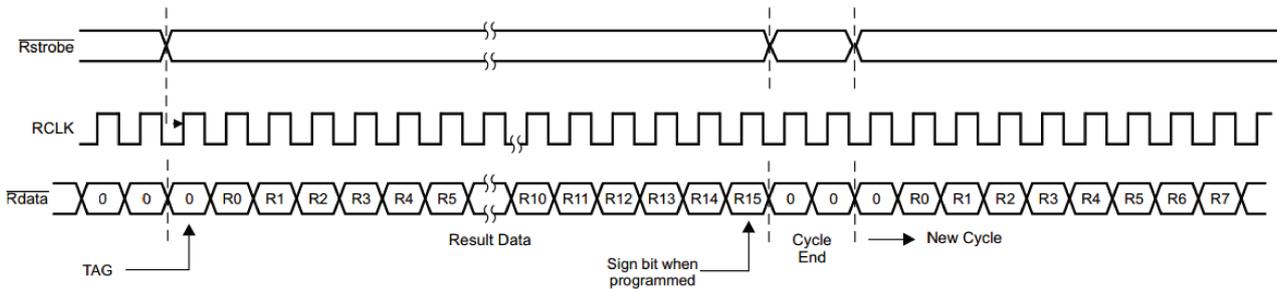


FIGURE 7.2 – Chronogrammes des signaux de l'interface des résultats en mode 16 bits

La latence  $T_{TDC}$  maximale et minimale nécessaire pour l'acquisition et la transmission des résultats est donnée par les équations (7.1) et (7.2)

$$T_{max} = T_{event} + 5 \times R_{ClkCycles} + 2 \times (Rdata_{Length} + 3) \times R_{ClkCycles} \quad (7.1)$$

$$T_{min} = T_{event} + 5 \times R_{ClkCycles} + (Rdata_{Length} + 3) \times R_{ClkCycles} \quad (7.2)$$

Avec  $T_{event} = 5 ns$  qui représente la durée minimale d'un événement et  $5 \times R_{ClkCycles}$  le nombre de cycles que prend la FIFO pour décaler des données à l'ALU.

Pour une fréquence d'horloge de résultats programmée à 150 MHz ( $R_{ClkCycles} = 6,66 ns$ ) et des données sur 40 bits, la latence maximale et minimale vaut successivement  $T_{max} = 611,67 ns$  et  $T_{min} = 325 ns$ .

La figure 7.3 décrit le graphique de transition de la machine à état de l'interface des résultats pour un canal quelconque. Elle est constituée de cinq états :

1. État d'attente du passage du signal "StrobeX" de l'état haut à l'état bas indiquant la présence des données dans le signal "RdataX"
2. État de lecture du bit TAG indiquant la validée des données
3. État de lecture de la sortie "RdataX" sur 40 ou 16 bits en fonction du mode d'utilisation
4. État de lecture du premier bit de fin de cycle
5. État de lecture du deuxième bit de fin de cycle

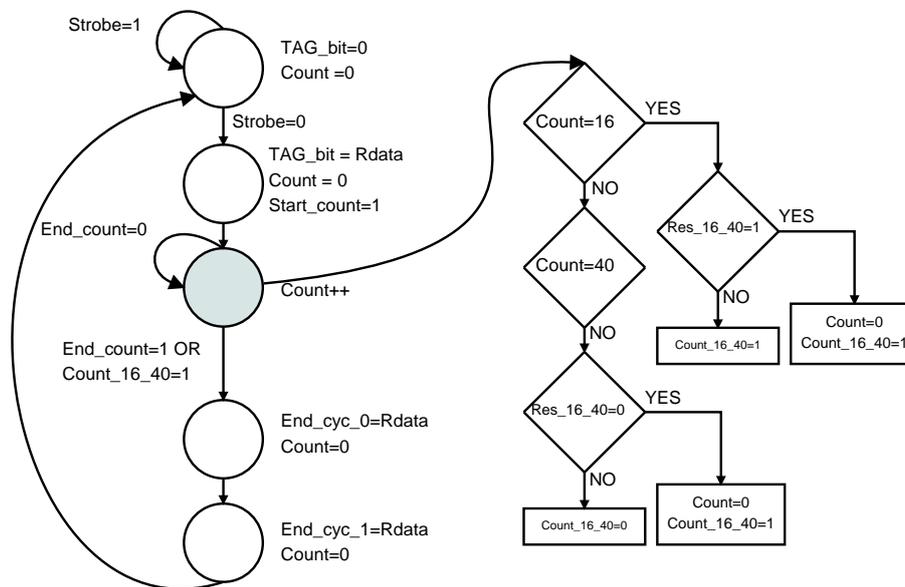


FIGURE 7.3 – Machine à état de l'interface d'acquisition des résultats avec deux modes d'utilisation (16 ou 40 bits) pour un seul canal

Durant l'état 3 (état de lecture du résultat), un ensemble de tests sont effectués (voir la figure 7.3) pour interrompre le compteur de résultat en fonction du mode de fonctionnement. Le mode est défini par le signal d'entrée "Res\_16\_40" : Lorsqu'il est à 1, la donnée est sur 16 bits sinon elle est sur 40 bits. En revanche, la taille de la donnée effective doit être configurée dans les registres de contrôle du circuit TDC par l'intermédiaire de l'interface de programmation. Pendant l'état (3) on récupère le résultat bit par bit en commençant par le bit de poids faible. Chaque bit transite dans un registre de données temporaires. La position des bits est définie par le compteur de bits. Le passage de la machine de l'état (3) à l'état (4) est réalisé par le passage à 1 du signal "End\_count" qui indique la fin du comptage (Arrêt du comptage à la valeur 16 ou 40), la donnée est ensuite mémorisée dans un signal temporaire sur 16/40 bits qui sera ensuite dupliquée dans le bus de sortie.

## 7.1.1.4 Implémentation en VHDL

## Entité

La figure 7.4 montre l'entité de l'interface d'acquisition de résultat. On constate l'utilisation de deux signaux de sortie Result\_start et Result\_end qui indiquent le début et la fin d'acquisition (voir le tableau 7.1). Ces derniers seront utilisés dans l'interface de mise en paquet et la transmission des données à l'ordinateur pour l'analyse et le traitement.



FIGURE 7.4 – Entité du contrôleur d'acquisition de résultats du circuit TDC

TABLE 7.1 – Description des différents signaux de l'interface d'acquisition de résultat

Signal	Direction	Description
Result_data_in	Entrée	Signal de la données "RdataX" activée au niveau haut
Result_strobe	Entrée	Signal "Rstrobe" activée au niveau bas
Result_ce	Entrée	Signal de validation du composant synchrone
Result_clk	Entrée	Signal d'horloge du résultat "RCLKX"
Result_rst	Entrée	Signal d'initialisation synchronisé, activé au niveau haut
Result_16_40	Entrée	Signal de contrôle du mode d'acquisition 16 ou 40 bits
Result_data_out	Sortie	Bus de sorties sur 40 bits du résultat
Result_start	Sortie	Signal indiquant le début d'acquisition de donnée (Vaut 1 lorsque durant l'acquisition du bit R0 du résultat)
Result_end	Sortie	Signal indiquant la fin d'acquisition de donnée il passe à 1 à la fin de l'acquisition (Lecture du dernier bit de résultat R39/R15)
Result_TAGbit	Sortie	Signal indiquant la validité de la donnée
Result_clk_out	Sortie	Image de l'horloge des résultats, il permet de synchroniser le bus de sortie pour un composant en amont du contrôleur

**Programme : Voir annexe II**

**Ressources**

Le tableau 7.2 montre les paramètres temporels et l’occupation en surface de la puce en fonction de la vitesse du synthétiseur. On constate que la fréquence minimale supportée par la puce est égale à 465,68 MHz pour la vitesse (-1). Elle augmente avec la vitesse. La même tendance pour le retard entre le signal d’entrée et la présence du résultat à la sortie, qui est de l’ordre de 400 ps pour la vitesse (-1). Pour conclure, l’architecture supporte un débit maximale de 465 Mbps pour la vitesse (-1), il est supérieur au débit des données issues du circuit TDC (300 Mbps de débit maximal). Le retard induit par le composant est négligeable et quasiment sans influence sur la donnée acquise car la donnée reçue est synchrone et dérivée d’une horloge précise (Horloge Maître du TDC).

TABLE 7.2 – Paramètres temporels et surface occupée en fonction du grade de vitesse de l’interface d’acquisition

Vitesse	$F_{max}(MHz)$	LUTs	Registres FF	IOB	Temps horloge (ns)	Retard IO(ns)
-1	465.68	222	177	88	1.33	0.40
-2	538.12	222	177	88	1.14	0.34
-3	628.43	222	177	88	0.96	0.28

**Simulation**

Pour illustrer le fonctionnement du composant on prends comme référence les paramètres suivants :

- Result\_ce = '1' ;
- Result\_rst = '0' ;
- Result\_data\_in = '1' ;
- Result\_16\_40 = '1' ou '0' ;

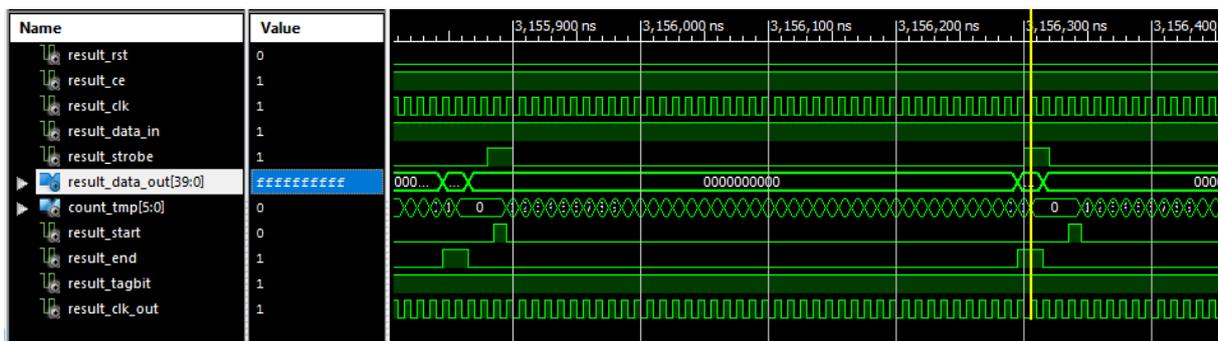


FIGURE 7.5 – Simulation de l’interface d’acquisition de données en mode 40 bits

La figure 7.5 montre les chronogrammes des signaux du composant *ResultHost* en mode 40 bits. On constate que la sortie *Result\_data\_out* est égale à x"0000000000" tout le temps, elle bascule à x"FFFFFFF" pendant que le signal *Result\_end* passe à 1 pour deux coups

d'horloge. On constate également que le signal Result\_start passe à 1 pour un coup d'horloge lorsque le signal Result\_strobe passe à 0. La figure 7.6 montre la durée des deux indicateurs de début et la fin d'acquisition.

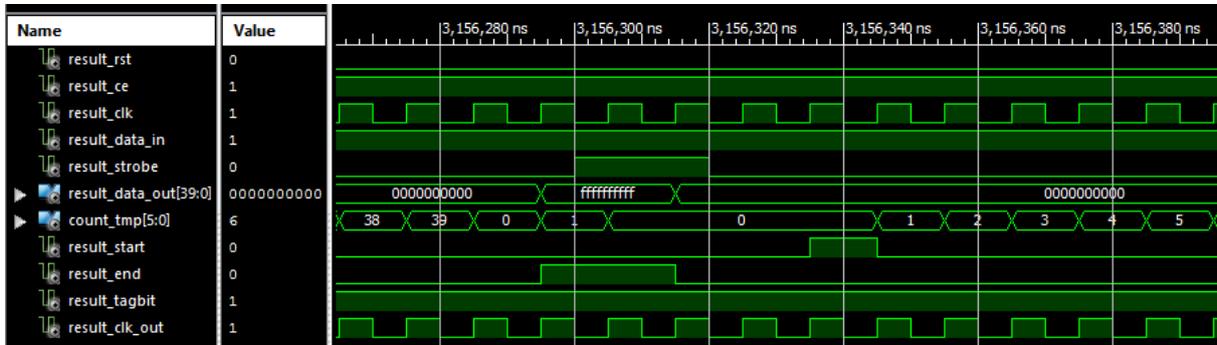


FIGURE 7.6 – Largeurs et allures des signaux en début et fin d'acquisition

La figure 7.6 illustre que le bus de sortie prend la valeur du registre interne lorsque le compteur passe à zéro avec une latence d'un coup d'horloge. En revanche l'indicateur de présence de donnée à la sortie (fin d'acquisition) est synchronisé avec les données.

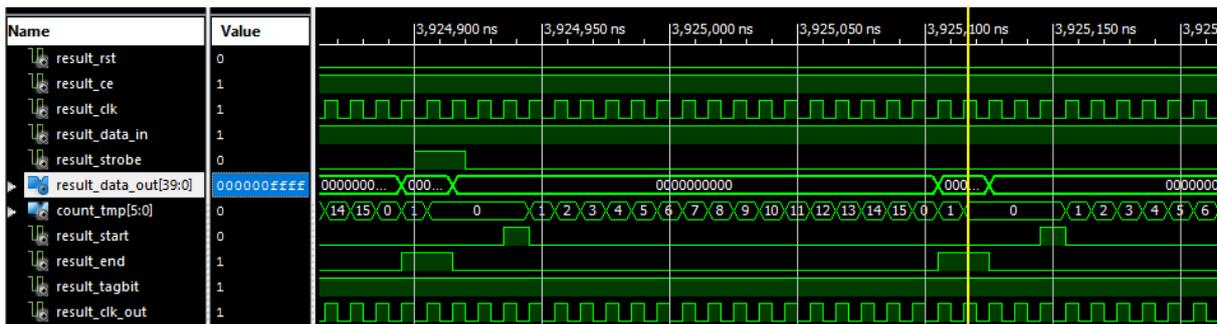


FIGURE 7.7 – Simulation de l'interface d'acquisition de données en mode 16 bits

La figure 7.7 met en évidence le mode 16 bits du bus de résultats. Il est clair que le compteur repasse à zéro à la valeur 15 ainsi et que le résultat prend la valeur x"0000FFFF", uniquement les 16 premiers bits sont remplis par la donnée issue de l'entrée Result\_data\_in (RdataX).

## 7.1.2 Interface série TDC-PC

### 7.1.2.1 Introduction

Après avoir récupérés les résultats avec le circuit TDC des quatre canaux (A à D), il est nécessaire de stocker les données pour l'analyse et le traitement différé ou en temps réel. Cette section sera dédiée au développement d'une interface permettant de transmettre en continue les données de mesure acquises du circuit TDC à un ordinateur via la liaison série. La figure 7.16 montre le schéma de base du processus de transmission. La section traite en détail le composant TDC2PC permettant l'acquisition, la mise en paquet et la transmission de données via la liaison série à l'ordinateur.

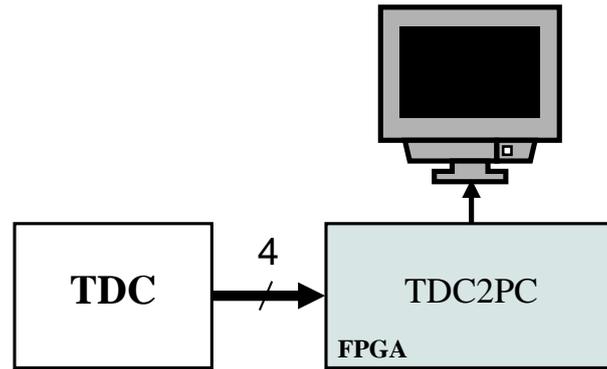


FIGURE 7.8 – Schéma bloc de l'interface de transmission de données entre le circuit TDC et l'ordinateur

### 7.1.2.2 Analyse de fonctionnement

L'interface des résultats décrit dans la section précédente permettait de récupérer le résultat de la mesure d'un canal du circuit TDC. La figure 7.17 montre le schéma bloc détaillé du circuit d'acquisition et de transmission des données à l'ordinateur. Il est constitué des blocs suivants :

1. *ResultHost* : Interface d'acquisition des résultats sur 16 ou 40 bits
2. *Format* : Composant de la mise en paquet de données par un ordre de priorité
3. *UART* : Contrôleur de liaison série asynchrone en mode transmission
4. *Ctrl* : Contrôleur permettant la gestion du trafic pour les différentes cadences des données

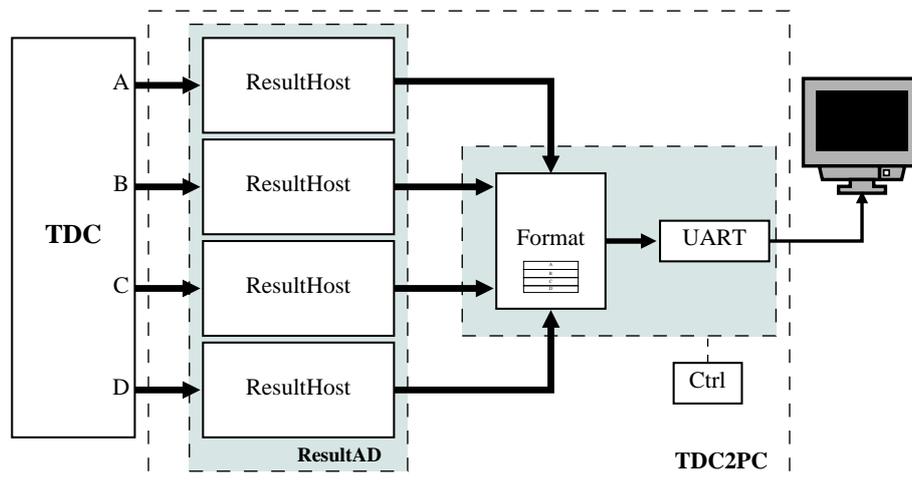


FIGURE 7.9 – Schéma bloc de l'interface de transmission de données entre le circuit TDC et l'ordinateur

*ResultHost* permet de récupérer quatre vecteurs de résultats sur 40 bits pour les canaux A à D d'une façon parallèle. Tous les canaux sont synchronisés par l'horloge des résultats RCLK. Les 4 vecteurs sont ensuite stockés en mémoire successivement par ordre de priorité.

Le contrôleur de transmission récupère les données dans une FIFO octet par octet pour la transmission via la liaison série. Il permet aussi la gestion de flux de données entre la mémoire et l'interface UART. En effet, elle interrompt l'écriture dans la mémoire durant la transmission.

### Formatage de données

L'étape de formatage de données consiste à la mise en paquet des données rapides issues du circuit TDC pour les quatre canaux. Ces derniers contiennent chacun 5 octets par canal, soit un total de 20 octets de données pour les 4 canaux. On a divisé les 40 bits en 5 octets parce que le contrôleur UART ne gère qu'un octet à la fois. La trame de données contient 28 octets au total : 20 octets de données pour les quatre canaux, 2 octets (x"AAAA") de début de trame et 2 octets (x"3333") de séparation entre chaque canal (voir la figure 7.17).

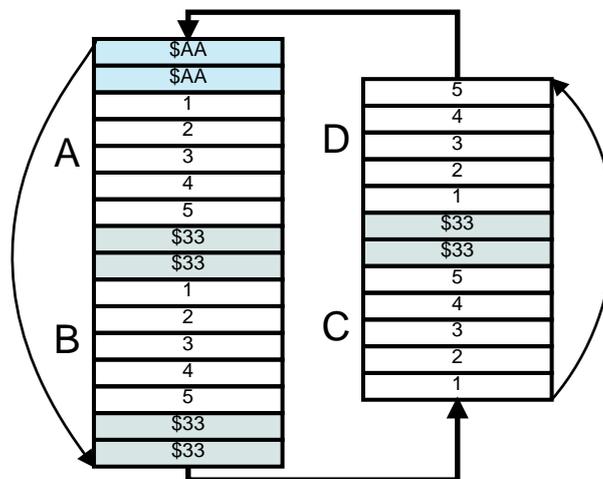


FIGURE 7.10 – Trame de données de quatre canaux

### Mise à jour de la trame de données

La trame de données est constituée de 28 mots sur 8 bits. Les cases de la mémoire de données (5x4) sont mises à jour canal par canal de façon continue. Le remplissage est effectué par ordre de priorité suivant : A puis B, C et D. La figure 7.18 présente la machine à état de la mise à jour de la mémoire. Chaque état de la machine assure la duplication du bus de résultats (sortie du composant "ResultHost") aux 5 cases mémoires successifs (de 1 à 5) en format LSB en premier (Fig. 7.18). Le transfert des octets de chaque canal est effectué en un seul coup d'horloge.

La figure 7.19 illustre la machine à état de la mise à jour de la mémoire des données. Les 4 premiers états assurent le transfert des 4 canaux à la mémoire, en revanche les deux derniers servent à générer le signal indiquant l'état de la mémoire (Mémoire pleine). Le transfert des données du bus à la mémoire est assuré par le passage à 1 du signal "End\_A", il représente l'indicateur de fin d'acquisition du résultat du composant *ResultHost*. On constate dans la machine une latence d'un coup d'horloge entre l'acquisition et le transfert, l'objectif de l'opération est de laisser le temps à la phase de transfert. Le transfert instantané n'est pas abouti (perte de données) d'après un ensemble de test sur carte qui ont été réalisés pendant la phase de configuration et validation du circuit. Le signal "MemFull" indique la transmission de la

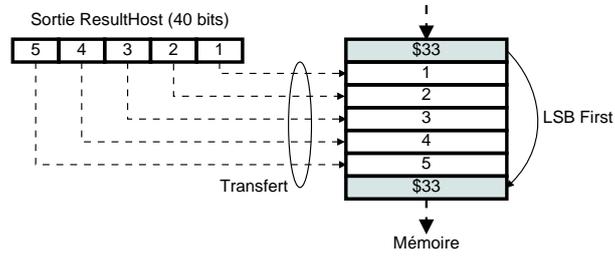


FIGURE 7.11 – Trame de données de quatre canaux

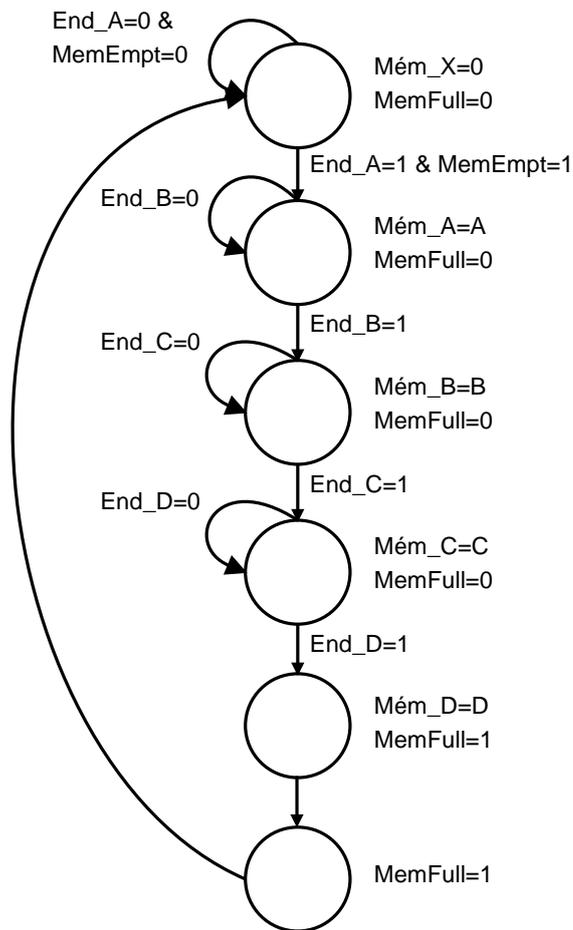


FIGURE 7.12 – Machine à état de la mise à jour de la mémoire

totalité des résultats à la mémoire.

Durant le premier état, la machine remet à zéro les cases mémoires des données puis reste en attente du signal de présence de résultat du canal A ( $End\_A=1$ ) et la transmission série effective de la totalité des données de la mémoire ( $MemEmpt=1$ ) (voir le programme en annexe). Une fois ses deux conditions réalisées, le processus de remplissage débute. Les deux indicateurs de l'état de la mémoire (mémoire vide ou plein) seront déployés dans le processus de contrôle de trafic entre la mémoire et le contrôleur de transmission. La figure 7.23 montre la machine à états du contrôleur de transmission.

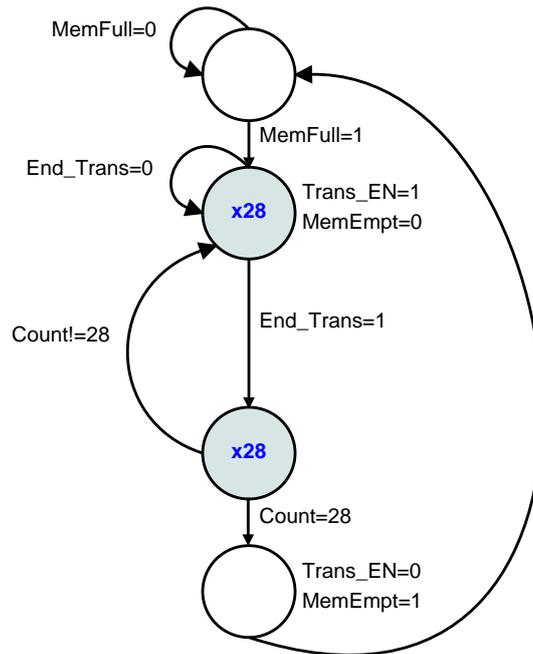


FIGURE 7.13 – Machine à état du contrôleur de transmission

L'interface UART dédiée à la transmission du contenu de la mémoire (28 octets) sur 8 bits avec une vitesse configurable (voir la section suivante). Le premier état est un état d'attente du signal indiquant la disponibilité des données en mémoire ( $MemFull=1$ ). Le deuxième et troisième état est des états de transmission des données. Durant ses derniers, la transmission est activée ( $Trans\_EN=1$ ). La fin de l'envoi d'un octet est indiquée par le passage à 1 du signal "End\_Trans". Cette opération est itérée pour chaque octet de la mémoire. Lorsque la totalité des données est transmise, l'indicateur de la mémoire vide passe à 1 ( $MemEmpt=1$ ). Pendant la phase de transmission l'accès à la mémoire est bloqué ( $MemEmpt=0$ ) pour maintenir les données intactes durant la transmission car le débit de transmission est plus long par rapport au débit d'acquisition.

### 7.1.2.3 Implémentation en VHDL

#### Entité

Le contrôleur UART 7.24 permet de transférer les données de mesure à l'ordinateur. Le mode de transmission de l'interface et de réception sont indépendants sur 8 bits. Le contrôleur

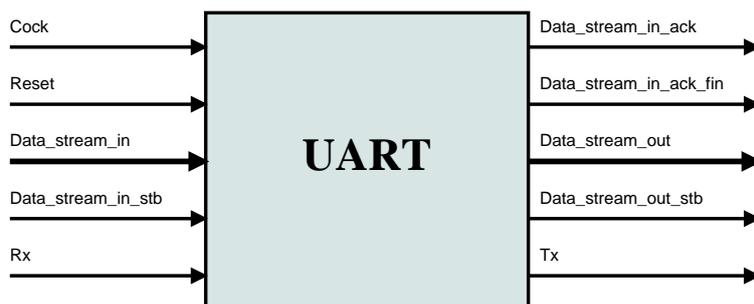


FIGURE 7.14 – Entité UART

TABLE 7.3 – Description des différents signaux de l'interface d'acquisition des résultats

Signal	Direction	Description
Clock	Entrée	Horloge de 100 MHz
Reset	Entrée	Entrée d'initialisation activée au niveau haut
Data_stream_in	Entrée	Donnée à transmettre sur 8 bits
Data_stream_in_stb	Entrée	Activation de l'interface en mode transmission activé au niveau haut
Rx	Entrée	Donnée série reçue
Data_stream_in_ack	Sortie	Indicateur de début de transmission
Data_stream_in_ack_fin	Sortie	Indicateur de fin de transmission
Data_stream_out	Sortie	Donnée reçue sur 8 bits
Data_stream_out_stb	Sortie	Activation de l'interface en mode réception, activé au niveau haut
Tx	Sortie	Donnée série transmise

intégré un bit de "start" et un bit de "stop" sans parité. Le paramètre vitesse en baud est déduit en précisant la valeur de la vitesse désirée en baud et la fréquence de l'horloge en Hertz. L'IP (Intellectual Property) du l'UART utilisé intègre le calcul automatique de la vitesse de transmission en baud. Pendant la phase de test, on va utiliser une fréquence de 100 MHz et une vitesse de 115200 baud.

**Programme : Voir l'annexe III**

### Ressources

On constate dans le tableau 7.5 indiquant les ressources du composant, que la fréquence minimale supportée par la puce est égale à 382,69 MHz pour la vitesse (-1). Elle augmente avec l'augmentation de la vitesse. La même tendance pour le retard minimal à l'entrée à l'arrivée de l'horloge. On conclut que l'architecture globale du système supporte le débit maximale de données à l'entrée (300 Mbps) pour divers niveau de vitesse (-1,-2 ou -3).

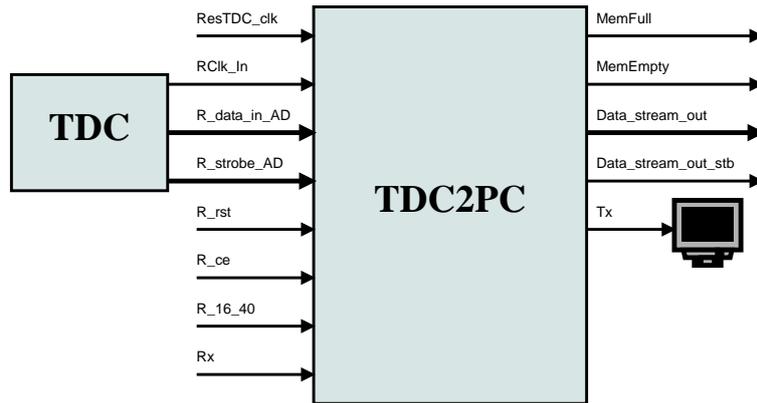


FIGURE 7.15 – Entité de l'interface d'acquisition et de transmission des mesures à l'ordinateur

TABLE 7.4 – Description des différents signaux de l'interface d'acquisition de résultat

Signal	Direction	Description
ResTDC_clk	Entrée	Horloge de l'interface TDC de 100 MHz
RClk_In	Entrée	Horloge des résultats issue du TDC (35.5,73...,300 MHz)
R_data_in_AD	Entrée	4 canaux de données série des résultats (RdataX)
R_strobe_AD	Entrée	4 signaux de présence de données dans les sorties Rdatax
R_rst	Entrée	Initialisation globale du composant
R_ce	Entrée	Activation du composant
R_16_40	Entrée	Sélection de mode d'acquisition 16 ou 40 bits
MemFull	Sortie	Indicateur de mémoire pleine
MemEmpty	Sortie	Indicateur de mémoire vide

TABLE 7.5 – Paramètre temporels et surface occupée en fonction du grade de vitesse de l'interface d'acquisition

Vitesse	$F_{max}(MHz)$	Registres FF	ADD/SUB	Temps horloge (ns)
-1	382.69	281	11	1.44
-2	444.91	281	11	1.24
-3	521.02	281	11	1.06

### 7.1.3 Interface série FPGA-TDC

#### 7.1.3.1 Introduction

L'interface de programmation permet de configurer les registres d'états et de contrôle du circuit TDC. On distingue deux modes de programmation : lecture ou écriture. Ces registres permettent de configurer le fonctionnement du composant. Tous les registres sont en mode lecture et écriture sauf les registres d'états. L'interface série dispose de 4 signaux : le signal d'initialisation "Reset" actif au niveau bas, "Hdata" : signal bidirectionnelle de la donnée série actif au niveau haut, "HCLK" : l'horloge synchrone avec la donnée et "Hstrobe" : indique le début du cycle de lecture ou d'écriture (voir la page 9 dans [160]). L'interface série peut

atteindre des débits importants (50 Mbps maximal). L'objectif principal de cette section et de développer en VHDL l'interface de programmation du circuit TDC avec ses différents modes d'utilisation.

### 7.1.3.2 Analyse de fonctionnement

Cette section sera consacrée à la description détaillée de l'interface de programmation du circuit TDC en mode lecture ou écriture 7.16. Elle permet d'adresser les registres de contrôles pour choisir un mode de mesure des intervalles du temps (40 bits, 16 bits, 300 MHz, 150 MHz, AB/CD, ...). Le composant contient deux blocs essentiels :

1. *Mode* : Sélection manuelle d'un mode fonctionnement du circuit TDC
2. *SerialHost* : Interface série sur 8 bits d'adresse et 16 bits de données en mode lecture/écriture. Ceci permet de mettre à jour les registres internes du TDC (registres d'état ou de contrôle).

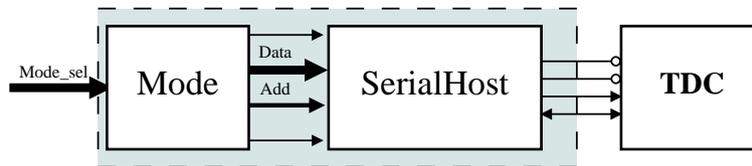


FIGURE 7.16 – Schéma bloc de l'interface de programmation du circuit TDC

La suite de la section est consacré au développement détaillé de ses deux composants.

### 7.1.3.3 Cycle d'écriture

La figure 7.17 illustre le cycle d'écriture dans le circuit TDC. Avec la présence de l'horloge HCLK. Le cycle d'écriture débute par le passage à 0 du signal Hstrobe (Hstrobe=0). Les premiers bits à la sortie Hdata définit le mode (Lecture=1, Écriture=0). Durant le cycle d'écriture le premier bit doit être à 0 (Hdata (1)=0). Le deuxième bit "parallèle" : Il définit le mode d'écriture dans les registres de même types pour divers canaux, si le bit est mis à 1, la donnée sera écrite en parallèle dans tous les registres de même type, sinon elle sera écrite uniquement dans le registre d'adresse indiquée dans la trame du cycle d'écriture. Les 8 bits du registre d'adresse seront ensuite véhiculés en série dans le signal de données Hdata en commençant par le poids fort (MSB First) puis par les 16 bits de données du même format. Un fois le dernier bit de données est envoyé (D0) durant le 26 coups d'horloge (voir la figure 7.17), un bit (w0) est envoyé pour valider les bits du cycle. Une fois les données validées, le signal "Hstrobe" repasse à 1. Le cycle d'écriture se termine par l'envoi de deux bits de fin de cycle, puis le cycle recommence. En résumé, un cycle complet nécessite 29 (2+8+16+1+2) coups d'horloge.

La machine à état du mode écriture (Fig. 7.18) décrit les chronogrammes de la figure 7.17. Chaque état est consacré à la transmission d'un ou plusieurs bits. Le cycle d'écriture est activé par la mise à 1 du signal "Start". Durant l'état d'attente l'horloge Hclk est désactivé, de plus le signal "Hstrobe" est initialisé à 1. Une fois le signal "Start" à 1, la machine envoie le bit indiquant le mode de transmission, dans le cas d'écriture ce bit est initialisé à 0 suivi du bit parallèle sinon à 0. Durant la transmission des registres d'adresse et de données, un compteur est

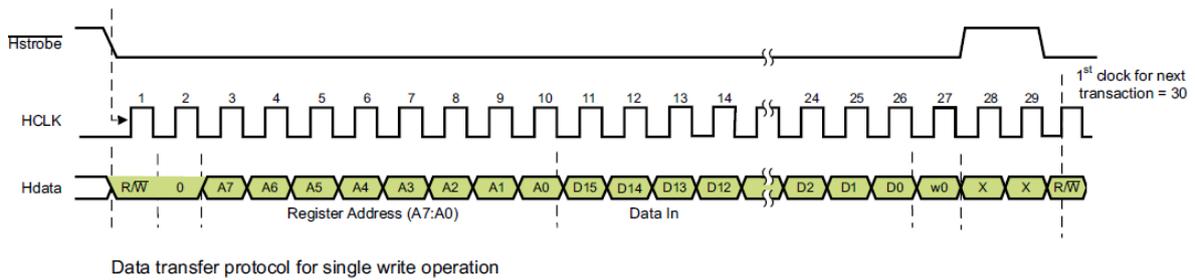


FIGURE 7.17 – Chronogrammes de l'interface série en mode écriture

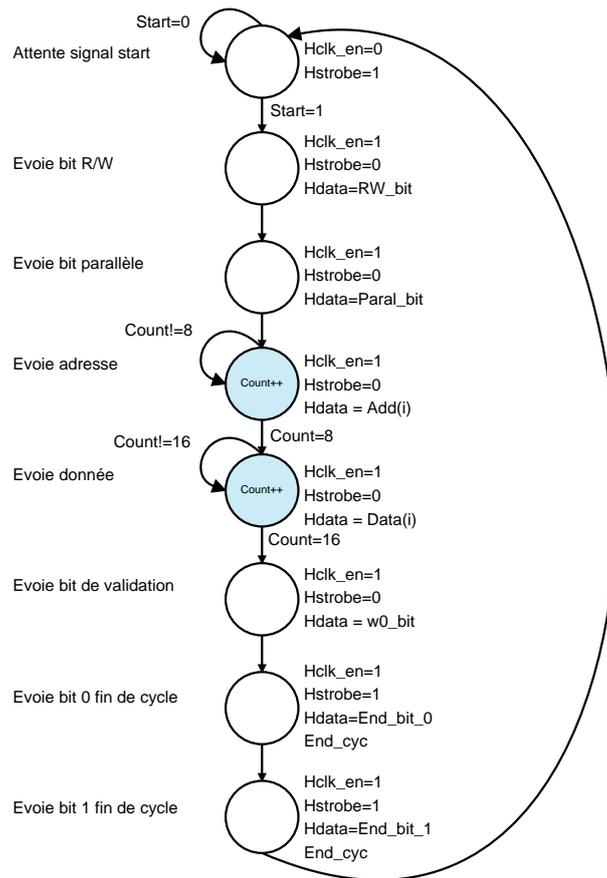


FIGURE 7.18 – Machine à état du mode d'écriture

utilisé pour transmettre bit par bit le contenu de ses derniers. La machine termine par l'envoi de bit de validation et deux bits de fin de cycle (voir la figure 7.18). Le signal "End\_cyc" est une sortie qui indique la fin du cycle. Ce dernier sera un indicateur de fin de cycle de programmation pour le composant en amont.

### 7.1.3.4 Cycle de lecture

Le cycle de lecture est quasiment identique à celui d'écriture. L'activation du mode d'écriture est indiqué par la mise à 1 du bit R/W ( $RW\_bit=1$ ) d'une part, d'autre part on constate l'absence du bit de validation des données ( $w0$ ) et l'apparition d'un bit  $r0$  qui indique au contrôleur interne du TDC le changement de sens des données : Pendant la transmission du registre d'adresse, le signal "Hdata" est en mode entrée, le signal bascule en mode sortie durant la réception de données. Cette opération prend un coup d'horloge (après l'envoi du bit  $r0$ )(voir Fig. 7.19). Le nombre de coup d'horloge pour un cycle complet reste intact pour les deux modes de fonctionnement. Durant ce mode, le bit parallèle doit être obligatoirement initialisé à 0 vue que le contenu des registres peut être différent pour les différents canaux.

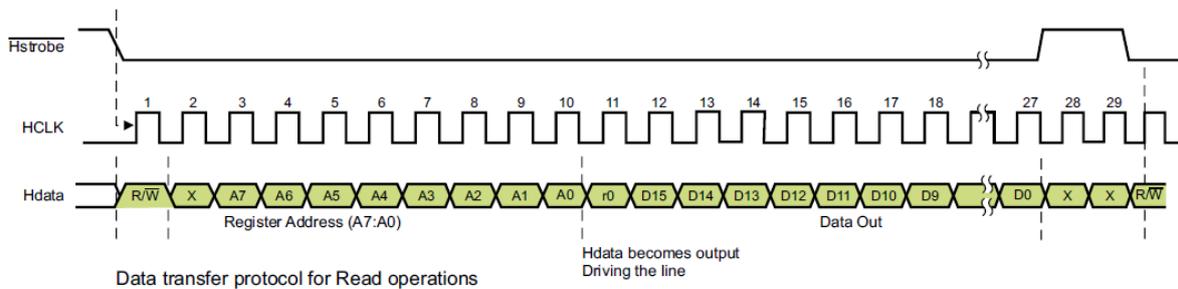


FIGURE 7.19 – Chronogrammes de l'interface série en mode lecture

La figure 7.20 montre la machine à état du mode de lecture des registres du circuit TDC. On observe l'apparition d'un état entre la transmission du registre d'adresse et celui des données, ce dernier sert à inverser le sens de données du signal Hdata. On constate également la suppression de l'état de validation des données (voir Fig. 7.20). Les autres états restent identiques à celui d'écriture. On va entamer dans la suite de la section, la description du bloc de sélection des différents modes de mesure du circuit TDC.

### 7.1.3.5 Les modes d'utilisation

Le composant TDC dispose de plusieurs registres de contrôle sur 16 bits. Ces derniers permettent de définir les paramètres de mesure des intervalles de temps. Pour simplifier la procédure de mesure, on a défini 8 modes d'utilisation essentiels. Chaque mode est caractérisé par les valeurs des registres de contrôle. Pendant la phase de définition des modes, on va utiliser l'interface série en mode écriture uniquement. Le tableau 7.6 illustre les registres de contrôles leurs descriptions et adresses. On distingue quatre registres essentiels 80h, 81h, x0h et x1h, avec x prend les valeurs 0, 2, 4 ou 6 pour les canaux A à D successivement. Les modes d'utilisation du circuit envisageable sont les suivants :

- Mode 0 : Mode initialisation des registres de contrôle

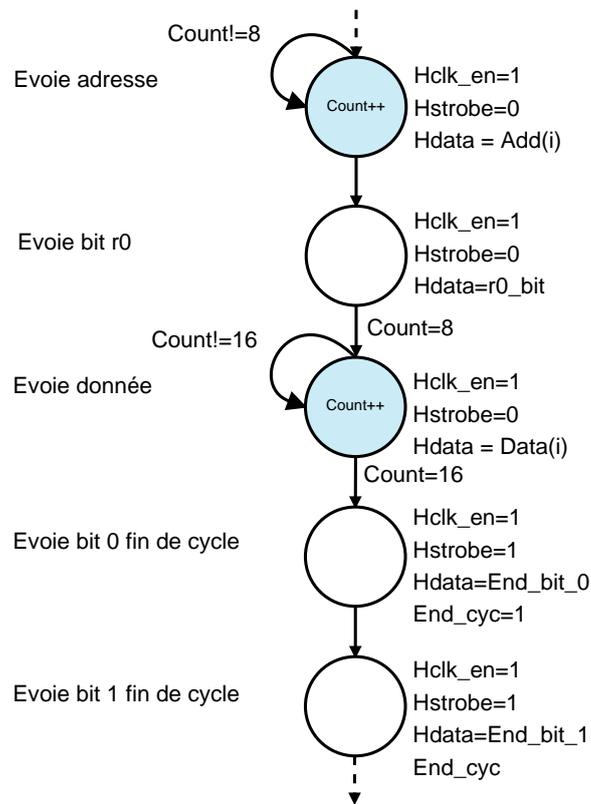


FIGURE 7.20 – Machine à état du mode de lecture

- Mode 1 : Mesure avec HCLK= 37,5 MHz sur 40 bits
- Mode 2 : Mesure avec HCLK= 75 MHz sur 40 bits
- Mode 3 : Mesure avec HCLK= 150 MHz sur 40 bits
- Mode 4 : Mesure avec HCLK= 300 MHz sur 40 bits
- Mode 5 : Mode silencieux
- Mode 6 : Désactivation des entrées Events/Sync
- Mode 7 : Mode d'initialisation (Reset)

TABLE 7.6 – Description des registres de contrôle du composant TDC

Adresse (Hexadécimal)	Description
80	Activation de l'horloge RCLK Activation du mode faible énergie Activation du signal de synchronisation (Sync) Définition de polarité du signal Sync Définition fréquence RCLK Définition taille des résultats Connections des canaux AB-CD Initialisation de la puce
81	Activation du mode silencieux Définition de la taille du compteur
00,20,40,60	Activation d'un canal Définition de la polarité des signaux Events Définition de retard des signaux Events
01,21,41,61	Définition des conditions de déclenchement des canaux

Le composant *SerialHost* illustré précédemment dispose de deux entrées : Entrée de la données sur 16 bits et celle d'adresse sur 8 bits. La programmation du circuit TDC consiste à envoyer quatre couples de registres (Adresse, Donnée) à l'interface série de programmation. Plus précisément les couples suivant (x0, Data(x0)), (x1, Data(x1)), (80, Data(80)) puis (81, Data(81)). La figure 7.21 décrit le processus séquentielle de configuration du composant. La transmission effective du contenu des registres est assurée par la mise à un de l'indicateur de fin du cycle "End\_cyc" du circuit *SerialHost*. Le processus est déclenché par la mise à 1 du signal de début de cycle "Start" (état d'attente), un indicateur de fin de cycle global est positionné à 1 vers la fin de cycle (état final).

### 7.1.3.6 Implémentation en VHDL

#### Entité

**Programme : voir l'annexe VI**

#### Ressources

Les tableaux 7.9 et 7.10 montrent les ressources en terme de surface et vitesse de l'interface de programmation pour les cibles FPGA Spartran-3A et Vertex-6. On constate que la fréquence

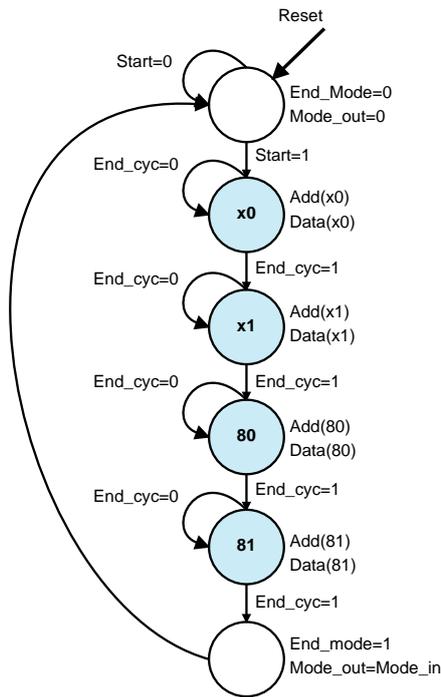


FIGURE 7.21 – Machine à état d’adressage des registres de contrôle (Définition du mode d’utilisation)

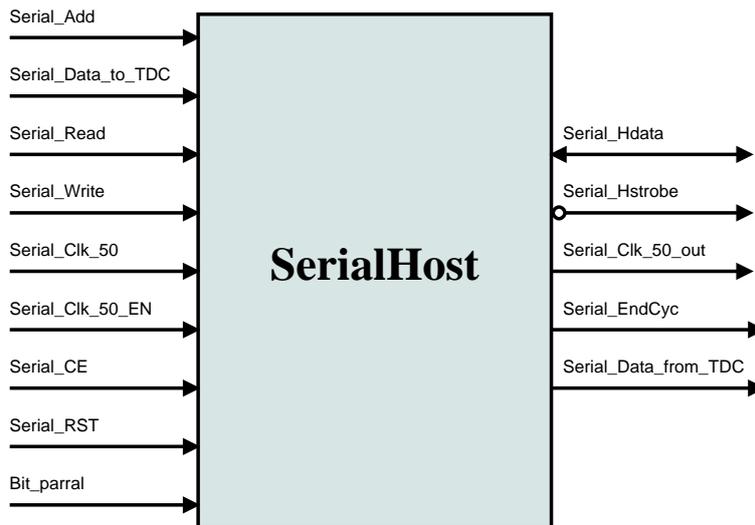


FIGURE 7.22 – Entité du composant SerialHost)

TABLE 7.7 – Description des signaux de SerialHost

Signal	Direction	Description
Serial_Add	Entrée	Bus d'adresse sur 8 bits
Serial_Data_to_TDC	Entrée	Bus de données sur 16 bits
Serial_Read	Entrée	Activation mode lecture activé NH
Serial_Write	Entrée	Activation mode écriture activé NH NH
Serial_Clk_50	Entrée	Horloge de transmission
Serial_Clk_50_EN	Entrée	Signal activation de l'horloge
Serial_CE	Entrée	Activation du composant activé NH
Serial_RST	Entrée	Signal de mise à zero du composant activé NH
Serial_Hdata	E/S	Signal donnée Hdata
Serial_Data_from_TDC	Sortie	Bus des données reçues sur 16 bits
Serial_EndCyc	Sortie	Indicateur fin de cycle de transmission
Serial_Clk_50_out	Sortie	Horloge image de l'horloge d'entrée
Serial_Hstrobe	Sortie	Signal Hstrobe
Bit_parral	Entrée	Signal de bit reverse

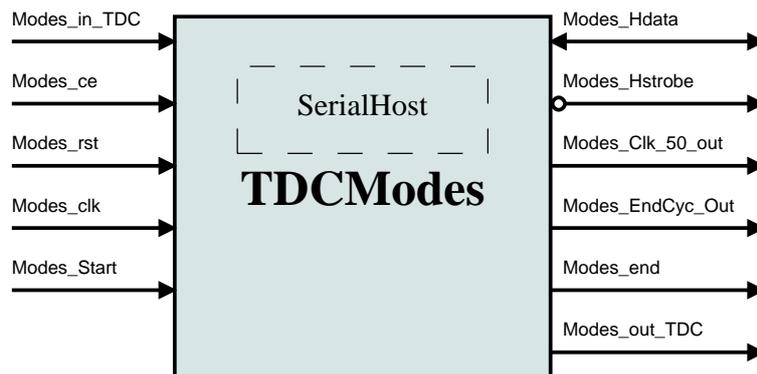


FIGURE 7.23 – Entité du composant Global TDCModes)

minimale supportée par les interfaces pour les deux puces sont supérieures à 225 MHz. L'interface série du TDC supporte une fréquence maximale de 50 MHz, donc l'architecture répond largement à la contrainte de fréquence. Pendant la phase de test, on va utiliser une carte à base de Spartran 3A pour configurer l'interface série du circuit TDC. Ce choix du à la logique LVCMOS33 utiliser par le TDC. En revanche, la carte de développement à base du Virtex-6 dédiée d'une logique LVDS25 configurable en LVCMOS, la dynamique de ce dernier n'est pas compatible avec l'interface série du TDC. Une autre solution possible est l'utilisation des composants d'adaptation des niveaux, cette solution est abandonnée car elle nécessite une carte d'adaptation de niveau.

### Simulation

Les figures 7.24 et 7.25 montrent les signaux de contrôle et d'états de l'interface série pour les modes 0 et 1. On constate que l'horloge "Clk\_50" est activée uniquement pendant le

TABLE 7.8 – Description des signaux de TDCModes

Modes_in_TDC	Entrée	Sélection de mode sur 3 bits
Modes_ce	Entrée	Activation du composant
Modes_rst	Entrée	Initialisation
Modes_clk	Entrée	Horloge d'entrée de l'interface série
Modes_Start	Entrée	Début cycle de programmation d'un mode
Modes_Hdata	E/S	Hdata
Modes_end	Sortie	Indication de fin de programmation d'un mode
Modes_out_TDC	Sortie	Mode activé
Modes_Clk_50_out	Sortie	Horloge de l'interface série
Modes_Hstrobe	Sortie	Hstrobe
Modes_EndCyc_Out	Sortie	Indication fin de cycle du SerialHost

TABLE 7.9 – Occupation de l'interface série dans la cible FPGA Spartran-3A

Vitesse	$F_{max}(MHz)$	LUT4	Registres FF	IOB	Retard (ns)
-4	225.24	71	45	15	7.21
-5	265.10	71	45	15	6.78

TABLE 7.10 – Occupation de l'interface série dans la cible FPGA Vertex-6

Vitesse	$F_{max}(MHz)$	LUT-FF	Registres FF	IOB	Retard (ns)
-1	586.16	65	49	15	1.38
-2	681.94	65	49	15	0.99
-3	802.69	65	49	15	0.84

cycle de programmation (durant la période de l'indicateur du fin du cycle de programmation "Modes\_endCyc\_out"). La fin d'envoi effective de chaque registre de contrôle (x0, x1, 80 et 80) (voir la figure 7.21) est indiquée par le signal "Modes\_end", le bus d'état présent indique l'état du registre en cours de transfert. Le signal "Hdata" contient les données séries du registre d'adresses, du contrôle et les modes de programmation (lecture, écriture et le mode parallèle). Dans les deux figures (Figure. 7.24 - 7.25), le premier bit est à '0' (mode écriture) durant le passage à zéro du signal "Hstrobe", suivi de '1' (mode parallèle), puis par l'adresse sur 8 bits et la donnée sur 16 bits. Les simulations du contrôleur de l'interface série décrivent parfaitement la synthèse théorique du composant.

## 7.2 Annexe I : Programme VHDL du générateur des intervalles $T_i$

```

001 library ieee;
002 use ieee.std_logic_1164.all;
003 use ieee.numeric_std.all;
004 use ieee.math_real.all;

```

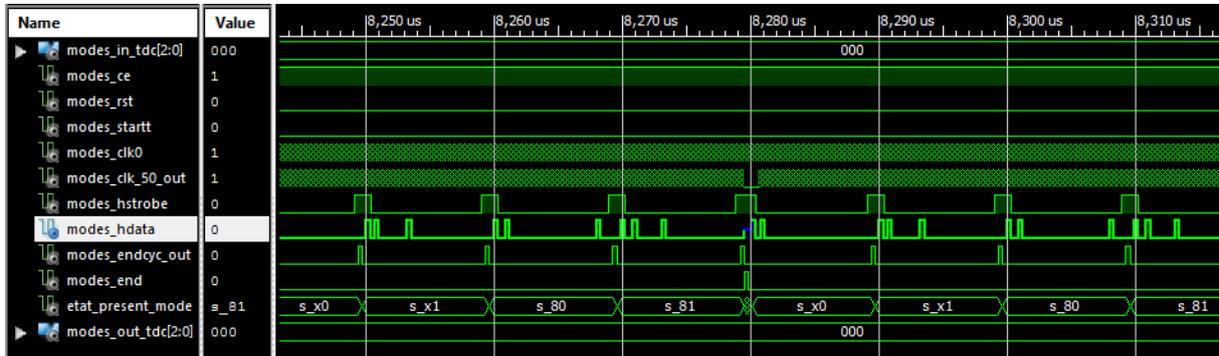


FIGURE 7.24 – Chronogrammes de l’interface série en mode initialisation (Mode 0)

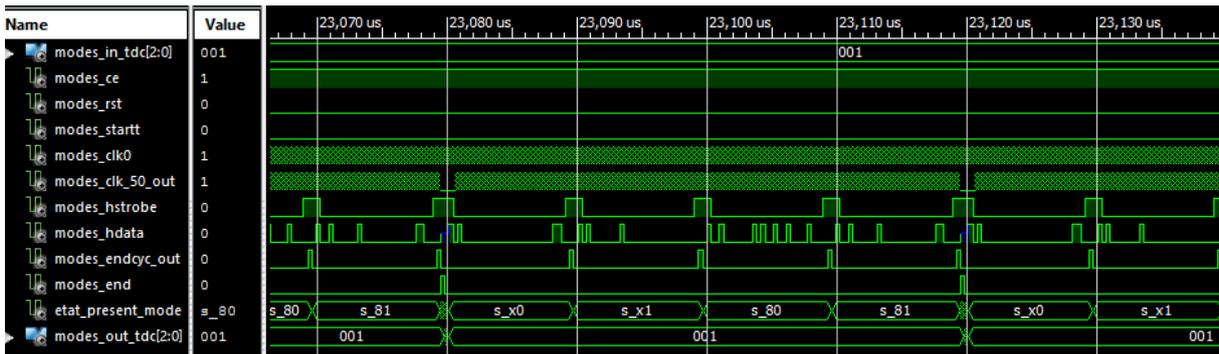


FIGURE 7.25 – Chronogrammes de l’interface série en mode de mesure avec HCLK = 37,5 MHz sur 40 bits (Mode 1)

```

005
006 -- Buffers
007 Library UNISIM;
008 use UNISIM.vcomponents.all;
009
010 entity Events_Gen is
011     Port ( EG_clk_200_ext : in  STD_LOGIC;
012           EG_ce           : in  STD_LOGIC;
013           EG_rst         : in  STD_LOGIC;
014           EG_modes       : in  STD_LOGIC_VECTOR (2 downto 0);
015           EG_events_p    : out  STD_LOGIC_VECTOR (3 downto 0);
016           EG_events_n    : out  STD_LOGIC_VECTOR (3 downto 0);
017           EG_sync_p      : out  STD_LOGIC;
018           EG_sync_n      : out  STD_LOGIC);
019 end Events_Gen;
020
021 architecture Behavioral of Events_Gen is
022
023 signal EG_sync_tmp           : STD_LOGIC := '0';
024 signal EG_events            : STD_LOGIC_VECTOR (3 downto 0);
025 signal EG_sync              : STD_LOGIC := '0';
026 signal Clk_EvSync           : STD_LOGIC := '0';
027 signal Clk_EvSync_tmp      : STD_LOGIC := '0';
028 signal count_EvSync        : STD_LOGIC_VECTOR (15 downto 0) := x"0000";
029 signal EG_clk_200_ext_tmp  : STD_LOGIC := '0';
030
031 type Etat_SyncEv is (
032     S_wait, S_syn_0, S_syn_1,
033     S_wait_EvA_0, S_wait_EvA_1, S_EvA_0, S_EvA_1,
034     S_wait_EvB_0, S_wait_EvB_1, S_EvB_0, S_EvB_1,
035     S_wait_EvC_0, S_wait_EvC_1, S_EvC_0, S_EvC_1,
036     S_wait_EvD_0, S_wait_EvD_1, S_EvD_0, S_EvD_1,
037     S_end_0, S_end_1);
038 Signal Etat_present_SyncEv, Etat_futur_SyncEv : Etat_SyncEv := S_wait;
039 begin
040     -- Instanciation des buffers Signle/LVDS
041     Sync_buf : OBUFDS
042         generic map (
043             IBUF_LOW_PWR => FALSE,
044             DIFF_TERM    => TRUE,
045             IOSTANDARD   => "LVDS_25"
046         )
047         port map (
048             O => EG_sync_p,
049             OB => EG_sync_n,
050             I => EG_sync
051         );
052     Ev0_buf : OBUFDS
053         generic map (
054             IBUF_LOW_PWR => FALSE,
055             DIFF_TERM    => TRUE,
056             IOSTANDARD   => "LVDS_25"
057         )
058     ...
059
060     -- Compteur de génération des evenements avec une fréquence de 200 MHz +

```

```

060  -- Le signal de synchronisation
061  p_count : process(EG_rst, EG_ce, EG_clk_200_ext_tmp)
062  begin
063      if EG_rst = '1' then
064          count_EvSync <= (others => '0');
065      elsif EG_clk_200_ext_tmp = '1' and EG_clk_200_ext_tmp'event then
066          if EG_ce = '1' then
067              count_EvSync <= count_EvSync + 1;
068          else
069              count_EvSync <= (others => '0');
070          end if ;
071      end if;
072  end process;
073
074  -- Modes
075  p_modes : process (EG_clk_200_ext_tmp, EG_modes, EG_ce )
076  begin
077      if EG_clk_200_ext_tmp = '1' and EG_clk_200_ext_tmp'event then
078          if EG_ce = '1' then
079              CASE EG_modes IS
080                  -- Mode 0
081                  WHEN "000" =>
082                      Clk_EvSync_tmp <= count_EvSync(5);
083                  -- Mode 1
084                  WHEN "001" =>
085                      Clk_EvSync_tmp <= count_EvSync(6);
086                  ...
087              END CASE;
088          else
089              Clk_EvSync_tmp <= '0';
090          end if ;
091      end if;
092  end process;
093  Clk_EvSync <= Clk_EvSync_tmp;
094  EG_clk_200_ext_tmp <= EG_clk_200_ext;
095
096  -- Process de mémorisation et passage l'état futur
097  P_Mem_etat : process (Clk_EvSync, EG_ce, EG_rst)
098  begin
099      if EG_rst = '1' then
100          Etat_present_SyncEv <= S_wait;
101      elsif Clk_EvSync'event and Clk_EvSync = '1' then
102          if(EG_ce = '1') then
103              Etat_present_SyncEv <= Etat_futur_SyncEv;
104          else
105              Etat_present_SyncEv <= S_wait;
106          end if ;
107      end if;
108  end process P_Mem_etat;
109
110  P_Logic_ES_SyncEv : process ( Etat_present_SyncEv, EG_ce )
111  begin
112      case Etat_present_SyncEv is
113          --wait
114          when S_wait =>
115              -- Logique des sorties

```

```
116         EG_events <=(others =>'0');
117         EG_sync <='0';
118         -- Passage de S_wait ----> S_syn_0
119         if EG_ce ='1' then
120             Etat_futur_SyncEv <= S_syn_0;
121         else
122             Etat_futur_SyncEv <= S_wait;
123         end if ;
124
125         --S_syn_0
126         when S_syn_0 =>
127             -- Logique des sorties
128             EG_events <=(others =>'0');
129             EG_sync <='1';
130             -- Passage de S_syn_0 ----> S_syn_1
131             Etat_futur_SyncEv <= S_syn_1;
132
133         --S_syn_1
134         when S_syn_1 =>
135             -- Logique des sorties
136             EG_events <=(others =>'0');
137             EG_sync <='1';
138             Etat_futur_SyncEv <= S_wait_EvA_0;
139         ...
140
141         --S_end_1
142         when S_end_1 =>
143             -- Logique des sorties
144             EG_events <=(others =>'0');
145             EG_sync <='0';
146             -- Passage de S_end_1 ----> S_wait
147             Etat_futur_SyncEv <= S_wait;
148
149         --Autres
150         when others =>
151             Etat_futur_SyncEv <= S_wait;
152         end case;
153     end process P_Logic_ES_SyncEv;
154 end Behavioral;
```

## 7.3 Annexe II : Programme VHDL de l'interface série des résultats TDC-FPGA

```
001 ...
002 entity ResultHost is
003     Generic( N_res : positive:= 40;
004             N_count : positive:=6
005             );
006     Port ( Result_clk      : in  STD_LOGIC:= '0';
007           Result_data_in  : in  STD_LOGIC:= '0';
008           Result_strobe   : in  STD_LOGIC:= '0';
009           Result_rst      : in  STD_LOGIC:= '0';
010           Result_ce       : in  STD_LOGIC:= '0';
011           Result_16_40    : in  STD_LOGIC:= '0';
```

```

012         Result_data_out : out STD_LOGIC_VECTOR (N_res -1 downto 0);
013         Result_start    : out STD_LOGIC:= '0';
014         Result_end      : out STD_LOGIC:= '0';
015         Result_TAGbit   : out STD_LOGIC:= '0';
016         Result_clk_out  : out STD_LOGIC:= '0'
017
018     );
019 end ResultHost;
020
021 architecture Behavioral of ResultHost is
022 ...
023 begin
024
025     ----- Signaux début et fin -----
026
027     -- Génération des signaux début et la fin
027     -- d'acquisition des données sur 16/40 bits
028     P_start_end : process (Etat_present_Result, Count_event_16_40,
028     Result_rst, Result_clk)
029     begin
030         if Result_rst = '1' then
031             Start_count <= '0';
032             End_count <= '0';
033         elsif Result_clk'event and Result_clk = '1' then
034             if Etat_present_Result = S_tag then
035                 Start_count <= '1';
036                 End_count <='0';
037             elsif Count_event_16_40 = '1' then
038                 Start_count <= '0';
039                 End_count <='1';
040             elsif Etat_present_Result = S_end_0 then
041                 End_count <='1';
042             else
043                 Start_count <= '0';
044                 End_count <= '0';
045             end if;
046         end if;
047     end process P_start_end;
048     Result_start <= Start_count;
049     Result_end <= End_count;
050     Result_end_tmp <= End_count;
051
052     ----- Résultats -----
053
054     -- Process de mémorisation et passage l'état futur
055     P_Mem_etat_Result : process (Result_rst, Result_ce, Result_clk)
056     begin
057         if Result_rst = '1' then
058             Etat_present_Result <= S_wait;
059         elsif Result_clk'event and Result_clk = '1' then
060             if(Result_ce = '1') then
061                 Etat_present_Result <= Etat_futur_Result;
062             else
063                 Etat_present_Result <= S_wait;
064             end if ;
065         end if;

```

### 7.3. ANNEXE II : PROGRAMME VHDL DE L'INTERFACE SÉRIE DES RÉSULTATS TDC-FPGA

---

```
066     end process P_Mem_etat_Result;
067
068     -- Compteur de nombre de bits en fonction du mode
069     P_count_incrim : process(Etat_present_Result, Result_clk, Result_16_40 )
070     begin
071         if Result_rst = '1' then
072             Count_event_16_40 <= '0' ;
073             Count_tmp <= (others =>'0');
074             Count_event_16_40 <= '0';
075         elsif (Result_clk'EVENT and Result_clk='1') then
076             if Etat_present_Result = S_data then
077                 Count_event_16_40 <= '0' ;
078                 if Etat_present_Result = S_data then
079                     Count_tmp <= Count_tmp +1 ;
080
081                 -- Mode 16 bits
082                 if (Count_tmp = Count_value_16) then
083                     if Result_16_40 = '1' then
084                         Count_tmp <= (others =>'0');
085                         Count_event_16_40 <='1';
086                     else
087                         Count_event_16_40 <='0';
088                     end if;
089
090                 -- Mode 40 bits
091                 elsif (Count_tmp = Count_value_40)
092             then
093                 if Result_16_40 = '0' then
094                     Count_tmp <= (others =>'0');
095                     Count_event_16_40 <= '1' ;
096                 else
097                     Count_event_16_40 <='0';
098                 end if;
099             end if;
100         else
101             Count_tmp <= (others => '0');
102         end if ;
103     else
104         -- Initialisation du compteur
105         Count_tmp <= (others =>'0');
106         Count_event_16_40 <= '0' ;
107     end if;
108 end process P_count_incrim;
109 -- Génération de l'horloge de sortie
110 Result_clk_out <= Result_clk;
111
112 -- Process registre d'état du compteur de bits de données
113 P_Logic_ES_Result : process (Result_strobe,Result_end_tmp,
113 Result_clk, Etat_present_Result, Count_event_16_40)
114 begin
115     case Etat_present_Result is
116         --wait
117         when S_wait =>
118
119             -- Passage de S_wait ----> S_tag
```

```

120         if Result_strobe = '0' then
121             Etat_futur_Result <= S_tag;
122         else
123             Etat_futur_Result <= S_wait;
124         end if ;
125     --S_tag
126     when S_tag =>
127         Result_TAGbit_tmp <= Result_data_in_tmp;
128         Etat_futur_Result <= S_data;
129     when S_data =>
130         if Result_end_tmp = '1' then
131             Etat_futur_Result <= S_end_0;
132         else
133             Etat_futur_Result <= S_data;
134         end if;
135
136         -- Saut dans le cas de la detction du
136         -- fin du compteur
137         if Count_event_16_40 = '1' then
138             Etat_futur_Result <= S_end_0;
139         else
140             Etat_futur_Result <= S_data;
141         end if ;
142     --S_end_0
143     when S_end_0 =>
144         -- Recupération de bit 0 du fin de cycle
145         Result_EndBit0 <= Result_data_in_tmp;
146         Etat_futur_Result <= S_end_1;
147     --S_end_1
148     when S_end_1 =>
149         -- Recupération de bit 1 du fin de cycle
150         Result_EndBit1 <= Result_data_in_tmp;
151         Etat_futur_Result <= S_wait;
152     --S_end_2
153     when Others =>
154         Etat_futur_Result <= S_wait;
155     end case;
156 end process P_Logic_ES_Result;
157 Count <= Count_tmp;
158
159 -- Récupération de TAGbit
160 Result_TAGbit <= Result_TAGbit_tmp;
161
162 -- Synchronisation et mise à jour des données de sortie
162 -- / donnée d'entrée
163 P_data_out : process(Result_clk, Result_rst, Count_event_16_40,
163 Etat_present_Result)
164 begin
165     if Result_rst = '1' then
166         Result_data_out_tmp <= (others => '0');
167     elsif Result_clk'event and Result_clk = '1' then
168         Result_data_out_tmp(Indice) <= Result_data_in_tmp;
169         Result_data_in_tmp <= Result_data_in;
170     end if;
171 end process P_data_out ;
172 -- Récupération de l'indice

```

```
173     Indice <= conv_integer(unsigned(Count));
174
175     -- Mise à jour du bus de sortie
176     Result_data_out <= Result_data_out_tmp when Result_end_tmp = '1' else
177     (others => '0');
178 end Behavioral;
```

## 7.4 Annexe III : Programme VHDL de l'interface série TDC-PC

```
001 architecture Behavioral of ResTDC_2_PC is
002
003     -- Lecteur TDC
004     COMPONENT ResultAD
005     PORT(
006         RClk_In          : IN std_logic;
007         R_data_in_AD     : IN std_logic_vector(3 downto 0);
008         R_strobe_AD      : IN std_logic_vector(3 downto 0);
009         R_rst            : IN std_logic;
010         R_ce             : IN std_logic;
011         R_16_40         : IN std_logic;
012         R_data_A        : OUT std_logic_vector(39 downto 0);
013         R_data_B        : OUT std_logic_vector(39 downto 0);
014         R_data_C        : OUT std_logic_vector(39 downto 0);
015         R_data_D        : OUT std_logic_vector(39 downto 0);
016         R_start_AD     : OUT std_logic_vector(3 downto 0);
017         R_end_AD        : OUT std_logic_vector(3 downto 0);
018         R_TAGbit_AD     : OUT std_logic_vector(3 downto 0);
019         R_clk_out       : OUT std_logic
020     );
021 END COMPONENT;
022
023     -- UART
024     COMPONENT uart
025     PORT (
026         clock            : IN std_logic;
027         reset            : IN std_logic;
028         data_stream_in   : IN std_logic_vector(7 downto 0);
029         data_stream_in_stb : IN std_logic;
030         data_stream_in_ack : OUT std_logic;
031         data_stream_in_ack_fin : OUT std_logic;
032         data_stream_out  : OUT std_logic_vector(7 downto 0);
033         data_stream_out_stb : OUT std_logic;
034         tx               : OUT std_logic;
035         rx               : IN std_logic
036     );
037 END COMPONENT;
038
039     -- Mémoire
040     signal Data_4_28 : Data_2D := (others => (others => '0'));
041     ..
042
043 begin
044
```

```

045     -- Initialisation de la trame
046     -- Header
047     Data_4_28(0) <= x"AA";
048     Data_4_28(1) <= x"AA";
049     -- Délimiteur
050     Data_4_28(7) <= x"33";
051     Data_4_28(8) <= x"33";
052     Data_4_28(14) <= x"33";
053     Data_4_28(15) <= x"33";
054     Data_4_28(21) <= x"33";
055     Data_4_28(22) <= x"33";
056     ...
057     -- Process de mise à jour de la mémoire
058     P_Logic_ES_MemSet : process ( R_end_AD, Etat_present_MemSet,
059     MemEmpty_tmp)
059     begin
060         case Etat_present_MemSet is
061
062             --wait
063             when S_wait =>
064
065                 -- Logique des sorties
066                 MemFull_tmp <= '0';
067                 -- Initialisation
068                 Data_4_28 (2 to 6) <= (x"00",x"00",x"00",x"00",x"00");
069                 Data_4_28 (9 to 13) <= (x"00",x"00",x"00",x"00",x"00");
070                 Data_4_28 (16 to 20) <= (x"00",x"00",x"00",x"00",x"00");
071                 Data_4_28 (23 to 27) <= (x"00",x"00",x"00",x"00",x"00");
072
073                 -- Passage de S_wait ----> S_ResA
074                 if MemEmpty_tmp ='1' then
075                     Etat_futur_MemSet <= S_ResA;
076                 else
077                     Etat_futur_MemSet <= S_wait;
078                 end if ;
079
080             --S_ResA
081             when S_ResA =>
082                 -- Logique des sorties
083                 MemFull_tmp <= '0';
084
085                 if R_end_AD(0) ='1' then
086                     -- Passage de S_wait ----> S_ResA
087                     Etat_futur_MemSet <= S_ResB;
088
089                     -- Récupération de résultat du canal A
090                     Data_4_28(2) <= R_data_A(7 downto 0);
091                     Data_4_28(3) <= R_data_A(15 downto 8);
092                     Data_4_28(4) <= R_data_A(23 downto 16);
093                     Data_4_28(5) <= R_data_A(31 downto 24);
094                     Data_4_28(6) <= R_data_A(39 downto 32);
095                 else
096                     Etat_futur_MemSet <= S_ResA;
097                 end if ;
098             ...
099         end case;

```

```
100     end process P_Logic_ES_MemSet;
101
102     -- Compteur + MmeEmpty
103     Data_uart_p :process (data_stream_in_ack_fin, ResTDC_clk,
103     Etat_present_MemSet ,Etat_present_MemUART    )
104     begin
105         if ResTDC_clk = '1' and ResTDC_clk'event then
106
107             -- Génération de l'indicateur de mémoire pleine
108             MemFull <= MemFull_tmp;
109
110             -- Initialisation de MemEmpty
111             if data_stream_in_ack_fin = '1' then
112                 Count_ram <= Count_ram + 1;
113             end if;
114             if Count_ram = b"11100" then -- 28 = x"1C"
115                 Count_ram <=(others => '0');
116                 End_Ram <= '1';
117             else
118                 End_Ram <= '0';
119             end if ;
120         end if ;
121     end process;
122     data_stream_in <= Data_4_28(conv_integer(Count_ram)) ;
123     ...
124
125     -- Instanciation du lecteur TDC
126     ResTDC: ResultAD PORT MAP(
127         RClk_In => RClk_In,
128         R_data_in_AD => R_data_in_AD,
129         R_strobe_AD => R_strobe_AD,
130         R_rst => R_rst,
131         R_ce => R_ce,
132         R_16_40 => R_16_40,
133         R_data_A => R_data_A,
134         R_data_B => R_data_B,
135         R_data_C => R_data_C,
136         R_data_D => R_data_D,
137         R_start_AD => R_start_AD,
138         R_end_AD => R_end_AD,
139         R_TAGbit_AD => R_TAGbit_AD,
140         R_clk_out =>R_clk_out
141     );
142
143     -- Instanciation du module UART
144     uart_tx_1: uart PORT MAP(
145         clock => ResTDC_clk,
146         reset => R_rst,
147         data_stream_in => data_stream_in,
148         data_stream_in_stb =>data_stream_in_stb ,
149         data_stream_in_ack =>data_stream_in_ack ,
150         data_stream_in_ack_fin =>data_stream_in_ack_fin ,
151         data_stream_out => data_stream_out,
152         data_stream_out_stb =>data_stream_out_stb ,
153         tx =>tx,
154         rx => rx
```

```

155     );
156
157 end Behavioral;

```

## 7.5 Annexe IV : Programme VHDL de l'interface de programmation du TDC

Déclaration des registres de contrôle :

```

01 Generic (   Len_data : positive:= 16;
02             Len_add  : positive:=8;
03             Num_chan : positive:=4
04             );
05
06 type CReg_X0 is record
07     En_ChX_0      : std_logic ;
08     ChX_IP_En_1   : std_logic ;
09     Pol_X_2       : std_logic ;
10     HOffRng0_2_X_79: std_logic_vector(2 downto 0);
11     HOffTm_En_X_10 : std_logic ;
12     PreCon_En_X_13 : std_logic ;
13 end record CReg_X0;
14
15 type CReg4_X0 is array (0 to Num_chan-1) of CReg_X0; -- 4x CReg_X0
16 signal CReg4_X0_AD : CReg4_X0 :=(others => ('0','0','0',
16 (others =>'0'),'0','0'));
17 signal Data_CReg_0_AD : Data_4_Chan := (others => (others => '0')); -- 4 x
18
19 type CReg_X1 is record
20     Arm_sel0_X_14      : std_logic_vector(3 downto 0);
21     ArmgCon_En_X_5     : std_logic ;
22 end record CReg_X1;
23
24 type CReg4_X1 is array (0 to Num_chan-1) of CReg_X1; -- 4x CReg_X0
25 signal CReg4_X1_AD : CReg4_X1 :=(others => ((others =>'0'),'0'));
26 signal Data_CReg_1_AD : Data_4_Chan := (others => (others => '0'));
27
28 type CReg_80 is record
29     Test_En_0      : std_logic ;
30     RESET_1       : std_logic ;
31     DDR_En_2      : std_logic ;
32     Connect_AB_3  : std_logic ;
33     Connect_CD_4  : std_logic ;
34     Rlength_57    : std_logic_vector(2 downto 0) ;
35     RCLK_sel_89   : std_logic_vector(1 downto 0) ;
36     OT_En_10     : std_logic ;
37     RST_OT_ALM_11 : std_logic ;
38     Sync_TS_Pol_12 : std_logic ;
39     Sync_IP_En_13 : std_logic ;
40     PWR_DN_14    : std_logic ;
41     RCLK_En_15   : std_logic ;
42 end record CReg_80;
43
44 signal CReg_80_all : CReg_80 := ('0', '0', '0', '0','0',
45 (others =>'0'),(others =>'0'),'0', '0', '0', '0','0','0');

```

## 7.5. ANNEXE IV : PROGRAMME VHDL DE L'INTERFACE DE PROGRAMMATION DU TDC

---

```
46 signal Data_CReg_80 : std_logic_vector(Len_data-1 downto 0)
46 :=(others =>'0');
47
48 type CReg_81 is record
49     CNT_Rng_12           : std_logic_vector(1 downto 0);
50     Quiet_Mod_3         : std_logic ;
51 end record CReg_81;
52
53 signal CReg_81_all      : CReg_81 := ((others =>'0'), '0') ;
54 signal Data_CReg_81    : std_logic_vector(Len_data-1 downto 0)
54 :=(others =>'0');
```

### Initialisation des registres d'adresse :

```
01 --X0
02 Reg_Add.CR_X0(0) <= x"00";
03 Reg_Add.CR_X0(1) <= x"20";
04 Reg_Add.CR_X0(2) <= x"40";
05 Reg_Add.CR_X0(3) <= x"60";
06 --X1
07 Reg_Add.CR_X1(0) <= x"01";
08 Reg_Add.CR_X1(1) <= x"21";
09 Reg_Add.CR_X1(2) <= x"41";
10 Reg_Add.CR_X1(3) <= x"61";
11 --80
12 Reg_Add.CR_80 <= x"80";
13 --81
14 Reg_Add.CR_81 <= x"81";
```

### Initialisation des registres de données en fonction des modes :

```
...
01 -- Modes
02 p_modes : process (Modes_in_TDC_tmp, Modes_clk )
03 begin
04     if Modes_clk = '1' and Modes_clk'event then
05         CASE Modes_in_TDC_tmp IS
06
07             -- Mode 0 : Initialisation des registres
08             WHEN b"000" =>
09                 -- Initialisation des registres
10                 Data_CReg_0_AD(0) <= (others =>'0');
11                 Data_CReg_1_AD(0) <= (others =>'0');
12                 Data_CReg_80 <= x"0002";
13                 Data_CReg_81 <= (others =>'0');
14
15             -- Mode 1 : Mesure avec 37.5 MHz, 40 bits
16             WHEN b"001" =>
17                 Data_CReg_0_AD(0) <= x"0003";
18                 Data_CReg_1_AD(0) <= x"0000";
19                 Data_CReg_80 <= x"A484"; --DDR ON
20                 Data_CReg_81 <= x"0006";
21
22             -- Mode 2 : Mesure avec 75 MHz, 40 bits
23             WHEN b"010" =>
24                 Data_CReg_81 <= x"0006";
25                 Data_CReg_80 <= x"A480"; --DDR OFF
26                 Data_CReg_1_AD(0) <= x"0000";
```

```

27             Data_CReg_0_AD(0) <=x"0003";
28
29             -- Mode 3 : : Mesure avec 150 MHz, 40 bits
30         WHEN b"011" =>
31             Data_CReg_81 <= x"0006";
32             Data_CReg_80 <= x"A684"; -- DDR ON
33             Data_CReg_1_AD(0) <= x"0000";
34             Data_CReg_0_AD(0) <=x"0003";
35
36             -- Mode 4 : : Mesure avec 300 MHz, 40 bits
37         WHEN b"100" =>
38             Data_CReg_81 <= x"0006";
39             Data_CReg_80 <= x"A680";
40             Data_CReg_1_AD(0) <= x"0000";
41             Data_CReg_0_AD(0) <=x"0003";
42
43             ...
44         WHEN OTHERS =>
45             Data_CReg_0_AD(0) <= x"0000" ;
46             Data_CReg_1_AD(0) <= x"0000";
47             Data_CReg_80 <= x"0000" ;
48             Data_CReg_81 <= x"0000";
49     END CASE;
50 end if;
51 end process;
52 ...

```

**Transmission de contenu des registres à l'interface série :**

```

01 ...
02 when S_wait =>
03     -- Logique des sorties
04     Modes_end <= '0';
05     Modes_out_TDC <= (others =>'0');
06
07     -- Désactivation de la transmission
08     Serial_Read <='0';
09     Serial_Write <='0';
10     Serial_Clk_50_EN <='0';
11     Serial_CE <='0';
12     Serial_RST<= '0';
13     Bit_parral <='0';
14
15     -- Passage de S_wait ----> S_x0
16     if Modes_Start ='1' then
17         Etat_futur_Mode <= S_x0;
18     else
19         Etat_futur_Mode <= S_wait;
20     end if ;
21
22     --S_x0
23 when S_x0 =>
24     -- Logique des sorties
25     Modes_end <= '0';
26     Modes_out_TDC <= Modes_in_TDC_tmp;
27
28     -- Activer la transmission des données au registre x0
29     Serial_Read <='0';

```

```
30     Serial_Write <='1';
31     Serial_Clk_50_EN <='1';
32     Serial_CE <='1';
33     Serial_RST<= '0';
34     Bit_parral <='1';
35
36     -- Transfert de données au circuit TDC
37     Serial_Add <= Reg_Add.CR_X0(0);
38     Serial_Data_to_TDC <= Data_CReg_0_AD(0);
39
40     -- Passage de S_x0 ----> S_x1
41     if Serial_EndCyc ='1' then
42         Etat_futur_Mode <= S_x1;
43     else
44         Etat_futur_Mode <= S_x0;
45     end if ;
46
47     --S_x1
48     when S_x1 =>
49         ...
```

Instantiation de l'interface série :

```
01     SPI1: SerialHost PORT MAP(
02         Serial_Add => Serial_Add ,
03         Serial_Data_from_TDC =>Serial_Data_from_TDC ,
04         Serial_Data_to_TDC =>Serial_Data_to_TDC ,
05         Serial_Read =>Serial_Read ,
06         Serial_Write =>Serial_Write ,
07         Serial_Clk_50 =>Serial_Clk_50 ,
08         Serial_Clk_50_EN =>Serial_Clk_50_EN ,
09         Serial_CE =>Serial_CE ,
10         Serial_RST =>Serial_RST ,
11         Serial_Hdata =>Modes_Hdata ,
12         Serial_EndCyc =>Serial_EndCyc ,
13         Serial_Clk_50_outt =>Modes_Clk_50_out ,
14         Serial_Hstrobee =>Modes_Hstrobe ,
15         Bit_parral =>Bit_parral
16     );
```

## 7.6 Annexe V : Programme VHDL de filtre de lissage

Les paramètres du programme illustré ci-dessous :

- D\_in\_len : Taille de bus d'entrée
- n\_shift :  $\log_2(M)$  égale à 3 pour  $M=8$
- Filter\_len : La taille du filtre (M)
- m\_virgule : Le nombre de bits consacrée à la virgule fixe (précision des résultats)

```
001     ...
002     entity Mean_Filter is
003         Generic(
004             D_in_len           : integer := 40;
```

```
005         n_shift           : integer := 3;
006         Filter_len        : integer := 8;
007         m_virgule         : integer := 0
008     );
009
010     Port (
011         MeanMean_rst       : in  STD_LOGIC;
012         MeanMean_clk       : in  STD_LOGIC;
013         MeanMean_ce        : in  STD_LOGIC;
014         MeanMean_D_outReady : out STD_LOGIC;
015         MeanMean_D_in      : in  STD_LOGIC_VECTOR (
016             D_in_len -1 downto 0);
017         MeanMean_D_out     : out STD_LOGIC_VECTOR (
018             (n_shift + D_in_len + m_virgule -1) downto 0)
019     );
020 end Mean_Filter;
021
022 architecture Behavioral of Mean_Filter is
023     ...
024 begin
025
026     -- Récupération de la donnée d'entrée
027     P_D_in : process( MeanMean_rst, MeanMean_clk)
028     begin
029         if MeanMean_rst = '1' then
030             MeanMean_D_in_tmp <= (others => '0') ;
031         elsif MeanMean_clk = '1' and MeanMean_clk'event then
032             MeanMean_D_in_tmp <= MeanMean_D_in;
033         end if;
034     end process;
035
036     -- Compteur des échantillons
037     P_count : process( MeanMean_rst, MeanMean_clk, Indice_tmp)
038     begin
039         if MeanMean_rst = '1' then
040             Indice_tmp <= 0;
041             Count_end <= '0';
042         elsif MeanMean_clk = '1' and MeanMean_clk'event then
043
044             -- Incrimentation du compteur
045             Indice_tmp <= Indice_tmp +1;
046
047             -- Indice fin du compteur & mémorisation
048             -- de l'enceins valeur
049             if Indice_tmp = Filter_len then
050                 Count_end <= '1';
051                 Indice_tmp <= 0;
052             elsif Indice_tmp = 0 then
053                 Count_end <= '0';
054                 Old_val_in <= MeanMean_D_in_tmp;
055             else
056                 Count_end <= '0';
057                 Old_val_in <= Old_val_in;
058             end if;
059         end if;
060     end process;
```

```
061
062     -- Calcul de résultat
063     P_calcul_mean : process( MeanMean_rst, MeanMean_ce ,
064     Count_end, MeanMean_clk)
065     begin
066         if MeanMean_rst = '1' then
067             -- initialisation
068             MeanMean_D_out_tmp <= (others => '0');
069             MeanMean_old_val <= (others => '0');
070             MeanMean_D_out_shift <= (others => '0');
071
072         elsif MeanMean_clk = '1' and MeanMean_clk'event then
073             if MeanMean_ce = '1' then
074
075                 -- Accumulateur
076                 Data_in_0(D_in_len + m_virgule -1 downto m_virgule )
077                 <= MeanMean_D_in_tmp;
078                 MeanMean_D_out_tmp <= MeanMean_D_out_tmp + Data_in_0 ;
079
080                 -- Initialisation du registre de calcul
081                 -- à la fin du compteur
082                 if Count_end = '1' then
083
084                     -- Mise à sero de la mémoire
085                     MeanMean_D_out_tmp <= (others => '0');
086
087                     -- Calcul de la moyenne (/N)
088                     MeanMean_D_out_shift(MeanMean_D_out_tmp'high
089                     - n_shift downto 0) <= MeanMean_D_out_tmp
090                     (MeanMean_D_out_tmp'high downto n_shift);
091
092                     -- Mémorisation de l'enceins valeur
093                     MeanMean_old_val(MeanMean_D_out_tmp'high
094                     - n_shift downto 0) <= MeanMean_D_out_tmp
095                     (MeanMean_D_out_tmp'high downto n_shift) ;
096
097                 end if;
098             else
099                 -- Initialisation
100                 MeanMean_D_out_tmp <= (others => '0');
101                 MeanMean_old_val <= (others => '0');
102                 MeanMean_D_out_shift <= (others => '0');
103             end if ;
104         end if;
105     end process;
106     MeanMean_D_out <= MeanMean_D_out_shift;
107     MeanMean_D_outReady <= Count_end;
108 end Behavioral;
```

## 7.7 Annexe VI : Programme VHDL de circuit de tri de trois canaux

000 ...

```

001  -- Lecture synchrone des entrées
002  D_in_1_tmp <= D_in_1;
003  D_in_2_tmp <= D_in_2;
004  D_in_3_tmp <= D_in_3;
005
006  In_1_indice_tmp <=      In_1_indice;
007  In_2_indice_tmp <=      In_2_indice;
008  In_3_indice_tmp <=      In_3_indice;
009
010  -- Lecture synchrone des données
011  P_sort: process(Rst, Clk)
012  begin
013      if Rst ='1' then
014          Out_1_tmp <= (others =>'0');
015          Out_2_tmp <= (others =>'0');
016          Out_3_tmp <= (others =>'0');
017
018          Out_1_indice_tmp <= In_1_indice_tmp;
019          Out_2_indice_tmp <= In_2_indice_tmp;
020          Out_3_indice_tmp <= In_3_indice_tmp;
021
022      elsif Clk='1' and Clk'event then
023
024          if (D_in_1_tmp>=D_in_2_tmp) and (D_in_2_tmp>=D_in_3_tmp) then
025              Out_1_tmp <= D_in_1_tmp;
026              Out_2_tmp <= D_in_2_tmp;
027              Out_3_tmp <= D_in_3_tmp;
028
029              -- Indice
030              Out_1_indice_tmp <= In_1_indice_tmp;
031              Out_2_indice_tmp <= In_2_indice_tmp;
032              Out_3_indice_tmp <= In_3_indice_tmp;
033
034
035          elsif (D_in_1_tmp>=D_in_3_tmp) and (D_in_3_tmp>=D_in_2_tmp) then
036              Out_1_tmp <= D_in_1_tmp;
037              Out_2_tmp <= D_in_3_tmp;
038              Out_3_tmp <= D_in_2_tmp;
039
040              Out_1_indice_tmp <= In_1_indice_tmp;
041              Out_2_indice_tmp <= In_3_indice_tmp;
042              Out_3_indice_tmp <= In_2_indice_tmp;
043
044
045          elsif (D_in_2_tmp>=D_in_1_tmp) and (D_in_1_tmp>=D_in_3_tmp) then
046              Out_1_tmp <= D_in_2_tmp;
047              Out_2_tmp <= D_in_1_tmp;
048              Out_3_tmp <= D_in_3_tmp;
049
050              Out_1_indice_tmp <= In_2_indice_tmp;
051              Out_2_indice_tmp <= In_1_indice_tmp;
052              Out_3_indice_tmp <= In_3_indice_tmp;
053
054          elsif (D_in_2_tmp>=D_in_3_tmp) and (D_in_3_tmp>=D_in_1_tmp) then
055              Out_1_tmp <= D_in_2_tmp;
056              Out_2_tmp <= D_in_3_tmp;

```

```
057         Out_3_tmp <= D_in_1_tmp;
058
059         Out_1_indice_tmp <= In_2_indice_tmp;
060         Out_2_indice_tmp <= In_3_indice_tmp;
061         Out_3_indice_tmp <= In_1_indice_tmp;
062
063     elsif (D_in_3_tmp >= D_in_1_tmp) and (D_in_1_tmp >= D_in_2_tmp) then
064         Out_1_tmp <= D_in_3_tmp;
065         Out_2_tmp <= D_in_1_tmp;
066         Out_3_tmp <= D_in_2_tmp;
067
068         Out_1_indice_tmp <= In_3_indice_tmp;
069         Out_2_indice_tmp <= In_1_indice_tmp;
070         Out_3_indice_tmp <= In_2_indice_tmp;
071
072
073     elsif (D_in_3_tmp >= D_in_1_tmp) and (D_in_2_tmp >= D_in_1_tmp) then
074         Out_1_tmp <= D_in_3_tmp;
075         Out_2_tmp <= D_in_2_tmp;
076         Out_3_tmp <= D_in_1_tmp;
077
078         Out_1_indice_tmp <= In_3_indice_tmp;
079         Out_2_indice_tmp <= In_2_indice_tmp;
080         Out_3_indice_tmp <= In_1_indice_tmp;
081     else
082         Out_1_tmp <= (others => '0');
083         Out_2_tmp <= (others => '0');
084         Out_3_tmp <= (others => '0');
085
086         Out_1_indice_tmp <= In_1_indice_tmp;
087         Out_2_indice_tmp <= In_2_indice_tmp;
088         Out_3_indice_tmp <= In_3_indice_tmp;
089     end if;
090 end if;
091 end process;
092
093 -- Affectation des sorties
094 Out_1 <= Out_1_tmp;
095 Out_2 <= Out_2_tmp;
096 Out_3 <= Out_3_tmp;
097
098 -- Indices après le tri
099 Out_1_indice <= Out_1_indice_tmp;
100 Out_2_indice <= Out_2_indice_tmp;
101 Out_3_indice <= Out_3_indice_tmp;
102
103 -- Génération synchrone de l'indicateur des résultats
104 P_d_out_ready: process(Rst, Clk)
105 begin
106     if Rst = '1' then
107         Data_ready_tmp <= '0';
108     elsif Clk = '1' and Clk'event then
109         Data_ready_tmp <= not(Data_ready_tmp);
110     end if; 112 ...
```

## 7.8 Annexe VII : Programme VHDL de circuit de calcul des écarts d'ajustement

Les paramètres du programme :

- Num\_T\_dxi : Taille des bus d'E/S (M)
- NumLinBit : Nombre de bits pour les canaux (N)
- NumAntenne : Nombre des canaux  $2^N$

```

001  ...
002  entity OffsCompute is
003      Generic(
004          Num_T_dxi : positive:= 40;
005          NumLinBit : positive:= 3;
006          NumAntenne: positive:= 8
007      );
008  Port (
009      clk           : in  STD_LOGIC;
010      ce            : in  STD_LOGIC;
011      rst           : in  STD_LOGIC;
012      T_in_i        : in  STD_LOGIC_VECTOR
013      (Num_T_dxi-1 downto 0);
014      Num_ligne     : in  STD_LOGIC_VECTOR
015      (NumLinBit -1 downto 0);
016      endCyc        : out STD_LOGIC;
017      startCyc      : out STD_LOGIC;
018      Out_test      : out STD_LOGIC_VECTOR (Num_T_dxi -1 downto 0);
019      d_x_i         : out STD_LOGIC_VECTOR (Num_T_dxi-1 downto 0));
020  end OffsCompute;
021  architecture Behavioral of OffsCompute is
022  ...
023  begin
024  P_Count: process(ce, rst, clk)
025  begin
026  if rst ='1' then
027      Count_T <= (others =>'0');
028      endCyc_tmp <='0';
029  elsif clk='1' and clk'event then
030      if ce ='1' then
031          Count_T <=Count_T + 1;
032          -- Fin cycle
033          if Count_T = MaxLigne then -- MaxLigne -1
034              Count_T <= (others =>'0');
035              -- Signal de fin de cycle
036              endCyc_tmp <= '1' ;
037          else
038              endCyc_tmp <='0';
039          end if ;
040          -- Début de cycle
041          if Count_T = 2 then -- MaxLigne -1
042              startCyc_tmp <= '1' ;
043          else
044              startCyc_tmp <='0';
045          end if ;

```

## 7.8. ANNEXE VII : PROGRAMME VHDL DE CIRCUIT DE CALCUL DES ÉCARTS D'AJUSTEMENT

---

```
044         else
045             Count_T <= (others =>'0');
046             endCyc_tmp <='0';
047         end if ;
048     end if;
049 end process;
050 startCyc <= startCyc_tmp;
051
052 P_T_1: process(ce, rst, clk)
053 begin
054     if rst ='1' then
055         T_in_i_1 <= (others =>'0');
056     elsif clk='1' and clk'event then
057         if ce ='1' then
058             T_in_i_1 <=T_in_i;
059         else
060             T_in_i_1 <= (others =>'0');
061         end if ;
062     end if;
063 end process;
064
065 P_d_x_1: process(ce, rst, clk ,endCyc_tmp )
066 begin
067     if rst ='1' or endCyc_tmp='1' then
068         d_x_i_1 <= (others =>'0');
069     elsif clk='1' and clk'event then
070         if ce ='1' then
071             d_x_i_1 <=d_x_i_InOut;
072         else
073             d_x_i_1 <= (others =>'0');
074         end if ;
075     end if;
076 end process;
077 endCyc <= endCyc_tmp;
078
079 -- Calcul
080 d_T_in <= T_in_i_1 - T_in_i;
081 d_x_i_InOut <= d_T_in + d_x_i_1;
082
083 -- Résultats de sortie
084 P_dataOut: process(rst, clk ,endCyc_tmp )
085 begin
086     if rst ='1' then
087         Out_test <= (others =>'0');
088         d_x_i <= (others =>'0');
089     elsif clk='1' and clk'event then
090         if endCyc_tmp='1' then
091             d_x_i <= (others =>'0');
092         else
093             Out_test <= d_x_i_InOut + T_in_i + 0;
094             if d_x_i_InOut(d_x_i_InOut'high)='1' then
095                 d_x_i <= d_x_i_InOut + 0; -- abs(0)
096             end if ;
097         end if ;
098     end if;
099 end process;
```

```
100 end Behavioral;
```

## 7.9 Annexe VIII : Programme VHDL de circuit de transfert des écarts d'ajustement

```
01 ...
02 -- Définition de la période de synchronisation
03 P_PeriodSync : process(Clk, PeriodSync_tmp)
04 begin
05     if Clk = '1' and Clk'event then
06         case PeriodSync_tmp is
07             when "000" =>
08                 Clk_PeriodSync <= Count(0);
09             when "001" =>
10                 Clk_PeriodSync <= Count(1);
11             when "010" =>
12                 Clk_PeriodSync <= Count(2);
13             when "011" =>
14                 Clk_PeriodSync <= Count(3);
15             when "100" =>
16                 Clk_PeriodSync <= Count(4);
17             when "101" =>
18                 Clk_PeriodSync <= Count(5);
19             when "110" =>
20                 Clk_PeriodSync <= Count(6);
21             when "111" =>
22                 Clk_PeriodSync <= Count(7);
23
24             when others =>
25                 Clk_PeriodSync <= Count(23);
26         end case;
27     end if;
28 end process P_PeriodSync;
29 ...
30 -- Registre d'état
31 P_Logic_ES_Mode : process (Clk_PeriodSync,
32 Etat_present_Adjust, Start)
33 begin
34     if Clk_PeriodSync'event and Clk_PeriodSync = '1' then
35         case Etat_present_Adjust is
36             ----- Wait
37             when S_wait =>
38                 LEN_DL_out_tmp <= x"f";
39                 dxi_DL_out_tmp <= x"000";
40                 EndCyc_tmp <= '0';
41
42                 if Start = '1' then
43                     Etat_futur_Adjust <= S_A;
44                 else
45                     Etat_futur_Adjust <= S_wait;
46                 end if ;
47             ----- CHA
48             when S_A =>
```

## 7.9. ANNEXE VIII : PROGRAMME VHDL DE CIRCUIT DE TRANSFERT DES ÉCARTS D'AJUSTEMENT

---

```
49         LEN_DL_out_tmp <= "0001";
50         dxi_DL_out_tmp <= dxi_A;
51         EndCyc_tmp <= '0';
52
53         if Start = '0' then
54             Etat_futur_Adjust <= S_wait;
55         else
56             Etat_futur_Adjust <= S_B;
57         end if ;
58         ----- CHB
59     when S_B =>
60         LEN_DL_out_tmp <= "0010";
61         dxi_DL_out_tmp <= dxi_B;
62         EndCyc_tmp <= '0';
63
64         if Start = '0' then
65             Etat_futur_Adjust <= S_wait;
66         else
67             Etat_futur_Adjust <= S_C;
68         end if ;
69         ----- CHC
70     when S_C =>
71         LEN_DL_out_tmp <= "0100";
72         dxi_DL_out_tmp <= dxi_C;
73         EndCyc_tmp <= '0';
74
75         if Start = '0' then
76             Etat_futur_Adjust <= S_wait;
77         else
78             Etat_futur_Adjust <= S_D;
79         end if ;
80         ----- CHD
81     when S_D =>
82         LEN_DL_out_tmp <= "1000";
83         dxi_DL_out_tmp <= dxi_D;
84         EndCyc_tmp <= '1';
85
86         if Start = '0' then
87             Etat_futur_Adjust <= S_wait;
88         else
89             Etat_futur_Adjust <= S_A;
90         end if ;
91     when others =>
92         LEN_DL_out_tmp <= LEN_DL_out_tmp;
93         Etat_futur_Adjust <= Etat_futur_Adjust;
94         EndCyc_tmp <= '0';
95     end case;
96 end if;
97 end process;
98 ...
```

# Bibliographie

- [1] Stanley C. Solomon. Optical aeronomy. U.S. National Report to the International Union of Geodesy and Geophysics, 1987 1990, 1991.
- [2] Travis A. Rector, Zoltan G. Levay and Lisa M. Frattare. The aesthetics of astrophysics : how to make appealing color-composite images that convey the science. astro-ph.IM, 2017.
- [3] I. Myserlis. A low frequency radio telescope first results of solar observations. Aristotle University of Thessaloniki, pages Page 1, Figure 1.1, 2011.
- [4] Astronomer's Toolbox, NASA. The electromagnetic spectrum. National Aeronautics and Space Administration, Goddard Space Flight Center, 2013.
- [5] Princeton University Press. An introduction to radio astronomy. Figure 1.1.
- [6] A. Wright. Single-dish radio astronomy. Astronomy Online course at Swinburne University of Technology, 2004.
- [7] M. Garrett. Introduction to radio interferometers. ERIS School - Bonn, ASTRON-Swinburne, 2007.
- [8] A. Crossley, A. Peck, J. Braatz and A. Bemis. Introduction to radio interferometry. National Radio Astronomy Observatory, Atacama Large Millimeter/submillimeter Array, 2015.
- [9] C. Caroublos. Sztrsauts radio solaires. Journal de Physique Colloque C2, supplément au no 11-12, Tome 34, pages C2-63, 1973.
- [10] F. Van Den Bosch. Chapter 27 - continuum emission mechanisms. Yale University - Cosmology, large scale structure and galaxy.
- [11] MIT Haystack Observatory. The hydrogen 21-cm emission line. Demystifying Scientific Data : RET 2006, Rev 2, 2006.
- [12] MIT Haystack Observatory. 21 cm line of atomic hydrogen. American Journal of Physics 62(12) :1077-1081 û December, 1994.
- [13] A. Paris. Hydrogen line observations of cometary spectra at 1420 mhz. The Center for Planetary Science, Volume 103 no2, Summer, 2017.
- [14] J. Robert, JR. Thornton, and A. Stockton. Optical and near-infrared spectroscopy of cygnus a. The Astronomical Journal, 118 : 1461E1467, 1999.
- [15] M. Elmouttie, R. F. Haynes and Others. The radio continuum structure of the edge-on spiral galaxy ngc 4945. R. Astron. Soc. 284, 830-838, 1997.

- [16] D. Neufeld, F. P. Goldsmith and Others. Search for interstellar lii in the milky way. astro-ph.GA 10 Feb, 2017.
- [17] F. Boulanger, T. Onaka and Others. Near-infrared spectroscopy of interstellar dust. EAS Publications Series, 46 399-405, 2011.
- [18] W. A. Stein and E. P. Ney. Continuum electromagnetic radiation from solar flares. Journal of Geophysical Research Volume 68, Issue 1, 2012.
- [19] V. Berk, T. Gordon Richards, A. Bauer, and al. Composite quasar spectra from the sloan digital sky survey. The Astronomical Journal 122 (2), 2001.
- [20] K.-L. Klein and A. Kerdraon. Solar physics at nançay radio observatory : Recent developments. IEEE General Assembly and Scientific Symposium, 2011.
- [21] C. Mercier and G. Chambe. Electron density and temperature in the solar corona from multifrequency radio imaging. Astronomy and Astrophysics 583, A101, 2015.
- [22] K.-L. Klein, S. Krucker, G. Lointier, and A. Kerdraon. Open magnetic flux tubes in the corona and the transport of solar energetic particles. Astronomy and Astrophysics 486, 589 596, 2008.
- [23] A. Kerdraon R. Weber, P. Coloc and A. Lecacheux. Rfi mitigation methods in low frequency bands for radio astronomy. Journée du CNFRS, page Métrologie et capteurs en électromagnétisme, 2004.
- [24] D. E. Taylor. Efficient implementation of cross-correlation in hardware. Norwegian University of Science and Technology, pages Tableau 2.1, page 4, 2014.
- [25] M. Pick, J-M Malherbe and A. Kerdraon. On the disk ha and radio observations of the 2003 october 28 flare and coronal mass ejection event. The Astrophysical Journal, 631 :L97 L100, page The American Astronomical Society, 2005.
- [26] ASTRON. Radio observatory-lofar science/lofar papers/an overview of lofar publications for the period of 2011 - 2017. LOFAR PAPERS, July 2017.
- [27] R. A. Fallows, W. A. Coles, D. McKay-Bukowski, and Others. Broadband meter-wavelength observations of ionospheric scintillation. Journal of GeophysicalResearch : Space Physics, 2014.
- [28] P. Zarka. Search of radio transient counterparts for gravitational waves with lofar. LESIA, IVeme Ecole de Physique des Astroparticules-Ondes gravitationnelles et Multi-Messagers-, OHP, 2013, 2013.
- [29] J. N. Girard, P. Zarka, C. Tasse, S. Hess, and the LOFAR Collaboration. Jupiter synchrotron imaging with lofar. Societee Francaise d’Astronomie et d’Astrophysique (SF2A) 2012, pages SF2A–2012 : Proceedings of the Annual meeting of the French Society of Astronomy and Astrophysics. Eds. : S. Boissier, P. de Laverny, N. Nardetto, R. Samadi, D. Valls–Gabaud and H. Wozniak, pp.681–686, 2012.
- [30] C. Vogt and Dwingeloo. A science case for an extended lofar. ASTRON and The Netherlands, 2006.
- [31] Philippe Zarka. Etudes planétaires et exoplanétaires avec le radiotélescope géant lofar. LESIA, USN, Observatoire de Paris, CNRS, PSL, Meudon.
- [32] the Low Frequency Array ASTRON, LOFAR. Netherlands institute for radio astronomy. [https ://www.astron.nl/general/lofar/lofar](https://www.astron.nl/general/lofar/lofar).

- [33] AD9430. 12 bit, 170/210 msp/s, 3.3 v a/d converter. Analog Devices, 2010.
- [34] A.A. Konovalenko. Low frequency radio astronomy with the existing and future radio telescopes. LOFAR meeting, APC Laboratory, Paris, 17-18 January, pages Institute of Radio Astronomy, Kharkov, Ukraine, 2008.
- [35] A.J.Boonstra, J.D. Bregman and A.A. Mohamoud. Lofar spectrum monitoring : dynamic range and spectral occupancy issues. ASTRON, Foundation for Research in Astronomy, The Netherlands, page 200, Enschede School of Engineering, The Netherlands.
- [36] ASTRON. Lofar signal path. Radio Observatory/LOFAR System Capabilities/LOFAR Signal Path.
- [37] Jurgen Morawietz and Menno Norden. The lofar receiver unit, from astronomical requirement to prototype. Page 10, 2005.
- [38] M.J. Norden and G.W. Kant. Rcu ii signal analysis and specification. ASTRON LOFAR-ASTRON-SRS-014.
- [39] M. de Vos. Lofar phase i baseline specification. ASTRON LOFAR-ASTRON-MEM-145.
- [40] W.Lubberhuizen and E. Kooistra. Rsp firmware functional specification. ASTRON LOFAR-ASTRON-SDD-001.
- [41] Integrated Device Tehnology. 3,3v cmos buffer clock driver. IDT49FCT3805 A.
- [42] A. van Houwelingen and G. Schoonderbeek. Detailed description of clock and sync. LOFAR-ASTRON-RPT-057.
- [43] Frequency Standards FS725. Benchtop rubidium frequency standard. Stanford Research Systems.
- [44] Andre W. Gunst (R, D), and G. Schoonderbeek. Update on pps synchronization, lofar status meeting. Stanford Research Systems, page Netherlands Institute for Radio Astronomy, ASTRON is part of the Netherlands Organisation for Scientific Research (NWO) 1.
- [45] P.J. Hall. The square kilometre array : an engineering perspective. International SKA Project Office, Dwingeloo, The Netherlands, pages ISBN 1-4020-3797-x, 2004.
- [46] G.W. Kant, D. Parbhu Patel and al. Electronic multi-beam radio astronomy concept : Embrace the european demonstrator for the ska program. ASTRON, 2004.
- [47] SKA Program Development Office. The square kilometre array. Jodrell Bank Centre for Astrophysics, 2011.
- [48] Richard T. Schilizzi, Peter E. F. Dewdney and al. The square kilometre array. Proc. of SPIE Vol. 7012 70121I-1, 2008.
- [49] J. Cordes. The square kilometre array. Project Description for Astro2010 US SKA Consortium, 2009.
- [50] S. Torchinsky. Dense aperture array for ska. SKA-EMBRACE, AAMID All-Hands Meetings, ASTRON, 2014.
- [51] Jader Monari. Embrace receiver. 8th SKADS Board Meeting, 2009.
- [52] G. Swarup, S. A. thakrishnan, V. K. Kapahi and V. K. Kulkarni. The giant metre-wave radio telescope. Current Science Vol 60 NO 2 25, 1991.
- [53] G. Swarup. Giant metrewave radio telescope (gmrt). Radio Interferometry Theory Techniques and Applications, 1991.

- [54] National Center for Radio Astrophysics. Giant metrewave radio telescope. Tata Institute of Fundamental Research, nra.tifr.res.in.
- [55] D. Anish Roshi. Chapter 9 correlator ii implementationr. Radio Interferometry : Theory Techniques and Applications, 1997.
- [56] D. Anish Roshi. Chapter 25 - the gmrt correlator. Radio Interferometry : Theory Techniques and Applications, 1997.
- [57] H. Reddy Suda and P. Ku mar Gupta. Powering real-time radio astronomy signal processing with gpus. NCRA, NVIDIA, India, 2010.
- [58] Rabi Swain and D. Anish Roshi. Direct digital synthesizer based clock/synchronization signal generation circuit for the gmrt correlator. Internal Technical Report, 1999.
- [59] Analog Device. Ad9851 cmos 180 mhz dds/dac synthesizer. Internal Technical Report, 1999.
- [60] Richard E. Hills and Anthony J. Beasley. The atacama large millimeter/submillimeter array. ALMA, Av El Golf 40, Piso 18, Santiago, Chile, 1995.
- [61] A. Wootten , K. Y. Lo and al. The atacama large millimeter/submillimeter array development program. Activity White Paper for Astro2010 Program Prioritization Panel, 2010.
- [62] Larry R. D’Addario. Timing and synchronization. ALMA Memo No. 298, 2000.
- [63] R. Perley, P. Napier, J. Jackson, B. Butler, B. Carlson and al. The expanded veryz large array. Proceedings of the IEEE | Vol. 97, No. 8, 2009.
- [64] R. Perley. The expanded very large array new capabilities for new science. National Radio Astronomy Observatory NRAO, 2012.
- [65] D. Backer, A. Bauermeister and al. The allen telescope array the first widefield, panchromatic, snapshot radio camera. Radio Astronomy Laboratory, University of California Berkeley CA, USA, 2009.
- [66] W.L. Urry. The ata imager update. URSI Meeting, Boulder, CO, 2004.
- [67] S.W. Ellingson, T.E. Clarke and al. The long wavelength array. Proceedings of IEEE, 2009.
- [68] G. B. Taylor. Science with the long wavelength array. The Metrewavelength Sky ASI Conference Series, 2014, Vol. 13, pp 349 354, 2014.
- [69] A. Konovalenko and al. The modern radio astronomy network in ukraine utr-2, uran and gurt. Exp Astron Springer, 2016.
- [70] Zakharenko V et al. Digital receivers for low-frequency radio telescopes utr-2, uran, gurt. Journal of Astronomical Instrumentation, World Scientific Publishing Company, 2017.
- [71] A. A. Stanislavsky et al. Heliograph of the utr-2 radio telescope. 2011.
- [72] J. Nieminen, L. Qian and R. Jantti. Network-wide time synchronization in multi-channel wireless sensor networks. Wireless Sensor Network, 2011, 3, 39-53, February 2011.
- [73] D. Djenouri and M. Bagaa. Synchronization protocols and implementation issues in wireless sensor networks : A review. IEEE Systems Journal, 10(2) :617–627, June 2016.
- [74] Bae, Shi-Kyu,, Chen, and Hsing-Chung. Performance evaluation of time synchronization protocols for wireless sensor networks. Springer Netherlands, pages 651–658, 2012.

- [75] C. Luong Nguyen, A. Mokraoui, P. Duhamel and L. Trung Nguyen. Improved time and frequency synchronization algorithm for 802.11a wireless standard based on the signal field. Journal on Electronics and Communications, IEEE, pages pp.40–49, 2013.
- [76] W-H.Tseng, S. Siu, S-Y. Lin, and C-S. Liao. Precise utc dissemination through future telecom synchronization networks. Precise UTC dissemination through future telecom synchronization networks, IEEE, April 2015.
- [77] Norman F. Ramsey. History of atomic clocks. JOURNAL OF RESEARCH of the National Bureau of Standards, Vol. 88, No. 5, September-October 1983, June 21, 1983.
- [78] P. Gill, H. Margolis, A. Curtis, H. Klein, S. Lea, S. Webster, and P. Whibberley. Optical atomic clocks for space. National Physical Laboratory, Hampton Road, Teddington, Middlesex, TW11 0LW, 2008.
- [79] H. Zhang, G. WuHongwei, L. Xinwan Li and J. Chen. High-precision ultralong distance time transfer using single-fiber bidirectional-transmission unidirectional optical amplifiers. IEEE Photonics Journal, High-Precision Ultralong Distance Time Transfer, Volume 8, Number 5, October 2016.
- [80] X. Zhang and J. Ye. Precision measurement and frequency metrology with ultracold atoms. National Science Review, Volume 3, Issue 2, 1 June 2016, Pages 189 200, March 2016.
- [81] E. Garone, A. Gasparri and F. Lamonaca. Clock synchronization protocol for wireless sensor networks with bounded communication delays. Automatica Volume 59, September 2015, Pages 60-72, 2015.
- [82] D.Fober, S. Letz and Y. Orlarey. Clock skew compensation over a high latency network. page 548 552, 2002.
- [83] K-L. Noh, Q. M. Chaudhari, E. Serpedin and B. W. Suter. Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks. IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 55, NO. 4, APRIL 2007, 2007.
- [84] C. J. Fidge. Timestamps in message-passing systems that preserve the partial ordering. Australian Computer Science Communications, Vol. 10, No. 1, pp. 56-66, February 1988, 1988.
- [85] A. G. Mungall, H. Daams, D. Morris and C. C. Costain. Performance and operation of the nrc primary cesium clock, csv. Springer-Verlag Metrologia, Volume 12, Number 3, 1975.
- [86] S. Peil, Thomas B. Swanson, J. Hanssen and J. Taylor. Microwave-clock timescale with instability on order of 10<sup>-17</sup>. Metrologia, Volume 54, Number 3, 2017.
- [87] L. Lorini, N. Ashby and Others. Recent atomic clock comparisons at nist. EDP Sciences, Springer-Verlag 2008, Eur. Phys. J. Special Topics 163, 19 35 (2008), 2008.
- [88] B. Jun Choi, H. Liang and Others. Dcs : Distributed asynchronous clock synchronization in delay tolerant networks. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, Volume : 23 Issue : 3 Page(s) : 491- 504, June 2011.
- [89] J. Zheng and A. Jamalipour. Wireless sensor networks a networking perspective. John Wiley and Sons, Inc., Hoboken, New Jersey, 2009.
- [90] W. Jiang, A. Kruszewski, J. P. Richard and A. Toguyeni. Networked control and observation for master-slave systems. Springer Science and Business Media, pp.31-54, 2009.

- [91] L. Chen and T. Zhu. Packet delay estimation algorithm against fifo waiting for asymmetric ieee 1588 system. Journal of Computers, VOL. 9, NO. 9, 2014.
- [92] F. Shi, X.Tuo, and Others. Multiple two-way time message exchange (ttme) time synchronization for bridge monitoring wireless sensor networks. Vittorio M. N. Passaro, May 2017.
- [93] B. Marques and M. Ricardo. Synchronization of application-driven wsn. Marques and Ricardo EURASIP Journal on Wireless Communications and Networking (2017) 2017 :37, 2017.
- [94] A. Mahmood, R. Exel and Others. Clock synchronization over ieee 802.11 a survey of methodologies and protocols. IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 13, NO. 2, APRIL 2017, 2017.
- [95] F. Sivrikaya and B. Yener. Time synchronization in sensor networks : A survey. IEEE Network, 2004.
- [96] F. Sivrikaya and B. Yener. Lightweight time synchronization for sensor networks. WSNA '03 Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications Pages 11-19, 2003.
- [97] D. L. Mills. Internet time synchronization : the network time protocol. IEEE Trans. Communications 39, 10 (October 1991), 1482-1493., 1991.
- [98] L. De jun, W. Gang and Others. Ieee 1588 based time synchronization system for a seafloor observatory network. Journal of Zhejiang University Science C, 2013.
- [99] D. Fober, Y. Orlarey and S. Letz. Real time clock skew estimation over network delays. GRAMME Laboratoire de recherche en informatique musicale Technical report TR-050601, 2005.
- [100] M. A. Lombardi, L. M. Nelson and Others. Time and frequency measurements using the global positioning system. 2001.
- [101] Michael A. Lombardi. The use of gps disciplined oscillators as primary frequency standards for calibration laboratories. 2008 NCSL International Workshop and Symposium 1, 2008.
- [102] P. Daly, I. Kitching, D. Allan and T. Pepler. Frequency and time stability of gps and glonass clocks. International Journal of Satellite Communications and Networking Volume 9, Issue 1 January/February 1991 Pages 11 22, 1991.
- [103] Peter Daly, Ian D. Kitching, David W. Allan and Trudi K. Pepler. Frequency and time stability of gps and glonass clocks. International Journal of Satellite Communications and Networking Volume 9, Issue 1 January/February 1991 Pages 11 :22, 1991.
- [104] R. Szplet, Z. Jachna, K. Rozyc and J. Kalisz. High precision time and frequency counter for mobile applications. WSEAS Transactions on Circuits and Systems 9(6) û June 2010, 2010.
- [105] R. Szplet, Z. Jachna, K. Rozyc and J. Kalisz. Portable time interval counter with picosecond precision. Proceedings of the 9th WSEAS Int. Conference on INSTRUMENTATION, MEASUREMENT, CIRCUITS and SYSTEMS ISSN : 1790-5117 53 ISBN : 978-960-474-174-8, 2010.
- [106] EPSON. Comparison of crystal oscillator and si-mems oscillators - a comparison of oscillator performance as required by electronic devices and communications system equipment. White Paper/Technical Note, 2014.

- [107] P. Banerjee , A. Chatterjee and A. Suman. Determination of allan deviation of cesium atomic clock for lower averaging time. INDIAN JOURNAL OF PURE and APPLIED PHYSICS Volume : 45 Issue : 12 Pages : 945-949, 2007.
- [108] W.J. Riley. Handbook of frequency stability analysis. NIST Special Publication 1065, 2008.
- [109] Pedro Manuel Oliveira Fernandes Moreira. Timing signals and radio frequency distribution using ethernet networks for high energy physics applications. Department of Electrical and Electronic Engineering University College of London, 2014.
- [110] Michael A. Lombardi. The use of gps disciplined oscillators as primary frequency standards for calibration laboratories. NCSL International Workshop and Symposium, 2008.
- [111] Gaetano Mileti et al. Laser-pumped rubidium frequency standards : New analysis and progress. IEEE Journal of Quantum Electronics, vol. 34, no 2, 1998.
- [112] Lindon L. Lewis. An introduction to frequency standards. Proceedings of the IEEE, vol. 79, no 7, 1991.
- [113] J. Kalisz. Review of methods for time interval measurements with picosecond resolution. 2004 BIPM and IOP Publishing Ltd Printed in the UK, 2003.
- [114] S. Grzelak, M.Kowalski, J. Czokow and M.Zielinski. High resolution time interval measurements applied to flow measurement. Metrol Meas. Syst. Vol. 21 2014 No. 1 pp. 77 : 84., 2014.
- [115] G. S. Jovanovic and M. K. Stojcev. Verniers delay line time to digital converter. Scientific Publications of the State University of Novi Pazar ser a Appl Math Inform and Mech vol 1, 1 2009, 11 :20, 2009.
- [116] C. Hsin Lin and M. Syrzycki. Single stage vernier time to digital converter with sub gate delay time resolution. Circuits and Systems, 2011, 2, 365 :371, 2011.
- [117] J. Kalisz, M Pawlowski and R. Pelka. Error analysis and design of the nutt time-interval digitiser with picosecond resolution. Journal of Physics E : Scientific Instruments, Volume 20, Number 11, 1987.
- [118] C-Y. Yang. A high-frequency cmos multi-modulus divider for pll frequency synthesizers. Analog Integr Circ Sig Process (2008) 55 :155 162, 2008.
- [119] K. Thamaraiselvan, C. Gayathri and N. Divya. A high speed cmos parallel counter using pipeline partitioning. International Journal of Engineering Research Volume No.2, Issue No. 8, pp : 491-495, 2013.
- [120] K-H. Cheng, C-L. Hung and C-W. Su. A sub-1v low-power high-speed static frequency divider. Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on, IEEE, 2007.
- [121] M. Rezvanyvardom, T. Ghanavati Nejad and E. Farshidi. A 5-bit time to digital converter using time to voltage conversion and integrating techniques for agricultural products analysis by raman spectroscopy. INFORMATION PROCESSING IN AGRICULTURE 1 (2014) 124 130, 2014.
- [122] T.Xia and J-C. Lo. Time-to-voltage converter for on-chip jitter measurement. IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 52, NO. 6, DECEMBER 2003, 2003.

- [123] J. Angevare, K. Blutman, A. Zjajo and N. Van Der Meijs. A cmos 0.23pj freeze vernier time-to-digital converter. 2013 NORCHIP, Vilnius, 2013, pp. 1-4., 2013.
- [124] C-T. Ko, K-P. Pun and A. Gothenberg. Vernier parallel delay-line based time-to-digital converter. Springer Science+Business Media, LLC 2011 Analog Integr Circ Sig Process (2012) 71 :151 153, 2012.
- [125] F. Cristian. Probabilistic clock synchronization. Springer-Verlag 1989 Distributed Computing (1989) 3 : 146-158, 1989.
- [126] D. Adithya C. Varma, P. Kumar Reddy and Gopinath. Performance comparison of physical clock synchronization algorithms. P K Reddy. M et al Int. Journal of Engineering Research and Applicationas ISSN : 2248-9622, Vol. 3, Issue 5, Sep-Oct 2013, pp.1355-1364, 2013.
- [127] T. Kunz and E. McKnight-MacNeil. Performance comparison of physical clock synchronization algorithms. International Journal of Autonomous and Adaptive Communications Systems 7(1-2) :128-150 û January 2014, 2014.
- [128] M. Maroti, B. Kusy, G. Simon and A. Ledecz. The flooding time synchronization protocol. Conference : Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys 2004, Baltimore, MD, USA, November 3-5, 2004, 2004.
- [129] C. Lenzen, P.p Sommer and R. Wattenhofer. Pulsesync : an efficient and scalable clock synchronization protocol. IEEE/ACM Transactions on Networking ( Volume : 23, Issue : 3, June 2015 ), 2014.
- [130] Texas Instrument. An-1728 iee 1588 precision time protocol time synchronization performance. Application Report, April 2013, 2012.
- [131] H. Cho, J. Jung, B. Cho and Others. Precision time synchronization using iee 1588 for wireless sensor networks. International Conference on Computational Science and Engineering, 2009.
- [132] I. Freire, I. Sousa and A. Klautau. Analysis and evaluation of end-to-end ptp synchronization for ethernet-based fronthaul. Global Communications Conference (GLOBECOM), 2016 IEEE., 2016.
- [133] B. Lv, Y. Huang, T. Li and Others. Simulation and performance analysis of the iee1588 ptp with kalman filtering in multi-hop wireless sensor networks. JOURNAL OF NETWORKS, VOL. 9, NO. 12, DECEMBER 2014, 2014.
- [134] M.Levesque and D. Tipper. A survey of clock synchronization over packetswitched networks. IEEE Communications Surveys and Tutorials 18(4) :2926 :2947, 2016.
- [135] J. Serrano, P. Alvarez, M. Cattin and Others. The white rabbit project. Proceedings of ICALEPCS2009, Kobe, Japan, 2009.
- [136] J. Serrano, M. Cattin, E. Gousiou and Others. The white rabbit project. Proceedings of IBIC2013, Oxford, UK, 2013.
- [137] G. Gong, S. Chen, Q. Du, J. Li and Y. Liu. Sub-nanosecond timing system design and development for lhaaso project. Proceedings of ICALEPCS2011, Grenoble, France, 2011.
- [138] EF. Dierikx, AE. Wallin and T. Fordell. White rabbit precision time protocol on long-distance fiber links. IEEE Trans Ultrason Ferroelectr Freq Control, Jan 2016, 2016.
- [139] M.J. Norden and G.W. Kant. Rcu ii signal analysis and specification. ASTRON LOFAR-ASTRON-SRS-014, page 10, 2007.

- [140] A. W. Gunst and G. W. Kant. Balancing bits and bandwidth for the a/d converters of lofar. ASTRON, LOFAR-ASTRON-MEM-121, 2003.
- [141] J. D Bregman. Design concepts for a sky noise limited low frequency array. Perspectives on Radio Astronomy Technologies for Large Antenna Arrays (ed. A.B. Smolders and M. P. van Haarlem) - ASTRON, The Netherlands, 1999.
- [142] Kent H. Lundberg and Vin Vq. Analog-to-digital converter testing. Massachusetts Institute of Technology.
- [143] B. XIA. Analog-to-digital interface design for wireless receivers. Office of Graduate Studies of Texas A and M University in partial fulfillment of the requirements, 2004.
- [144] R. Thilak Rajan and M. Bentum. Synchronization for space based ultra low frequency interferometry. IEEEAC Paper 2244, Version v4 , Updated January 6, 2013.
- [145] C.Marque. Environnement coronal des filaments quiescents et eruptifs : observations radio, euv, et coronographiques. HAL Id : tel-00011487, 2006.
- [146] Stephen M. Whitea, Namir E. Kassimb and William C. Ericksonb. Solar radioastronomy with the lofar (low frequency array) radio telescope. Remote Sensing Division, Naval Research Laboratory, Code 7213, Washington D.C.20375-5351, USA, 2009.
- [147] Mike Garrett. Radio astronomy. ASTRON and Leiden, page 60, 2007.
- [148] Gerard Borg. Radiation and antennas. ANU The Australian National University, Embedded Wireless, 2015.
- [149] D. Redmayne, E. Trelewicz and A. Smith. Understanding the effect of clock jitter on high speed adcs. Linear Technology, Design Note 1013.
- [150] T. Neu. Clock jitter analyzed in the time domain, part 1. Texas Instruments Incorporated, Data Acquisition, 2009.
- [151] Z. Kulka. Sampling jitter in audio a/d converters. RCHIVES OF ACOUSTICS 36, 4, 831 849, 2011.
- [152] Todd R. Hunter and Peter J. Napier. Antennas and receivers in radio interferometry. Proceedings of the 2014 Synthesis Imaging Workshop held in Socorro, NM on May 13-20, 2014.
- [153] H. Weibel. High precision clock synchronization according to iee 1588 - implementation and performance issues. Embedded World, pp.22-24, 2005.
- [154] I. Freire, I. Sousa and A. Klautau. Analysis and evaluation of end-to-end ptp synchronization for ethernet-based fronthaul. Global Communications Conference (GLOBECOM), 2016 IEEE, 2016.
- [155] A. Kerdraon and J.-M. Delouis. Coronal physics from radio and space observations. IEEE General Assembly and Scientific Symposium, 2011.
- [156] El Houssain Ait Mansour, K.-L. Klein, Bruno Da Silva et al. Sub-nanosecond time synchronization mechanism for radio interferometer array. 2017 IEEE International Workshop on Measurement and Networking (M and N) 27-29 Sept, 2017.
- [157] G. Gong, S. Chen, Q. Du, J. Li and Y. Liu. Sub-nanosecond timing system design and development for lhaaso project. ICALEPCS2011, Grenoble, France, 2011.
- [158] M. J. Lopez, J. L. Gutiérrez Rivas and J. Daaz Alon. A white-rabbit network interface card for synchronized sensor networks. SENSORS, 2014 IEEE, 2014.

- [159] J. Serrano, M. Cattin and E. Gousiou. The white rabbit project. IBIC2013, Oxford, UK, 2013.
- [160] Texas Instruments. Ths788 quad-channel time measurement unit (tmu). SLOS616D, 2015.
- [161] ON Semiconductor. Mc10ep195, mc100ep195 programmable delay chip.
- [162] IDT Integrated Device Technology. 2 :4, lvpecl output fanout buffer. 8SLVP1204I.
- [163] IDT Integrated Device Technology. 2 :4, lvpecl output fanout buffer. 8SLVP1204I.
- [164] XC6VLX240T. Ic fpga 600 i/o 1156fcbg. Xilinx Inc, 2015.
- [165] Elbert V2. Spartan 3a xc3s50a fpga development board. numato.
- [166] Texas Instruments. 1 :3 lvpecl clock buffer with programmable divider. CDCP1803, 2013.
- [167] Texas Instruments. Ultra-low jitter clock generator with two independent plls, eight outputs and integrated eeprom. LMK03328, 2016.
- [168] ML605. Evaluation board featuring virtex-6 xc6vlx240t-1ffg1156 fpga. Xilinx Inc.

# Table des figures

1.1	Observation des images radio de la couronne solaire le 24 février 2006 des fréquences (de gauche à droite) 150,9 MHz, 228 MHz, 327 MHz, 408 MHz et 432 MHz . . . . .	16
1.2	Images de la couronne solaire au début du cycle solaire pour les deux fréquences 410 MHz et 150 MHz observées le 27 mai 2004 [21]. . . . .	17
1.3	La température de brillance du soleil $T_b$ (MK) en fonction de la fréquence (MHz). A gauche : 7 jours d'observations en 2004 après la maximum du cycle. A droite : 10 jours d'observations en 2008-2009 durant le minimum du cycle solaire [21] . . . . .	17
1.4	La densité d'électron en fonction de la distance au centre du soleil pour les 6 fréquences (comprises entre 150 MHz et 432 MHz) dans les deux directions EW (Est-Ouest) et NS ( Nord-Sud) [21] . . . . .	18
1.5	Contours d'iso-intensité à mi-puissance (iso-contours colorés) de quelques sursauts solaires de type III observés par le Radiohéliographe de Nançay superposés aux magnétogrammes de la photosphère et à l'extrapolation dans la couronne des lignes de champ ouvertes vers l'espace interplanétaire (courbes en blanc). . . . .	19
1.6	Observations des sursauts de types II, III et IV le 3 novembre 2003 aux fréquences discrètes 432 MHz, 410 MHz, 327 MHz, 236 MHz et 164 MHz obtenues par le NRH (en haut) et les observations de spectrographie du soleil entier par OSRA (AIP Potsdam)(en bas) . . . . .	20
1.7	Schéma bloc de la chaîne radiofréquence d'une antenne NS (Nord-Sud) du Radiohéliographe de Nançay . . . . .	23
2.1	Architecture globale du réseau LOFAR : BF (30-80 MHz) d'un diamètre de 60m et HF (110-250 MHz) avec d'un diamètre de 50 m [31]. . . . .	26
2.2	Arborescence du système d'acquisition et du transfert de données de LOFAR . . . . .	29
2.3	Architecture interne de la carte RSP . . . . .	30
2.4	Circuit de distribution d'horloge et de datation du récepteur LOFAR . . . . .	31
2.5	Schéma bloc du générateur FS725 dédié de trois sorties à 10 MHz et trois signaux PPS asservi par une source GPS . . . . .	32
2.6	Le réseau phasé à deux polarisations à la station EMBRACE de Westerbork . . . . .	33
2.7	Architecture du récepteur de la station EMBRACE [51] . . . . .	33
2.8	Location et architecture de l'instrument GMRT [54] . . . . .	35
2.9	Architecture basique du numériseur du GMRT avec 32 MHz de bande [57] . . . . .	35

3.1	Horloges $C_i$ en fonction du temps $t$ avec divers offsets et skews . . . . .	44
3.2	Compensation de l'erreur du "skew" (A) de l'offset (B) d'une horloge dans un système de synchronisation. (1) : Horloge avant compensation et (1') après compensation de l'horloge (1) par rapport à l'horloge de référence (2). $T_{sync}$ la période de resynchronisation du système. . . . .	45
3.3	Schéma basique de mesure de temps . . . . .	49
3.4	Principe de circuit de la mesure du temps . . . . .	51
3.5	Chronogrammes de la mesure du temps d'intervalle suivant la méthode Nutt . . . . .	53
3.6	Schéma bloc du principe du compteur et les chronogrammes des signaux de mesure du temps . . . . .	54
3.7	Schéma de principe de la conversion temps-tension . . . . .	54
3.8	Schéma de principe de la technique de mesure du temps de Vernier . . . . .	55
3.9	Chronogrammes de la technique de mesure de l'intervalle de temps $T_i$ de Vernier . . . . .	56
3.10	Schéma bloc de la méthode lignes à retard numérique . . . . .	56
3.11	Architecture basique d'un système Maître/Esclaves . . . . .	57
3.12	Requête Maître/ Esclave de l'algorithme de Cristian . . . . .	57
3.13	Architecture du protocole de synchronisation PTP . . . . .	60
3.14	DMTD analogique . . . . .	63
3.15	Signaux d'entrées et de sorties du circuit de mesure de phase analogique pour divers rapport de fréquence . . . . .	65
3.16	Retard du filtre passe bas . . . . .	66
3.17	Histogramme du PPS entre le maître et l'esclave . . . . .	67
4.1	$T_{sky}(K)$ en fonction de la fréquence (Hz) (voir la figure 3 dans [139]). La température $T_{sky}(K)$ est proportionnelle à $\lambda^{2.55}$ dans la bande NRH (entre 150 MHz et 450 MHz environ). . . . .	70
4.2	$P_{sky}(W)$ en fonction de la fréquence $f_{low}$ (Hz) pour différentes bandes de fréquence ( $300 MHz \leq B \leq 500 MHz$ ) . . . . .	71
4.3	Le nombre de bits requis $n_r$ par le convertisseur A/D dans les cas du soleil actif (courbe verte) et calme (courbe bleue) en fonction de la fréquence basse $f_{low}$ (Hz). Soleil calme (courbe bleue). . . . .	75
4.4	Le SNR (dB) en fonction de la fréquence $f_{in}$ du convertisseur A/D pour chaque valeur du jitter (ps) . . . . .	76
4.5	Le temps de la gigue $t_j(ps)$ en fonction du nombre de bits $n_s$ pour différents nombres de bits requis $n_r$ par le convertisseur A/D pour une fréquence maximale $f_{in}$ de 450 MHz . . . . .	77
4.6	La gigue maximum en fonction du nombre de bits pour une fréquence maximale de 450 MHz . . . . .	79
4.7	Schéma de principe d'un interféromètre radio . . . . .	80
4.8	L'erreur différentielle $d\tau_g(s)$ en fonction du taux de corrélation exprimé en pour cent . . . . .	83
5.1	Modèle du retard entre le maître et l'esclave (Antenne). . . . .	86

5.2	Les chronogrammes des horloges $Clk_i$ dérivées de l'horloge Maître $Clk_M$ avant (A) et après (B) le processus d'ajustement. A-1, A-2 : Horloges au niveau du maître et des esclaves avant le processus d'ajustement successif B-1, B-2 : horloges au niveau du Maître vu du coté des esclaves après l'ajustement en utilisant le circuit du retard programmable DL avec une valeur égale à $\frac{d_{xi}}{2}$ pour chaque horloge.	89
5.3	Pseudo-code de l'algorithme de synchronisation du temps d'un réseau d'antennes multi-reparties avec la prise en compte de l'asymétrie du retard aller-retour . . .	90
5.4	Topologie en étoile du réseau d'antennes du Radiohéliographe de Nançay avec une distance différente entre le maître et chaque esclave ( $d_{MM}^i \neq d_{MM}^{i+1}$ ). . . . .	92
5.5	Période de synchronisation ( $T_{Sync}$ ) en nombre d'itérations en fonction du nombre d'antennes ( $N$ ) pour différentes valeurs de ( $\mu$ ). $\mu = 10^{-6}$ (courbe rouge), $\mu = 10^{-7}$ (courbe bleue) et $\mu = 10^{-8}$ (courbe verte). . . . .	93
5.6	Allure de l'erreur de l'algorithme $\mu_e(ps)$ après le calcul et l'ajustement des offsets en fonction du nombre d'itérations ( $l$ ) pour $N=100$ et $\mu = 10^{-5}$ . . . . .	94
5.7	Les retards $d_{MS}^i$ , $d_{SM}^i$ et $d_{MM}^i$ en fonction de ( $i$ ) ( $i = 1, 2, \dots, 1000$ ) et leurs distributions dans le temps . . . . .	95
5.8	Les retards au niveau des esclaves après l'ajustement des offsets en utilisant l'algorithme ( $\frac{d_{MM}^i + d_{xi}}{2}$ ) et l'ajustement réel ( $d_{MS}^i + \frac{d_{xi}}{2}$ ) pour une liaison asymétrique ( $d_{MS}^i \sim \mathcal{N}(\mu = 568, 02ns, \sigma = 18, 65ps) < d_{SM}^i \sim \mathcal{N}(\mu = 568, 04ns, \sigma = 15, 56ps) \neq \frac{d_{MM}^i}{2}$ ) au début de la minimisation de l'erreur (première itération) et avec $10^4$ itérations pour $N = 10^3$ , $\mu = 10^{-5}$ . . . . .	96
5.9	Mêmes paramètres de simulation de la figure 5.8 pour une liaison parfaitement symétrique ( $d_{MS}^i \sim \mathcal{N}(\mu = 568.02ns, \sigma = 18.65ps) < d_{SM}^i \sim \mathcal{N}(\mu = 568.04ns, \sigma = 15.56ps) \neq \frac{d_{MM}^i}{2}$ ). . . . .	97
5.10	$\mu_e(ps)$ et l'écart-type $\sigma_e(ps)$ avant et après ajustement en fonction du nombre d'itération ( $l$ ) et du paramètre d'algorithme ( $\mu$ ) pour $N=1000$ . . . . .	98
5.11	Schéma bloc du principe de calibration . . . . .	100
5.12	$\mu(ps)$ et $\sigma(ps)$ de l'erreur de mesure après calibration en fonction de $\alpha$ . . . . .	103
6.1	Schéma bloc du système de mesure et d'ajustement des offsets en temps réel . .	105
6.2	Schéma de principe du système de mesure des intervalles de temps. . . . .	107
6.3	Schéma bloc détaillé du système de mesure et d'acquisitions des données rapides issues du circuit TDC . . . . .	107
6.4	Schéma bloc du générateur des intervalles de temps programmables . . . . .	109
6.5	Schéma bloc du générateur des intervalles de temps programmables . . . . .	109
6.6	Machine à état du générateur des signaux des événements et de synchronisation (Events + Sync) avec le tableau des états des sorties sous la forme $Ev_3Ev_2Ev_1Ev_0Sync$ . . . . .	110
6.7	Entité du générateur des intervalles de temps pour quatre canaux différentiels pour le circuit TDC . . . . .	111
6.8	Chronogramme des signaux Events et Sync pour le mode (1) . . . . .	112
6.9	Chronogramme des signaux Events et Sync pour le mode (2) . . . . .	112
6.10	Schéma bloc du circuit de calcul des écarts d'ajustement . . . . .	114
6.11	Chronogrammes des retards dans un trajet aller-retour . . . . .	115
6.12	Schéma bloc du circuit de calibration . . . . .	116

6.13	Mesures d'intervalles de temps de 1 ns avec une gigue de 120 ps (1 sigma) en fonction de la taille du filtre M. . . . .	118
6.14	La réponse fréquentielle du filtre H(f) en fonction de la fréquence pour divers tailles de filtre M . . . . .	119
6.15	Architecture basique de calcul par bloc de la moyenne . . . . .	119
6.16	Architecture globale de calcul par bloc de la moyenne . . . . .	120
6.17	Entité du filtre de lissage du bruit de mesure . . . . .	121
6.18	Chronogrammes du filtre moyennneur d'une taille égale à 8 . . . . .	121
6.19	Ressources normalisées en fonction de la taille du bus de données à l'entrée . . . . .	123
6.20	Machine à état du soustracteur avec mémoire . . . . .	124
6.21	Entité circuit de compensation de $d_0^i$ . . . . .	124
6.22	Schéma bloc du modèle VHDL du circuit de calibration pour quatre canaux du modèle théorique illustré dans la figure 6.12 . . . . .	125
6.23	Données des canaux A à D avant la calibration pour les deux phases (0 et 1). $d_1^i = 0$ durant la phase 0 (absence du câble) . . . . .	126
6.24	Données des canaux A à D à la sortie du circuit de calibration en utilisant le modèle matlab . . . . .	127
6.25	Données des canaux A à D à la sortie du circuit de calibration en utilisant le modèle VHDL . . . . .	128
6.26	Distribution des données dû canal A pour les modèles Matlab et VHDL dans le régime permanent ( $1.5 \times 10^5 \leq n \leq 4 \times 10^5$ ) . . . . .	129
6.27	Entité du composant de tri de quatre canaux . . . . .	131
6.28	Entité du composant de tri de trois canaux . . . . .	131
6.29	Chronogrammes des retards mesurés et leurs tris pour les quatre canaux . . . . .	132
6.30	Chronogrammes de simulation de la détection de la valeur maximale des quatre canaux de mesures . . . . .	133
6.31	Chronogrammes d'ajustement des offsets des horloges . . . . .	134
6.32	Schéma bloc du circuit de calcul des écarts d'ajustement des horloges . . . . .	135
6.33	Entité du circuit de calcul des écarts d'ajustement des horloges $d_{xi}$ . . . . .	136
6.34	Chronogrammes de simulation de l'entité OffsComp . . . . .	137
6.35	La fréquence maximale de l'architecture (MHz) en fonction de la taille des bus d'E/S pour N=[3, 8 or 16 bits] . . . . .	138
6.36	Nombre de registres Flip-Flop en fonction de la taille des bus d'E/S, M pour N=[3, 8 ou 16 bits]. . . . .	139
6.37	Schéma bloc d'adaptation des résolutions TDC/DTC . . . . .	139
6.38	Comparaison de la sortie $S_3$ entre le calcul avec Matlab et le calcul en virgule fixe . . . . .	140
6.39	Schéma bloc de la partie matérielle sur carte et FPGA du circuit de transfert des écarts de calibration $\frac{d_{xi}}{2}$ . . . . .	141
6.40	Machine à état du contrôleur d'ajustement des offsets des horloges pour quatre canaux . . . . .	142
6.41	Chronogrammes des signaux du circuit de transfert des écarts d'ajustement aux circuits DTC . . . . .	143
6.42	Architecture globale du système de synchronisation . . . . .	143
6.43	Carte du circuit TDC de mesures des retards aller/retour pour quatre canaux . . . . .	144
6.44	Carte du Maître . . . . .	146
6.45	Mise en cascade de 2 cellules DTC . . . . .	148

6.46	Schéma interne du buffer 1 : vers 4 . . . . .	148
6.47	Carte distributrice de l'horloge 1 vers 4 . . . . .	149
6.48	Kit de développement FPGA Vertex-6 . . . . .	150
6.49	Kit de développement FPGA Spartran 3A [165] . . . . .	150
6.50	Schéma bloc de l'esclave dans l'architecture globale du système de synchronisation	151
6.51	Carte esclave (Vue de dessus) . . . . .	152
6.52	Schéma bloc de la PLL . . . . .	153
6.53	Architecture du circuit de mesure et de calibration de la chaîne. . . . .	154
6.54	Architecture matérielle du système. . . . .	154
6.55	Gigue de la sortie différentielle de la PLL intégrée dans la carte maître avec une fréquence de 200 MHz. La distribution en rose indique le skew entre C1 et F1 (50 ps/div, 200 mV/div). La distribution en bleu indique le skew entre C2 et F1 (50 ps/div, 500 mV/div). . . . .	156
6.56	Intervalle de temps $T_i$ entre Start(SYNC) et Stop (Events) du canal A Mode (0) et le "skew" entre les deux signaux au niveau du Maître (Sorties du générateur) (100 ps/div, 1V/div) . . . . .	157
6.57	Intervalle de temps $T_i$ entre Start (SYNC) et Stop (Events) du canal A Mode (0) et le "skew" entre les deux signaux au niveau de l'esclave (100 ps/div, 1V/div)	157
6.58	Distributions des retards $d_{MM}^i(s)$ pour les quatre canaux pour le mode 2 (avant calibration) . . . . .	158
6.59	Distributions des retards de la chaîne $d_0^i(s)$ pour les quatre canaux pour le mode 2	159
6.60	Distribution de l'erreur $e_{MM}^i(s)$ pour les quatre canaux pour le mode 2 après calibration . . . . .	160
6.61	Erreur quadratique moyenne(EQM) avant et après la phase de calibration pour les modes (2, 3 et 4) pour les quatre canaux . . . . .	162
6.62	Gigue de l'horloge de la fréquence de référence du circuit TDC à 200 MHz (20 ps/- div, 500 mV/div) . . . . .	163
6.63	Distribution des retards $d_{MM}^i(s)$ pour les quatre canaux pour le mode 2 (avant calibration) . . . . .	164
6.64	Distribution des retards de la chaîne $d_0^i(s)$ pour les quatre canaux pour le mode 2	165
6.65	Distribution de l'erreur $e_{MM}^i(s)$ pour les quatre canaux pour le mode 2 après calibration . . . . .	166
6.66	Erreur quadratique moyenne(EQM) avant et après la phase de calibration pour les modes (2,3 et 4) pour les quarts canaux . . . . .	167
7.1	Chronogrammes des signaux de l'interface des résultats en mode 40 bits . . . . .	172
7.2	Chronogrammes des signaux de l'interface des résultats en mode 16 bits . . . . .	172
7.3	Machine à état de l'interface d'acquisition des résultats avec deux modes d'uti- lisation (16 ou 40 bits) pour un seul canal . . . . .	173
7.4	Entité du contrôleur d'acquisition de résultats du circuit TDC . . . . .	174
7.5	Simulation de l'interface d'acquisition de données en mode 40 bits . . . . .	175
7.6	Largeurs et allures des signaux en début et fin d'acquisition . . . . .	176
7.7	Simulation de l'interface d'acquisition de données en mode 16 bits . . . . .	176
7.8	Schéma bloc de l'interface de transmission de données entre le circuit TDC et l'ordinateur . . . . .	177

7.9	Schéma bloc de l'interface de transmission de données entre le circuit TDC et l'ordinateur . . . . .	177
7.10	Trame de données de quatre canaux . . . . .	178
7.11	Trame de données de quatre canaux . . . . .	179
7.12	Machine à état de la mise à jour de la mémoire . . . . .	179
7.13	Machine à état du contrôleur de transmission . . . . .	180
7.14	Entité UART . . . . .	181
7.15	Entité de l'interface d'acquisition et de transmission des mesures à l'ordinateur .	182
7.16	Schéma bloc de l'interface de programmation du circuit TDC . . . . .	183
7.17	Chronogrammes de l'interface série en mode écriture . . . . .	184
7.18	Machine à état du mode d'écriture . . . . .	184
7.19	Chronogrammes de l'interface série en mode lecture . . . . .	185
7.20	Machine à état du mode de lecture . . . . .	186
7.21	Machine à état d'adressage des registres de contrôle (Définition du mode d'utilisation) . . . . .	188
7.22	Entité du composant SerialHost) . . . . .	188
7.23	Entité du composant Global TDCModes) . . . . .	189
7.24	Chronogrammes de l'interface série en mode initialisation (Mode 0) . . . . .	191
7.25	Chronogrammes de l'interface série en mode de mesure avec HCLK= 37,5 MHz sur 40 bits (Mode 1) . . . . .	191

# Liste des tableaux

1.1	Source et le type d'émission engendrée . . . . .	16
2.1	Résumé des caractéristiques techniques de l'instrument LOFAR . . . . .	27
2.2	Les bandes de base utilisées par LOFAR . . . . .	28
2.3	Caractéristiques clés de la carte TDS . . . . .	30
2.4	Les spécifications d'EMBRACE . . . . .	34
2.5	Les spécifications de GMRT . . . . .	36
2.6	La bande de base des instruments et la fréquence $F_s$ . . . . .	37
2.7	La résolution des convertisseurs A/D . . . . .	37
2.8	La précision à l'application de l'horloge (Offset) . . . . .	38
3.1	Évolution des standards du temps et de la fréquence . . . . .	51
3.2	Erreur d'offset ( $\mu, \sigma$ )(ns) du protocole PTP/PPS en fonction du type de lien de communication entre les nœuds et la durée de la période de synchronisation . . . . .	61
4.1	La surface effective $A_{eff}$ ( $m^2$ ) pour les antennes du NRH . . . . .	73
4.2	La température $T_s(k)$ du signal en sortie des antennes RH . . . . .	73
4.3	La puissance $P_s(dBm)$ du signal en sortie des antennes RH dans une bande de B=300 MHz cas du soleil calme . . . . .	74
4.4	La puissance $P_s(dBm)$ du signal en sortie des antennes RH dans une bande de B=300 MHz cas du soleil actif . . . . .	74
4.5	Les spécifications du nouveau système numérique du Radiohéliographe de Nancy . . . . .	84
5.1	La précision globale de l'algorithme $\mu_0(ps)$ après l'ajustement des offsets en fonction de $\mu$ et $N$ . . . . .	99
6.1	Le nombre de coups d'horloge durant l'intervalle de temps pour différents canaux	109
6.2	Description des différents signaux du générateur . . . . .	111
6.3	La fréquence maximale d'horloge supportée par le générateur et les ressources en surface en fonction de la technologie de la puce (-1,-2 ou -3) . . . . .	112
6.4	Effet de la taille du filtre M sur la restitution de la mesure réelle . . . . .	118
6.5	Description des signaux du moyeneur . . . . .	121
6.6	Description des signaux du moyeneur . . . . .	124
6.7	Description des signaux du composant de tri de trois canaux . . . . .	130

---

6.8	Description des différents signaux de circuit de calcul des écarts d'ajustement des offsets des horloges . . . . .	136
6.9	Caractéristiques temporelles pour N=3 bits & M=40 bits, Vitesse (-1) . . . . .	137
6.10	Composition de la carte illustrée dans la figure 6.42 . . . . .	145
6.11	Caractéristiques principales du circuit DTC . . . . .	147
6.12	Composition de la carte esclave . . . . .	152
6.13	Intervalles de temps $T_i(\mu s)$ pour les quatre canaux A à D en fonction du mode du générateur de signaux Start et Stop . . . . .	155
6.14	Caractéristiques de la technique proposée. (*) : Précision simulée dans le cas symétrique, (**) : Précision simulée dans le cas asymétrique, (***) : Précision mesurée sans minimisation de l'erreur. . . . .	168
7.1	Description des différents signaux de l'interface d'acquisition de résultat . . . . .	174
7.2	Paramètres temporels et surface occupée en fonction du grade de vitesse de l'interface d'acquisition . . . . .	175
7.3	Description des différents signaux de l'interface d'acquisition des résultats . . . . .	181
7.4	Description des différents signaux de l'interface d'acquisition de résultat . . . . .	182
7.5	Paramètre temporels et surface occupée en fonction du grade de vitesse de l'interface d'acquisition . . . . .	182
7.6	Description des registres de contrôle du composant TDC . . . . .	187
7.7	Description des signaux de SerialHost . . . . .	189
7.8	Description ds signaux de TDCModes . . . . .	190
7.9	Occupation de l'interface série dans la cible FPGA Spartran-3A . . . . .	190
7.10	Occupation de l'interface série dans la cible FPGA Vertex-6 . . . . .	190

# Glossaire

NRH	RadioHéliographe de Nançay
RH	RadioHéliographe
BF	Basse Fréquence
HF	Haute Fréquence
OL	Oscillateur Locale
RF	Radio Fréquence
A/D	Analog-to-Digital
ADC	Analog-to-Digital Converter
LOFAR	LOW Frequency ARray
Gbps	Giga bit per second
ASTRON	Astronomy institute Netherlands Institute Radio Astronomy
GPS	Global Position System
Tb/s	Tera bit per second
CPU	Central Processing Unit
GPU	Graphic Processing Unit
LB	Low Band
HB	High Band
RSP	Remote Station Processor
LBH	Low Band for Higher band
LBL	Low Band for Lower band
FPGA	Field Programmable Gate Array
LCU	Local Control Unit
RCU	ReCeiver Unit
TDS	Timing Distribution Subrack
CEP	CEntral Processor
PPS	Pulse Per Second
TDS	Timing Distribution Subrack
VCXO	Voltage Controlled Crystal Oscillator
PLL	Phase loop Locked
EMBRACE	Electronic Multi-Beam Radio Astronomy ConcEpt
SKA	Square Kilometre Array
Le GMRT	Giant Meter-wave Radio Telescope
FFT	Fast Fourier Transform
ALMA	Atacama Large Millimeter/submillimeter Array

EVLA	Expanded Very Large Array
ATA	Allen Telescope Array
LWA	Leak Wave Antenna
UTR2	Ukranian T-shaped Telescope 2nd geeneration
PRS	Primary Reference Source
PRC	Primary Reference Clock
Rb	Ribudium
TAI	Temps Atomique International
BIH	Bureau International de l'Heure
NTP	Network Time Protocol
EPTMA	Exponential Peak Tolerant Midpoint Algorithm
DUT	Device Under Test
TUR	Test Uncertainty Ratio
UTC	Temps Universel Coordonné
TIC	Time Interval Counter
TDC	Time-to-Digital Converter
LSB	Less Significant Bit
MSB	Most Significant Bit
DTC	Digital-to-Time Converter
DL	Delay chip
CMOS	Complementary Metal-Oxide-Semiconductor
HC	Heure Courante
FTSP	Flooding Time Synchronization Protocol
PTP	Precision Time Protocol
TWTT	Two-Way Time Transfer
WR	White Rabbit
LHAASO	Large High Altitude Air Shower Observatory
DDMTD	Digital Dual Mixer Time Diference
LP	Low Pass
ENOB	Effective Number Of Bits
NS	Nord-Sud
Ext2	Extension 2
SFU	Sun Flux Unit
SNR	Signal Noise Ratio
GEG	Gaussian Error Generator
VHDL	Very High-Level Design Language
FF	Flip-Flop
LUT	LukUp Table
IOB	Input Output Bufer
FIFO	Fisrt In First Out
PC	Personnel Computer
UART	Universal Asynchronous Receiver Transmitter
RCLK	Result Clock
EN	Enable

SPI	Serial Protocol Interface
NECL	Negatif Emitter Coupled Logic
PECL	Positif Emitter Coupled Logic
LVDS	Low Voltage Differential Signaling
LVPECL	Low Voltage PECL
LVC MOS	Low Voltage CMOS
PCIe	Peripheral Component Interconnect Express
DDR3	Double Data Rate 3rd generation
MMCM	Mixed Mode Clock Manager
RAM	Random Access Memory
MAC	Universally Administred Adress
EQM	Erreur Quadratique Moyenne



# Index

## A

Absence du câble 111, 203  
ADC 29, 45, 133  
Algorithm 81, 83, 84, 85  
Algorithme de Berkley 2, 48  
Algorithme de Cristian 2, 47  
Algorithme de synchronisation 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90  
Alimentation 43, 134  
ALMA 29, 30, 31, 32, 193  
Amplitude 55, 56, 71, 103  
Antenne 74, 133, 202, 25, 62, 91  
Applique au NRH 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72  
Astron 191, 192, 193, 198

## B

Bande numérisée 1, 29  
Bits 21, 26, 27, 29, 64, 65, 66, 67, 68, 72, 93, 94, 95, 104, 105, 106, 107, 109, 115, 119, 120, 121, 122, 124, 129, 130, 134, 152, 153, 154, 155, 156, 157, 158, 160, 161, 162, 164, 165, 167, 168, 169, 173, 174, 179, 180, 204, 205, 207  
Bruit 43  
Bruit de circuit de mesure 110  
Buffer 104, 130

## C

Câbles 126  
calcul 91, 99, 126, 182, 187  
Calcul et ajustement 99, 101, 103, 105, 107, 109, 111, 113, 115, 117, 119, 121, 123  
Calibrage 3, 122

Calibration 3, 88, 89, 90, 91, 92, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 126, 139, 141, 142, 144, 146, 147  
Calibration des retards 2, 87, 88, 89, 90  
Calme 62, 64  
Canal 95, 99, 113, 119, 139, 140, 141, 142, 144, 145, 146, 147  
Canaux 93, 109, 110, 111, 112, 113, 116, 119, 120, 124, 157, 204  
Cellules DTC 130, 204  
Chile 193  
Circuit de distribution d'horloge et de datation du LOFAR 24, 25, 201  
Circuit TDC 96, 126, 129, 136  
Clock 43, 44, 45, 46, 93, 126, 176, 178  
CMOS 193  
compteur 57, 104, 177  
Contrôle et acquisition 3, 92, 93, 95, 97  
Conversion temps-tension 2, 43, 45  
Convertisseur 21, 29, 61, 67, 31  
Corrélateur 29

## D

d'ajustement 77, 185  
d'horloge 110, 155  
d'horloges 37, 49  
delays 78, 83, 84  
Description des signaux d'acquisition 154, 161, 208  
Description des signaux du moyennneur 105, 109, 207  
Désynchronisation 42, 94

Des écarts de calibration 123, 204  
 Distribution de l'erreur 141, 146  
 Distributions dans le temps 80, 82, 202  
 DTC 3, 91, 92, 99, 120, 122, 123, 126, 129, 204  
 Délai 111, 112

## E

EMBRACE 1, 25, 30, 31  
 Erreur 72, 123, 142, 147, 51, 207, 78, 83, 84, 85, 90  
 EQM 142, 147, 204, 205  
 Esclaves 3, 47, 49, 74, 78, 91, 92, 125, 127, 134, 202  
 Est-West 12, 201

## F

Filtrage des mesures 3, 102, 105  
 FPGA 29, 92, 93, 94, 123, 126, 135, 156, 199

## B

Galaxies 9  
 GHz 9, 25, 26, 29, 31, 35, 60, 72, 127, 129, 130, 132, 134, 202, 208  
 Gigue d'horloge 72  
 GMRT 1, 30, 31, 193  
 GPS 23  
 GPU 21  
 Générateur 88, 96, 135, 204  
 Générateur des intervalles de temps 3, 93

## H

Haute Fréquence 54, 68  
 Horloges 36, 50, 92, 117, 130

## I

IEEE 49, 194, 195, 197, 198, 199  
 Implémentation 3, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 150  
 Implémentation en VHDL 3, 4, 96, 160, 167  
 Instrumentation du temps 2, 39, 40, 41, 42, 43, 44, 45, 46

Instrumentation et synchronisation 1, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58  
 Interfaces de communication 4, 151, 153, 155, 157, 159, 161, 163, 165, 167  
 Interfaces de de communication 152, 154, 156, 158, 160, 162, 164, 166, 168  
 Intervalle 41, 101, 41, 137, 204

## J

Jitter 26, 29, 65, 126

## K

Kerdrakon 192, 198, 199  
 Kilometre 193  
 Klein 192

## L

La densité d'électrons 12, 201  
 La phase 54, 108, 110  
 La précision 2, 40, 51  
 La puissance 63, 207  
 La température 60, 63, 87, 202, 207  
 La variance 41, 81

## M

Master 52, 74, 79, 91, 117, 126, 133, 135  
 Matlab 111, 113, 123  
 Max 51, 67, 107, 114  
 Mbps 92, 93, 115, 131, 160  
 Mean 78, 105, 109, 182  
 Mesure lente 43  
 Mesurer 43, 87  
 Mesures 89, 102, 139, 140, 141, 144, 145, 146

## N

Nutt 43, 57

## O

Observation du soleil 1, 10, 11, 12  
 Observations d'un sursaut 14, 201  
 Offset 36, 53, 54, 78, 54, 78, 83, 84, 99  
 Oscillateurs locaux 26, 29

## P

Phase 45, 100, 101, 108, 110, 111, 135

PLL 126, 129, 133, 134, 135  
Polarisation 26, 29  
Pour une liaison asymétrique 80, 83, 203  
ppb 33  
ppm 131  
PPS 34, 39  
Précision 31, 129

## Q

Quartz 42

## R

Radiohéliographe de Nançay 60, 61, 62, 63, 64,  
65, 66, 67, 73  
Radio interféromètre 191, 193  
Retard aller 74, 77, 80  
Résolution 26, 29, 44, 122

## S

SFU 61, 62, 63, 72, 207  
SKA 193  
Skew 23, 29, 34, 35, 36, 37, 38, 93, 117, 127,  
130, 133, 136, 201  
Slave 74, 79, 82, 83, 84, 91, 92, 126, 133, 135  
SNR 26, 61, 65  
Sort 78, 114, 184  
Synchronisation du temps 33, 38

## T

TDC 42, 91, 92, 93, 94, 99, 102, 124, 126, 127,  
129, 135, 156, 157, 161, 162, 166, 168, 179, 180,  
181  
Temps 36, 38, 45, 55, 56, 101, 102, 123, 139,  
140, 141, 144, 145, 146  
Time-interval 92  
Time-to-Digital Converter 42, 92

## U

UART 93, 157, 160, 176, 177, 178

## V

Valeur du jitter 65, 202  
VHDL 112, 113, 123  
Vitesse 97, 120, 131, 155, 207

## W

WSRT 30

## X

Xilinx 199

## Z

Zarka 192

## Résumé

Le Radiohéliographe de Nançay est le seul instrument dédié à l'imagerie du soleil en ondes décimétriques-métriques. Il fonctionne sur le principe de l'interférométrie, en utilisant 47 antennes essentiellement réparties sur des axes est-ouest (3,2 km) et nord-sud (2,5 km). Cette étude a pour but d'explorer un nouveau concept de numérisation propre à la radioastronomie du futur, appliquée ici à l'interférométrie solaire. Elle porte sur la numérisation rapide d'un système synchronisé en sortie d'antennes. Ces aspects "numérisation rapide" et "synchronisation" sont d'une importance capitale pour les prochains radiotélescopes du futur. Ils permettent de simplifier les chaînes de réception radiofréquence et de diminuer la consommation électrique ainsi que les coûts d'entretien et de la maintenance. L'application à l'observation du soleil comporte cependant des contraintes originales, comme la grande dynamique des signaux, qui ne sont pas prises en compte actuellement dans les études en cours pour les radiotélescopes du futur. Le radiohéliographe actuel a une chaîne de réception analogique avec une numérisation centralisée. La commutation entre les différentes fréquences dans la bande 150-450 MHz est réalisée d'une façon analogique et temporelle. Ceci nécessite beaucoup de calibrations analogiques et oblige de figer la gamme des fréquences (10 fréquences avec une largeur de bande de 1 MHz). De plus, en interférométrie métrique, les très grandes longueurs de câbles coaxiaux sont onéreuses. Les signaux transmis des antennes au récepteur sont toujours sources d'erreurs et des fluctuations importantes réduisent l'information radiofréquence. Apporter une numérisation complète de la bande (300 MHz) permet d'avoir de la souplesse dans le traitement et l'analyse des données (résolution fréquentielle et la possibilité d'observer plusieurs bandes simultanément, traitement des parasites). Ceci engendre la nécessité d'avoir une très grande précision des horloges (0,7 ps d'erreur de phase) pour cadencer des ADC (Analog-to-Digital-Converter) à large bande (de fréquence 1 GHz d'horloge). L'objectif principal de la thèse est d'étudier la synchronisation pour l'application à un réseau d'antennes multi-réparties. Les résultats et les concepts étudiés ont d'autres applications dans les grands projets européens et internationaux.

## Mots Clés

Radiohéliographe, Numérisation rapide, Synchronisation d'horloge, Antennes multi-réparties, Architecture numérique, Observation du soleil

## Abstract

The Nançay Radioheliograph is the only instrument dedicated to imaging of the solar corona imaging in the 150-450 MHz frequency band. It operates on the principle of interferometry, using 47 antennas essentially distributed on east-west (3.2 km) and north-south (2.5 km) axes. This study aims to explore a new technical concept for future radio astronomy, applied to a solar interferometer. It deals with the rapid digitization of a synchronized system at the antennas sides. High speed digitization and high accuracy synchronization are most important aspects for future radio telescopes. They make it possible to simplify radiofrequency reception chains and reduce their power consumption, as well as maintenance costs and complexity. The application to the observation of the sun, however, has some original constraints, such as the great dynamics of the signals, which are not taken into account in the current studies for future radio telescopes. The current radio telescope has an analog receiver with a centralized digitization. The switching time between each frequency (10 frequencies with 1 MHz bandwidth) in the 150-450 MHz band introduce latency in solar images processing, and also decrease the signal-to-noise ratio. In addition, in metric interferometry, the long coaxial cables in which the signal is transported from the antennas to the receiver always cause significant errors and fluctuations in the radiofrequency reception chains. Providing full digitization of the band (300 MHz) allows more flexibility in data processing and analyzing (frequency resolution and the ability to observe multiple bands simultaneously). This required high clock accuracy (0.7 ps of jitter) for ADCs clocks (1 GHz clock frequency). Therefore, the main objective of this thesis is to reach a sub-ns global time synchronization of distributed networks as the Nançay Radioheliograph. The results and the concepts studied here can be of use in major European and international projects.

## Keywords

Radioheliograph, High speed digitizer, Clock synchronization, Multi-Distributed Antennas, Digital Architecture, Solar observation