

**CONSERVATOIRE NATIONAL DES ARTS ET METIERS  
PARIS**

---

**MEMOIRE**

**Présenté en vue d'obtenir**

**Le DIPLÔME D'INGENIEUR C.N.A.M.**

**en**

**ELECTRONIQUE**

**par**

**Abdallah HAMDY**

---

**Conception et Réalisation d'un  
Système Numérique Rapide pour  
la Protection d'un Accélérateur Linéaire**

**Soutenu le 9 juillet 2004**

**JURY**

**PRESIDENT : H. VU-THIEN (CNAM)**

**MEMBRES : M. LUONG (CEA)  
A. FRANCE (CEA)  
S. SCHREIBER (DESY)  
A. LENTZ (CNAM)  
C. ALEXANDRE (CNAM)**





Cavité RF pour accélérateur linéaire



## Remerciements

Je dédie ce travail d'abord à mes parents pour leur dévouement dont je serai l'éternel débiteur, ma femme et amie, de la première heure du CNAM à la dernière et je le souhaite pour le reste de mes jours. Merci à toute ma famille et à mes amis pour leur soutien sans faille.

Je souhaite aussi dire toute ma reconnaissance aux professeurs du CNAM que j'ai rencontré dans mon cursus, qu'ils soient encore en fonction ou à la retraite. S'ils ne sacrifiaient pas de leurs temps pour venir le Week-end et le soir, je n'aurais peut être pas eu la chance d'aboutir à ce mémoire.

Ce mémoire a été réalisé au LESAR (Laboratoire d'Etude des Structures Accélématrices et de Radiofréquence), je remercie vivement toutes les personnes qui ont participé à cette étude. En particulier, MM. Poitevin, Novo, Denis, Napoly, Jablonka, Desmons, France et Luong.

Je tiens à remercier aussi MM. Kalmykow, Ballester, France, Sinnana, Uriot, Joly, Hanus, Long, Monnereau, Jacques, Boudigou, Porcher et Berry ainsi que Mmes Cotteverte et Régnaud.

Enfin, nos collègues Allemands qui ont permis une intégration facile dans leur groupe: MM. Nölle, Wendt, Werner, Schreiber et Schreiter.



## Table des matières

<b>Glossaire</b> .....	6
<b>Table des Illustrations</b> .....	8
<b>Introduction</b> .....	10
1 Présentation du DAPNIA/SACM .....	10
2 Présentation du projet TESLA et de l'installation TTF .....	11
2.1 Le projet TESLA .....	11
2.2 L'installation TTF .....	13
2.3 Le système de protection du linac .....	14
<b>Partie 1 : Etude et conception du système de protection du linac</b> .....	16
1. Principe du système et description des signaux .....	16
1.1 Signaux de contrôle et leur distribution .....	17
1.2 Les capteurs : tores de DESY .....	20
1.3 Choix des convertisseurs analogique-numérique (en Anglais : ADC) .....	22
1.4 Réalisation et test d'un prototype de module ADC .....	23
2. Traitement Numérique .....	34
2.1 Les calculs et la décision : DSP ou FPGA .....	34
2.2 Outil de développement : ISE .....	39
2.3 Outil de vérification : MODELSIM .....	44
2.4 Conception d'une carte d'évaluation FPGA .....	46
2.5 Vérification et validation de base .....	47
3. Interfaces et paramétrages .....	50
3.1 Signaux d'alertes (Interlocks) et d'états (Status) .....	51
3.2 Communication avec le système de contrôle-commande .....	51
4. Conception du système complet .....	53
4.1 Description des fonctions intégrées de la carte TPS .....	53
<b>Partie 2 : Implémentation et Validation</b> .....	56
1 Description et simulation des modules internes .....	56
1.1 Le module AUTORESET .....	57
1.2 Le module BTC (Conversion de binaire à complément à deux) .....	59
1.3 Le module CALCQx&TRANS $\mu$ C .....	60
1.4 Le module ADDRESS CODEC .....	65
1.5 Le module PARAMETERS INTERFACE .....	66
1.6 Le module CHARGE VALIDATION .....	67
1.7 Le module SINGLE MODE .....	68
1.8 Le module SLICE MODE .....	69
1.9 Le module INTEGRATION .....	73
2 Essais de la carte TPS (Toroid Protection System) .....	74
2.1 Test de configuration du FPGA .....	74
2.2 Signaux d'horloges .....	74
2.3 Test des ADC et du DAC .....	76
2.4 Evaluation des ADC .....	77
2.5 Test de la phase d'étalonnage .....	79
3 Essais de la carte à DESY .....	82
3.1 Essais de l'étalonnage du système .....	83
3.2 Essais des modes Single Bunch et Charge Validation .....	85
<b>Conclusion</b> .....	88
<b>Annexes</b> .....	90
Annexe A : Tores (DESY) .....	92
Annexe B : Schéma prototype ADC .....	96
Annexe C : Publication présentée à la Diagnostic particle accelerator conference 2003 .....	104
Annexe D : Poster présenté à l'I.T.R.P. (International Technology Recommendation Panel) .....	110
Annexe E : Architecture logique du système .....	114
Annexe F : Exemple de listing de banc de test VHDL pour la simulation des modules .....	116
de traitement numérique .....	116
Annexe G : Listing du programme en langage C du microcontrôleur ADUC812 pour l'étalonnage de la carte .....	124
TPS .....	124
Annexe H : Schéma de la carte TPS .....	134

Annexe I : Synoptique de la carte TPS .....	142
<b>Bibliographie</b> .....	<b>144</b>



## Glossaire

ADC	Analog to Digital Converter
ALS	Accélérateur Linéaire de Saclay
ASIC	Application Specific Integrated Circuit
BG	Bunch Gate
BIC	Beam Interlock Concentrator
BIS	Beam Interlock System
Bunch	paquet d'électrons
CEM	Compatibilité ElectroMagnétique
CLB	Configurable Logic Block
CLKD	Downstream Clock
CLKD_B	Downstream Bottom ADC Clock
CLKD_T	Downstream Top ADC Clock
CLKU	Upstream Clock
CLKU_B	Upstream Bottom ADC Clock
CLKU_T	Upstream Top ADC Clock
CLKUa, CLKU <sub>b</sub> , CLKU <sub>c</sub>	Horloges du FPGA
CRC	Cycle Redundancy Check
DAC	Digital to Analog Converter
DOOCS	Distributed Oriented Object Control System
DPDSEL	Downstream Programmable Delay Selection
DSP	Digital Signal Processor
DW_B	Downstream Bottom digital value
DW_T	Downstream Top digital value
FPGA	Field Programmable Gate Array
HIZ	High Impedance
JTAG	Joint Test Access Group
LC	Logic Cell
Linac	Linear Accelerator
LSB	Least Significant Bit
Macro-impulsion	train de paquet d'électrons
MSB	Most Significant Bit
PCB	Printed Circuit Board
PDG	Programmable Delay Generator
QD	Charge du tore aval
QU	Charge du tore amont
RAM	Random Access Memory
RF	Radiofréquence
ROM	Read Only Memory
RTL	Register Transfer Level
SDF	Standard Delay Format
SRAM	Static Random Access Memory
TPS	Toroid Protection System
TSD	Toroid Signal Downstream
TSU	Toroid Signal Upstream
TTF2	Tesla Test Facility phase 2
TTL	Transistor Transistor Logic
UP_B	Upstream Bottom digital value
UP_T	Upstream Top digital value

UPDSEL  
VHDL  
Language

Upstream Programmable Delay Selection  
Very high speed integrated circuit Hardware Description

## Table des Illustrations

Figure 1 : Vue d'artiste du Synchrotron SOLEIL actuellement en construction près du CEA/SACLAY .....	10
Figure 2 : Schéma simplifié de TESLA .....	12
Figure 3 : Plan de TTF phase 1 – longueur totale ~ 120 m.....	13
Figure 4 : Plan de TTF phase 2 – longueur totale ~ 255 m.....	13
Figure 5 : Disposition des tores pour la protection machine, seules les tores en rouge sont concernés .....	14
Figure 6 : Simulation du comportement d'un tore.....	16
Figure 7 : Principe du double échantillonnage.....	17
Figure 8 : Représentation des macro-impulsions avec les signaux de synchronisation .....	18
Figure 9 : Position du système de protection par rapport à l'accélérateur TTF2 .....	18
Figure 10 : Chronogrammes des signaux en entrée du TPS.....	19
Figure 11 : Capteur tore déshabillé, un unique enroulement en lamelle pour chaque sortie.....	20
Figure 12 : Tore de mesure pré-montés (DESY) et Amplificateur 100 MHz-40 dB-4voies (DESY).....	21
Figure 13 : Oscillations en sortie des tores à plusieurs enroulements .....	21
Figure 14 : Mesure en sortie des tores à enroulement unique de DESY .....	22
Figure 15 : Photos des prototypes de module ADC .....	24
Figure 16 : Synoptique du prototype de module ADC pour la sécurité différentielle.....	24
Figure 17 : Montage pour sonder les sorties des prototypes de module ADC .....	25
Figure 18 : Montage de mesure d'une tension continue pour les prototypes de modules ADC .....	26
Figure 19 : Signaux d'horloges sur les modules ADC .....	27
Figure 20 : Circuit de décalage en tension des modules ADC .....	27
Figure 21 : Plage d'entrée des ADC .....	28
Figure 22 : Timing interne de l'AD9501 .....	29
Figure 23 : AD9501 configuration type .....	29
Figure 24 : Mesures de signaux dynamiques avec les modules ADC.....	30
Figure 25 : Signal carré à l'entrée du module ADC.....	31
Figure 26 : Front de montée échantillonné par le module ADC .....	32
Figure 27 : Signal triangle à l'entrée du module ADC .....	32
Figure 28 : Echantillonnage de quelques points d'un signal triangle .....	33
Figure 29 : Balayage sur 10 ns avec le second module ADC modifié .....	33
Figure 30 : Synoptique de l'ADSP-21990 d'Analog Device .....	34
Figure 31 : Structure de base d'un FPGA, Configurable Logic Block.....	36
Figure 32 : Structure de base d'un CLB : Logic Cell.....	37
Figure 33 : Bloc d'entrée/sortie d'un FPGA : IOB .....	37
Figure 34 : Réseau de distribution global des horloges dans le SPARTAN2 .....	38
Figure 35 : Exemple d'un fichier VHDL pour un codeur .....	39
Figure 36 : Exemple d'entrée de design avec ECS .....	40
Figure 37 : Flot d'entrée de design pour ISE .....	40
Figure 38 : Flot d'implémentation de design pour FPGA.....	41
Figure 39 : Mode de configuration Master Serial .....	43
Figure 40 : Organigramme de configuration du FPGA.....	44
Figure 41 : Etapes de simulation pour design en langage HDL .....	45
Figure 42 : Carte d'évaluation.....	46
Figure 43 : Montage de test pour carte d'évaluation FPGA.....	47
Figure 44 : Module FPGA de calcul arithmétique pour la carte d'évaluation.....	48
Figure 45 : Simulation post routage d'un module de calcul arithmétique.....	49
Figure 46 : Face avant de l'instrument virtuel pour le test de la carte d'évaluation, résultat d'une multiplication.....	49
Figure 47 : Sécurité différentielle et son environnement .....	51
Figure 48 : Carte « Toroid Protection System ».....	53
Figure 49 : Architecture interne du FPGA pour la sécurité différentielle de TTF2 .....	55
Figure 50 : Générateur de stimuli intégré à ISE Webpack.....	57
Figure 51 : Module Autoreset .....	57
Figure 52 : Simulation fonctionnelle du module Autoreset .....	58
Figure 53 : Simulation post placement routage du module Autoreset .....	59
Figure 54 : Module CALCQx&TRANS $\mu$ C .....	60
Figure 55 : Synchronisation des valeurs numériques et calcul de la charge amont et de la charge aval corrigée .....	61
Figure 56 : Simulation de la partie CALCQx .....	62

Figure 57 : Zoom sur la simulation de la partie CALCQx .....	62
Figure 58 : Partie TransuC.....	63
Figure 59 : Simulation de la partie TransuC, zoom sur la phase d'acquisition.....	64
Figure 60 : Simulation de la partie TransuC, zoom sur la lecture de la sortie ADCVAL.....	65
Figure 61 : Module Address codec .....	65
Figure 62 : Simulation du module ADDRESS CODEC .....	66
Figure 63 : Module PARAMETERS INTERFACE.....	66
Figure 64 : Simulation du transfert des paramètres.....	67
Figure 65 : Module CHARGE VALIDATION.....	68
Figure 66 : Simulation du mode CHARGE VALID .....	68
Figure 67 : Module SINGLE MODE.....	69
Figure 68 : Simulation du mode SINGLE.....	69
Figure 69 : Module SLICE MODE.....	70
Figure 70 : Simulation du mode SLICE, zoom sur les signaux inhibiteurs de la sortie .....	72
Figure 71 : Simulation du mode SLICE, zoom sur la sortie d'alerte.....	73
Figure 72 : Module INTEGRATION.....	73
Figure 73 : Simulation du mode INTEGRATION.....	74
Figure 74 : Horloge des ADCs amonts de la carte TPS .....	75
Figure 75 : Horloges pour la synchronisation interne des modules de l'architecture FPGA .....	75
Figure 76 : Configuration FPGA pour les acquisitions ADC vers DAC.....	76
Figure 77 : Montage Acquisition d'un sinus.....	76
Figure 78 : Test échantillonnage .....	77
Figure 79 : Montage pour l'évaluation des ADC avec une tension continue .....	77
Figure 80 : Configuration FPGA pour l'évaluation des ADCs niveau 2 .....	78
Figure 81 : Configuration FPGA pour l'évaluation des ADCs niveau 1 .....	78
Figure 82 : Montage pour les tests d'étalonnage du système .....	80
Figure 83 : Configuration FPGA pour l'étalonnage du système .....	80
Figure 84 : Etalonnage, cas avec les amplitudes des signaux tores à peu près égales.....	81
Figure 85 : Etalonnage, cas avec les amplitudes des signaux tores différentes.....	81
Figure 86 : Site de mesure à DESY pour les essais avec faisceaux du système TPS.....	82
Figure 87 : Montage d'isolation galvanique des signaux de contrôle de la carte TPS.....	83
Figure 88 : Mesure des signaux BG, horloge 9 MHz et sortie des tores T1 et T2 sur l'accélérateur TTF2.....	85
Figure 89 : Montage de test des modes Single et Charge Validation.....	86
Figure 90 : Configuration du FPGA pour les essais des modes Single et Charge Validation.....	87

# Introduction

## 1 Présentation du DAPNIA/SACM

Au sein du DAPNIA (Département d'Astrophysique, de Physique Nucléaire et de l'Instrumentation Associée), le SACM (Service des Accélérateurs, de Cryogénie et de Magnétisme) est spécialisé dans le développement de nouvelles technologies pour la conception et la réalisation des accélérateurs.

Chaque application nécessite des caractéristiques de faisceau particulières : nature des particules, intensité, énergie, luminosité, dimension. La recherche et le développement des accélérateurs visent à obtenir des performances accrues ou une meilleure efficacité pour les applications futures des accélérateurs. Elles s'articulent autour de quatre pôles de recherche :

- 1) La dynamique de faisceaux, première étape dans la conception d'un accélérateur, permet de modéliser le comportement des particules accélérées, et ainsi de prévoir les éléments à insérer dans la machine en fonction de la qualité de faisceau désirée.
- 2) L'utilisation d'un supraconducteur pour la réalisation des cavités radiofréquence accélératrices est l'un des développements majeurs en R&D des accélérateurs. Cette technique permet d'accéder à des performances accrues et à un moindre coût d'exploitation.



Figure 1 : Vue d'artiste du Synchrotron SOLEIL actuellement en construction près du CEA/SACLAY

- 3) Les systèmes radiofréquence car chaque application peut nécessiter un développement de composants spécifique : RF de puissance (alimentations, modulateurs, sources, bancs de puissance), RF bas niveau (électronique RF analogique et digitale, mesure, régulation, diagnostics faisceaux).
- 4) L'étude et la réalisation des sous-ensembles d'un accélérateur prêts à être installés (sources, injecteurs, diagnostics faisceau, cryomodules complets avec cavités, coupleurs, système d'accord à froid, etc...).

Les compétences acquises par le passé nous ont permis de nous spécialiser selon trois grands axes : haute énergie avec des gradients accélérateurs supérieurs à 35 MV/m, cycles utiles élevés jusqu'au continu, faisceaux à haute intensité. Tous les projets en cours ou à venir nécessitent la maîtrise de l'un ou plusieurs de ces points, et le SACM qui regroupe des compétences dans tous ces domaines peut s'adapter de façon très polyvalente aux différents types de machines, que ce soit des machines de type TESLA (collisionneur  $e^+e^-$  haute énergie pour la physique des particules), des accélérateurs de protons à haute intensité avec le projet IPHI (applications possibles en physique nucléaire, production de neutrons, réacteurs hybrides, transmutation des déchets nucléaires) ou bien des sources lumineuses de type rayonnement synchrotron comme dans le projet SOLEIL (Figure 1), ou de type laser à électrons libres comme XFEL.

## 2 Présentation du projet TESLA et de l'installation TTF

### 2.1 Le projet TESLA

Le projet Tera Electron Volt Energy Superconducting Linear Accelerator (TESLA) est actuellement en cours d'étude à Hambourg au sein du Deutsches Elektronen-Synchrotron (DESY), dans le cadre d'une collaboration internationale [1]. TESLA (Figure 2) devrait ouvrir de nouveaux horizons pour la recherche fondamentale en physique des particules, mais aussi tenter de répondre à des questions que l'homme n'a jamais cessé de se poser :

- Qu'est ce que la matière ?
- De quoi est elle composée ?
- Comment s'est elle formée ?
- Qu'est ce que le vide ?
- Qu'est ce que l'espace et le temps ?

#### a) Caractéristiques générales

Longueur total du linac :	33 Km (incluant 2 sections accélératrices de 15 Km)
Tunnel diamètre interne :	approx. 5 mètres
profondeur sous-terre :	10 – 30 mètres
Point d'interaction :	unique
Nombre de hall cryogénique :	7
Surface du site de recherche :	approx. 40 hectares

#### b) Caractéristiques techniques

Energie (Point d'Interaction) :	500 GeV à 1 TeV
Nombres de cavités accélératrices :	19712
Température des cavités :	-271°C
Précision d'alignement :	0.5 millimètres
Longueur d'onde des rayons X :	0.5 Å à 5 nm

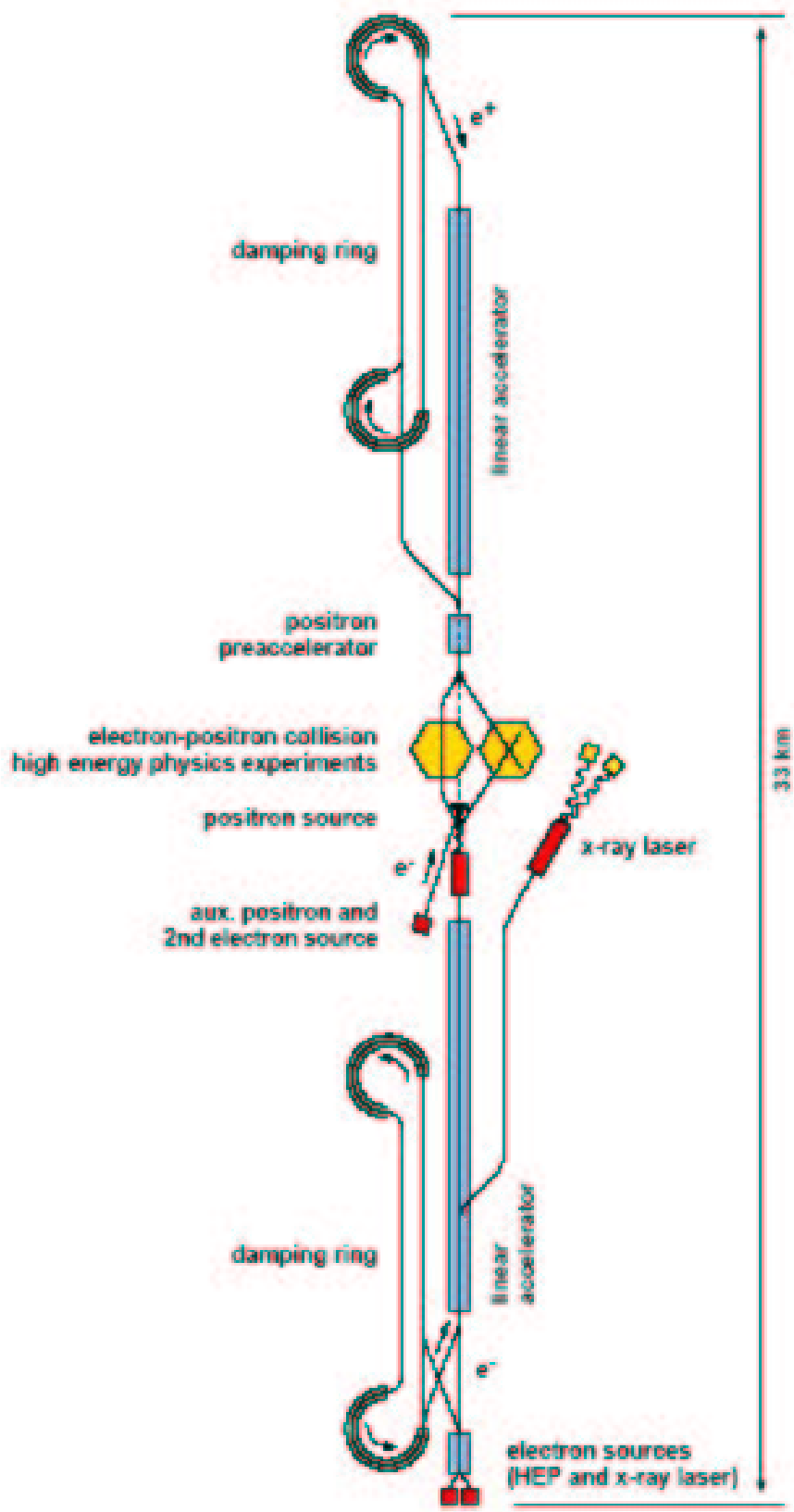


Figure 2 : Schéma simplifié de TESLA

## 2.2 L'installation TTF

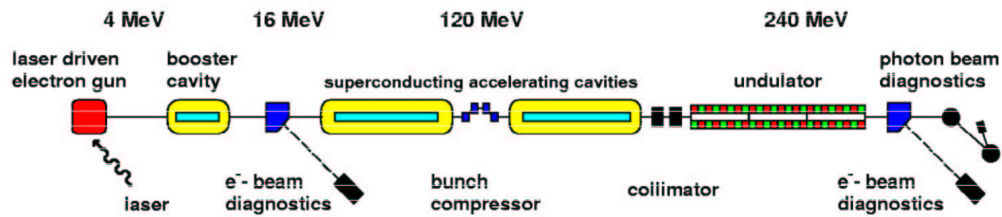


Figure 3 : Plan de TTF phase 1 – longueur totale ~ 120 m

La démonstration de faisabilité de TESLA [2] consiste en la réalisation d'un prototype (projet Tesla Test Facility –TTF à DESY, Hambourg). Il inclut des installations pour la préparation et la mesure des structures accélératrices supraconductrices de 1.3 GHz, un linac basé sur ces structures accélératrices, la cryogénie, les sources RF, les diagnostics et d'autres systèmes de logistique appropriés à cette démonstration.

Le but initial de TTF phase 1 (Figure 3) était d'atteindre 15MV/m, ce qui était sensiblement plus haut que les performances des cavités accélératrices mesurées dans les laboratoires de R&D sur les cavités. Le corps principal du linac était composé de quatre cryomodules de huit mètres chacun contenant les cavités neuf-cellules fonctionnant à 1.3 GHz. Le but premier de ces cavités est d'atteindre un gradient accélérateur conduisant à une énergie des paquets d'électrons (bunches) en bout de linac de 240 MeV.

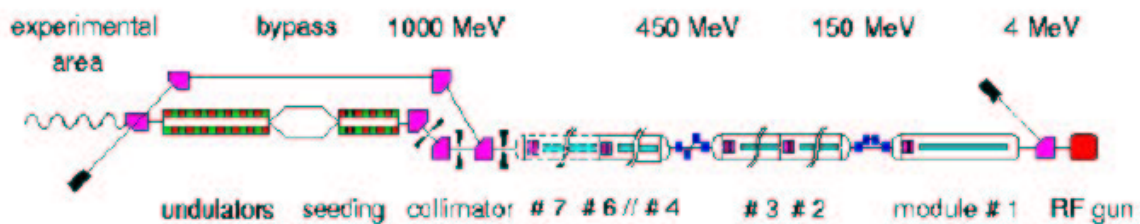


Figure 4 : Plan de TTF phase 2 – longueur totale ~ 255 m

Il faudra viser les 25 MV/m sur TTF2 (Figure 4) pour que l'énergie collectée en sortie soit de 800 MeV, voire 1GeV si l'on arrive à optimiser les performances des cavités au delà des 25 MV/m. Des secteurs d'analyse de faisceau en énergie et émittance seront prévus pour la caractérisation du faisceau injecté.

L'accélérateur linéaire TTF2 est situé à DESY, et incorpore les composants et l'appui technique fournis par les établissements de collaboration (12 pays participants en Europe, Asie et en Amérique du Nord). Les objectifs principaux sont :

- Test des composants pour TESLA et X-RAY FEL (Free Electron Laser)
- Fonctionnement en 4ème génération de source lumineuse pour produire des radiations SASE (Self Amplified Spontaneous Emission) dans la gamme des longueurs d'onde de 100 à 6 nm.



Cette machine sera capable de réaliser des trains d'impulsions faisceau, macro-impulsions (en Anglais : macropulses) de 800  $\mu$ s à une fréquence de répétition de 5 Hz pour des paquets d'électrons (bunches) espacés de 110.8 ns. L'énergie contenue dans ces macro-impulsions est assez élevée pour causer de sérieux dégâts en cas de perte du faisceau. C'est pourquoi, la protection rapide de la machine devient une préoccupation importante.

D'autres systèmes plus lents, avec un temps de réponse de l'ordre de la minute, basés sur la dosimétrie des rayonnements seront mis en place. Ces systèmes utilisent des fibres optiques et des photo-multiplieurs (PM). La fibre optique émet de la lumière lorsqu'elle est traversée par des neutrons, les PM convertissent cette lumière en signal électrique et permettent la mesure du rayonnement. Mais le temps de traitement est trop important, seule la mesure différentielle du courant du faisceau en amont et en aval par deux tores (transformateurs de courant) permet d'assurer une protection rapide.

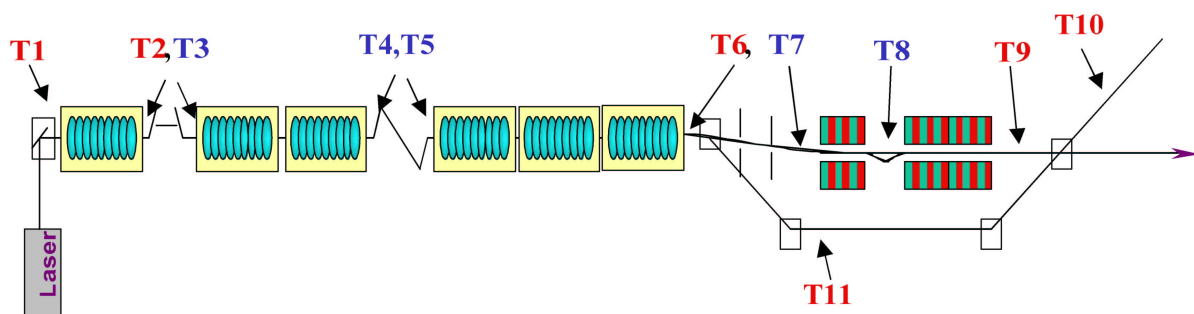


Figure 5 : Disposition des tores pour la protection machine, seules les tores en rouge sont concernés

Tore Amont	Nom	Position	Tore aval	Nom	Position	Commentaires
T1	Toroid/3Gun	1,25 m	T9	Toroid/12Exp	244,97 m	Ligne de faisceau par FEL
T1	Toroid/3Gun	1,25 m	T11	Toroid/16ByP	161,254 m	Ligne de faisceau par Bypass
T2	Toroid/2UBC2	20,548 m	T10	Toroid/?Dump	248,9 m	Dump atteint par FEL
T2	Toroid/2UBC2	20,548 m	T10	Toroid/?Dump	248,9 m	Dump atteint par Bypass

Tableau 1 : Paires de tores concernés par la sécurité différentielle

### 2.3 Le système de protection du linac

Le CEA, en particulier le SACM est en charge de différentes instrumentations pour l'accélérateur TTF2, la sécurité différentielle est une part non négligeable de ces tâches.

Le système de détection des pertes de faisceau est basé sur une mesure de l'intensité du courant faisceau à l'aide de tores (Figure 5). Le principe du système de protection différentielle est de comparer la charge des paquets d'électrons mesurée par des capteurs (Tableau 1) le long du linac. Lorsque le système détecte une variation de charge qui dépasse un seuil, un signal est envoyé au système de contrôle du canon pour inhiber le faisceau.

Un registre d'état est alors mis à jour pour indiquer les causes de l'événement qui a déclenché l'alerte. Ce registre peut être lu à tout moment par le réseau de commande de la machine (DOOCS pour Distributed Object Oriented Control System). Le DOOCS peut aussi lire et écrire les paramètres de seuil et de correction de la sécurité différentielle.

Ce mémoire présente l'étude et la réalisation du système de protection. Dans la première partie, nous introduisons le principe du système, le choix des composants et leurs évaluations à l'aide de prototypes, la communication avec le système de contrôle et enfin la conception de la carte TPS (Toroid Protection System). La seconde partie décrit les architectures et les bancs de tests VHDL utilisés pour simuler le système complet puis sa validation en laboratoire. Finalement, les essais sur l'accélérateur TTF2 à DESY sont présentés.

# Partie 1 : Etude et conception du système de protection du linac

## 1. Principe du système et description des signaux

Il s'agit de la réalisation des tiroirs d'électronique comparant les signaux de 4 paires de tores de mesure de courant faisceau, et engendrant un signal d'arrêt faisceau en cas de perte de courant détectée. Quatre acquisitions sur les tores amont et aval d'une même paire pour un même paquet d'électrons sera transmise à un ordinateur pour effectuer la décision. Chaque couple de tores permettra de connaître la charge d'un paquet d'électrons à deux endroits différents le long du linac. Le système de protection par 4 paires de tores se situera dans une baie, un tiroir par paire et un tiroir de distribution de l'horloge et du signal porte (BG) et un tiroir de protection d'une paire de tore de rechange. Dans cette étude, on présentera le système de protection pour une paire de tore.

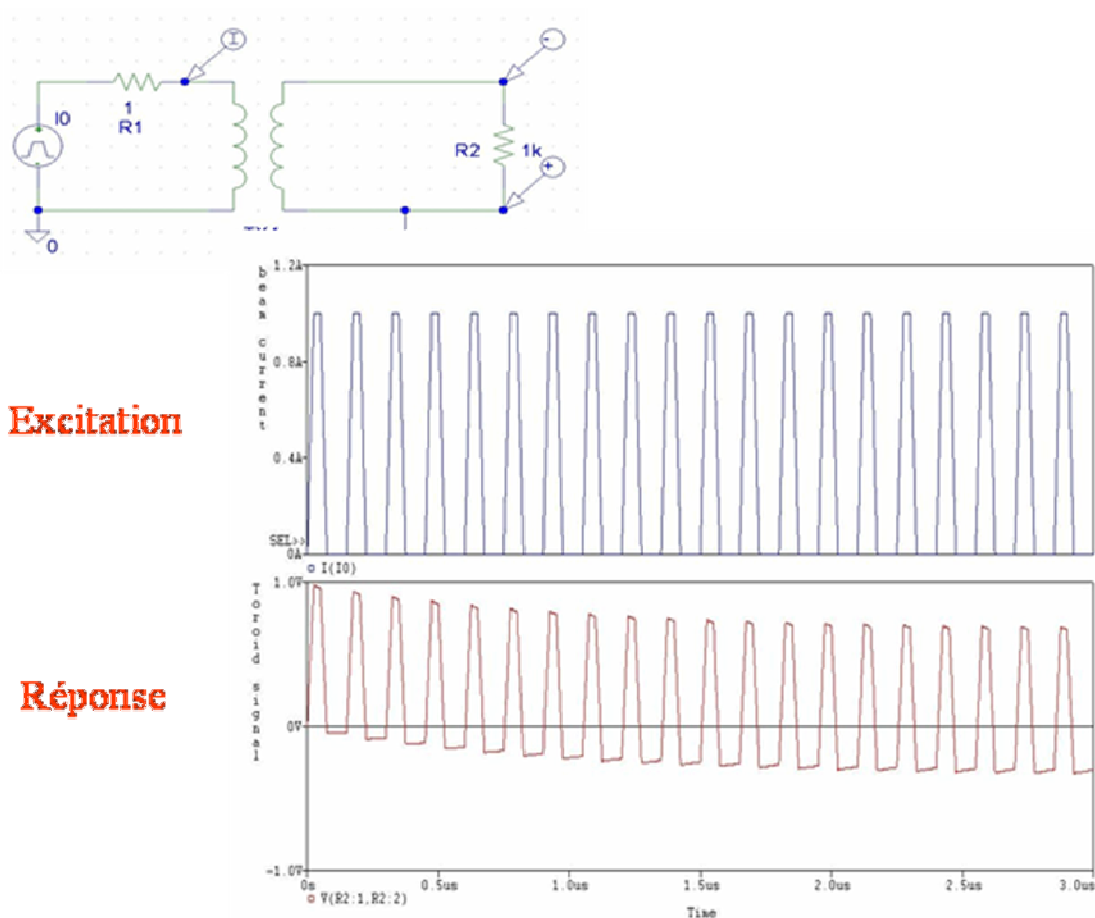


Figure 6 : Simulation du comportement d'un tore

Le filtrage de la valeur moyenne du signal mesuré par les transformateurs de courant est un phénomène appelé « droops ».

D'après le montage (Figure 6) :  $I_{R2} = n I_{R1}$  avec  $n$  : rapport des enroulements.

Et  $I_{R1} = K \exp(-t/\tau)$  avec  $\tau = L/R$  ( $L$  self du bobinage et  $R$  résistance parasite de la bobine).

D'où  $V_{R2} = K' \exp(-t/\tau)$ , ce qui explique la décroissance au secondaire du transformateur.

Pour être opérationnel dès le premier paquet d'électrons, nous prendrons deux échantillons (Figure 7) par sortie des capteurs tores.

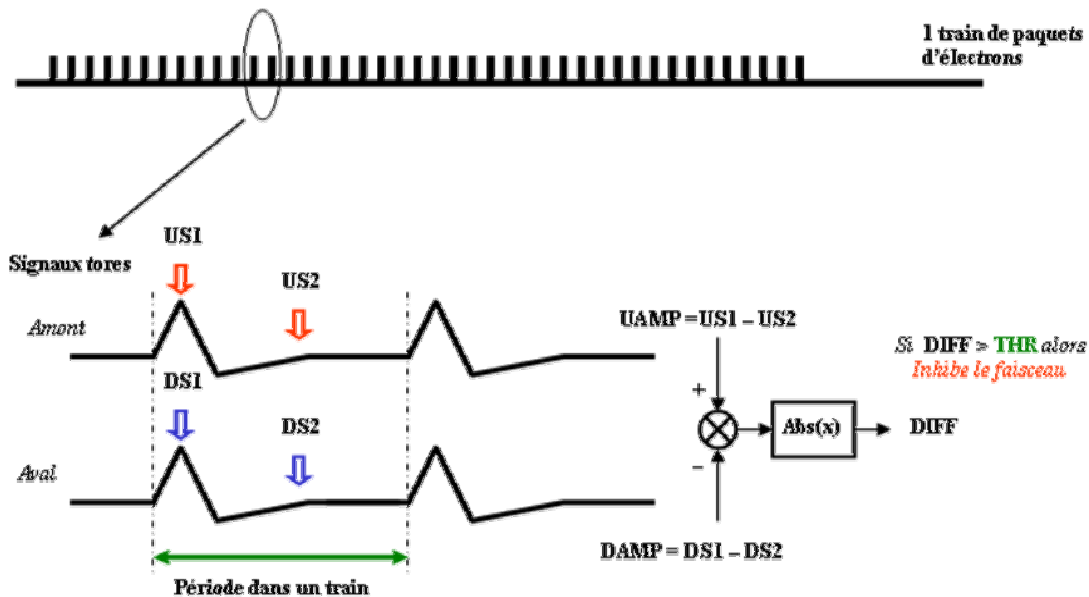


Figure 7 : Principe du double échantillonnage

La différence absolue mesurée entre tore amont et aval permet de connaître précisément la perte de faisceau ou si le faisceau arrose le tore aval. Si des électrons entre en collision avec une paroi métallique comme le tube de transport de faisceau au niveau du tore, il y a émission secondaire d'un nombre  $N$  d'électrons supérieur au nombre d'électrons émetteurs. Le signal mesuré par les tores arrosés est plus fort, on observe alors une amplification du courant de faisceau dû à une déviation de trajectoire des électrons. Dans la suite, la charge  $QU$  est proportionnelle à  $U_{AMP}$  (Figure 7) pour le tore en amont et la charge  $QD$  est proportionnelle à  $D_{AMP}$  pour le tore en aval.

### 1.1 Signaux de contrôle et leur distribution

Le système fonctionne à fréquence fixe pour l'échantillonnage et le traitement des données au sein du FPGA : 9.0277 MHz, un sous multiple (1/144) de la fréquence des ondes électromagnétique accélératrices des paquets d'électrons (bunches) qui est de 1.3 GHz. Cette fréquence est aussi la fréquence de répétition maximum des paquets d'électrons au sein d'un train d'impulsion. Les paquets d'électrons ayant des largeurs d'environ 30 ns, l'horloge doit être la plus précise possible, avec un minimum de jitter.

Le signal porte (en Anglais : Bunch Gate) est un signal qui indique la présence des paquets d'électrons (Figure 8), très important pour la sélection des échantillons en vue du traitement numérique puisqu'il n'y a pas de trigger.

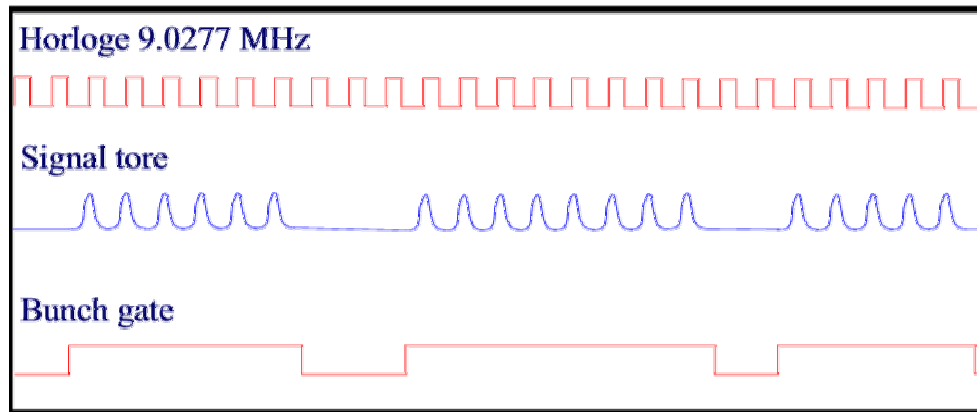


Figure 8 : Représentation des macro-impulsions avec les signaux de synchronisation

Les longueurs des câbles sont ajustées pour que les délais entre tores amont et aval soient réglés pour être multiples de 110,8 ns (même phase que l'horloge 9,0277 MHz). L'erreur de synchronisation acceptable  $\delta$  (Figure 9) est de l'ordre de 1 ns. Les câbles coaxiaux induisent des déformations du signal dues aux pertes intrinsèques. Pour avoir des formes de signaux à peu près identiques à l'entrée de la sécurité différentielle, il faut que les câbles soient approximativement de la même longueur.

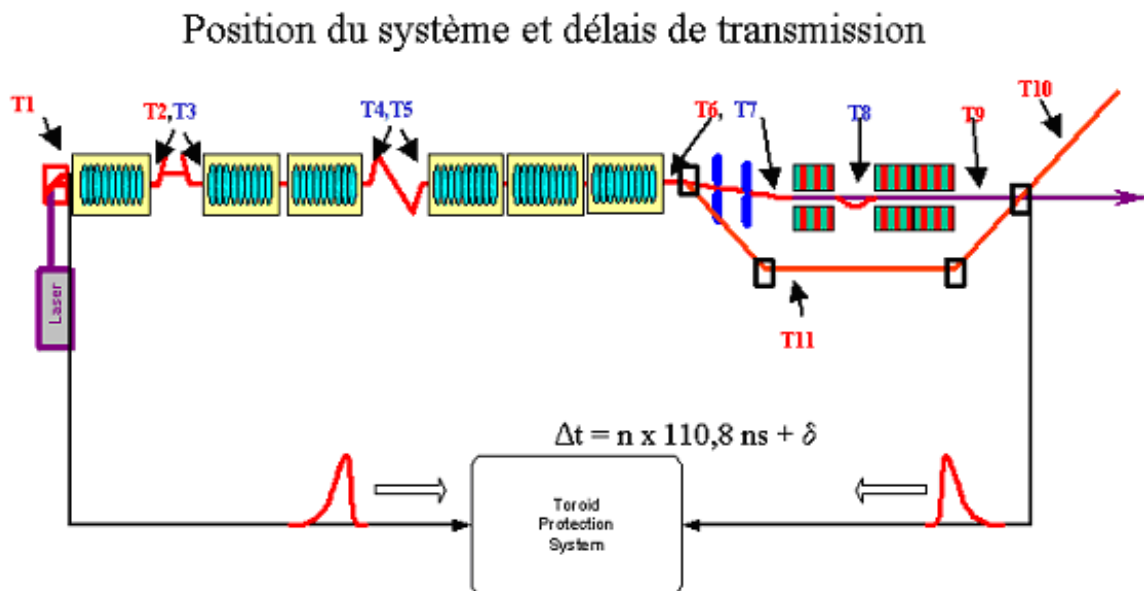


Figure 9 : Position du système de protection par rapport à l'accélérateur TTF2

Le tiroir de distribution (en Anglais : splitter) doit régler les retards des horloges pour chaque signal tore de façon à ce que  $\xi$  (Figure 10) soit dans la plage d'ajustement des délais programmables pour échantillonner la valeur crête. Le tiroir de distribution a pour objectifs de synchroniser les signaux de contrôle pour chaque tiroir de sécurité différentielle de manière à avoir la disposition suivante :

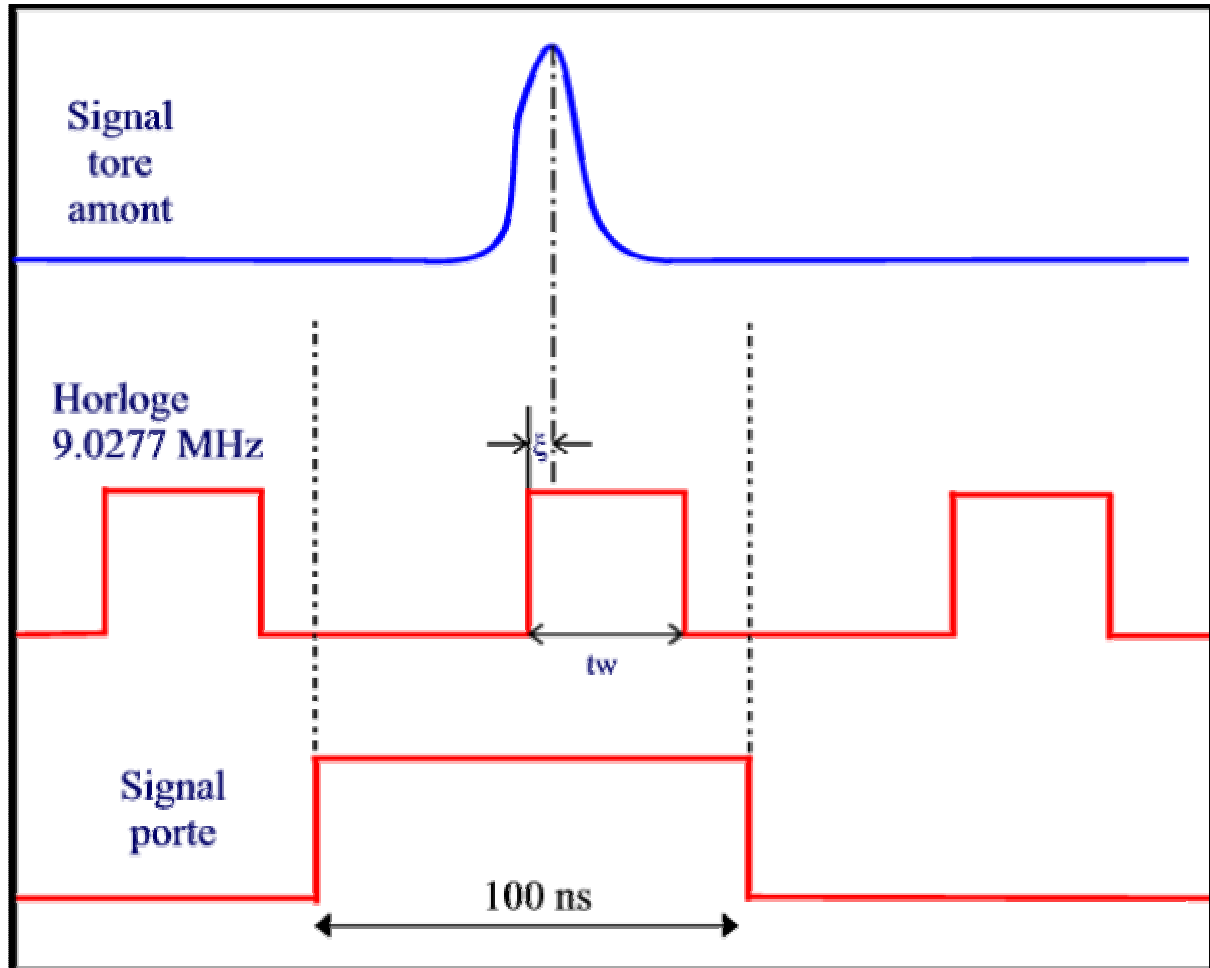


Figure 10 : Chronogrammes des signaux en entrée du TPS

Définition des modes de protection:

1. *Mode Absence de charge ou absence de signal porte faisceau ou en anglais, Charge validation or missing bunch gate:* compare la charge mesurée sur le tore amont avec un seuil, si le BG est présent et que la charge est insuffisante ou si le BG est absent et la charge est suffisante, un signal d'alerte est envoyé.
2. *Mode de comparaison instantanée paquet par paquet ou en anglais, Single mode :*

$$\left( \frac{|QD - QU|}{QU} \right)_i > T_{sgl}, T_{sgl} \cong 25 \%$$

Ce mode compare la différence de charge des tores amont et aval à un seuil, s'il y a dépassement un signal d'alerte est envoyé. Les pertes importantes doivent être détectés rapidement et le faisceau arrêté aussi vite que possible.

### 3. Mode moyennage par tranche ou en anglais, mode Slice :

$$\sum_{j=i-l}^i \left| \frac{QD - QU}{QU} \right|_j > T_{\text{Slice}}, T_{\text{Slice}} \cong O(1\%)$$

Ce mode effectue une moyenne glissante de la différence de charge des tores amont et aval le long du train de paquet d'électrons sur N paquets et compare celle-ci à un seuil pour déclenche de l'alerte s'il y a dépassement.

### 4. Mode Intégration (moyennage sur une macro-impulsion) ou en anglais Integration mode :

$$\sum_{i=0}^n |QD - QU|_i > T_{\text{int}}; T_{\text{int}} \cong O(10 \text{ nC})$$

Ce dernier mode somme les différences de charge des tores amont et aval sur le train entier de paquet d'électrons, il envoi un signal interlock dès que la somme dépasse un seuil. Le nombre de paquets d'électrons en opération nominale de l'accélérateur est de l'ordre de 7200, c'est pourquoi les erreurs statistiques correspondant au bruit dans les mesures électroniques joueront un rôle mineur et une résolution d'environ  $10^{-3}$  peut être envisagée.

## 1.2 Les capteurs : tores de DESY

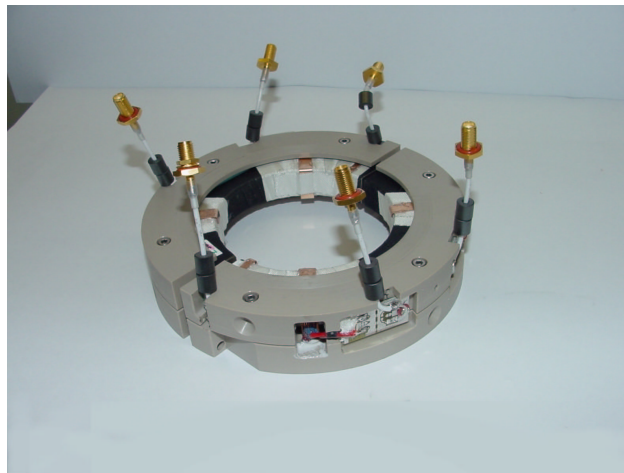


Figure 11 : Capteur tore déshabillé, un unique enroulement en lamelle pour chaque sortie

Les tores actuels (Figure 11) n'ont qu'un seul enroulement pour éviter les problèmes d'oscillations. Il y a quatre sorties avec un seul enroulement pour les utilisateurs et 2 entrées pour les signaux d'étalonnage. Leur bande passante est d'  $\sim 198$  MHz. La sensibilité est de 8 mVpp/nC. Les tores ont été conçus mécaniquement pour faciliter leur montage sur l'accélérateur par l'intermédiaire de deux demi coquilles.

Les sorties des tores sont combinées deux à deux pour n'avoir qu'une sortie coaxiale qui attaque un filtre passe-bas de fréquence de coupure 15 MHz. Elle est amplifiée par un amplificateur (Figure 12) de 4 voies de sorties permettant une dynamique de +/- 2 Volts. Le facteur de conversion est de 500 mV/nC en sortie de l'amplificateur.

Les anciens systèmes de protection étaient complètement analogiques et moins rapide qu'aujourd'hui, ils ne permettaient pas de détecter les pertes de faisceau paquet par paquet. La

détection se faisait macro-impulsion par macro-impulsion. Les tores de mesure précédents avaient plusieurs enroulements pour obtenir une self importante. Ce qui générant une constante de temps ( $\tau = L/R$ ) suffisamment grande pour représenter la macro-impulsion. Les oscillations (Figure 13) sont dues aux capacités parasites entre les enroulements des tores.

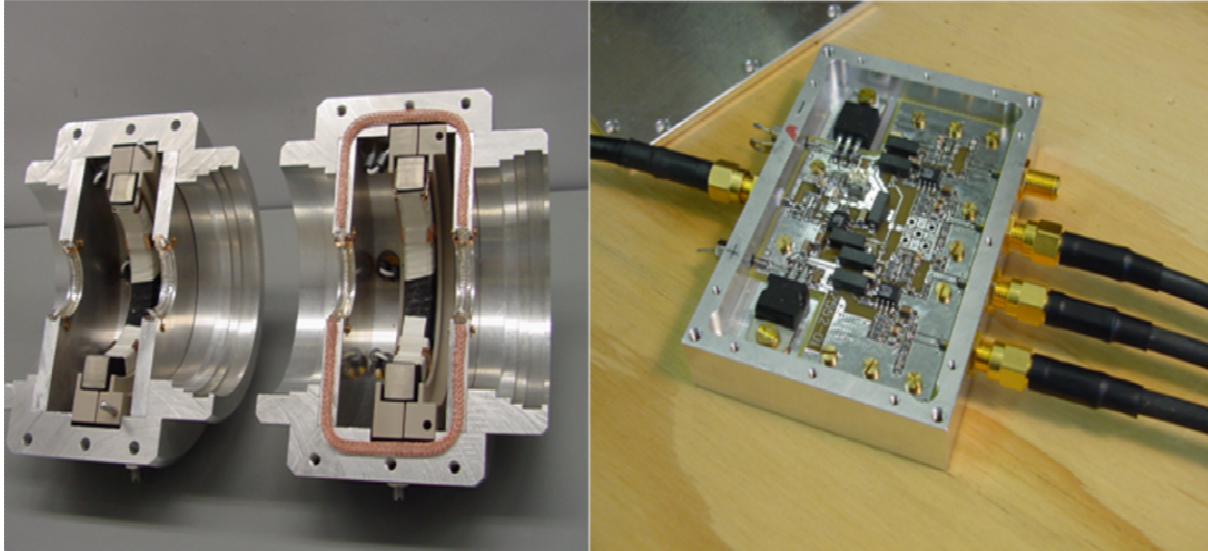


Figure 12 : Tore de mesure pré-montés (DESY) et Amplificateur 100 MHz-40 dB-4-voies (DESY)

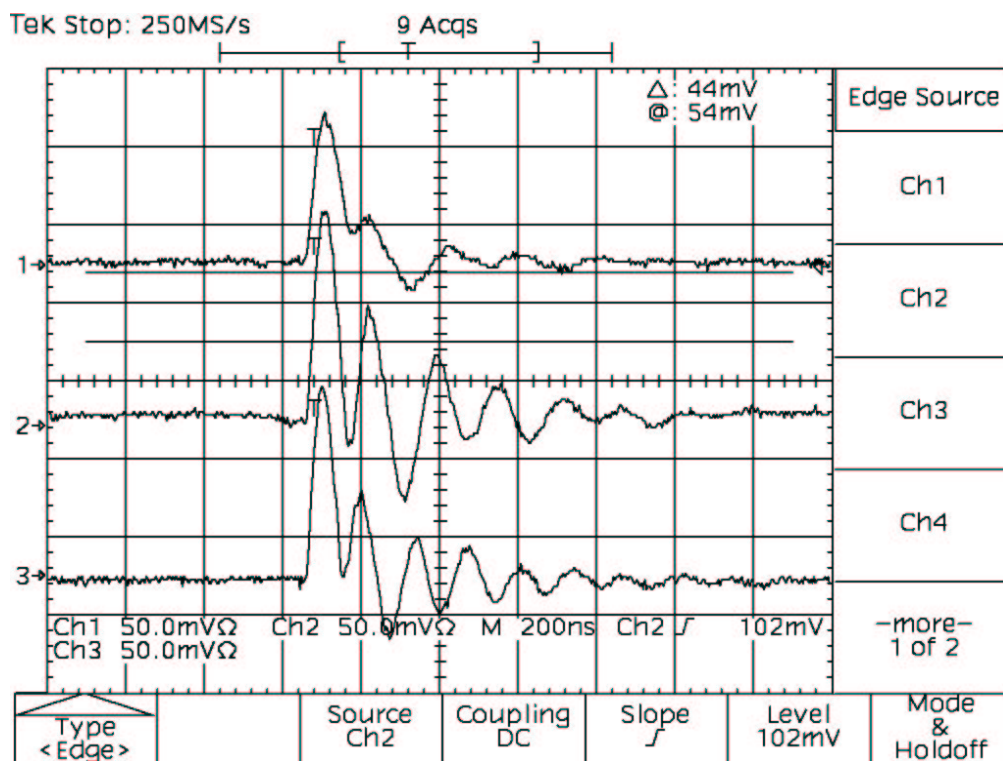


Figure 13 : Oscillations en sortie des tores à plusieurs enroulements



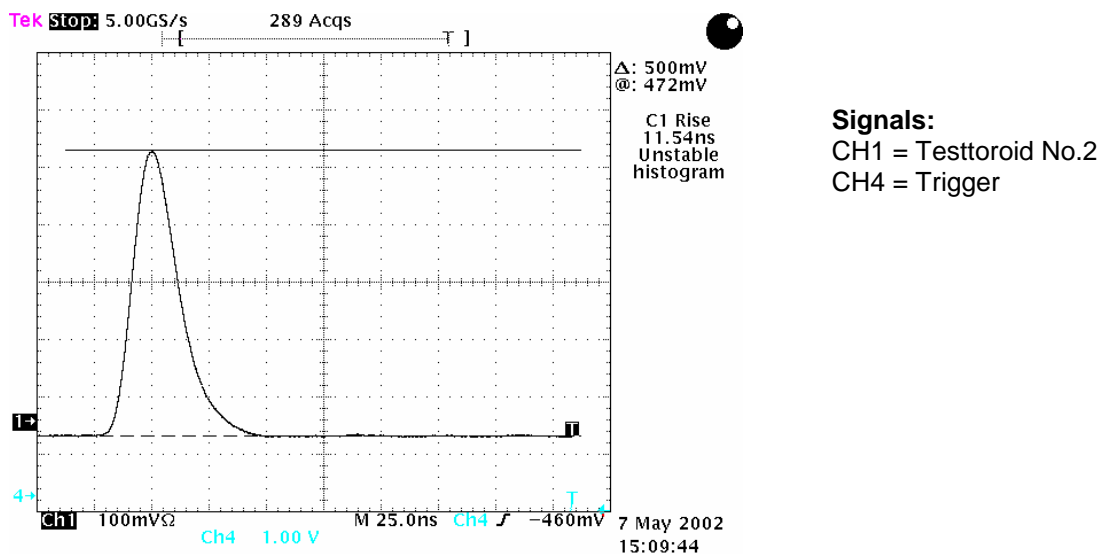


Figure 14 : Mesure en sortie des tores à enroulement unique de DESY

Les mesures des signaux tores (Figure 14) montrent une bonne qualité du signal en sortie des amplificateurs. Le bruit étant faible, nous pouvons envisager des possibilités de réglage des seuils de déclenche d'alerte plus bas que ceux prévus à l'origine (10% en mode paquet par paquet au lieu de 25% dans les premières spécifications). Bien entendu, les mesures ci-dessus ne tiennent pas compte de la transmission sur câble coaxial de grande longueur (100 à 150 m).

### 1.3 Choix des convertisseurs analogique-numérique (en Anglais : ADC)

Il existe 4 grandes familles d'ADC [3] : les flash, les pipelines, les convertisseurs à approximations successives ou SAR et les sigma-deltas. Chaque méthode de conversion dispose d'avantages et d'inconvénients propres qu'il convient de cerner afin de choisir le composant adéquat.

Le convertisseur flash applique la tension analogique d'entrée à  $2^n - 1$  de comparateurs de tension en parallèle, les entrées de ces derniers étant espacées les unes des autres de l'équivalent d'un LSB, équivalent à la pleine échelle divisée par  $2^n$ . Ensuite, il s'agit de décoder les sorties binaires des comparateurs pour les transformer en un mot de  $n$  bits. Son avantage principal est sa rapidité, puisqu'en théorie l'ensemble de la conversion est effectué en un seul cycle. Mais cette structure ne permet pas d'avoir une grande résolution dû à son grand nombre de comparateurs. Limités à 8 voire 10 bits, les flash sont utilisés pour l'acquisition de données haute vitesse (radars par exemple).

Le convertisseur pipeline repose sur  $p$  étages de conversion, chacun étant basé sur un CAN flash à  $m$  bits effectuant une boucle de comparaison avec un CNA  $m$  bits et un échantillonneur bloqueur. Le premier étage sert à calculer les  $m$  premiers bits de la conversion et passe le reste du signal d'entrée à l'étage suivant. Le principal atout des convertisseurs A/N pipeline est d'atteindre des résolutions bien supérieures à celles des modèles flash tout en étant moins gourmand en comparateurs. Au détriment, bien sûr, du temps de conversion totale, qui passe de 1 cycle à  $p$  cycles. Ces CAN se retrouvent principalement dans les systèmes de communications (stations de base, modems, équipements réseaux), de test et d'instrumentation rapide.

Les CAN SAR (*successive approximation register*) comparent la tension d'entrée à la moitié de la pleine échelle, ce qui donne le MSB, puis à un quart ou à trois quart de la pleine échelle, et ainsi de suite jusqu'à atteindre une résolution de  $n$  bits au bout de  $n$  cycles. Ils sont destinés pour des conversions haute résolution à bande passante limitée. Ils sont très utilisés pour les contrôles moteurs.

Les CAN sigma-delta ont une structure assez différente des précédentes, elle met en jeu un intégrateur, un comparateur et un convertisseur N/A 1 bit dans la boucle de comparaison. Opérant en sur échantillonnage ( $f \gg f_{\text{Nyquist}}$ ), le bruit est rejetée à des fréquences élevées. Un filtre numérique en sortie suffit alors à supprimer la majeure partie de ce bruit, permettant d'atteindre des résolutions allant de 14 à 24 bits. Très présent dans les applications audio, ils sont aussi utilisés en mesure (pression, poids, température).

Les mesures de pertes faisceau pour la sécurité différentielle de TTF exige une importante résolution compte tenu des spécifications émises par les experts en diagnostique faisceau. Dans nos spécifications de départ avec nos collègues allemands, nous cherchions un ADC de résolution  $> 12$  bits, de gamme d'entrée analogique 4 Volts crête à crête, de fréquence d'échantillonnage au minimum de 9 MHz et de bande passante  $> 50$  MHz. Nous avons en course principalement l'ADS5421 de Burr Brown racheté par Texas Instruments et l'AD9240 d'Analog Device.

	<b>ADS5421</b>	<b>AD9240</b>
Résolution (bits)	<b>14</b>	<b>14</b>
Nb de cycle pipeline	<b>10</b>	<b>3</b>
Fé (MSPS)	<b>40</b>	<b>10</b>
Bande passante (MHz)	<b>500</b>	<b>70</b>
Rapport signal à bruit (dB)	<b>75 (à 10 MHz)</b>	<b>75 (à 5 MHz)</b>
ENOB (bits)	<b>12.2 (à 1 MHz)</b>	<b>12.2 (à 5 MHz)</b>

Tableau 2 : Comparatif des ADCs ADS5421 et AD9240

Fé : Fréquence d'échantillonnage. ENOB : Nombre de bits garantie comme étant exacte par le constructeur.

Ces deux ADCs (Tableau 2) sont assez proche du point de vue des performances recherchées. Mais l'ADS5421 demande un temps de latence avant de sortir le premier échantillon de 10 cycles. Ce qui correspond à un retard d'environ  $1.1 \mu\text{s}$ , trop long pour un système de protection rapide à double échantillonnage. C'est donc après un compromis sur la résolution et la vitesse de conversion que le choix s'est porté sur un ADC 14 bits : AD9240.

#### 1.4 Réalisation et test d'un prototype de module ADC

Les modules ADC (Figure 15) sont conçus pour réaliser trois fonctions :

- l'acquisition et la conversion numérique sur 14 bits des signaux en provenance des tores,
- la remise en forme du signal d'horloge,
- l'ajustement des horloges d'échantillonnages par un délai programmable (PDG).

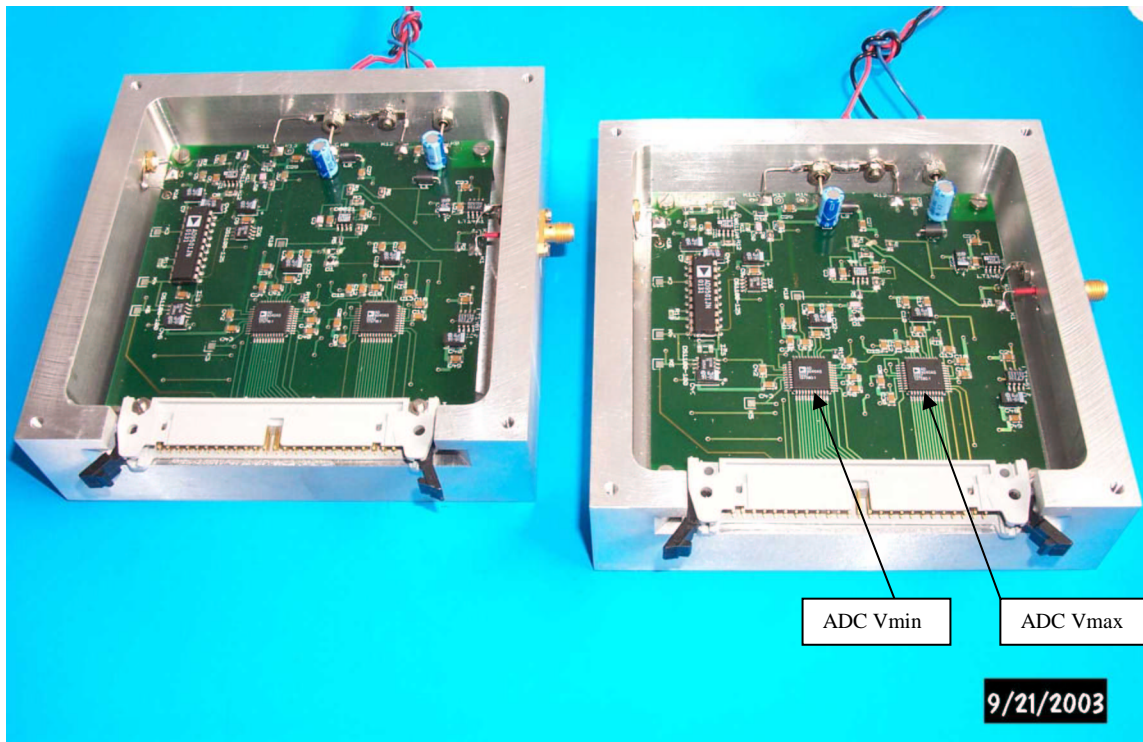


Figure 15 : Photos des prototypes de module ADC

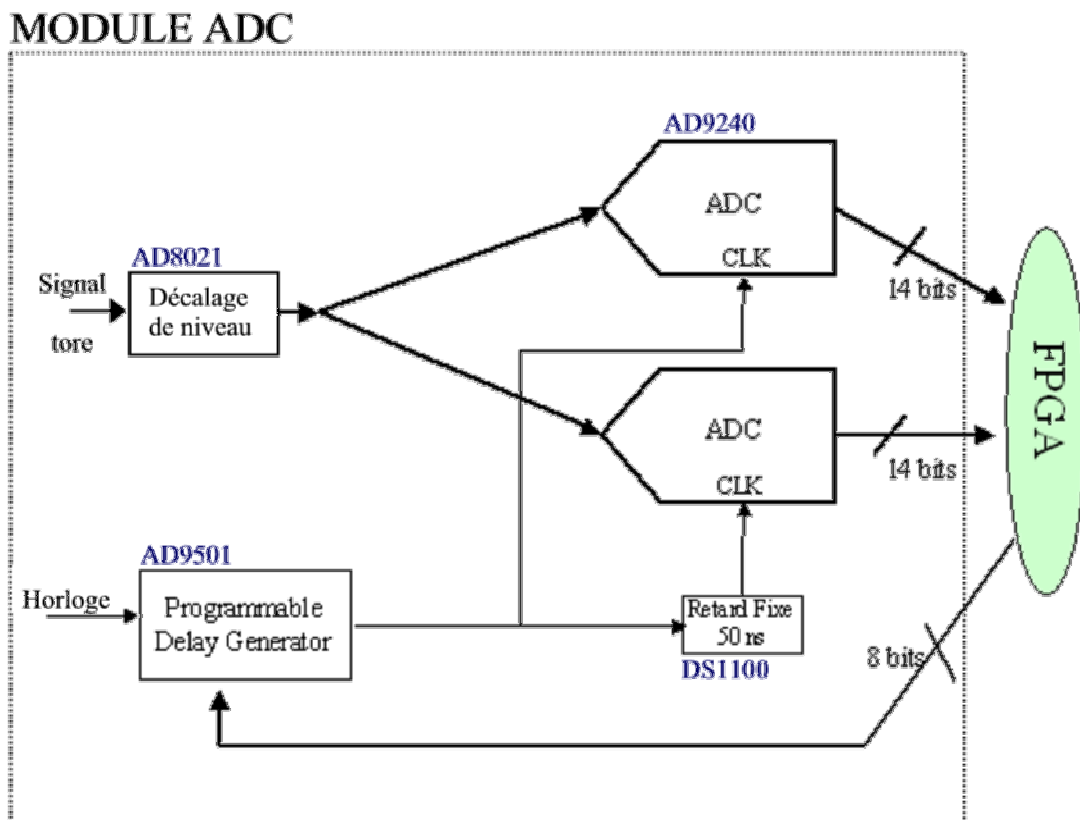


Figure 16 : Synoptique du prototype de module ADC pour la sécurité différentielle

Le module ADC est conçu pour échantillonner les signaux sur un tore et les transmettre sous forme numérique au FPGA. L'ajustement de l'horloge pour prélever l'échantillon TOP se fait avec un PDG (Programmable Delay Generator) piloté par le FPGA et par un délai fixe de 50 ns pour la valeur BOT.

Les mesures ont été réalisées sur les deux prototypes de module ADC (Figure 16). L'interface « lecture des sorties numériques et commande du délai programmable » (Figure 18) consiste en un petit circuit contenant des connecteurs facilitant l'accès avec une sonde aux bits de sortie des ADC. Les mesures de chaque bit des sorties ADC sont faites à l'aide de sonde de bande passante 500 MHz.

Les sorties des ADC se présentent sous format binaire de base quelle que soit la gamme d'entrée. Les mesures sont réalisées en sondant bit après bit et en convertissant en tension les valeurs binaires.

$$V_{\text{range}} = 2 * V_{\text{ref}} = 2 * 2.048\text{V} = 4.096\text{V}$$

$$\text{Le bit de poids faible (LSB) vaut donc : } V_{\text{range}}/2^{14} \sim 250 \mu\text{V}.$$

Nous avons piloté le PDG par une carte multifonction (P6040E) de National Instruments à travers un châssis PXI, mais dès le départ le niveau de bruit pour une mesure de tension continu était très élevé : plus de 15.75 mV de fluctuations, c'est à dire seulement 8 bits de précision au lieu de 14. Nous avons donc réalisé un montage pour la commande des 8 bits du PDG (Programmable Delay Generator) à l'aide d'interrupteurs mini-dips selon le montage (Figure 17).

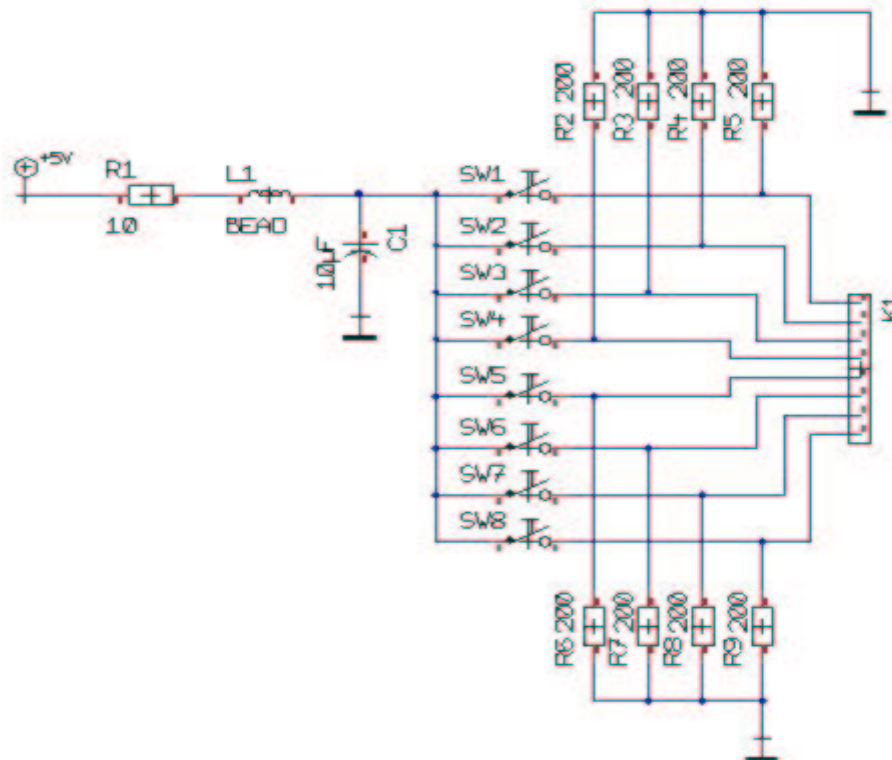


Figure 17 : Montage pour sonder les sorties des prototypes de module ADC

#### 1.4.1 Mesure d'une tension continue

A l'aide d'une pile 1.5V, nous avons réalisé un montage (Figure 18) nous permettant de faire varier une tension continue la moins bruitée possible. Le splitter, qui est un ensemble de 8 amplificateurs suiveurs en parallèle, a été réalisé pour distribuer l'horloge aux différents modules du système. L'entrée Trig0 du prototype conduit à un trigger de Schmidt pour améliorer les fronts de montée du trigger. Trig1 est la sortie de ce trigger. CLK\_H et CLK\_L sont respectivement l'horloge de l'ADC\_TOP (échantillon max) et de l'ADC\_BOT (échantillon min).

Pour chaque mesure (Tableau 3, Tableau 4), nous scrutons chaque bit de chaque ADC avec la sonde et nous notons les valeurs des bits stables et le nombre de bits instables. Nous calculons ensuite l'erreur de précision pour la tension mesurée. Soit l'exemple de calcul de l'erreur Err pour 5 bits instables exprimés en décimal :

$$Err = (\text{Valeur maximale en décimal des bits instables}) * \text{quantum}$$

$$Err = 31 * 250 \mu V = 7.75 \text{ mV}$$

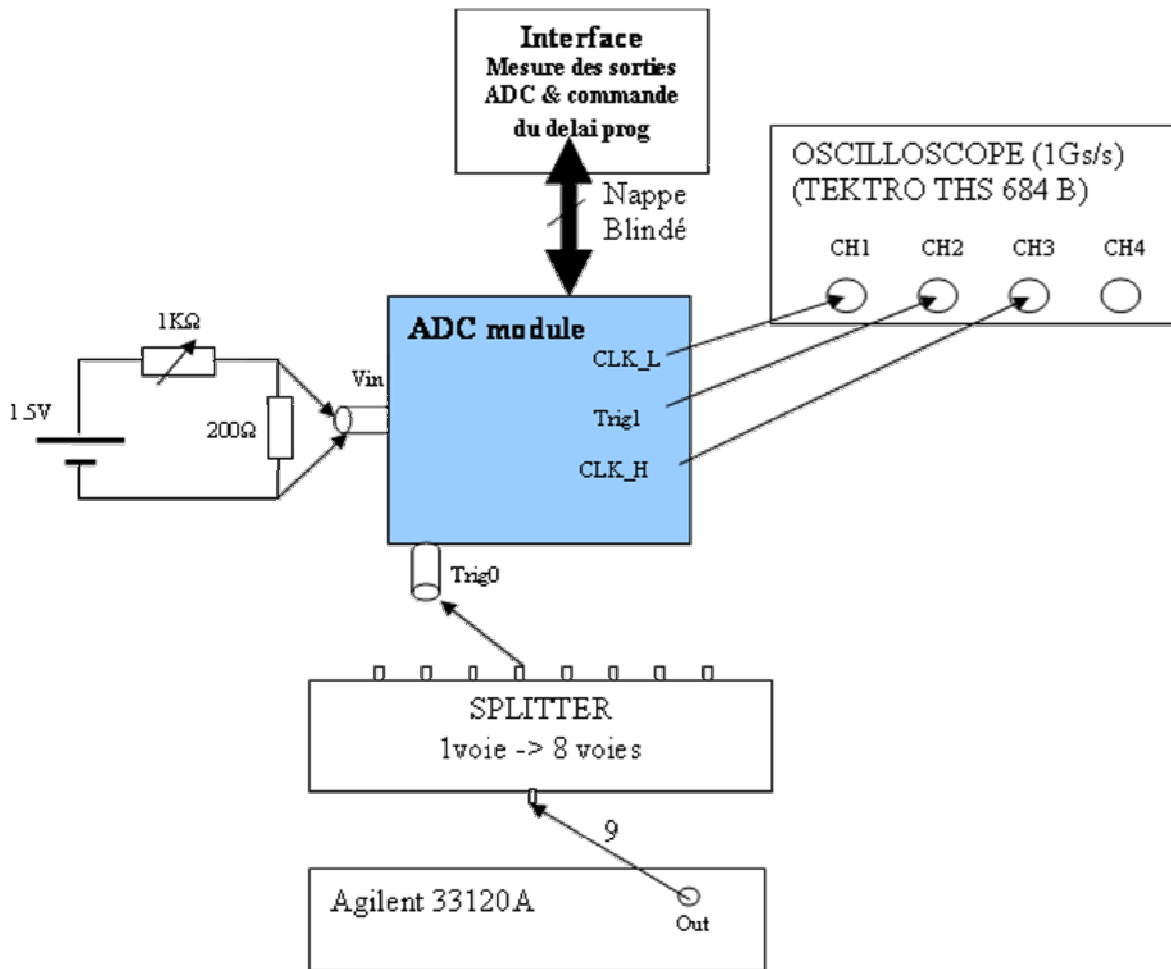


Figure 18 : Montage de mesure d'une tension continue pour les prototypes de modules ADC

Tension Entrée (V)	ADC	ADC module #1	
		S <sub>n</sub> (V)	Err (mV)
1	Top	0.966	15.75
	Bot	0.968	3.75
0.5	Top	0.502	7.75
	Bot	0.485	3.75
-0.5	Top	-0.481	3.75
	Bot	-0.516	1.75
-1	Top	-0.976	31.75
	Bot	-1.013	3.75

Tableau 3 : Mesure d'une tension continue sur le module ADC #1

Dans le Tableau 3,  $S_n$  est la Sortie numérique, correspondant aux bits stables. L'erreur Err représente les bits instables. On observe des fluctuations sur les bits de poids faible en sorties des ADC qui ne correspondent pas aux valeurs attendues ( $10 < ENOB < 11$ ). Pour une tension négative de 1 Volt en entrée, on mesure un bruit de l'ordre de 30 mV en sortie des convertisseurs dans le pire cas et cela malgré l'éviction du châssis de commande numérique du PDG.

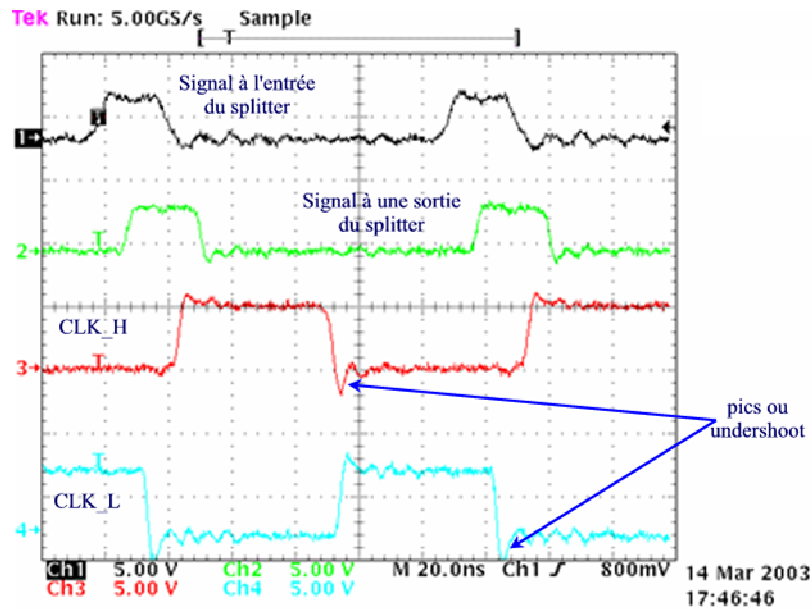


Figure 19 : Signaux d'horloges sur les modules ADC

Les pics (Figure 19) peuvent être à l'origine de raies à hautes fréquences perturbatrices de l'électronique. Pour étudier le problème du bruit en sortie des ADC, on utilise un analyseur de spectre (HP4195A) et on vient sonder aux entrées des ADC alors que le signal à l'entrée est continu. Avec 0 Volt à l'entrée, on mesure une raie à la fréquence des horloges ADC (9MHz) d'amplitude  $P = -40\text{dBm}$  sur une entrée  $V_{INA}$  d'un des deux ADC de la carte.

$$P_{dBm} = 10 \log [P(W)/1mW], \text{ d'où } P_{dB} = P_{dBm} - 30\text{dB} = -70\text{dB} \Leftrightarrow P = 0.1\text{mW}.$$

$$P = V^2(\text{crête})/R \text{ avec } R = 50\Omega.$$

Le bruit dû à l'horloge vaut  $\sqrt{50 \times 10^{-(\frac{-40-30}{10})}} = 2.2\text{mV rms}$  ou  $6.2\text{mV crête à crête}$ . Cette raie était à  $-65\text{dBm}$  à l'entrée du boîtier du module ADC soit  $125\mu\text{V}$ . C'est donc, notre carte et plus précisément les circuits d'horloges qui génèrent ce bruit.

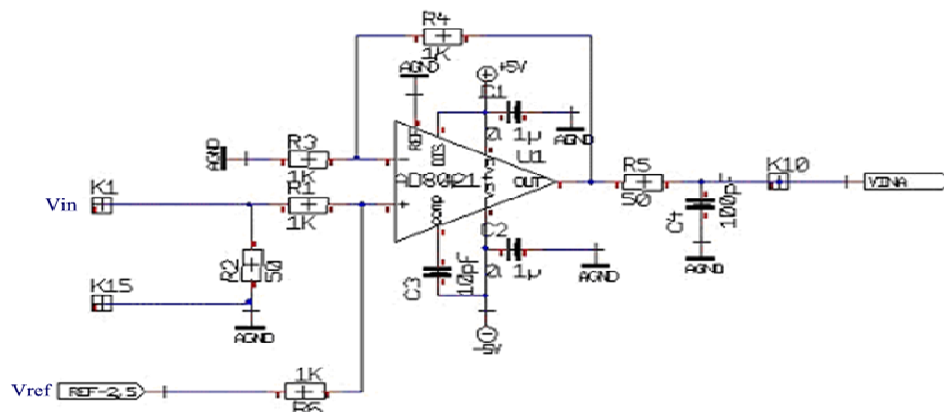


Figure 20 : Circuit de décalage en tension des modules ADC

Le circuit (Figure 20) permet le décalage en tension des signaux tores pour être dans la plage d'entrée de l'ADC (Figure 21). Le montage de décalage en tension a pour sortie

$$V_{out} = V_{ref} + V_{in}$$

Or cet amplificateur a une plage de sortie limitée à 3,5V maximum pour une alimentation +/- 5V, ce qui permet une gamme d'entrée du module de +/-1V ( $1\text{ V} + V_{Ref} = 3.5\text{ V}$ ). Toutefois, on peut sensiblement dépasser cette plage en augmentant jusqu'à +/-5,3V l'alimentation de la carte.

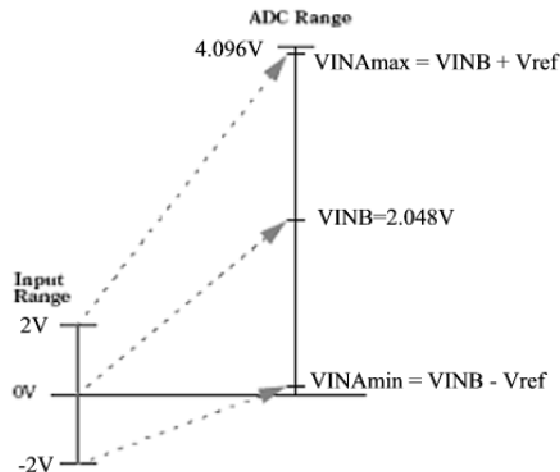


Figure 21 : Plage d'entrée des ADC

Au niveau du PCB, une piste du signal d'horloge des ADCs se trouvant en bordure des 2 plans de masse (analogique, digital) a été déroutée pour éviter de passer près des boîtiers DILs de l'AD9501, source de rayonnement. On élimine les résistances  $50\Omega$  d'adaptation, autre source de rayonnement dû au courant les traversant. Nous les avons mis à l'extérieur de la boîte. On ajoute des résistances en série sur les pistes d'horloge pour filtrer les hautes fréquences et atténuer les fronts de montée.

L'autre souci était dû au PDG (Programmable Delay Generator), celui-ci ne permettait de balayer qu'une plage de 4.4 ns à 9 MHz. Si l'on programme un retard supérieur, l'AD9501 devient instable.

*Calcul de la plage de balayage du PDG (Figure 22) à une fréquence donnée :*

*Retard minimum :  $t_{PD} = 13\text{ ns}$ , dû à la circuiterie interne de l'AD9501.*

*Retard programmé :  $t_D = [(Valeur\ digitale)/256] * R_{SET} (C_{EXT} + 8.5\text{ pF}) \cdot 3.84$  ici 4.4 ns.*

*Retard total :  $t_{PD} + t_D$*

*Retard de propagation du reset :  $t_{RD}$*

*Temps de stabilisation du générateur de la rampe :  $t_{LRS}$*

*Largeur de la sortie du PDG :  $t_W$*

*A 9 MHz, d'après la Figure 22,  $T = t_{PD} + t_D + t_W - t_{RD} + t_{LRS}$  avec  $T$  la période du trigger en ns.*

*A cette fréquence  $t_{LRS} - t_{RD} \sim 42.6\text{ ns}$ .*

*$110 = 13 + t_D + 50 + 42.6 \Rightarrow$  d'où  $t_D = 4.4\text{ ns}$ .*

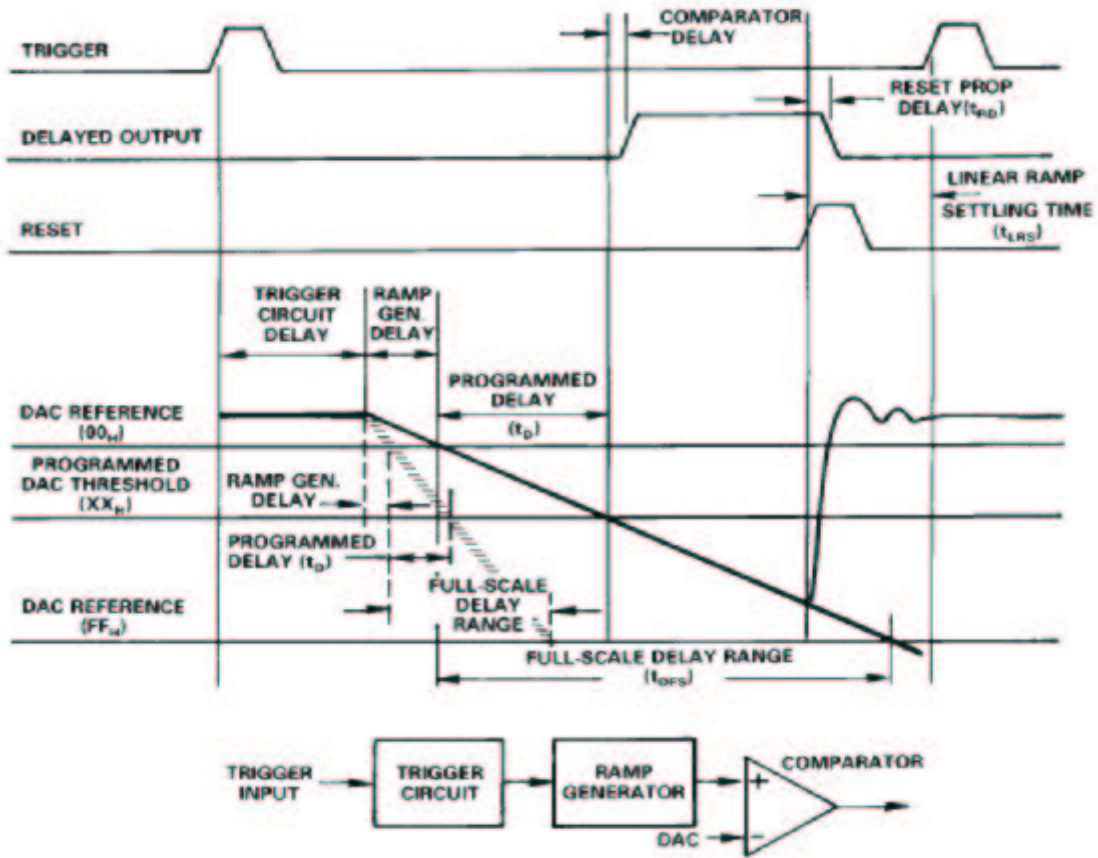


Figure 22 : Timing interne de l'AD9501

La première cause de cette faible plage est le délai fixe ajouté entre la sortie et l'entrée Reset (Figure 23) de l'AD9501 de 50 ns pour répondre aux caractéristiques des horloges des ADC (largeur de créneau minimum de 45 ns). Sans cela, le signal retardé en sortie du PDG avait une largeur de  $t_{RD} \sim 18$  ns, insuffisante pour les ADC. La seconde cause était la résolution de 77 ps pour une plage totale  $t_{DFS}$  de 20 ns réglée par un couple  $R_{SET}$ ,  $C_{EXT}$ .

$$(t_{DFS}) = R_{SET} (C_{EXT} + 8.5 \text{ pF}) \cdot 3.84$$

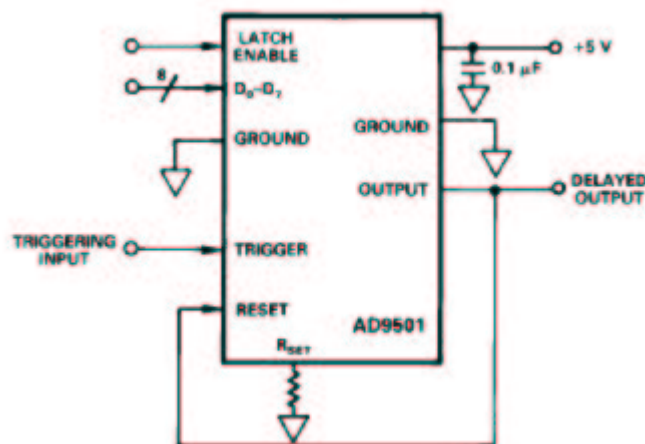


Figure 23 : AD9501 configuration type

Pour améliorer les performances des cartes ADC (Tableau 4), nous avons modifié les caractéristiques comme suit :



- Pour augmenter la plage d'entrée des ADC, nous avons diminué  $V_{ref}$  de 2.5 V à 2.048 V, ce qui donne une plage de +/- 1.5V en entrée du prototype.
- Le délai fixe dans la boucle PDG (sortie et entrée du Reset de l'AD9501) est passé de 50 à 35 ns et la plage  $t_{DFS}$  a doublé (pas :156 ps), donnant un balayage possible d'environ 20 ns à 9 MHz.

Tension Entrée (V)	ADC	ADC module #1		ADC module #2	
		$S_n$ (V)	Err (mV)	$S_n$ (V)	Err (mV)
1	Top	1.002	3.75	1.023	1.75
	Bot	0.970	3.75	1.023	1.75
0.5	Top	0.494	3.75	0.495	1.75
	Bot	0.483	3.75	0.494	3.75
-0.5	Top	-0.502	3.75	-0.498	3.75
	Bot	-0.474	3.75	-0.498	3.75
-1	Top	-1.01	3.75	-1.002	3.75
	Bot	-0.976	3.75	-1.013	1.75

Tableau 4 : Mesures de tension continue après les modifications sur les 2 modules ADC

En comparaison du Tableau 3, les résultats sont meilleurs. Le nombre de bits stables est passé de 8 à 10. Cela est plus proche des résultats escomptés.

#### 1.4.2 Mesure de signaux dynamiques périodiques

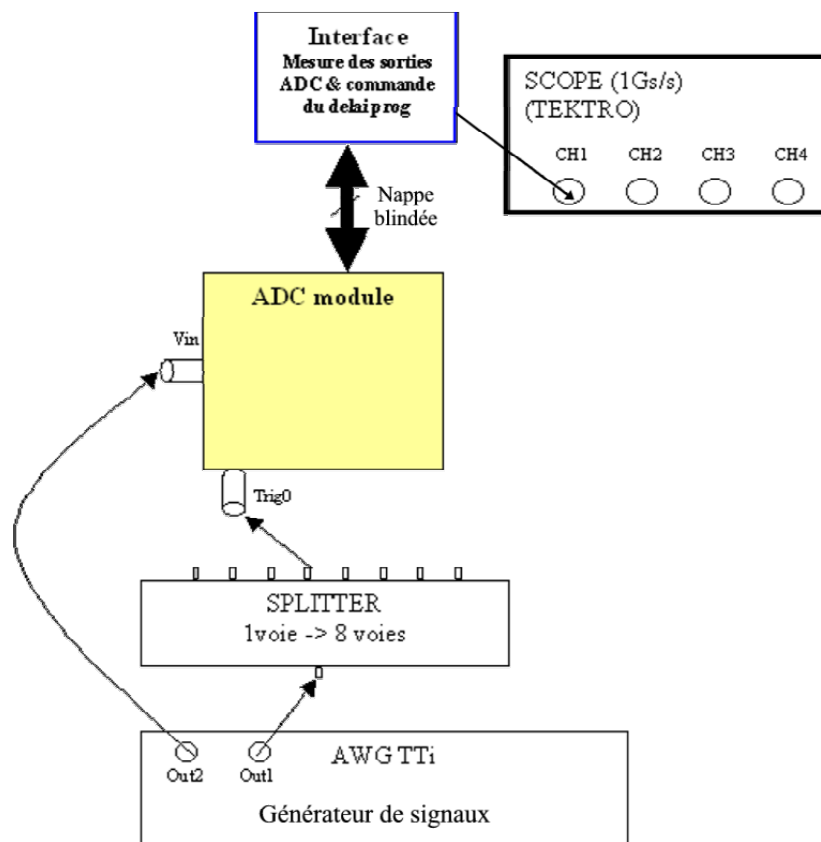


Figure 24 : Mesures de signaux dynamiques avec les modules ADC

L'objectif de ce test est la mesure d'un signal à la fréquence de l'horloge et balayage de celui-ci à l'aide du délai programmable. Ces mesures ont été faites avec le premier module ADC avant que les modifications apportées suite aux mesures d'une tension continue ne soient effectuées.

A l'aide du montage (Figure 24), nous avons mesuré deux formes de signaux : un carré et un triangle. Le générateur de signaux est réglé pour avoir un signal d'horloge attaquant le splitter à 9 MHz de type TTL, ce signal est synchronisé avec la voie principale qui doit toujours rester dans la gamme  $\pm 1$  V. Le retard est programmé pour  $t_D = 0$  ns, des câbles sont alors ajoutés pour positionner le retard entre l'horloge et le signal à échantillonner de façon à ce que l'horloge soit en avance d' $\sim 4$  ns de la zone à balayer sur le signal  $V_{in}$ .

a) Balayage sur le front de montée d'un signal carré avec le PDG à 9 MHz

Les curseurs (Figure 25) indiquent la zone balayée par le module ADC, ci-dessous nous visualisons le signal d'entrée  $V_{in}$ .

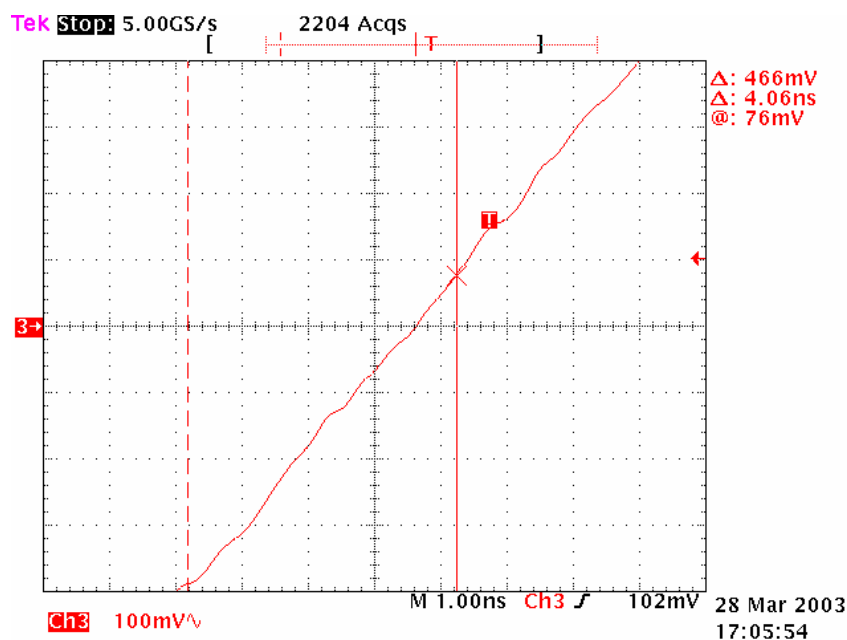


Figure 25 : Signal carré à l'entrée du module ADC

Les croix rouges (Figure 26) indiquent les sorties digitales converties en tension. Le pas du retard sur le signal d'horloge est de 231 ps. Comme sur la Figure 25, l'amplitude balayée est d' $\sim 500$  mV pour une plage d' $\sim 4.4$  ns.

b) Vérification du délai entre l'ADC\_TOP et l'ADC\_BOT

On génère un signal triangulaire sur la voie principale d'amplitude 2 V crête à crête et l'on ajuste l'horloge de l'ADC\_TOP pour échantillonner l'amplitude maximum. Une fois de plus, les curseurs (Figure 27) sont placés pour montrer la zone à échantillonner.

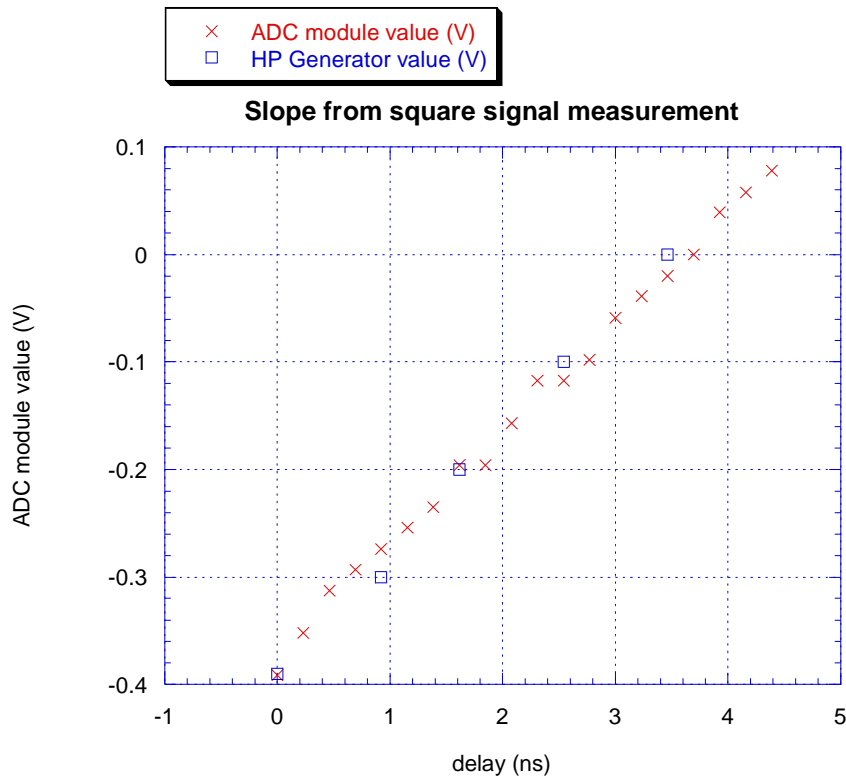


Figure 26 : Front de montée échantillonné par le module ADC

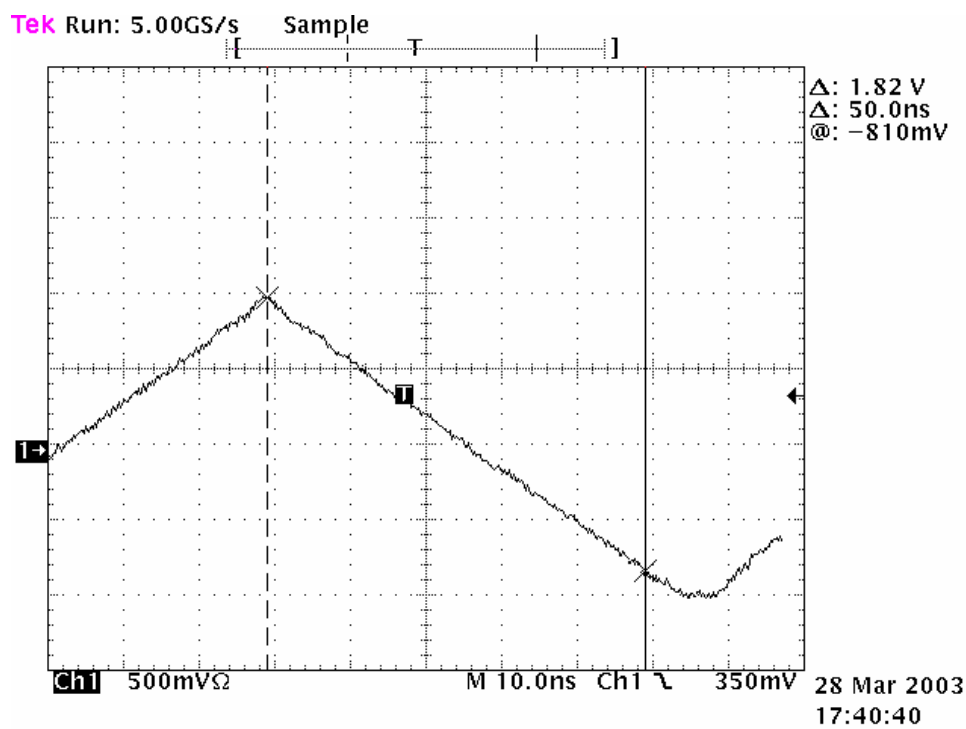


Figure 27 : Signal triangle à l'entrée du module ADC

Quelques points (Figure 28) suffisent à vérifier que le module ADC prélève la valeur maximum et la valeur minimum 50 ns plus tard.

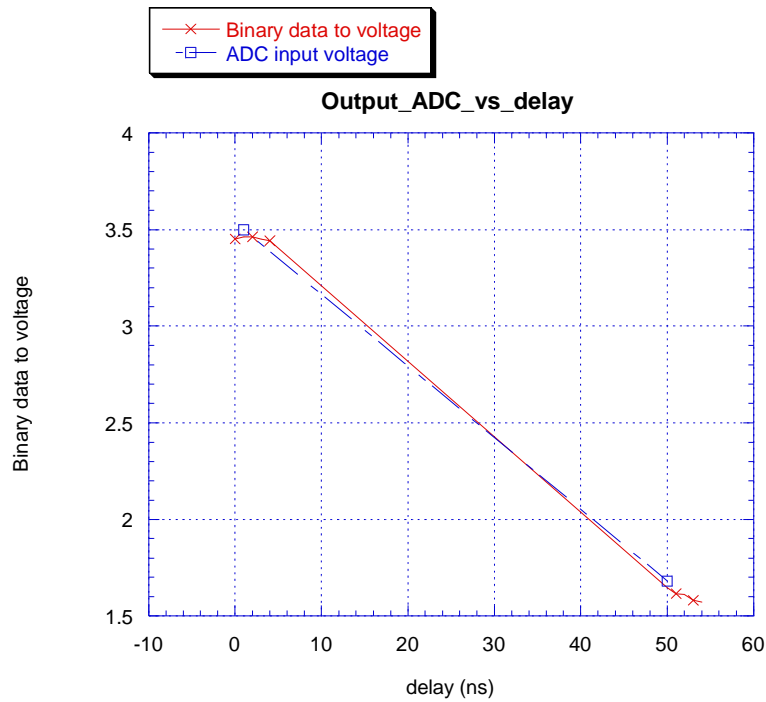


Figure 28 : Echantillonnage de quelques points d'un signal triangle

Le signal (Figure 28) est bien décalé autour de la tension  $V_{ref}$  de 2.5V pour être dans la plage d'entrée des ADC (0 à  $2V_{ref}$ ). Il y a bien décalage de 50 ns entre la valeur TOP et BOT.

c) Balayage d'un signal carré après modification sur les modules ADC

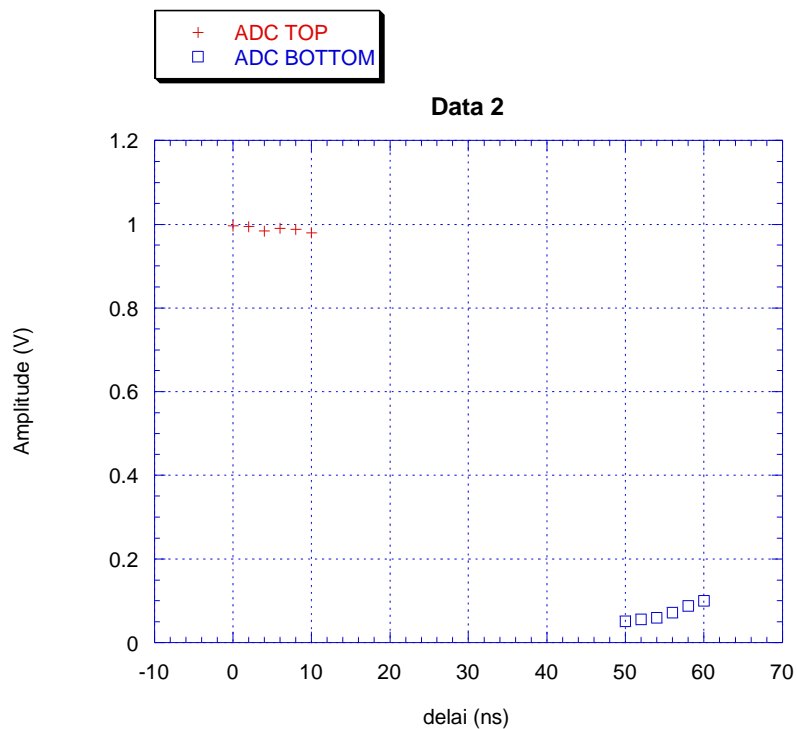


Figure 29 : Balayage sur 10 ns avec le second module ADC modifié

On vérifie (Figure 29) que l'on peut parcourir une plage de plus de 4 ns après les modifications.

Cette étude a permis de dégrossir la partie analogique de la sécurité différentielle. Par la suite, nous avons gardé les ADC AD9240 et le montage de décalage en tension en alimentant les amplificateurs opérationnels en +/-12V pour avoir une plus grande dynamique de sortie. Il est aussi possible, en ajoutant un autre amplificateur attaqué sur l'entrée inverseuse et connecté à l'entrée  $V_{INB}$ , d'utiliser les ADC en mode différentielle pour améliorer la mesure en terme de bruit et de distorsion.

Pour la carte de protection finale, nous avons choisi des délais programmables DS1020 de Dallas Semiconductor en boîtiers CMS et disposant d'une interface de programmation série, moins bruyant par le fait qu'il n'utilise que 2 pistes pour la liaison série (SCLK, SDATA) et moins complexe d'utilisation que l'AD9501.

Au niveau du PCB, nous avons tenu compte du routage des pistes d'horloges pour qu'elles soient le plus loin possible de la zone analogique. Ces pistes ont toutes reçues des résistances d'amortissement en série. Les circuits d'horloges ont été éloignés des ADC. Enfin, les charges d'adaptation ont toutes été prévues à l'extérieur du boîtier.

## 2. Traitement Numérique

### 2.1 Les calculs et la décision : DSP ou FPGA

Au cœur du système de protection se trouve le calculateur, il doit être précis et rapide. C'est pourquoi dans un premier temps, l'étude du DSP [4] s'est imposé, plus particulièrement un composant d'Analog Device : l'ADSP-21990 (Figure 30). Ce composant intègre 8 voies ADC 14 bits/20 MSPS multiplexés et la possibilité d'avoir 2 voies simultanées, il dispose aussi d'une interface UART.

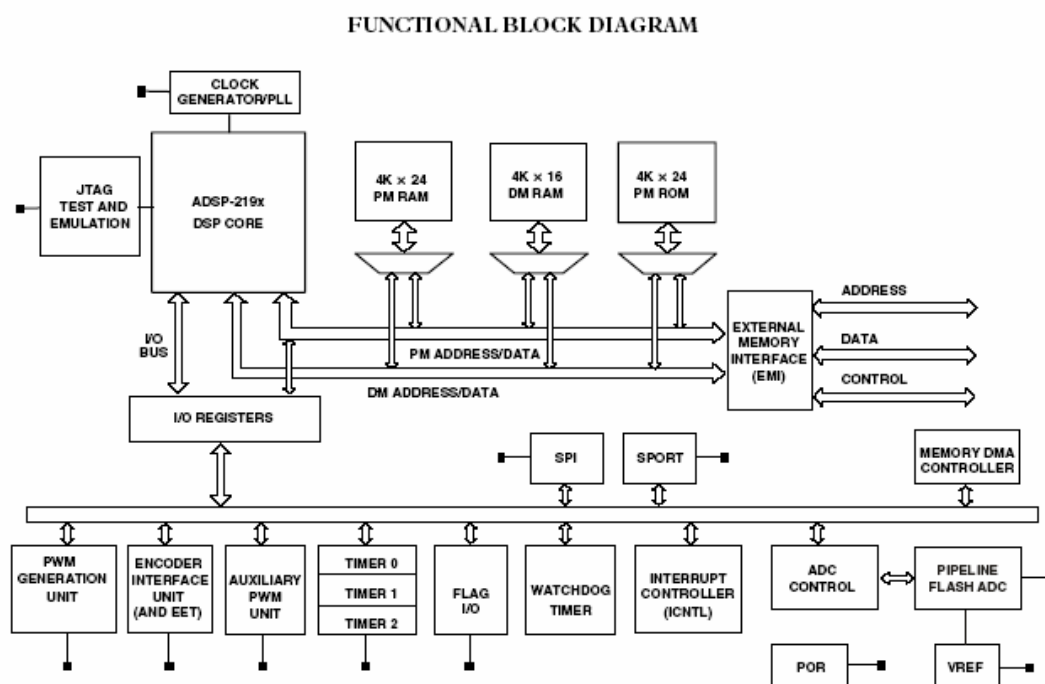


Figure 30 : Synoptique de l'ADSP-21990 d'Analog Device

Malgré la puissance de calcul à virgule fixe du DSP, Les ADC pipeline de 6 cycles intégrés ne permettait d'avoir les données échantillonnées qu'au bout de ~ 740 ns. Reste ensuite le traitement des données qui pouvait être rapide mais demandait quelques cycles d'horloge. Les DSP offrent aussi une programmation plus facile en langage C dans un environnement pour l'émulation et le débogage. Mais le parallélisme de traitement des données est limité (cf. p.23 « définition des modes de protection »).

Le choix s'est alors porté sur les FPGA, qui ont pour premier avantage la possibilité d'être optimisé pour l'application voulue : *la sécurité rapide de perte faisceau* par exemple. En effet, ils permettent de programmer la totalité du circuit : les fonctions logiques, mais également les flip-flops, les reset, l'interconnexion des modules, les entrées/sorties.

Les FPGA sont des circuits très complexes et très grands, qui peuvent avoisiner plusieurs millions de transistors avec les technologies actuelles. Il existe principalement trois types de FPGA :

- FPGA programmés par RAM, telles que fabriqués par Xilinx Corporation ou Altera,
- FPGA programmés par mémoire non-volatile, par exemple EEPROM, fabriqués par Lattice Semiconductor,
- FPGA programmés par fusible, produits par Actel.

Chaque méthode a ses avantages et ses inconvénients. Tout dépend du type d'application souhaitée. Par exemple, les FPGA programmés par RAM ou EEPROM sont très utiles pour les prototypes ou les applications commerciales. Par contre, on utilise les FPGA avec fusible pour les applications embarquées ayant besoin de beaucoup de fiabilité. Quoiqu'il en soit, le type de programmation (RAM, ROM ou fusibles) n'influe pas sur la manière dont on conçoit le FPGA et ne se répercute pas au niveau du VHDL.

Nous avons choisi les FPGA de Xilinx car il propose un large choix de composants. L'outil de développement (ISE Webpack) complet est bien intégré sous une seule interface, il permet une utilisation plus facile pour l'étude d'un projet de conception ainsi que sa simulation. Au contraire d'Altera par exemple, dont les outils de développement étaient séparés et plus difficile à utiliser. Reste aussi des outils de développements tel que Synopsys qui permet la synthèse et la simulation sur des cibles FPGA de différents fabricants mais très coûteux.

ISE est gratuit pour la gamme d'entrée de Xilinx (les Spartan , tous les CPLD et quelques Virtex), cela implique des projets de taille synthétisable sur ces cibles. Dans notre cas, nous n'avons pas rencontré de limitation pour la partie développement du projet, les limitations sont apparues en simulation.

Nous avons choisit le plus grand circuit disponible en boîtier CMS de la famille Spartan [5] : le XC2S200. Nous avons volontairement éviter les circuits ayant plus d'entrées/sorties mais vendus sous des boîtiers BGA (broches de type matrice à bille en dessous du circuit) pour faciliter la réalisation du PCB, malgré le fait que la largeur des broches du boîtier CMS soient de 220  $\mu\text{m}$ . Ce circuit possède 208 broches, donnant la possibilité d'avoir 140 entrées/sorties et 200 000 portes réalisant 1176 CLBs et 75264 RAM bits.

Le FPGA est constitué de CLB (Configurable Logic Block), qui sont des éléments logiques génériques programmables (Figure 31). Ces CLBs sont interconnectés grâce à un

réseau dense de canaux de routage qui parcourent tout le circuit. On trouve aussi des modules qui gèrent chacune des entrées/sorties du circuit.

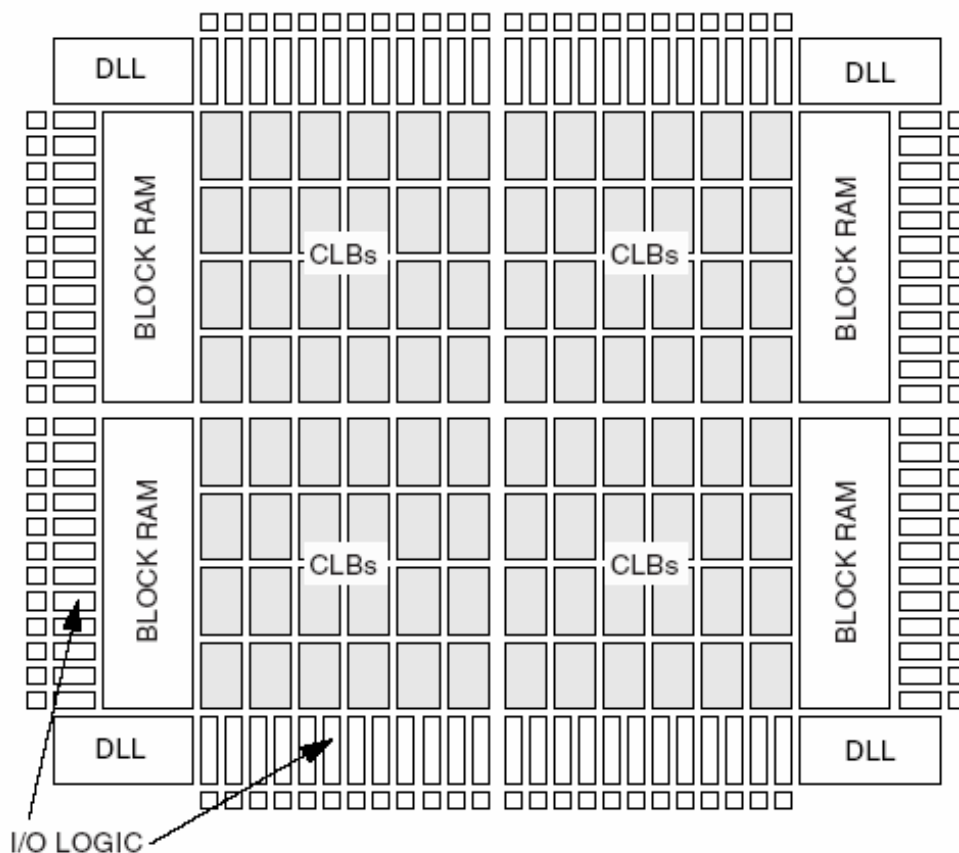


Figure 31 : Structure de base d'un FPGA, Configurable Logic Block

Chaque CLB est constitué de quatre cellules LCs (Logic Cells) qui intègre chacune un générateur de fonction à quatre entrées, une logique de propagation de la retenue (carry logic pour les calculs arithmétiques), et un élément de mémorisation. Ces CLBs sont organisés par 2 tranches similaires de deux LCs (Figure 32).

Les IOBs (Input/Output Block) permettent l'interconnexion entre la logique interne et les circuits externes. (Figure 33). Les trois registres IOB peuvent fonctionner comme une bascule sur niveau ou comme flip-flop sur front. Ils ont une horloge (CLK) en commun et chacun un Clock Enable (CE). De plus, ils partagent un signal Set/Reset (SR) configurable pour chaque registre en position synchrone set ou reset et asynchrone preset ou clear.

Chaque entrée ou sortie de l'architecture doit être connectée à un buffer d'entrée (IBUF) ou de sortie (OBUF) permettant l'interconnexion à la logique interne. Chaque sortie inclut un buffer à sortie trois état, le signal peut être routé directement au buffer de sortie ou via un flip-flop de sortie. Il est possible de configurer une entrée/sortie ou un vecteur d'entrée/sortie en mode bidirectionnel (INOUT), cela est possible en se positionnant en haute impédance lorsqu'un hôte veut écrire sur les broches bidirectionnelles.

La synchronisation externe du SPARTAN2 est optimisée pour 4 entrées d'horloge pour la synchronisation interne des données. Ces quatre entrées utilisent des buffers d'entrées appelés « Global buffers » (IBUFG) dont les sorties sont connectées à un réseau de

distribution globale des horloges du FPGA (Figure 34). Nous les avons toutes utilisées pour le projet final.

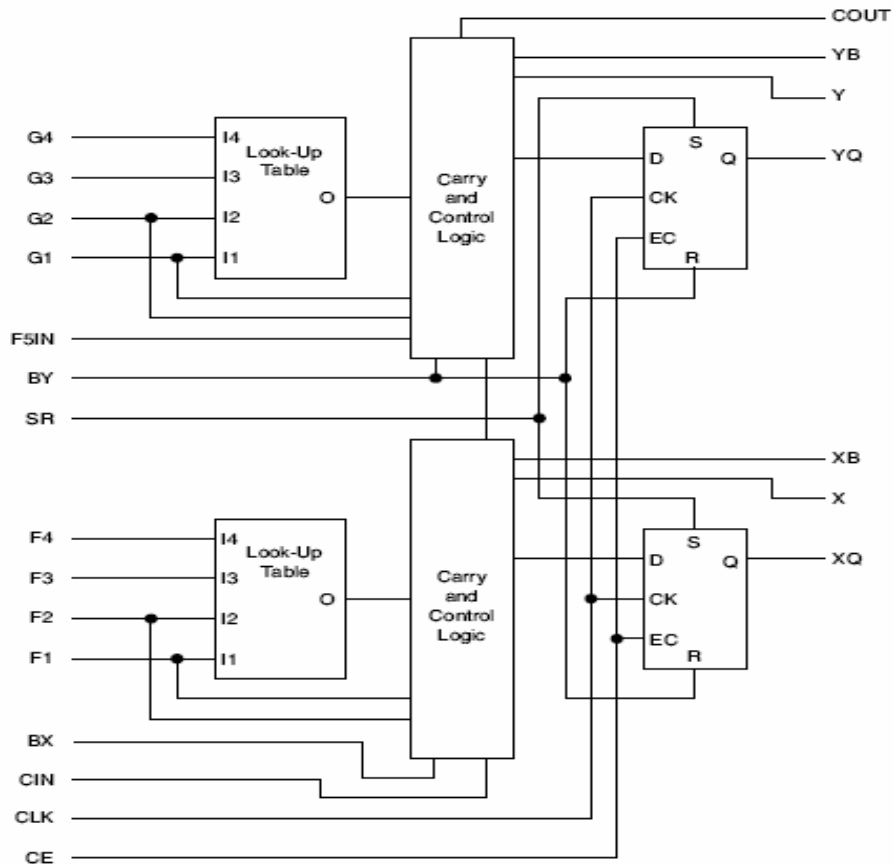


Figure 32 : Structure de base d'un CLB : Logic Cell

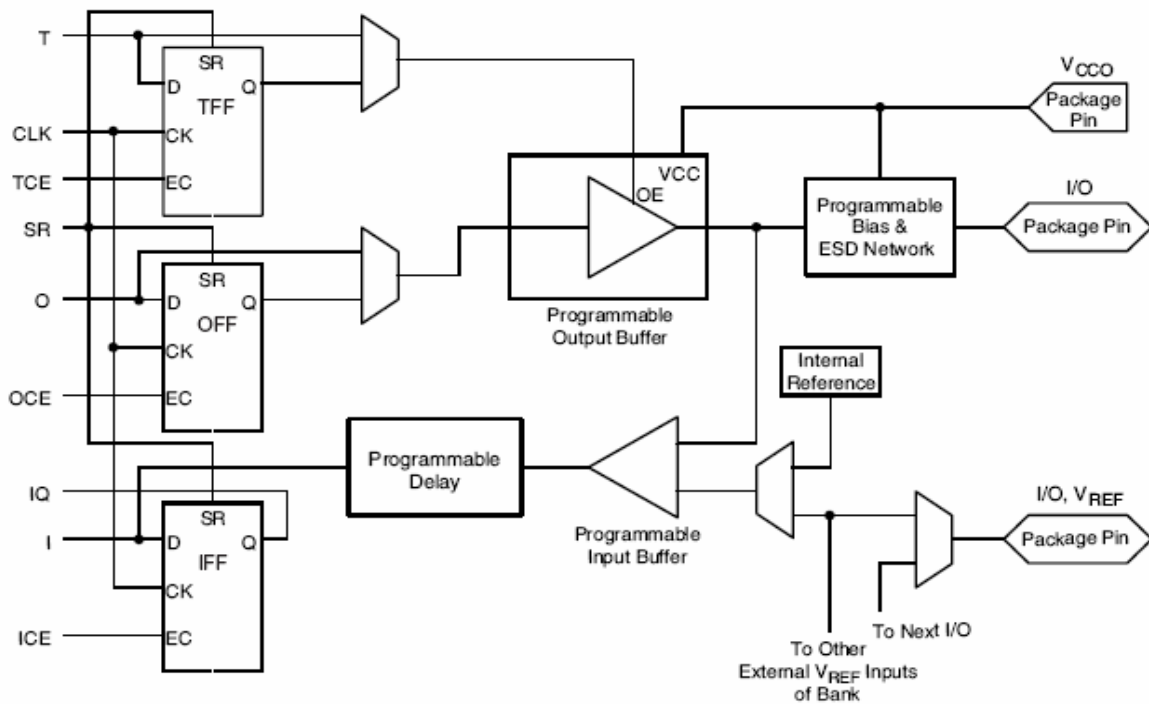


Figure 33 : Bloc d'entrée/sortie d'un FPGA : IOB



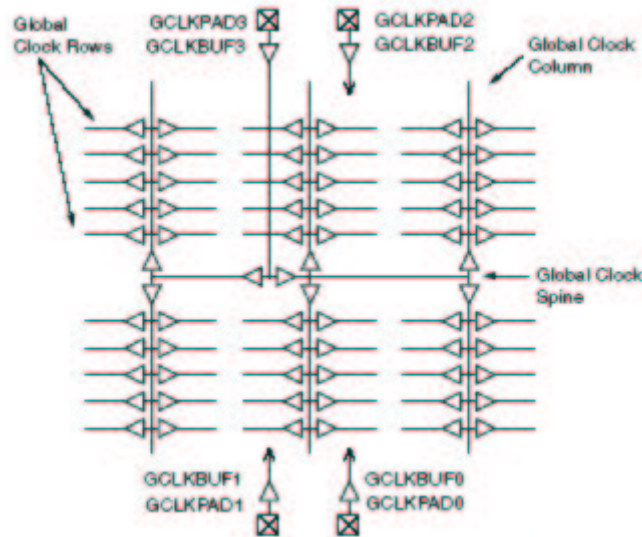


Figure 34 : Réseau de distribution global des horloges dans le SPARTAN2

La famille SPARTAN offre, en outre, la possibilité d'utiliser 16 standards de pilotage d'entrées/sorties (E/S). Cela est dû à la prolifération de nombreux standard d'alimentation en très basse tension (Tableau 5). Sur notre FPGA, nous avons 8 banques d'E/S configurables avec 8 standards différents. Le choix se fait à travers la broche Vcco de chaque banque d'E/S.

I/O Standard	Input Reference Voltage ( $V_{REF}$ )	Output Source Voltage ( $V_{CCO}$ )	Board Termination Voltage ( $V_{TT}$ )
LVTTL (2-24 mA)	N/A	3.3	N/A
LVC MOS2	N/A	2.5	N/A
PCI (3V/5V, 33 MHz/66 MHz)	N/A	3.3	N/A
GTL	0.8	N/A	1.2
GTL+	1.0	N/A	1.5
HSTL Class I	0.75	1.5	0.75
HSTL Class III	0.9	1.5	1.5
HSTL Class IV	0.9	1.5	1.5
SSTL3 Class I and II	1.5	3.3	1.5
SSTL2 Class I and II	1.25	2.5	1.25
CTT	1.5	3.3	1.5
AGP-2X	1.32	3.3	N/A

Tableau 5 Standard d'entrées/sorties compatibles avec le Spartan2

Dans notre projet, nous n'utilisons que le standard LVTTL. Le microcontrôleur, les horloges et d'autres signaux connectés au FPGA sont en logique TTL. Dans la famille SPARTAN, seul le SPARTAN2 tolère le format TTL, c'est aussi pour cette raison que nous l'avons choisi.

## 2.2 Outil de développement : ISE

Le logiciel ISE WebPack [6] permet de programmer une large gamme de composants de Xilinx. Les composants sont de plusieurs familles : CPLD, FPGA et des mémoires. Cependant pour pouvoir avoir accès à toute la bibliothèque du fabricant, il faut acheter une licence. Ce logiciel permet de réaliser le design, le synthétiser, l'implémenter et l'optimiser pour la cible choisi. Enfin, il génère un fichier de configuration pour la programmation de la cible.

Il est possible de concevoir les architectures des fonctions à réaliser en VHDL (Figure 35), Verilog ou par l'intermédiaire d'une interface graphique en utilisant les fonctions élémentaires disponibles. Mais surtout, nous pouvons combiner toutes ces méthodes. L'architecture réalisée et compilée, un logiciel intégré (HDL bench) permet la création de banc de test automatique en vue d'une simulation et d'une vérification.

### 2.2.1 Entrée de design

Les langages de description de systèmes matériels (VHDL, Verilog) sont saisis sous un éditeur de texte intégré :

```

-----
-- Title       : 4 bits CODER
-- Project     : TTF2 secdiff
-----
-- File        : C4_2.vhd
-- Author      : Abdallah Hamdi <ahamdi@cea.fr>
-- Company     : CEA/Saclay
-- Date        : 2004/02/25
-----
-- Description : code 4 bits in 2 bits.
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;

entity C4_2 is
    Port ( E : in std_logic_vector(3 downto 0);
          S : out std_logic_vector(1 downto 0));
end C4_2;

architecture Behavioral of C4_2 is

begin

process(E)
    Begin
        case E is

            when "0001" => S <= "00"; -- SUMSB
            when "0010" => S <= "01"; -- SULSB
            when "0100" => S <= "10"; -- SDMSB
            when "1000" => S <= "11"; -- SDLSE
            when others => S <= "11"; -- Default statement added (for values XX, UU, ZZ...)

        end case;

    end process;

end Behavioral;

```

Figure 35 : Exemple d'un fichier VHDL pour un codeur

L'autre possibilité est de décrire son architecture par l'interface graphique Engineering Capture System (ECS) :

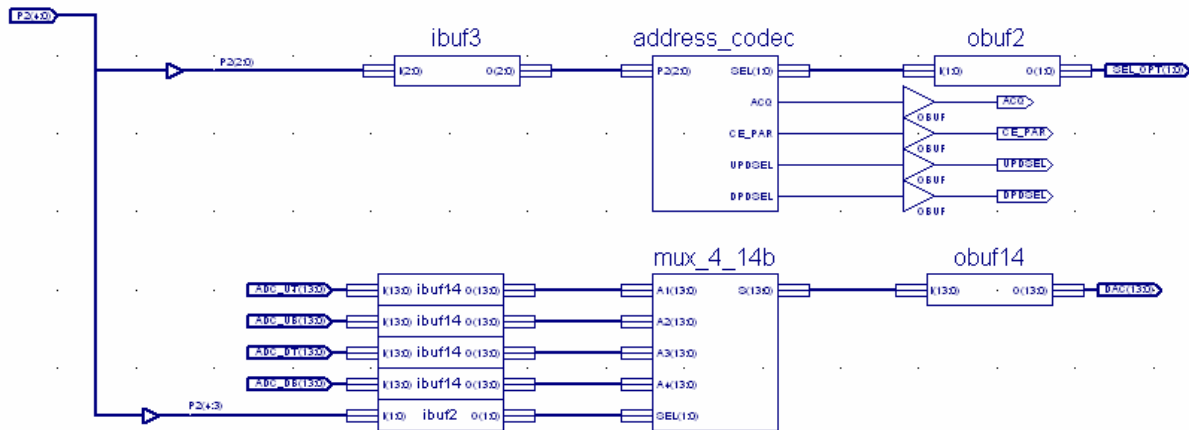


Figure 36 : Exemple d'entrée de design avec ECS

Cette interface (Figure 36) procure une bibliothèque de primitives (ET, OU) et de fonctions macros (ensemble de primitives réalisant une fonction) synthétisables pour une famille donnée de composant Xilinx.

### 2.2.2 Synthèse

La synthèse consiste à traduire une description textuelle d'une fonction en une interconnexion de modules physiques [7]. Ces modules correspondent à des cellules d'une librairie définie pour un composant cible donné. Ce niveau de description est appelé RTL pour Register Transfer Level, ce vocable désigne la nature synchrone des systèmes décrits (mémoires + combinatoires).

Les étapes d'entrée de design et de synthèse permettent de générer un fichier de netlist (Figure 37) EDIF (Electronic Data Interchange Format).

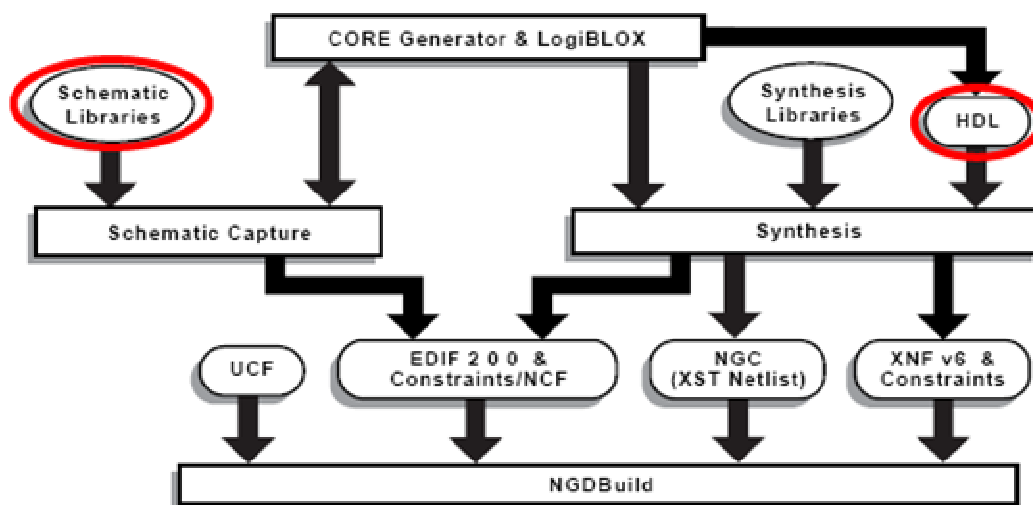


Figure 37 : Flot d'entrée de design pour ISE

### 2.2.3 Implémentation de design

Elle consiste en plusieurs étapes (Figure 38) que l'on peut résumer par :

a) *La traduction* : génère, à partir du fichier de NETLIST et du fichier de contrainte (UCF pour User Constraints File), un fichier de description logique NGD (Native Generic Database).

b) *Le mapping* : à partir du fichier NGD, effectue une vérification logique du design (Design Rule Check) et réalise les correspondances entre les fonctions logiques à réaliser et les composants de base utilisables sur la cible FPGA (IOB, CLB et autres). Le résultat est un fichier NCD (Native Circuit Description).

c) *Le Placement-routage* : cette phase consiste à disposer les cellules nécessaires à la réalisation de la fonction, de telle sorte que les contraintes imposées par le concepteur soient respectées :

- optimisation en surface,
- optimisation en vitesse (délai de propagation entre porte logique),
- combinaison des deux.

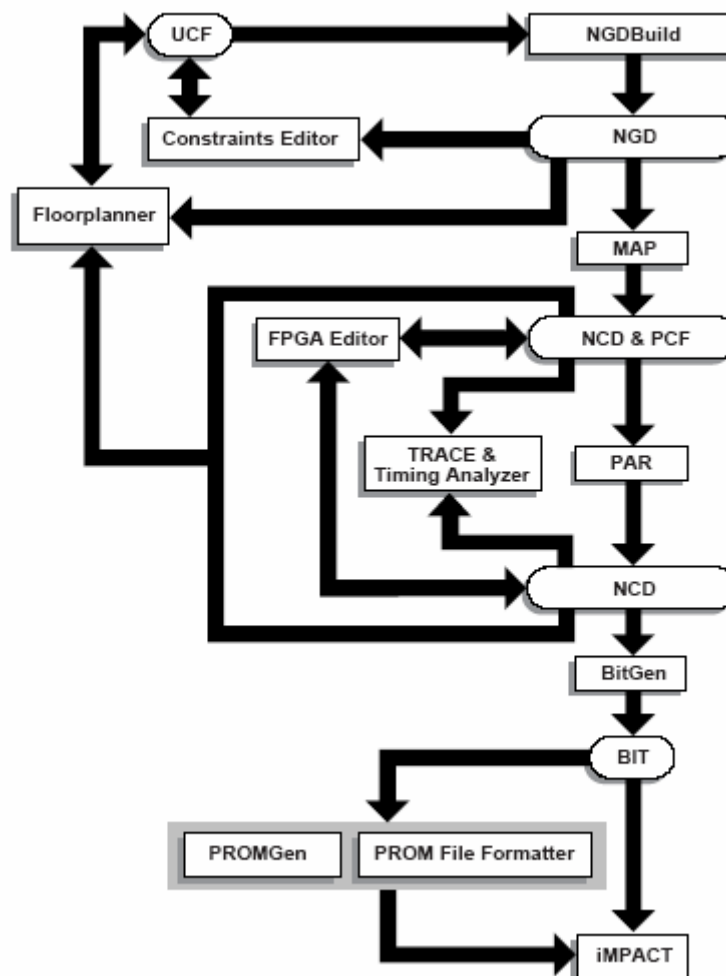


Figure 38 : Flot d'implémentation de design pour FPGA

### 2.2.4 La configuration du FPGA

Il y a deux phases nécessaires pour la configuration d'un FPGA Xilinx [8] :

*Phase 1* : Implémentation à l'aide d'outil logiciel qui permettent d'adapter l'architecture à la cible FPGA souhaitée en produisant un fichier de configuration bitstream (\*.bit).

*Phase 2* : Configuration qui consiste à télécharger les données de programmation (\*.bit) dans le FPGA en utilisant une source externe, comme une PROM ou un microprocesseur.

Les FPGA SPARTAN2 sont volatiles car basé sur une technologie SRAM. Le composant perd sa configuration si l'alimentation est coupée. Typiquement, les FPGA utilisent une mémoire externe, telle qu'une PROM pour prévenir des risques de coupure électrique dans les environnements industriels. Néanmoins, les FPGA peuvent aussi être programmés à l'aide d'un câble sans mémoire externe en utilisant le mode JTAG.

Modes de configuration possible pour un FPGA:

- **JTAG** : est un mode standard (IEEE 1149.1 ou 1532) de programmation série. Une logique externe, microprocesseur ou autres, est utilisée pour piloter les 4 broches spécifiques, Test Data In (TDI), Test Mode Select (TMS), Test Data Out (TDO) et Test Clock (TCK). Les données sont chargées à la fréquence de 1 bit/coup d'horloge TCK.
- **SelectMAP et Slave-Parallel** : ces modes permettent l'écriture ou la lecture en parallèle sur des ports de largeur 8 bits. Pour ce faire, une horloge externe doit être fourni par un microprocesseur, un câble de transfert ou un autre FPGA. Ce mode est utilisé quand la vitesse de configuration est un critère important.
- **Master-Serial** (Figure 39) est le mode le plus simple de configuration FPGA. Le FPGA charge les données de configuration d'une PROM. Le FPGA utilise son oscillateur interne pour cadencé le transfert d'1 bit/CCLK.
- **Slave-Serial** utilise aussi une horloge externe et permet le téléchargement en série de plusieurs FPGA. La configuration peut se faire à partir d'un microprocesseur, d'un câble de téléchargement ou d'un autre FPGA. Le taux de transfert est aussi d'1 bit/CCLK.
- **Express mode** est un mode analogue au Slave-Serial mis à part que le transfert est fait octet/CCLK au lieu d'1 bit/CCLK. Il nécessite aussi une horloge externe.

L'utilisation d'une ISP-PROM permet de reconfigurer un minimum de 100 000 fois le système. Pour ce projet, nous avons choisit la configuration via une PROM car nous n'avons pas de grande série à réaliser, le coût par unité de carte n'est pas le critère le plus important. Nous avons opter pour le mode Master-Serial car adapter à l'utilisation d'une PROM externe.

Le programme est chargé automatiquement à la mise sous tension (Powerup) du système ou sur commande de configuration en mettant la broche « *Program* » du FPGA à l'état bas. Le mode de configuration est sélectionné par l'état des trois broches de mode FPGA (M0, M1, M2). En mode Master-Serial, le FPGA charge automatiquement le programme de configuration d'une mémoire externe.

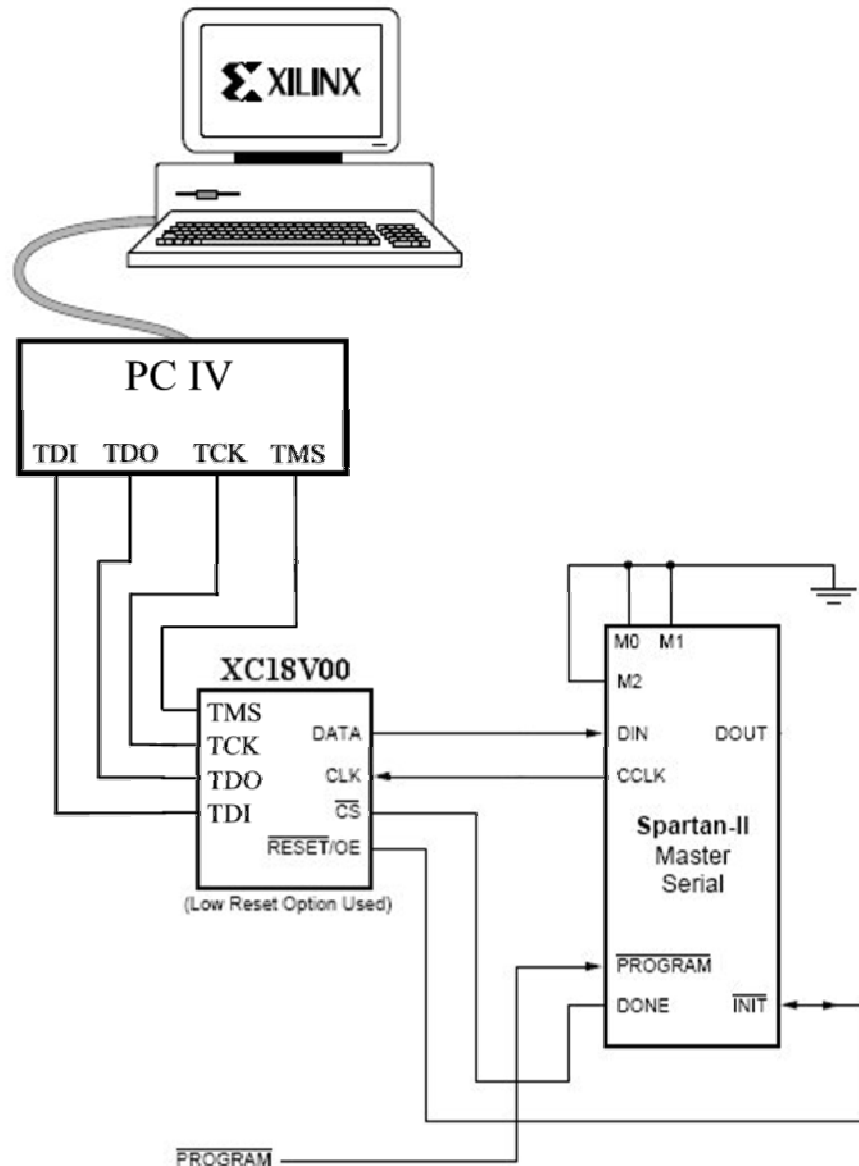


Figure 39 : Mode de configuration Master Serial

L'interface « Parallel Cable IV » permet de mettre en forme les signaux de commandes du microprocesseur. En plus des quatre signaux (TDI, TDO, TCK, TMS), le PCIV doit être connecté à la masse et à la tension de référence  $V_{cc0}$  de la PROM.

A la mise sous tension ou Powerup (Figure 40) ou sur commande de reconfiguration (« Program »=0), le FPGA vérifie que les banques contenant les broches de configuration (Bank 2) et du cœur du circuit ( $V_{ccint}$ ) sont alimentés. Pendant l'effacement de la zone mémoire de configuration, la broche « Init » est maintenue à l'état bas. Dès que « Init » repasse à l'état haut, le FPGA entre en mode Master Serial car les broches de mode sont à l'état bas ( $M0=0$ ,  $M1=0$ ,  $M2=0$ ). Des données sont lues de la PROM séquentiellement sur une seule ligne de données (DIN). La synchronisation est fournie par le FPGA sur chaque front de montée du signal CCLK. Finalement, une vérification appelé CRC (Cycle Redundancy Check) basé sur la comparaison d'un mot contenu dans le fichier de configuration et un mot calculé par le FPGA sur la trame des données reçu, active le mode opérationnel du FPGA.

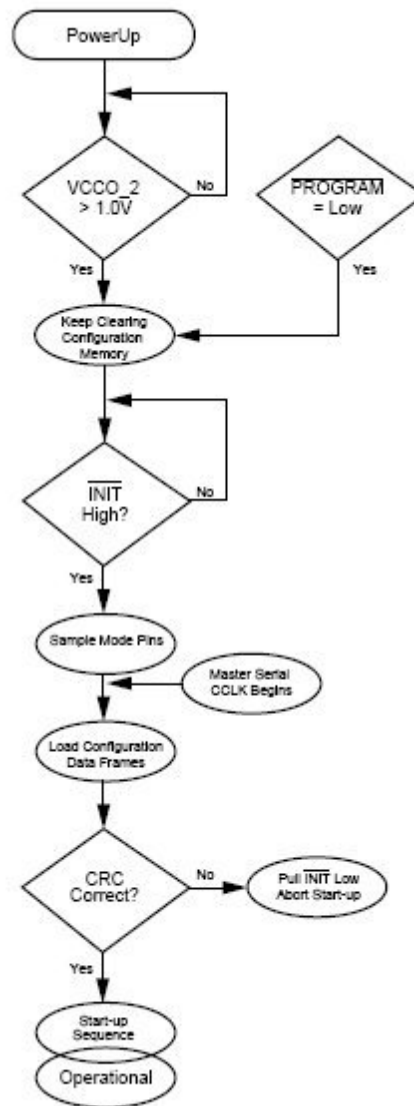


Figure 40 : Organigramme de configuration du FPGA

### 2.3 Outil de vérification : MODELSIM

Cette application est un simulateur pour les langages de description matériels (VHDL, Verilog) conçu par Mentor Graphics. Une version gratuite Modelsim MXE (Modelsim Xilinx Edition) développée en partenariat avec Alliance Foundation, éditeur d'ISE, permet d'évaluer ce logiciel jusqu'à 500 lignes de code sans dégradation des performances. Après cette limite, il est possible de travailler mais à la condition d'être patient.

Pour vérifier le bon fonctionnement d'un design (Figure 41), un concepteur doit passer par trois phases :

- la simulation comportementale (RTL simulation),
- la simulation fonctionnelle (functional simulation),
- la simulation post placement-routage (timing simulation).

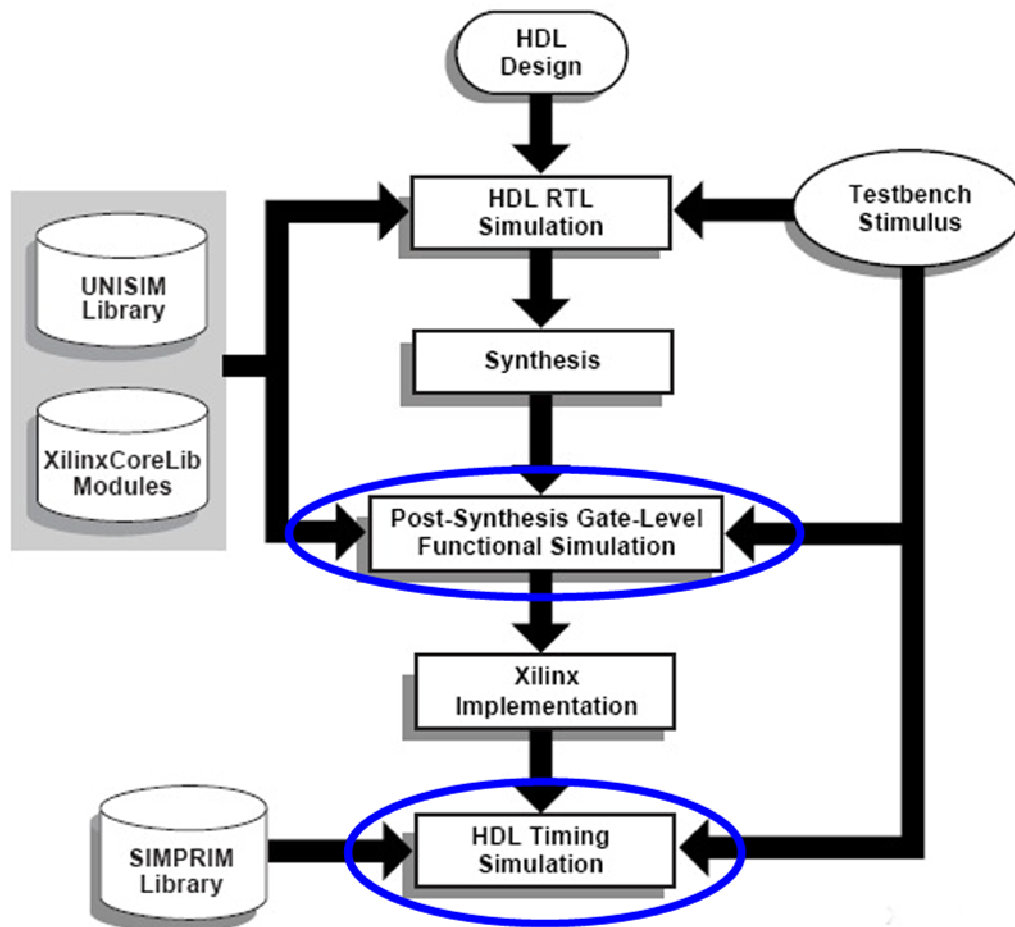


Figure 41 : Etapes de simulation pour design en langage HDL

Après la synthèse du design (Figure 41), ISE crée un modèle structurel en VHDL grâce à son synthétiseur à partir du fichier NETLIST. Il génère de plus un fichier SDF (Standard Delay Format) en utilisant les timings extraits de l'analyse physique après l'étape de placement routage.

### 2.3.1 La simulation comportementale

Le banc de test VHDL s'applique à un modèle au niveau RTL avant la synthèse, ce modèle n'utilise pas les composants spécifiques à la cible mais permet de vérifier la syntaxe et la fonctionnalité souhaitée. De plus, ce modèle est portable car indépendant d'une cible. De cette manière, il est possible de se fabriquer une librairie de fonction réutilisable sur différentes cibles FPGA, c'est un des points fort du VHDL. Dans notre cas, nous ne souhaitons utiliser qu'une cible et nous voulions que notre circuit soit opérationnel le plus tôt possible. C'est pourquoi nous avons évité cette étape pour passer directement à la simulation fonctionnelle du modèle structurel.

### 2.3.2 La simulation fonctionnelle ou post placement routage sans timing

Le banc de test s'applique au modèle structurel créé par le synthétiseur. Cette simulation ne prend pas en compte les informations de timing, elle ne fait pas appel au fichier SDF, c'est à dire que tous les calculs et traitements se font de manière instantanée. Dès qu'une entrée change, la sortie est mise à jour sans délai. Les seuls phénomènes temporels



sont ceux liés aux registres et autres mémoires qui sont cadencés par une horloge de synchronisation des données. Cette simulation n'est pas proposée par MXE, nous l'avons réalisée car elle simule le comportement de la cible sans les informations de retard de propagation dû au routage.

### 2.3.3 La simulation post placement routage

Celle-ci utilise le fichier structurel créé après l'étape de placement routage et intègre le fichier SDF. Le fichier SDF contient une description des retards estimés des instances du design. Typiquement, il y a trois valeurs de retard (Min, Typ, Max) mais la version d'ISE que nous avons ne fournit que les valeurs maximums des retards. S'il fallait choisir, c'est cette simulation pour la vérification du design que nous aurions prise car c'est la seule qui garantisse un succès après configuration du FPGA.

## 2.4 Conception d'une carte d'évaluation FPGA

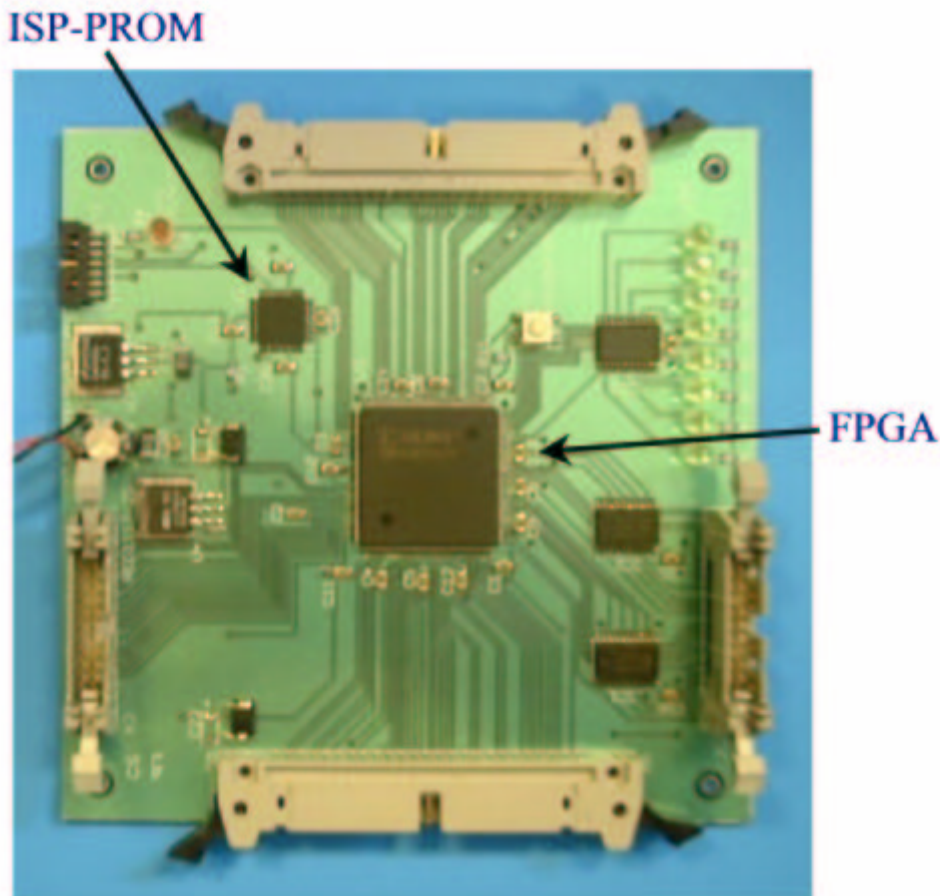


Figure 42 : Carte d'évaluation

Dans un premier temps, nous avons étudié le comportement de composants FPGA à l'aide des outils fournis par les différents fabricants, de façon à choisir une technologie. Le choix du fabricant fait, il fallait concevoir une carte de test permettant l'application du type de configuration choisie et la validation d'architectures réalisant des traitements partiels du système (opérations arithmétiques signées, logique séquentielle et combinatoire).

### Caractéristiques de la carte d'évaluation :

- FPGA SpartanII : XC2S200PQ208-I5
- ISP-PROM : XC18V02VQ44
- 8 LEDs pilotée par le FPGA pour permettre la vérification rapide des fonctionnalités à tester.
- Reconfiguration au vol par un bouton poussoir
- Configuration en mode JTAG de la PROM
- 2 buffers externes 16 bits en technologie LVTTL (3.3V) bien que le FPGA tolère le 5V en TTL. Un buffer de sortie prévu pour conduire les signaux sur des nappes de plus ou moins grande longueur vers la carte de National Instruments PCI6532 de 32 E/S TTL (5V).

## 2.5 Vérification et validation de base

Les objectifs sont la mise en œuvre des outils Xilinx pour la programmation du FPGA et le test de quelques fonctions de base pour les différents modes de fonctionnement de la sécurité différentielle.

### a) Configuration

Pour la cible FPGA choisit : XC2S200 , il faut une capacité d'au moins 1335840 bits de configuration, nous avons utilisé une PROM XC18V02 de chez Xilinx avec une capacité de 2097152 bits de configuration.

Nous avons choisit de télécharger les configurations (bitstream) dans l'ISP-PROM au moyen d'un câble parallèle : le *Parallel Cable IV* Nous utilisons le mode JTAG pour la programmation via le logiciel fourni par Alliance Foundation : *Impact* intégré dans les menus d'ISE WebPack.

### b) montage et test

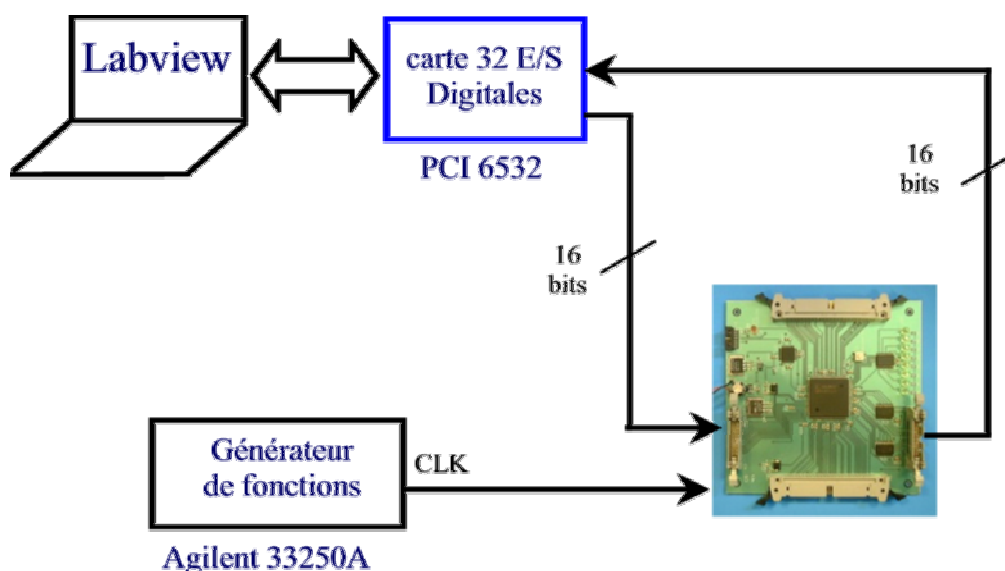


Figure 43 : Montage de test pour carte d'évaluation FPGA

Nous ne présentons ici qu'un test de la carte d'évaluation. Ce test englobe les points que nous voulions éclaircir à l'aide de cette carte :

- la mise en œuvre du FPGA,
- la configuration en mode JTAG de la PROM,
- la configuration en mode MASTER SERIAL du FPGA,
- l'implémentation de calculs arithmétiques nécessaire au traitement de la sécurité différentielle.

Pour évaluer le FPGA, nous avons développé un module de calcul arithmétique (Figure 44) sur 8 bits en nombre signé sous ISE, il consistait à réaliser 4 opérations arithmétiques successives (addition, soustraction, multiplication de l'entrée 1 par l'entrée 2 et une division par une valeur multiple de  $2^n$  de l'entrée 1). Les opérations étaient cadencées par deux compteurs en cascade effectuant une division par 8 de l'horloge externe.

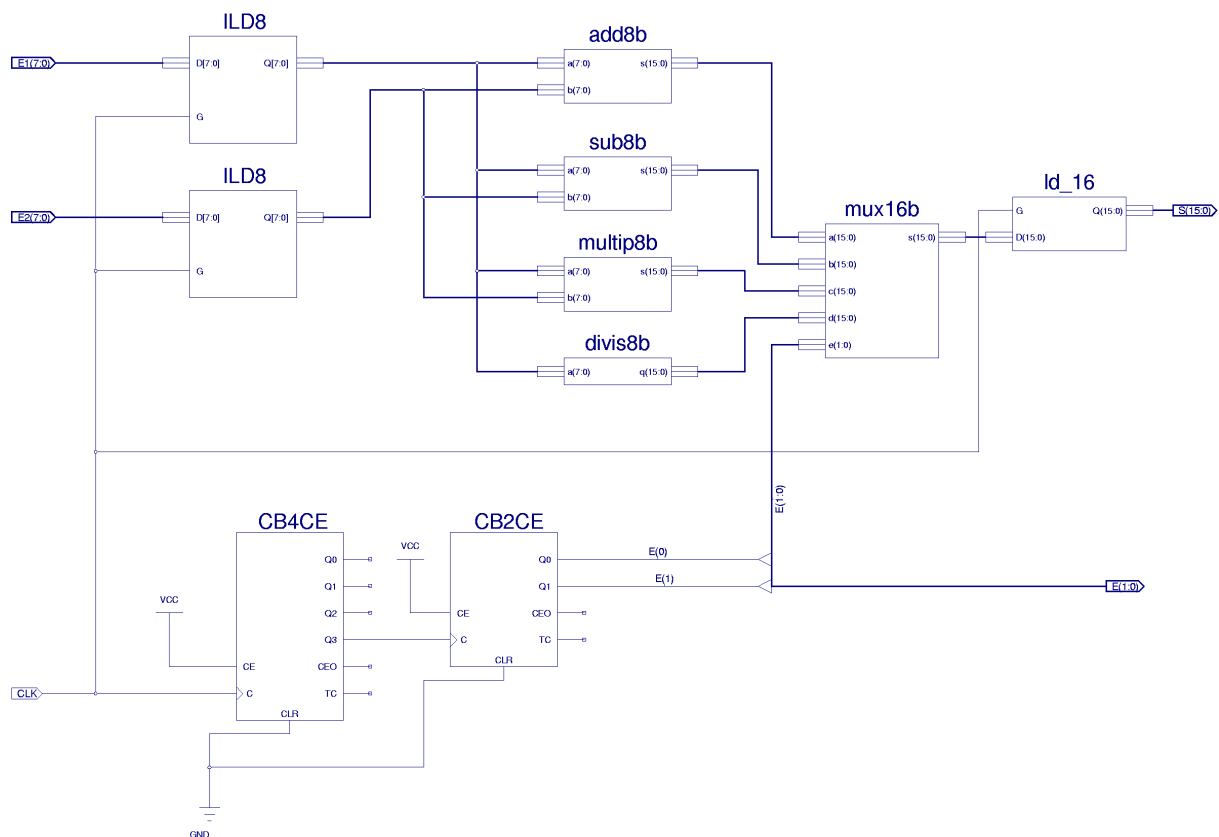


Figure 44 : Module FPGA de calcul arithmétique pour la carte d'évaluation

Avant chaque configuration du FPGA, les designs sont validés par la simulation post placement routage. Les simulations passées avec succès ont toujours été synonymes de réussite des tests matériels.

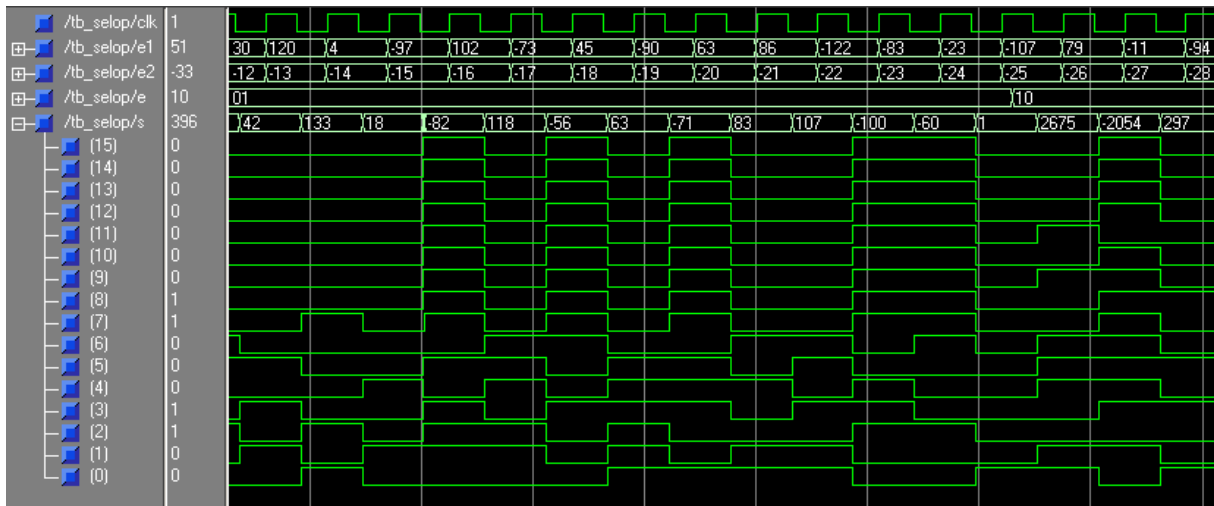


Figure 45 : Simulation post routage d'un module de calcul arithmétique

Les données sont cadencées en entrée sur front montant et en sortie sur front descendant de l'horloge CLK.

L'instrument virtuel (VI) de Labview fait la conversion, du format décimal au format binaire de base, des données saisies par l'opérateur. Puis il génère les données converties sur les ports de sorties de la carte PCI 6532. Alors, le FPGA (Figure 43) effectue le calcul et renvoie le résultat sur la carte PCI6532. Le VI acquiert les données numériques en provenance du FPGA et réalise la conversion en décimale signée avant de les afficher.

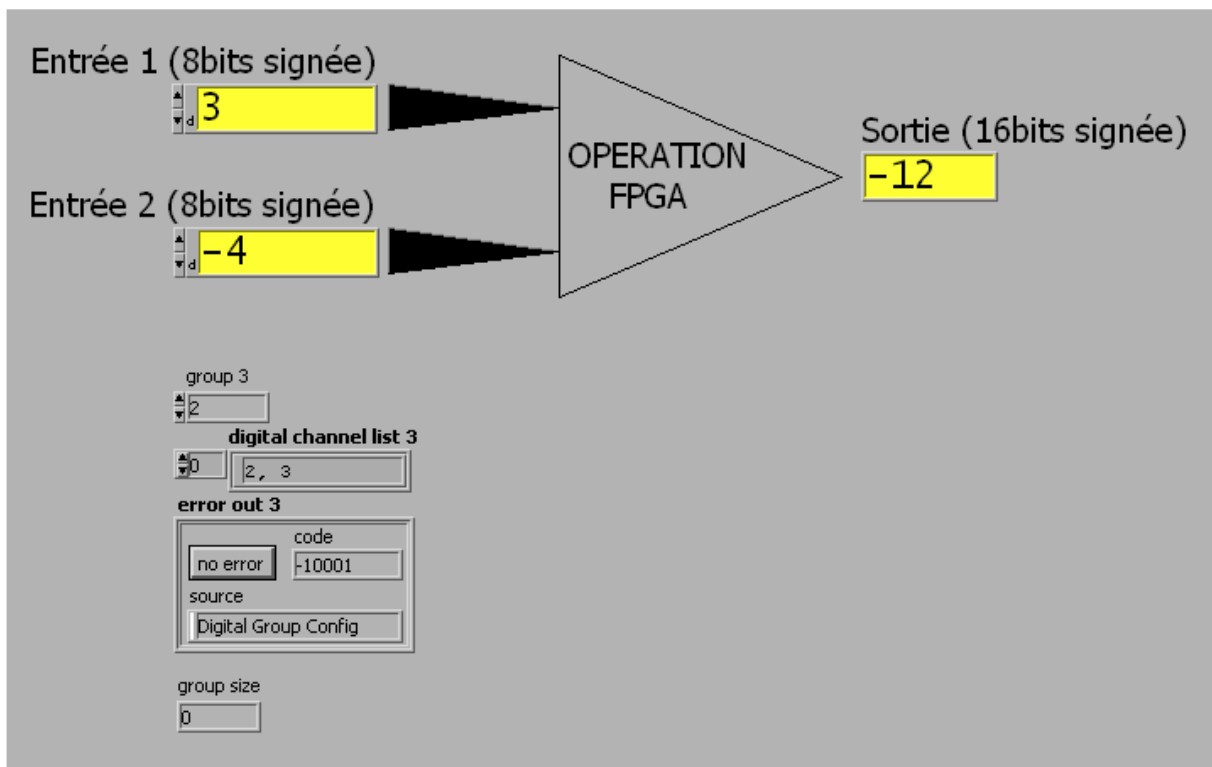


Figure 46 : Face avant de l'instrument virtuel pour le test de la carte d'évaluation, résultat d'une multiplication

D'autres applications ont été téléchargées et vérifiées au sein de cette carte comme par exemple un accumulateur additionneur synchronisé.

La réalisation de la carte a été très ralentie par des problèmes de fourniture de composants (3 mois pour le câblage de la carte). Malgré cela, cette phase a été franchie avec succès. En effet, les buts fixés pour cette carte d'évaluation : validation du système de configuration et implémentation de calculs de base ont été atteints.

C'est aussi pendant ces tests que nous avons été confrontés au problème de la division qui implique de s'intéresser au codage des nombres à virgules. L'opérateur de division disponible en langage VHDL de synthèse ne permet que les divisions par un nombre multiple de 2. Cela va de soit lorsqu'on sait que cette opération est réalisé par des registres à décalage. Une autre possibilité consistait à acheter des IP (Intellectual Property) de Xilinx. Ces modules, prêts à l'emploi, réalisent les fonctions recherchées (FFT, calcul sur des nombres à virgule, filtrage numérique) pour une cible donnée.

### 3. Interfaces et paramétrages

Le DOOCS (Figure 47) est le système principal de contrôle commande. Il a une structure de serveur connecté à un ensemble de station de travail SUN, SOLARIS et de PC sous UNIX. Développé par la collaboration TESLA en langage C++ dans le cadre du GAN (Global Accelerator Network).

Ce système doit intégrer toutes les informations de paramétrages et de diagnostics de l'accélérateur. Pour que le contrôle soit plus fiable, il est important de centraliser les informations en provenance de chaque cellule de contrôle et d'en donner l'accès rapidement. Ce contrôle sera possible en local, par un réseau LAN et par Internet pour les utilisateurs de la collaboration.

Ceci est encore plus vrai pour les systèmes de sécurité de la machine, la sécurité différentielle procure une interface au DOOCS pour le paramétrage des seuils de comparaison pour chaque mode d'alarme. De plus, un registre d'état du système sera accessible en lecture à tout moment à travers le BIS. Le BIS peut aussi ordonner un étalonnage du système pour le calcul du facteur de correction et l'ajustement des horloges d'échantillonnage des ADCs. La dernière interface, et non la moindre, est la commande d'inhibition du faisceau, elle se fait par une ligne RS-422 pour plus de fiabilité et est transmise au BIC.

Il y a plusieurs entités qui communiquent avec la sécurité différentielle (Figure 47) :

- le BIS : Beam Interlock System,
- le BIC: Beam Interlock Concentrator,
- le DOOCS: Distributed Oriented Object Control System.

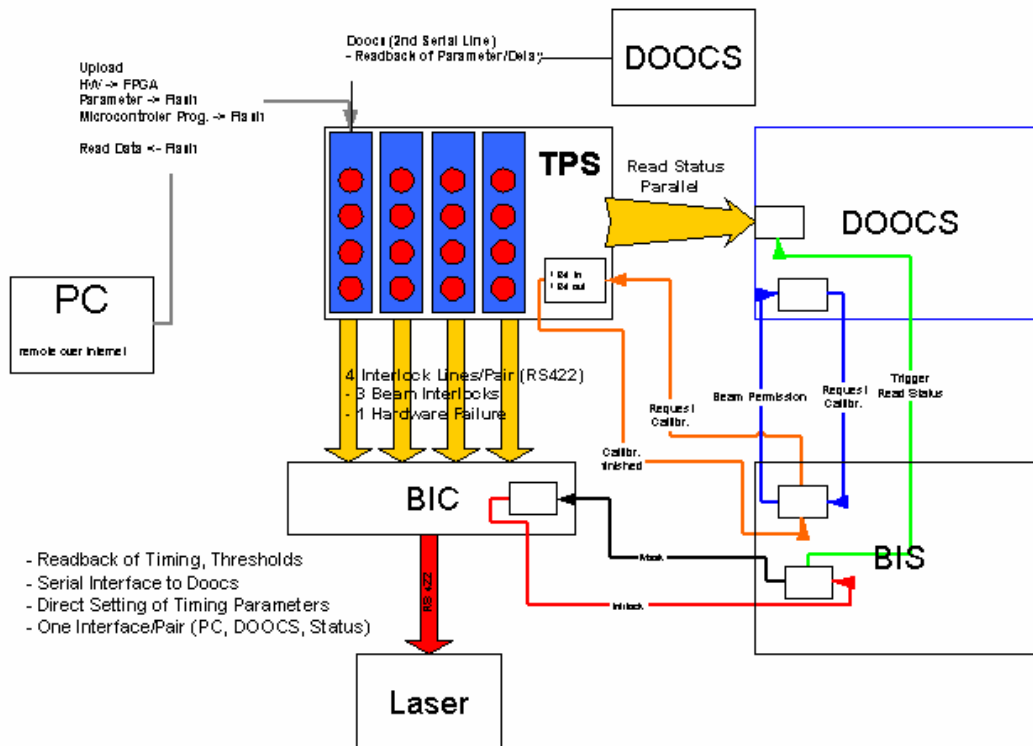


Figure 47 : Sécurité différentielle et son environnement

### 3.1 Signaux d'alertes (Interlocks) et d'états (Status)

Les Signaux d'alarme sont envoyés au BIC sous forme différentielle par liaison RS-422. L'interface est assurée par un driver de ligne différentielle 4 voies : DS26C31T. Les experts en diagnostic autour des accélérateurs estiment que pour un signal de protection, une simple liaison TTL est insuffisante à cause des problèmes de CEM.

Le registre d'état 8 bits du système et la commande CALREQ et CALDONE seront transmis en parallèle au Beam Interlock System par des opto-coupleurs photodarlington : 4N32 pour isoler galvaniquement la protection machine.

### 3.2 Communication avec le système de contrôle-commande

Utilisation d'un microcontrôleur : l'ADUC812 pour la communication des paramètres de seuil et du facteur de correction du tore aval au FPGA. La programmation est faite en langage C à l'aide d'un compilateur, émulateur et simulateur « Keil DK51 ». En plus de la communication, l'ADUC812 effectue l'étalonnage et le calcul du facteur de correction.

*Caractéristique technique :*

- 8 canaux ADC 12 bits 200 KSPS multiplexé
- 8 K octets de mémoire programme flash
- 640 octets de mémoire donnée flash
- 256 octets de mémoire donnée RAM
- adressage pour 16 Mo de mémoire donnée externe
- adressage pour 64 Ko de mémoire programme externe
- 9 sources d'interruption
- protocoles UART et SPI intégrée

*Les fonctions du microcontrôleur :**a) Transfert des paramètres :*

Le  $\mu$ C reçoit les paramètres de seuil du DOOCS par liaison RS-232. Il les stocke et les mets à jour dans le FPGA par une liaison série. Il calcule après étalonnage le facteur de correction du tore aval.

*b) Etalonnage :*

Le signal CALREQ (Calibration Request) est connecté à une broche configurée en interruption dans le microcontrôleur. Ce signal n'est transmis que lorsque le faisceau est en mode single shot : un seul paquet d'électrons est accéléré par macro-impulsion. A ce moment, la routine d'interruption réalise alors une série d'acquisition de couple (échantillons numériques, valeur binaire du délai correspondant) qui seront stockées dans la RAM externe. Ensuite, les calculs suivants sont effectués :

- le premier algorithme recherche le délai optimal pour l'acquisition de la valeur Top des tores amont et aval.
- le second calcule la différence de charge entre les tores amont et aval et en déduit un facteur de correction.

Enfin, le microcontrôleur écrit un octet sur le port P2 pour signifier la fin de l'étalonnage au FPGA qui repasse alors en mode sécurité différentielle.

*c) RAZ et contrôle du mode de fonctionnement FPGA :*

Le microcontrôleur commande l'entrée Master Reset MRST permettant la remise à zéro des sorties des registres du FPGA sauf les paramètres de seuils et le facteur de correction qui doivent être écrasés pour être modifiés. Les modes de fonctionnement du FPGA : Sécurité différentielle ou étalonnage sont sélectionnés à l'aide d'octets écrits sur P2.

#### 4. Conception du système complet

Au lieu de quatre cartes initialement : 2 modules ADC, un module de traitement numérique et une carte d'interface avec le système de contrôle (DOOCS et BIS), nous avons décidé de réaliser un système réunissant l'essentiel des fonctions sur une seule carte (Figure 48).

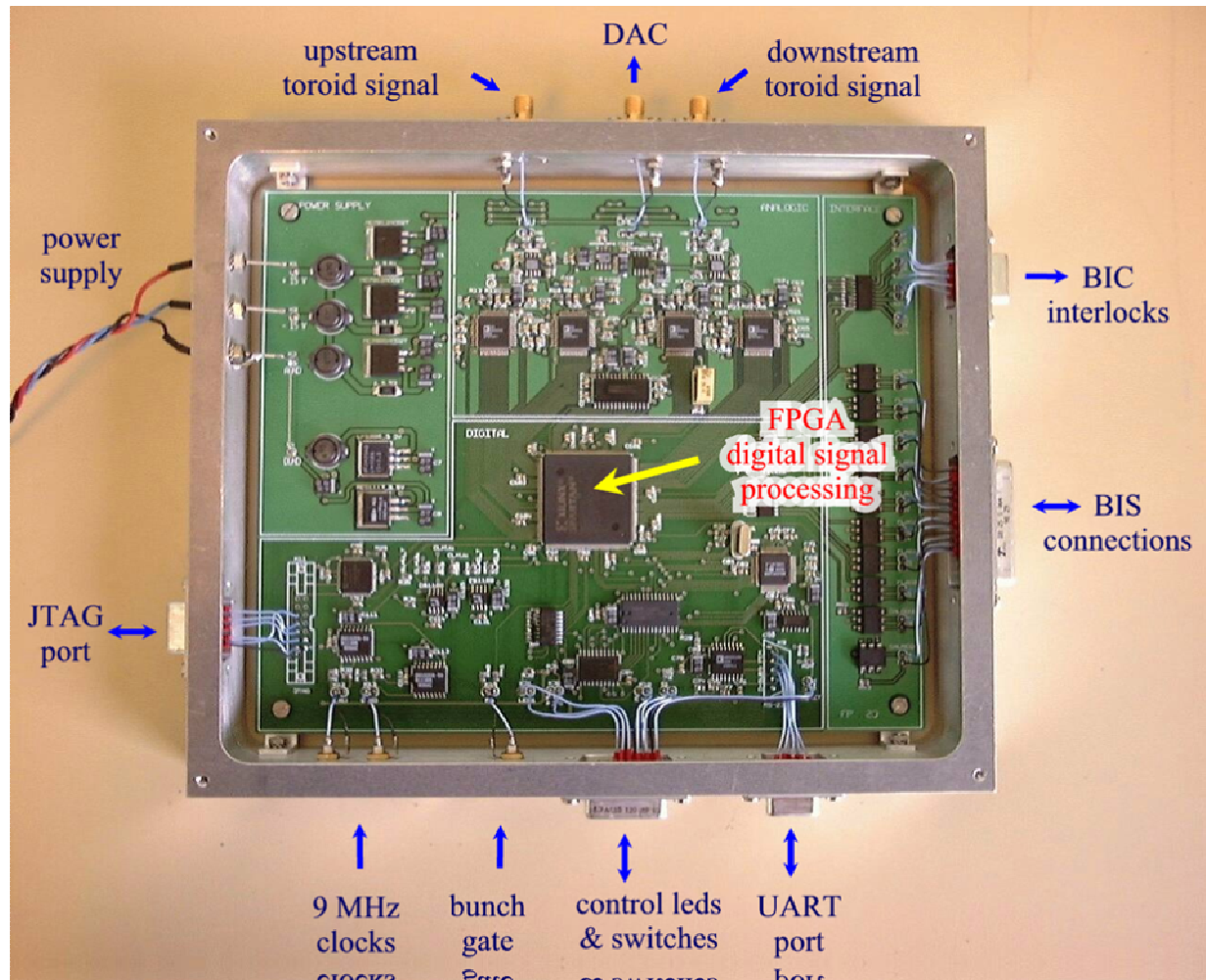


Figure 48 : Carte « Toroid Protection System »

##### 4.1 Description des fonctions intégrées de la carte TPS

Le synoptique (Annexe I : Synoptique de la carte TPS) décrit l'architecture générale du système. Il présente tous les composants en jeu dans la sécurité différentielle et les connexions entre eux.

##### *Echantillonnage*

4 ADCs AD9240 pipeline assurent la conversion analogique numérique pour les signaux tores amont et aval, respectivement TSU et TSD.



## *Synchronisation*

Nous utilisons deux retard programmables DS1020 et deux retard fixe DS1100 de Dallas Semiconductor pour ajuster les quatre horloges CLKU\_T, CLKU\_B, CLKD\_T, CLKD\_B des ADCs. Leur programmation est faite par le microcontrôleur en utilisant une ligne série (SCLK, SDATA) et le port P2 pour l'adressage. Les PDG (Programmable Delay Generator) choisis pour la carte TPS ont une résolution de 0.5 ns pour une plage maximum de balayage de 45 ns. Les retards entre les échantillons TOP et BOT sont fixés à ~ 80 ns par les délais fixes DS1100. Les horloges de synchronisation des modules internes du FPGA sont dérivées de l'horloge CLKU\_T de la manière suivante :

$CLKU_a = CLKU_T + 20 \text{ ns}$ ,

$CLKU_b = CLKU_T + 60 \text{ ns}$ ,

$CLKU_c = CLKU_T + 100 \text{ ns}$ .

## *Traitement numérique*

Nous avons utilisé les composants et la configuration similaire à ceux de la carte d'évaluation (Figure 42) : un SPARTAN2 XC2S200 et une ISP-PROM XC18V02. La programmation de la PROM se fait par la liaison JTAG et le téléchargement des données de configuration FPGA selon le mode MASTER SERIAL.

L'étude des scénarios tenant compte de tous les signaux a été menée dans un premier temps à la main sur un grand nombre de chronogrammes au format A3. Cette étude a dégagée une vue d'ensemble de l'architecture à implémenter, nous avons alors réalisé un schéma global (Annexe E : Architecture logique du système) en n'utilisant que des fonctions logiques simples (registres, portes logiques) et des opérateur arithmétiques.

A partir de cet ensemble de fonctions, nous avons découpé l'architecture complète de traitement en modules (Figure 49) situé à un même niveau hiérarchique. Ces modules englobent différentes fonctions de base du traitement numérique pour la sécurité différentielle. Cette découpe a été faite pour simplifier la validation de chaque module mais aussi pour faciliter les essais sur la carte, en ne configurant que certains modules dans le FPGA.

### ➤ *Interfaces BIC et BIS*

Un driver de ligne quatre entrées au format RS-422 transmet les signaux de commande d'inhibition du faisceau vers le BIC. Huit opto-coupleurs sont utilisés pour l'écriture du registre d'état. Notamment, ils permettent un découplage galvanique entre la carte et le BIS. Deux autres opto-coupleurs permettent la commande d'étalonnage du système.

### ➤ *Interfaces DOOCS*

Tous les paramètres de seuil pour le traitement du FPGA seront mémorisés dans la mémoire flash du microcontrôleur ADUC812 en passant par la liaison RS-232. De plus, les paramètres d'ajustement des délais programmables et de la correction des défauts intrinsèque de la ligne faisceau seront aussi stockés dans la mémoire flash. Une mémoire RAM externe de 32 Ko est ajoutée pour la phase d'étalonnage et de calcul du facteur de correction. Cette RAM

étant adressée sur 15 bits, deux ports du microcontrôleur sont utilisés pour l'adressage (port0 : octet LSB et port2 : octet MSB). Une LATCH sur 8 bits permet le maintien de l'octet LSB pour pouvoir atteindre toute la capacité de la RAM.

### ➤ Débogage

En cas de besoin, pour fournir des informations sur la charge mesurée ou s'assurer du bon fonctionnement des quatre ADCs, un DAC 14 bits THS5671 de Texas Instruments est intégré au système pour pouvoir relire les ADCs en passant via le FPGA.

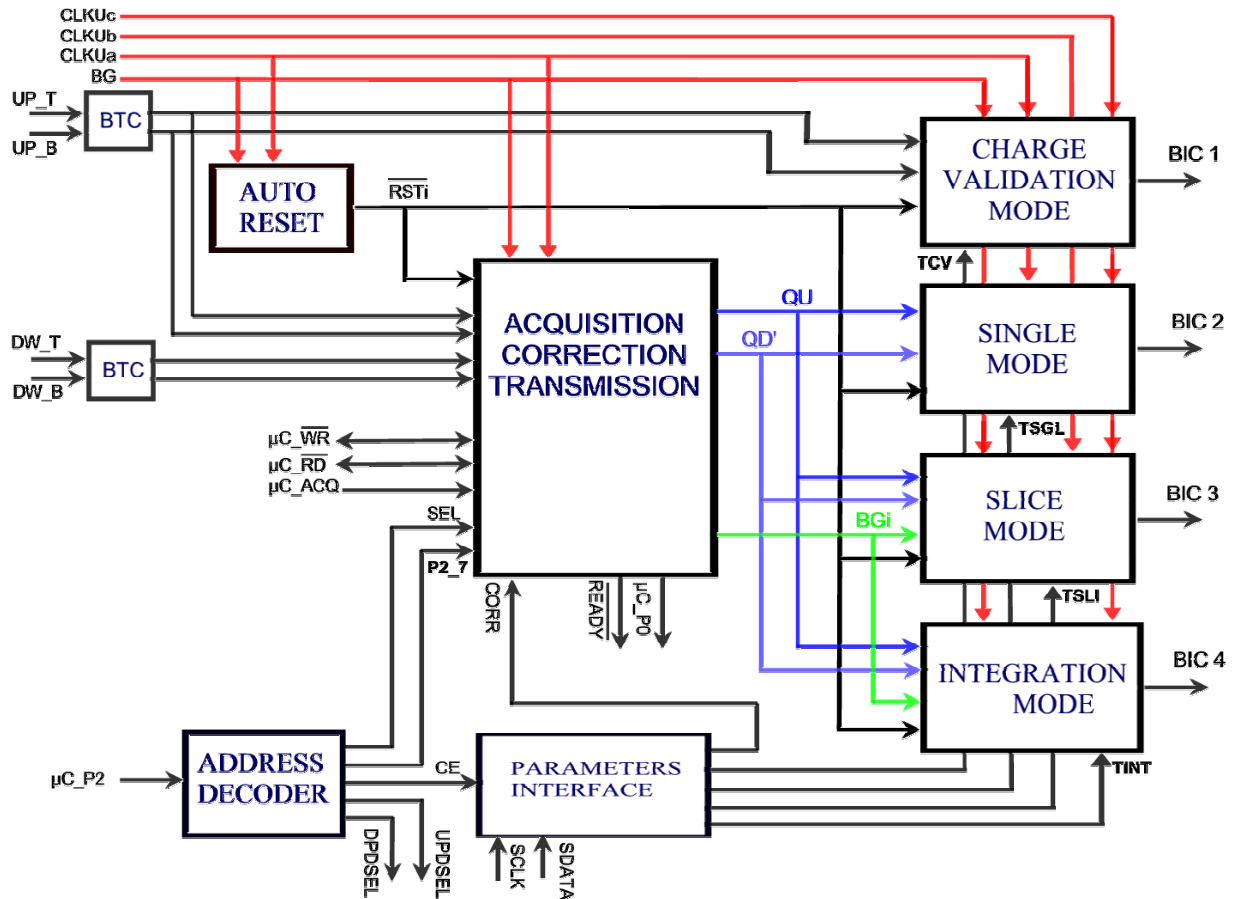


Figure 49 : Architecture interne du FPGA pour la sécurité différentielle de TTF2

# Partie 2 : Implémentation et Validation

## 1 Description et simulation des modules internes

Il était possible de réaliser chaque module complètement en VHDL, mais notre but était d'utiliser le plus possible de primitives ou de macro présentes dans la bibliothèque de symboles de Xilinx. Quand la fonction recherchée n'était pas disponible, alors nous la réalisons en VHDL, cela pour accroître la lisibilité, faciliter la maintenance et la vérification lors de la synthèse puis de l'implémentation.

Pendant la conception des modules, nous avons vérifié que leur synthèse était possible. Le langage VHDL [9] n'a pas été défini pour la synthèse mais bien pour la simulation. Il en résulte que toutes les constructions VHDL ne peuvent pas être traduites en primitives matérielles. Il suffit, pour s'en convaincre de prendre les exemples de types pointeurs (access dans le langage VHDL) ou fichier.

Chaque module est testé par l'intermédiaire d'un banc de test en VHDL dédié (testbench). Les tests présentés ici ont pour objectifs de valider la fonction du module pour des scénarios réalistes du point de vue des opérations sur la machine.

Pour chaque test, nous avons besoin de deux fichiers VHDL :

- le fichier module\_timesim.vhd généré par ISE,
- le testbench en VHDL.

Le fichier module\_timesim.sdf, contenant les retard de propagation des instances du design, est utilisé par le simulateur pour la simulation post placement routage. De plus, il faut ajouter, pour chaque module, le lien avec les IOBs du FPGA. Cela est fait à l'aide des composants Ibuf et Obuf (Input ou Output Buffer) sous l'éditeur ECS.

Pour faciliter la création du testbench et des stimuli, nous utilisons une application d'ISE : HDL Bencher (Figure 50). Cet outil crée un fichier testb.vhw en VHDL et réalise l'instanciation et la configuration du composant pour le test. Il est alors possible d'éditer et de modifier ce fichier pour y ajouter des signaux de vérification des sorties.

Les inconvénients de cet outil sont la difficulté d'indiquer un temps précis pour la fin de la simulation et l'impossibilité d'introduire des variables de vérifications. Pour avoir la maîtrise du testbench, nous récupérons le fichier testb.vhw que nous modifions sous Modelsim pour l'adapter à nos besoins. La compilation du fichier contenant la description structurelle, à l'aide de primitives de la librairie Simprim, et du testbench est alors la dernière étape avant de lancer une simulation.

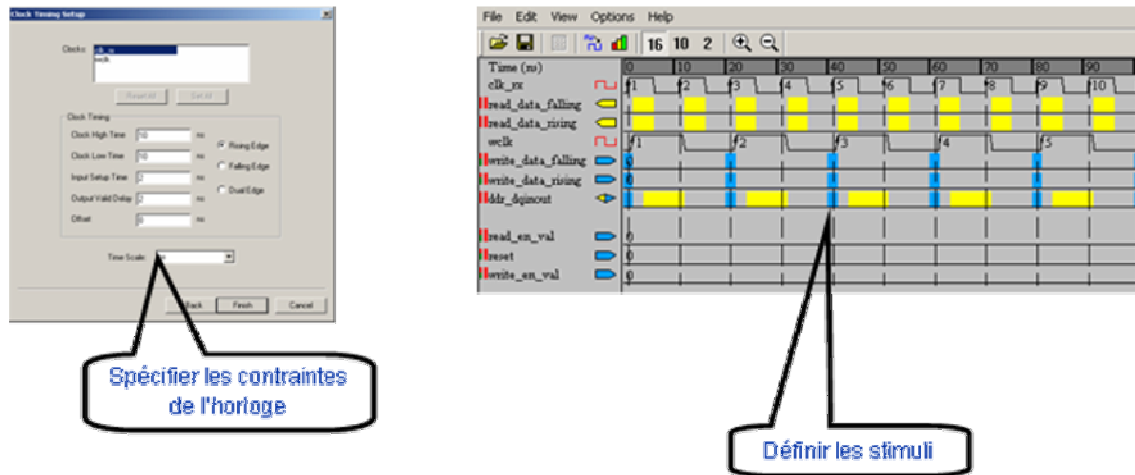


Figure 50 : Générateur de stimuli intégré à ISE Webpack

### 1.1 Le module AUTORESET

Ce module a pour fonction de générer automatiquement un signal de remise à zéro (RAZ) de l'architecture interne FPGA après une durée de 2 ms de la fin de la macro-impulsion. La RAZ est forcée lorsqu'un signal MRST (Master Reset) externe est appliqué.

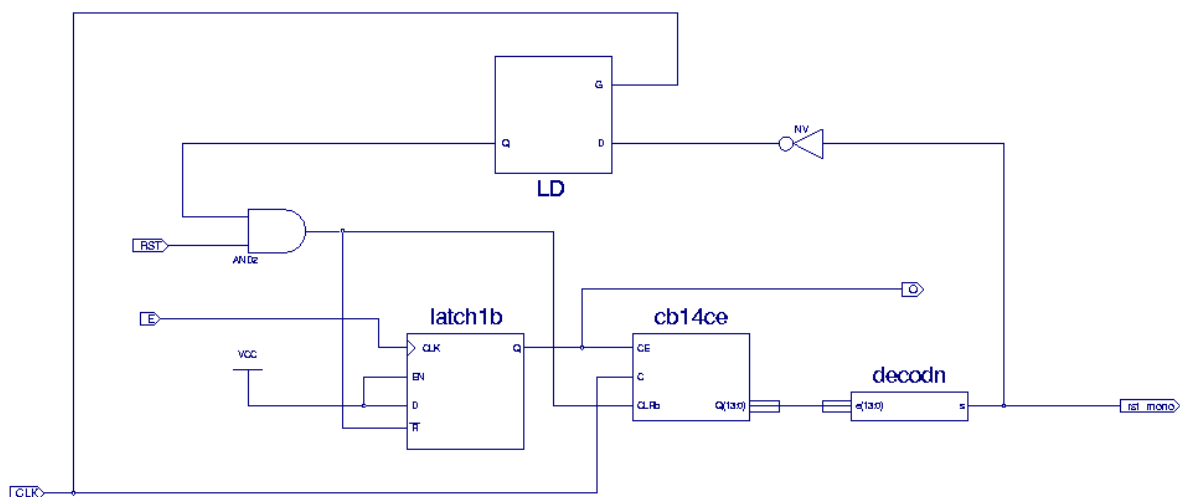


Figure 51 : Module Autoreset

Le comptage est autorisé lorsqu'une macro-impulsion a débuté et s'est achevée, la sortie du compteur est comparée à une valeur représentant le nombre de coup d'horloge nécessaire pour avoir une durée de 2 ms. Ici, l'horloge est de période 110.8 ns, d'où un comptage de 18050 période. Il faut donc un compteur sur 15 bits.

Le décodeur génère deux commandes de reset l'une après l'autre:

- Un reset interne (RSTi) pour tous les autres modules
- Un reset pour lui-même

Les 2 RAZ, défini ci-dessus, ne doivent pas être faites en même temps car le signal généré serait trop bref pour être efficace. C'est pourquoi, on retarde la seconde RAZ par l'intermédiaire d'une bascule D.

La simulation fonctionnelle est, en générale, celle qui demande le plus d'aller-retour entre le design et le testbench. C'est une phase de mise au point, comme pour un test matériel.

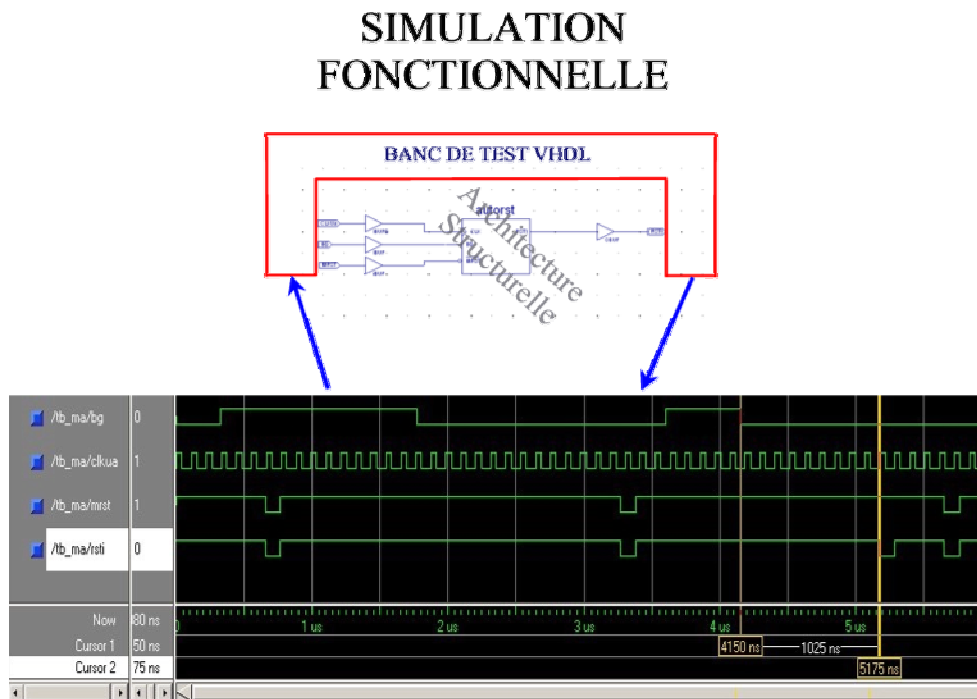


Figure 52 : Simulation fonctionnelle du module Autoreset

Ici (Figure 52), aucun signal de vérification n'a été calculé, nous procédons par contrôle visuel, c'est pourquoi la durée après la fin de la macro-impulsion est fixée à 10 coups d'horloges au lieu de 18051 (2 ms). Le signal BG ci-dessus représente une macro-impulsion, hors pendant cette dernière on génère un reset externe (MRST). Ce qui a pour conséquence d'inhiber le déclenchement du module. Cela s'explique par le fait que la bascule à l'entrée du module est déclenchée sur front montant du signal BG.

Pour le second scénario commençant après 3  $\mu$ s, le signal RSTi de RAZ interne intervient bien 10 fronts montant après la fin du train. Le temps mesuré de 1025 ns correspond à  $9 \times 110$  ns + durée (fin de la macro-impulsion jusqu'au premier coup d'horloge CLKUa). Une vérification avec la durée nominale de 2 ms, au lieu de 990 ns dans la simulation présentée ici, a été réalisée avec succès.

Lorsque c'était possible, nous avons réduit les durées des monostables pour gagner du temps pendant les phases d'analyse et d'optimisation de l'ensemble des modules. A chaque fois, après avoir terminé les mises aux points des modules, nous avons réalisé une simulation post placement routage avec les spécifications nominales pour la sécurité différentielle de TTF2.

## SIMULATION POST ROUTAGE

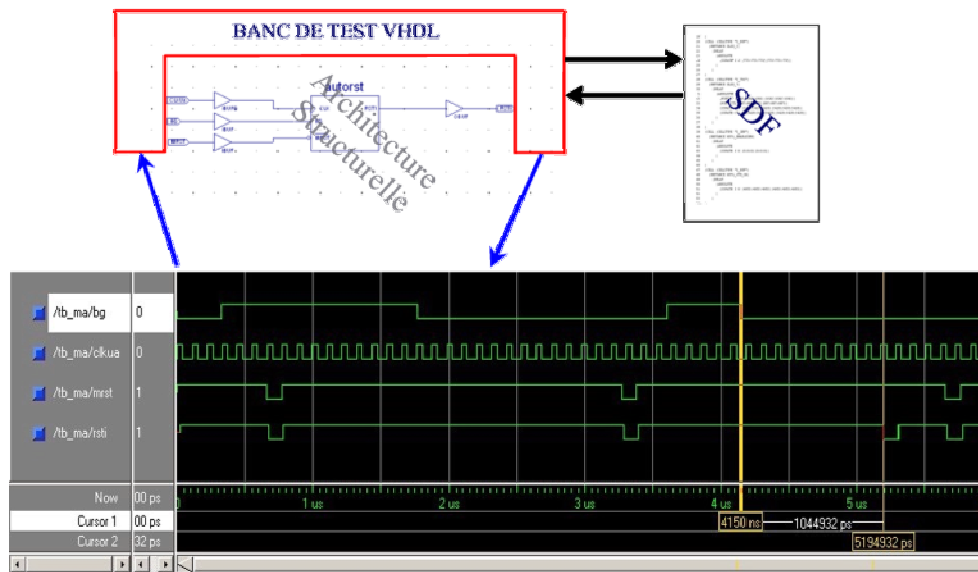


Figure 53 : Simulation post placement routage du module Autoreset

La durée entre la fin de la macro-impulsion et la génération du signal RSTi (Figure 53) est plus importante (~ 1045 ns) car la simulation a tenu compte des retards de propagation des signaux dus aux routages au sein du composant. Par la suite, toutes les simulations présentées seront celles effectuées après le placement routage avec le fichier SDF. Nous avons aussi arrondi la période des paquets d'électrons à 110 ns au lieu de 110.8 ns.

### 1.2 Le module BTC (Conversion de binaire à complément à deux)

Tous les calculs sont effectués sur des nombres signés car les données en entrée du système peuvent être négative. Les ADCs présentent leurs sorties en binaire de base pour toutes les gammes d'entrées. Ce module très simple convertit les sorties des ADC en complément à deux. Grâce au codage interne de l'AD9240, il suffit d'inverser le MSB de la sortie des ADC (Tableau 6).

Entrée signaux tores(V)	Sortie digitale des ADCs (binaire)	Décimale	Sortie complémentée à 2 par le FPGA	Décimale
2.048	11 1111 1111 1111	16383	01 1111 1111 1111	8191
0	10 0000 0000 0000	8192	00 0000 0000 0000	0
-2.048	00 0000 0000 0000	0	10 0000 0000 0000	-8192

Tableau 6 : Format de sortie des ADCs et correspondance en complément à 2

### 1.3 Le module CALCQx&TRANSμC

Ce module (Figure 54) est l'un des plus complexes car il regroupe plusieurs fonctions qu'il était très difficile à séparer. Il a pour fonctions principales :

- le transfert de la charge amont et de la charge aval corrigée aux autres modules,
- le transfert de ces mêmes charges sans correction au microcontrôleur.

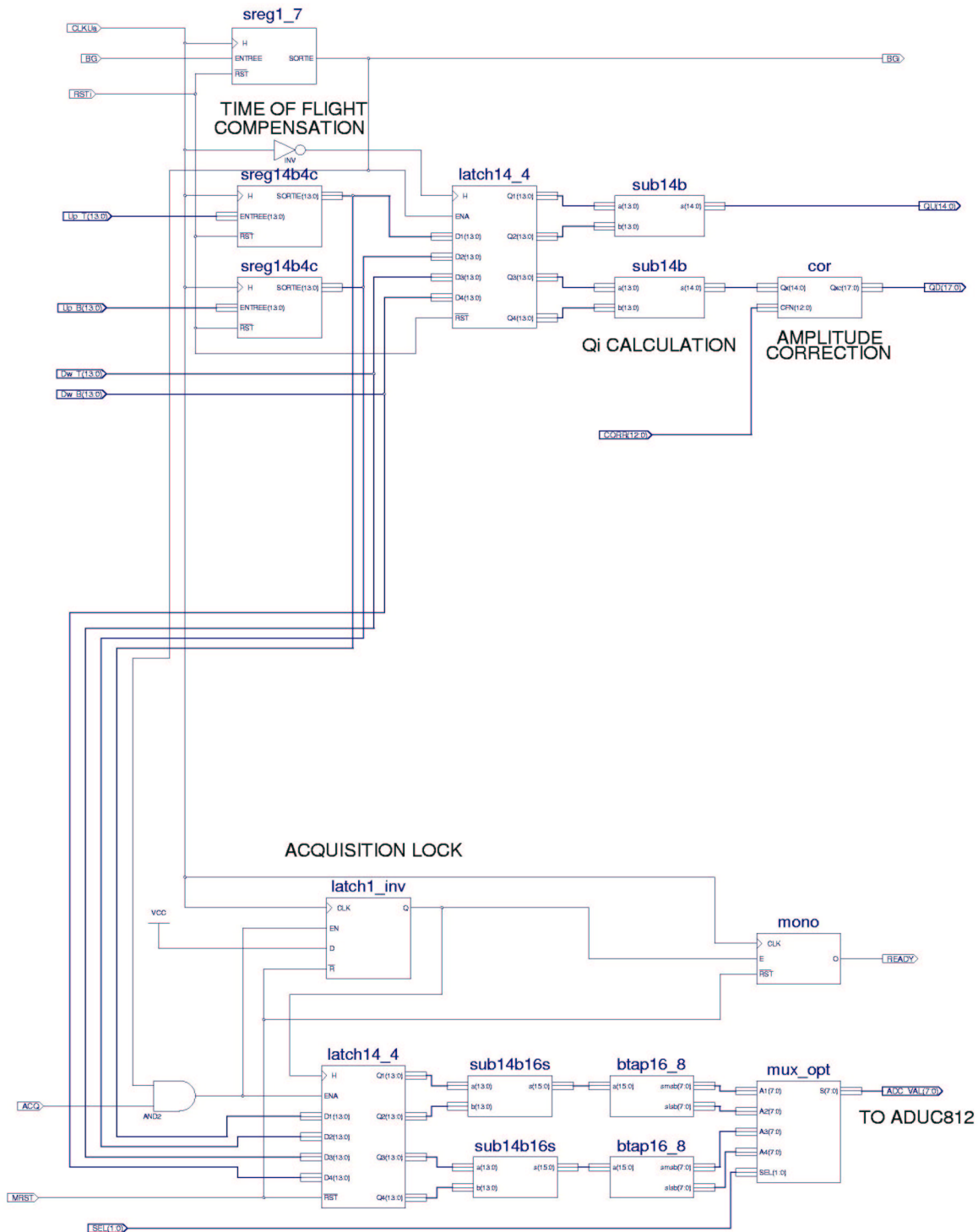


Figure 54 : Module CALCQx&TRANSμC

Il se divise en deux parties, une partie mise en œuvre pendant la phase de protection machine (Figure 55) et l'autre utilisée pour étalonner le système (Figure 58).

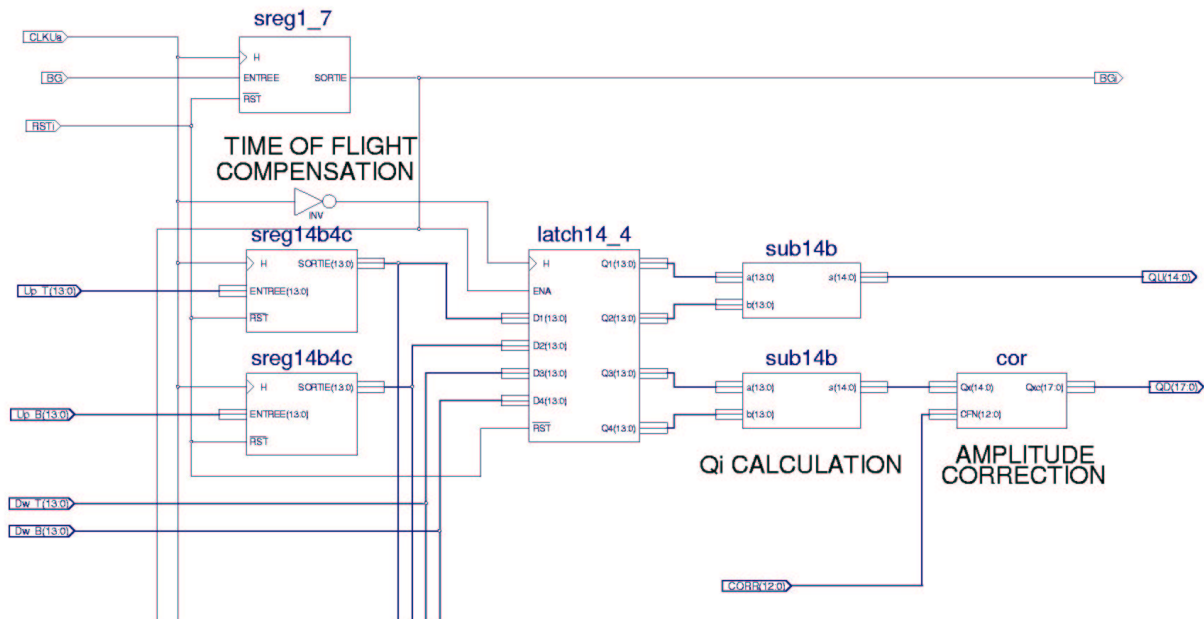


Figure 55 : Synchronisation des valeurs numériques et calcul de la charge amont et de la charge aval corrigée

La première partie consiste à synchroniser les entrées en tenant compte de la latence des ADC et du temps de vol ( $N$ ) des paquets d'électrons entre les tores amont et aval. Le facteur  $N$  correspond au retard exprimé en multiple de la période d'horloge à 9.0277 MHz.

En sortie de la LATCH mémorisant les valeurs  $UP\_T$ ,  $UP\_B$ ,  $DW\_T$  et  $DW\_B$ , un soustracteur calcule les différences proportionnelles à la charge des tores amonts et avals :

$$QU = UP\_T - UP\_B \text{ et } QD_1 = DW\_T - DW\_B$$

Nous devons tenir compte des imperfections des chaînes de mesures du faisceau, cela concerne surtout l'électronique qui se trouve derrière les tores (combiner, atténuateur, filtre, amplificateur) mais aussi les câbles coaxiaux. Pour cela, lors de l'étalonnage du système, nous calculons un coefficient  $\alpha$  :

$$\alpha = QU / QD_1 \text{ avec } 1 < \alpha < 2$$

Dans le FPGA, nous ne travaillons qu'avec des nombres entiers. C'est pourquoi, nous multiplions  $\alpha$  par 1024 et nous tronquons les chiffres restant après la virgule. La valeur 1024 ( $2^{10}$ ) est choisie pour avoir une précision d'au moins  $10^{-3}$  et permettre la division dans le FPGA (rappel : la division n'est permise que pour des valeurs en  $2^n$ ).

$$CORR = \alpha * 1024$$

La valeur maximale de CORR est  $2 * 1024 = 2048$ . Toutes nos opérations sont faites sur des nombres signés. Pour atteindre 2048 en nombre signé, CORR doit être écrit sur 13 bits. Dans ce cas, la valeur maximale est :

$$2^{(n-1)} - 1 = 2^{12} - 1 = 4095$$

Ce facteur de correction est ensuite appliqué à QD :

$$QD = (QD_1 * CORR) / 1024$$



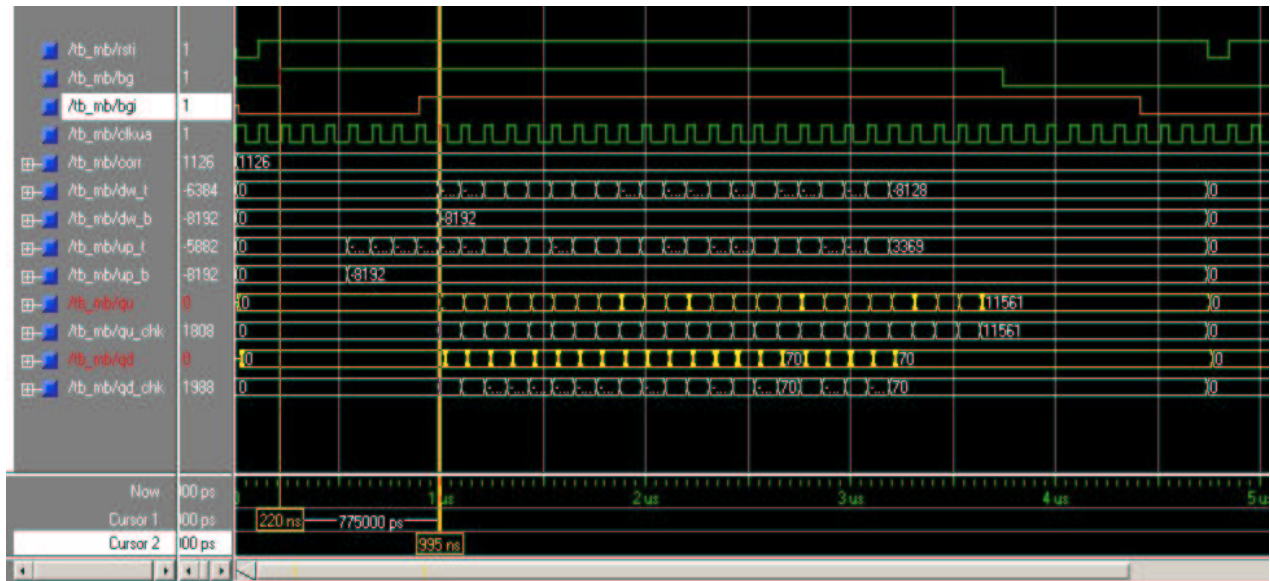


Figure 56 : Simulation de la partie CALCQx

Dans la simulation (Figure 56), le temps de vol vaut  $N=4$  (c'est à dire un retard supposé entre le tore amont et aval de 440 ns). Nous observons QU et QD et BGi (BG retardé de 770 ns pour tenir compte du retard entre les tores et de la latence des ADCs de 3 cycles d'horloge). Le facteur CORR est de 1126, ce qui correspond à  $\alpha \sim 1.1$ . On vérifie que :

- QU et QD sortent  $(4+3)*110 \text{ ns} + 5 \text{ ns} = 775 \text{ ns}$  après BG, les 5 ns correspondent au délai entre le front montant du BG et le front montant de CLKUa.
- les valeurs calculées QU\_chk et QD\_chk par le testbench correspondent aux sorties QU et QD du module,
- la synchronisation des signaux tore amont, tore aval et BG est réalisée.

Si l'on zoom :

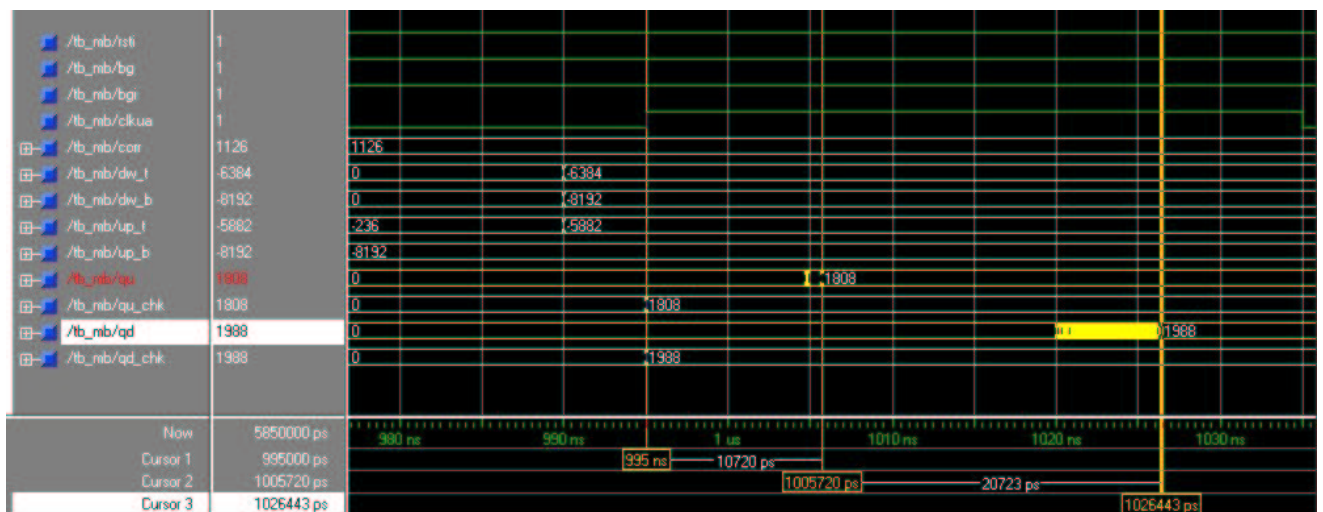


Figure 57 : Zoom sur la simulation de la partie CALCQx

On observe les retards entre le front montant de l'horloge CLKUa et les sorties QU et QD (Figure 57) dues à la propagation entrées-sorties de la bascule 14 bits et aux calculs. De 10.7 ns pour QU et de 31.4 ns pour QD. La durée est plus longue pour QD car ayant une multiplication et une division en plus de la soustraction et de l'extension de signe.

Vérification du résultat de QD pointé par le curseur (Figure 57) :

$$D = ((DW\_T - DW\_B) * CORR) / 1024 = ((-6384 + 8192) * 1126) / 1024 = 1988$$

La seconde partie permet le transfert des acquisitions au microcontrôleur octets par octets :

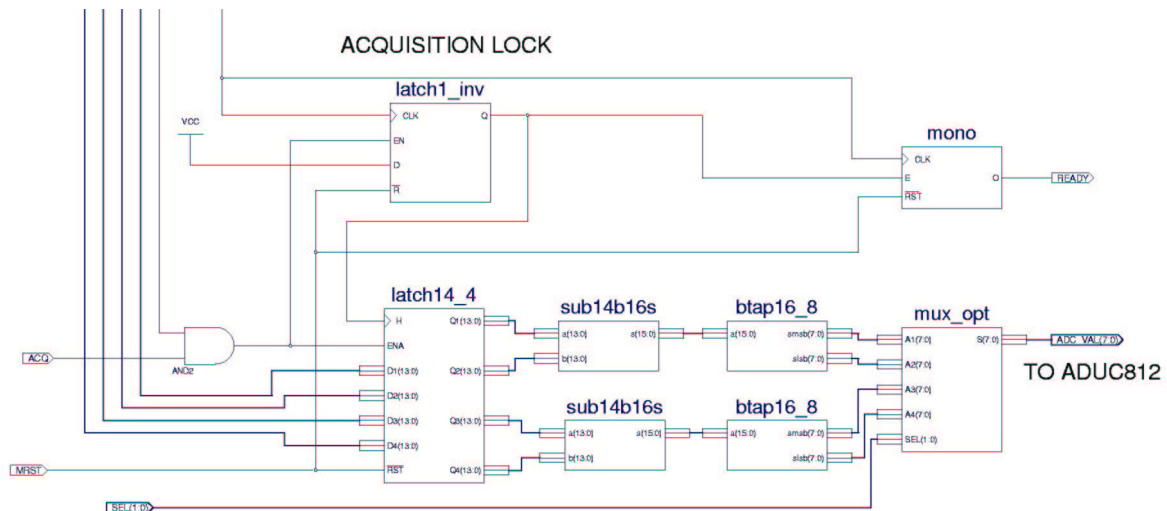


Figure 58 : Partie TransµC

Cette partie (Figure 58) n'est utilisée que lorsque le BIS demande un étalonnage. Cette commande est alors prise en charge par un sous programme pointé par une interruption du microcontrôleur. Les données amont et aval sont synchronisées par le même biais que pour la partie CALCQx.

Le signal ACQ et le signal BGi activent la bascule latch1\_inv, déclenchée sur front descendant de l'horloge CLKUa. Cette bascule permet de s'assurer que le temps de pré positionnement de la bascule 14 bits, à déclenche sur front, est respecté. Mais surtout, elle permet de verrouiller l'acquisition jusqu'au prochain signal BG. Pour déverrouiller les bascules, il faut impérativement activer le signal MRST. Après l'acquisition de la bascule 14 bits, le calcul de la charge QU et QD est effectué et étendu sur 16bits puis divisé en 2 octets MSB et LSB.

Les quatre octets sont placés à l'entrée d'un multiplexeur (4à1) adressé par le port P2 du microcontrôleur pour être transmis sur le port P0. Le FPGA transmet un signal (front descendant du signal READY) au microcontrôleur pour indiquer la disponibilité de l'acquisition. Le signal READY passe à l'état haut lorsque la bascule 14 bits est validé et est maintenu à l'aide d'un monostable pour que les calculs et le formatage soient faits avant que le microcontrôleur ne commande la lecture.

Une combinaison du MSB du port P2 et du signal read RD, qui se trouve à un niveau hiérarchique supérieur de l'architecture, permet de s'assurer qu'il n'y a pas de conflit sur le port P0. Le port P0 reçoit la valeur ADCVAL (Figure 58). Mais si le microcontrôleur utilise P0 pour adresser la RAM externe ou pour une écriture sur ce port, il faut que le FPGA mette P0 en Haute Impédance (HIZ) pendant ce temps. Sinon, il y a un conflit.

Nous avons simulé l'étalonnage qui se fera en mode single shot, un paquet par macro-impulsion. Nous ne visualisons (Figure 59) qu'une partie des sorties du module.

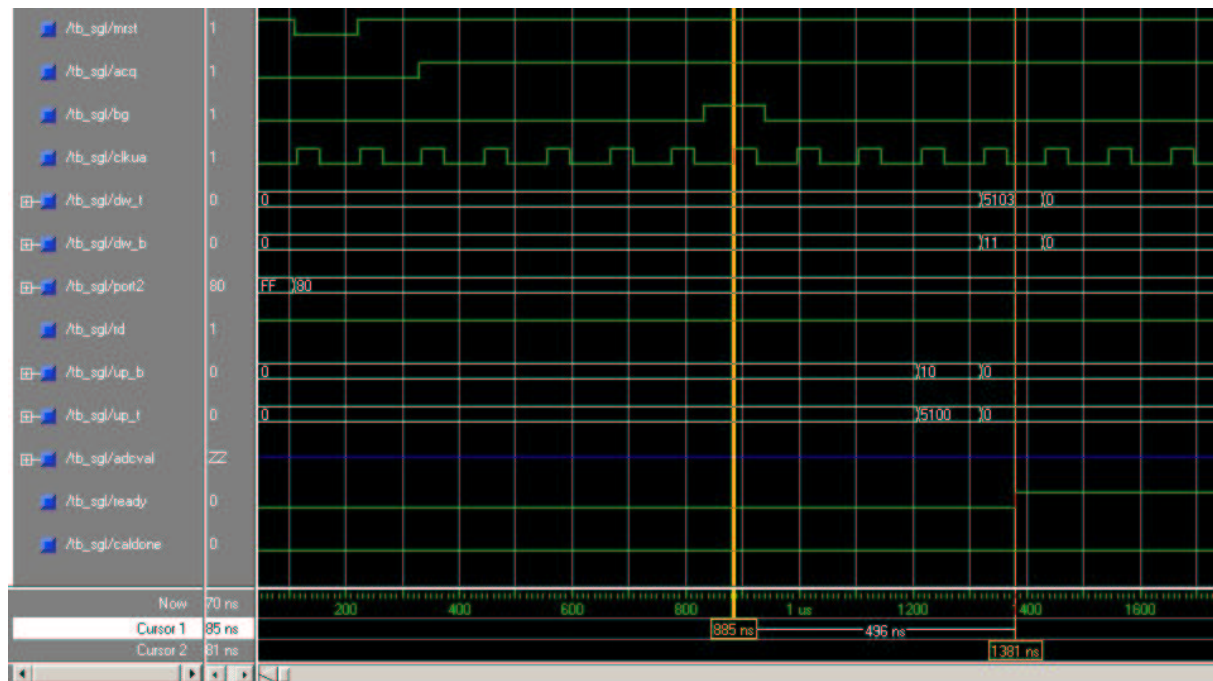


Figure 59 : Simulation de la partie TransuC, zoom sur la phase d'acquisition

Nous vérifions que la latence de 3 cycles des ADCs (330 ns) et le temps de vol entre les tores amonts et avals (110 ns), soit un total de 440 ns, sont pris en compte avant l'acquisition des sorties ADCs. Cette acquisition est représentée par le signal READY. Les curseurs (Figure 59) mesurent un délai de 496 ns entre le moment où BG est échantillonné et le moment où READY passe à l'état haut. Ce délai intègre les 440 ns décrit plus haut. Il reste 56 ns correspondant d'une part au 40 ns d'attente entre les fronts montants et descendants de l'horloge CLKUa. D'autre part, à 16 ns de retard de propagation dans le monostable « mono » (Figure 58).

La Figure 60 montre la suite de la simulation de la Figure 59. Nous simulons la lecture faite par le microcontrôleur après un front descendant du signal READY survenu 1 ms après l'acquisition de la bascule 14 bits (Figure 58). Dans ce scénario (Figure 60), le MSB du port P2 est toujours à l'état haut, pas d'écriture dans la RAM externe ni par conséquent sur le port P0. Le signal RD conditionne la sortie ADCVAL qui est transmise sur le port P0. A l'état haut de RD, ADCVAL est mis en haute impédance. Pour lire, le microcontrôleur active RD et positionne P2 pour adresser les octets de poids fort puis de poids faible de QU et QD.

Sur la Figure 59, on vérifie que :

$$QU = UP\_T - UP\_B = 5100 - 10 = 5090 \text{ en format décimal ou } 13E2 \text{ en hexadécimal}$$

$$QD = DW\_T - DW\_B = 5103 - 11 = 5092 \text{ ou } 13E4$$

Ce qui correspond (Figure 60) aux valeurs acquises par le module.

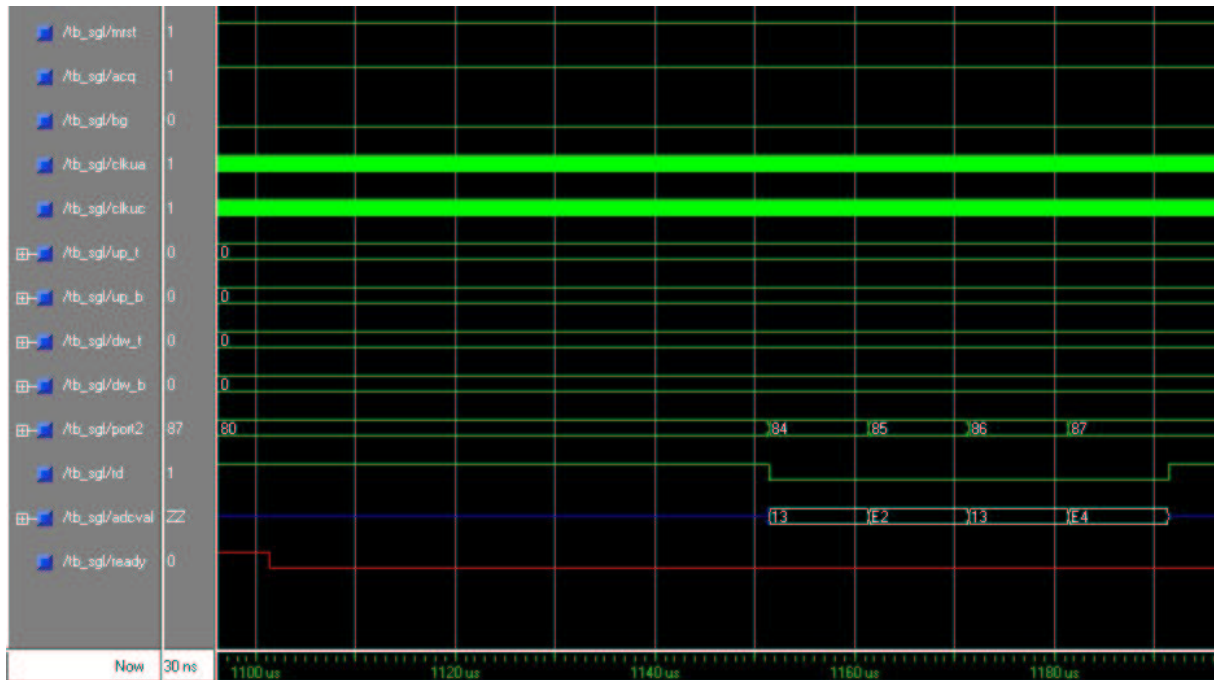


Figure 60 : Simulation de la partie Trans $\mu$ C, zoom sur la lecture de la sortie ADCVAL

## 1.4 Le module ADDRESS CODEC

La communication entre le microcontrôleur et le FPGA ne pouvait se faire sans une fonction de codage et de décodage d'adresse.

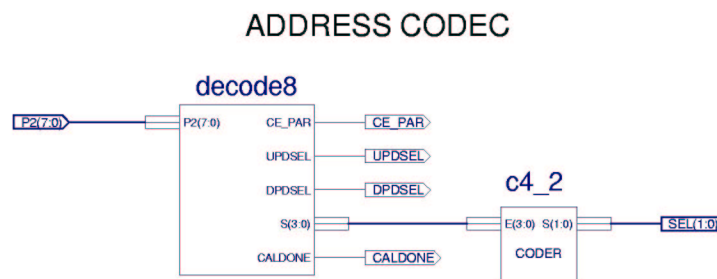


Figure 61 : Module Address codec

Le port P2 est utilisé dans le module ADDRESS CODEC (Figure 61) pour adresser les éléments suivants :

- le module Parameters Interface avec CE\_PAR,
- les deux délais programmables par UPDSEL et DPDSEL,
- le multiplexeur 4 à 1 du module TRANS $\mu$ C (Figure 58),
- le signal CALDONE qui indique au BIS la fin de l'étalonnage

Adresse sur le port P2 (binaire)	Donnée ou élément activé
10000000	Module Parameters Interface
10000001	PDG du tore amont
10000010	PDG du tore aval
10000100	MSB de la charge amont
10000101	LSB de la charge amont
10000110	MSB de la charge aval
10000111	LSB de la charge aval
10001000	CALDONE

Tableau 7 Données et éléments adressés par le port P2

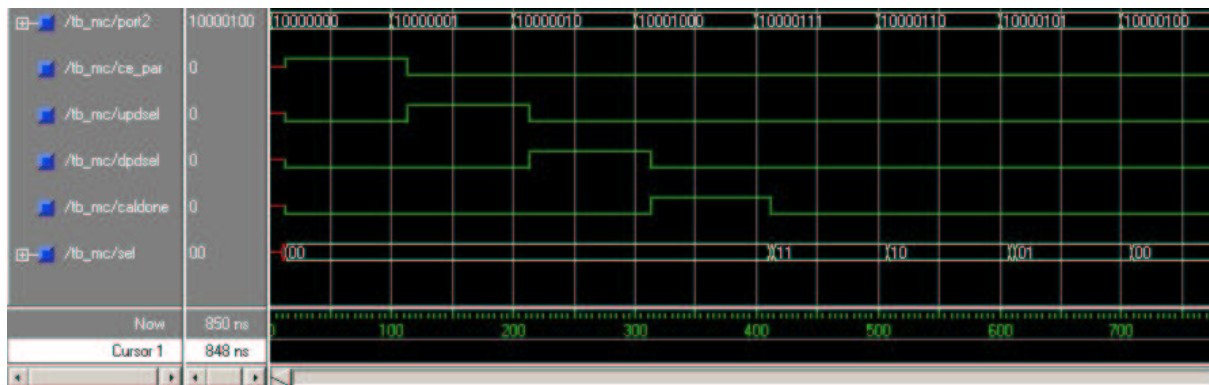


Figure 62 : Simulation du module ADDRESS CODEC

Cette vérification très simple est strictement visuelle. Les adresses (Tableau 7) sur P2 sont décodées. Les sélections des délais programmables et du module de transfert de paramètre sont réalisées comme prévu. On peut noter (Figure 62) l'instabilité pendant quelques nanosecondes des sorties SEL à cause des phénomènes de commutations au niveau des portes logiques ainsi que le retard de propagation des portes qui se manifeste par un retard entre la commande d'adressage et la sortie adressée.

### 1.5 Le module PARAMETERS INTERFACE

La mise à jour des seuils et du facteur de correction ainsi que leur mémorisation est une part importante du système.

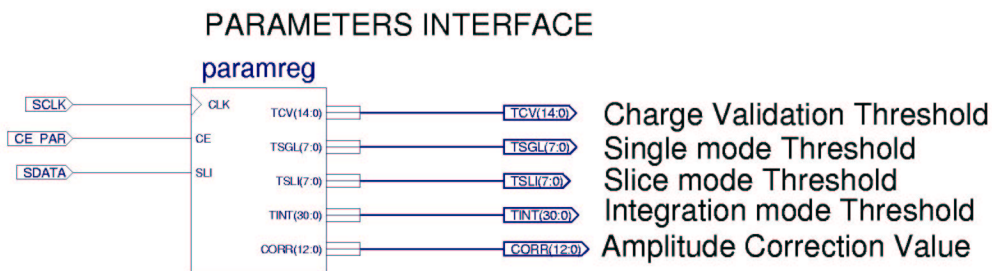


Figure 63 : Module PARAMETERS INTERFACE

Ce module (Figure 63) est un registre à décalage à entrée série et sortie parallèle. Les lignes SCLK et SDATA proviennent du microcontrôleur, l'entrée de sélection est validée par le module Address Codec. Le mode de transfert des données peut être asynchrone, la trame à écrire comporte les données suivantes :

- seuil du mode de Validation de Charge sur 15 bits (TCV)
- seuil du mode Single sur 8 bits (TSGL)
- seuil du mode Slice sur 8 bits (TSLI)
- seuil du mode Intégration sur 31 bits (TINT)
- facteur de correction sur 13 bits (CORR)

Nous avons donc une trame de 75 bits (Figure 64) pour toutes ces données. Il n'y a pas d'entrée de RAZ car ces paramètres ne seront pas souvent modifiés. Ce module doit absolument être inactivée par l'entrée CE\_PAR après l'écriture du dernier bit de la trame.

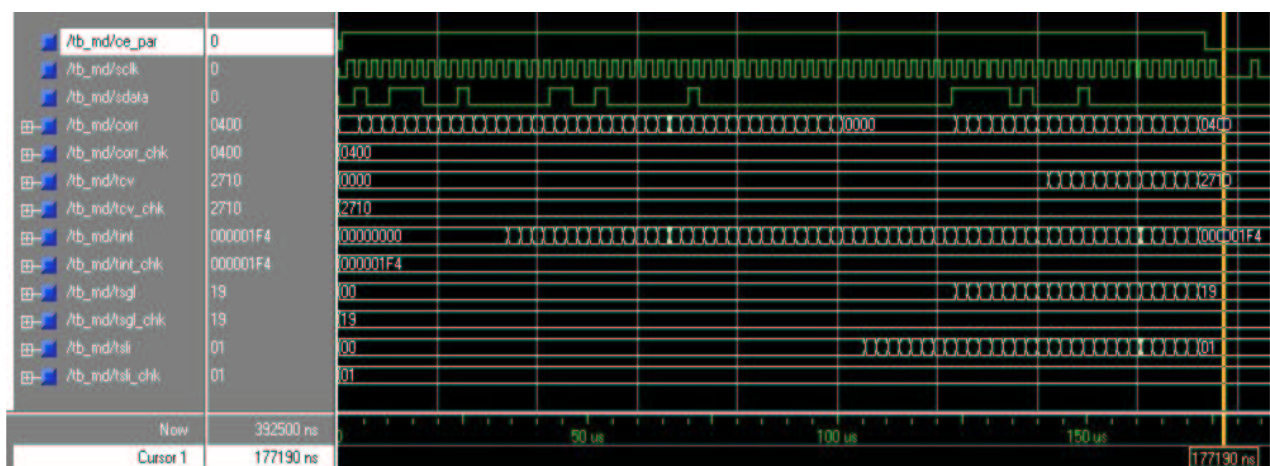


Figure 64 : Simulation du transfert des paramètres

L'écriture des seuils dans le FPGA se fait par une liaison série (SCLK, SDATA). Le port P2 sélectionne le module par l'entrée CE\_PAR, puis la trame de 75 bits est écrite en commençant par le MSB de TCV et en finissant par le LSB de CORR avant la désélection du module. Le curseur positionné en fin de trame montre la concordance des données de vérification (SORTIE\_CHK) écrites dans la trame et des sorties du module. Pour faciliter la vérification, nous avons cadencé les bits à une fréquence fixe. Nous avons contrôlé que après la désélection du module, les valeurs en sorties restaient inchangées, même si l'horloge SCLK est reste active.

## 1.6 Le module CHARGE VALIDATION

Ce module (Figure 65) a pour rôle de déclencher une alarme dans les deux cas suivants :

- la charge du paquet du tore amont est inférieure à un seuil,
- la charge est suffisante mais le signal porte faisceau (BG) est absent.

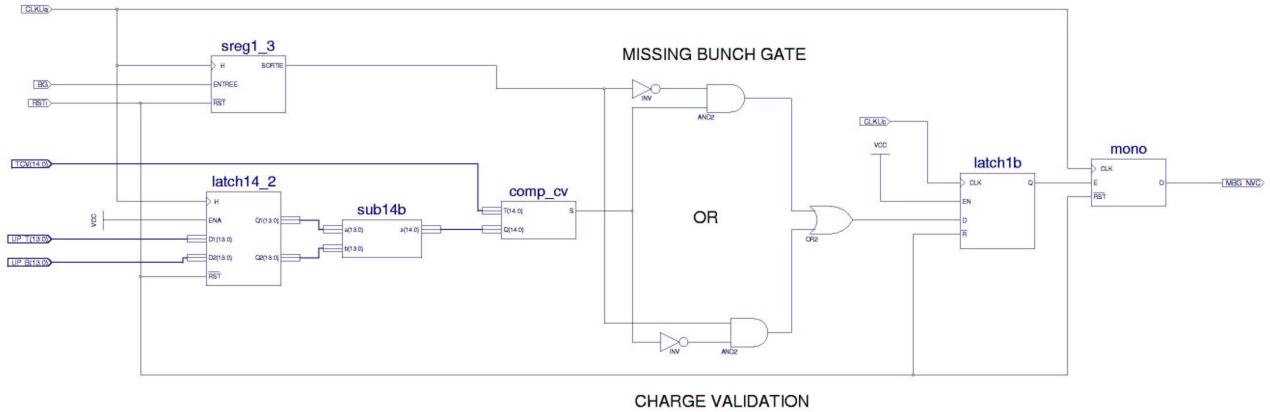


Figure 65 : Module CHARGE VALIDATION

Le signal BG est retardé de trois coups d’horloge pour être synchroniser avec les sorties ADC TOP et BOT du tore amont. Le calcul et l’extension de signe sur 15 bits de la charge sont suivis du comparateur au seuil TCV. Un monostable maintiens la sortie à l’état haut pendant 1 ms, temps nécessaire au BIC pour reconnaître le signal d’interlock. Ce monostable sera présent sur chaque module déclencheur d’interlock.

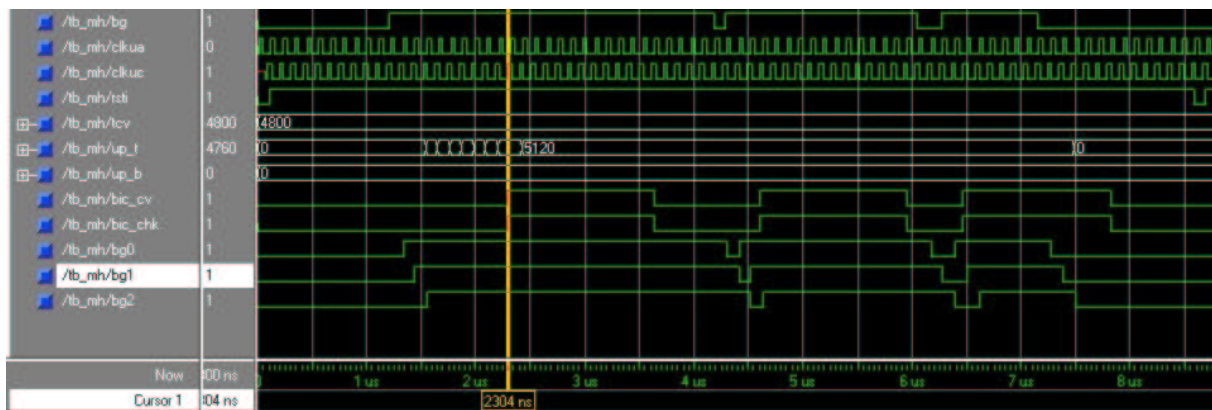


Figure 66 : Simulation du mode CHARGE VALID

Dans le premiers cas de déclenche d’alarme (Figure 66), le seuil de la charge vaut 4800 en décimal, ce qui correspond à  $4800 * 250\mu V = 1.2V$  ou  $2.4nC$ , sachant que la charge nominale des paquets sera de  $3nC$ . Dès que la charge est inférieure et que le BG est valide, l’interlock est généré. Dans les cas suivants de génération d’interlocks, la charge de  $2.56nC$  (5120 en décimal) est suffisante mais le BG est absent d’où émission d’un interlock. BG0, BG1 et BG2 sont des signaux du testbench, BG2 représente la sortie du registre pipeline SREG1\_3 (Figure 65).

### 1.7 Le module SINGLE MODE

Ce module (Figure 67) protège la machine paquet par paquet, c’est le mode de protection de l’accélérateur le plus rapide et le plus important. L’alarme est déclenchée dès que la différence de charge entre le tore amont et aval dépasse un seuil.

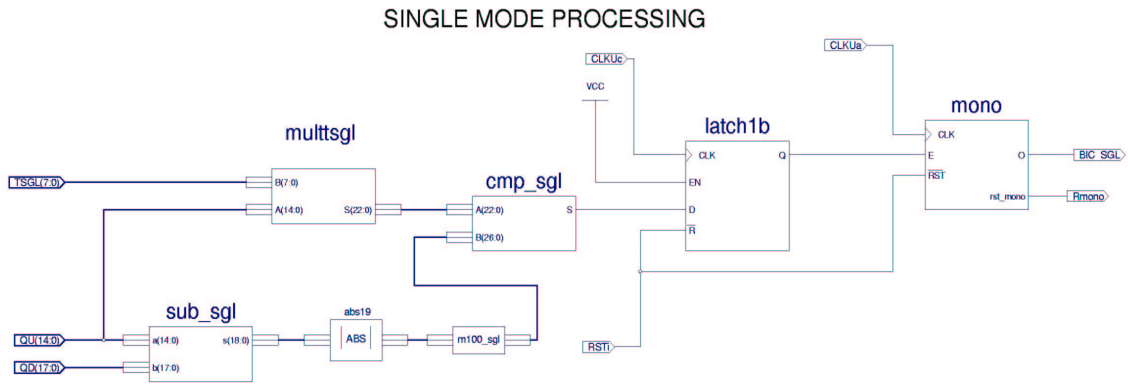


Figure 67 : Module SINGLE MODE

L'opération de division étant gênante car non faisable sur les entiers différents de  $2^n$ , nous avons donc effectué le calcul équivalent suivant :

$$\frac{|QD - QU|}{QU} > Tsgl \Leftrightarrow |QD - QU| \cdot 100 > QU \times Tsgl$$

avec  $Tsgl$  exprimé en %

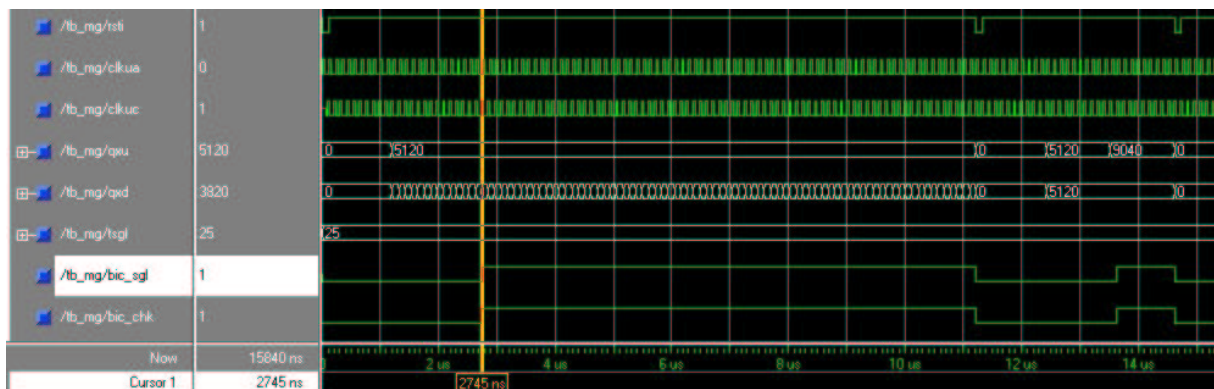


Figure 68 : Simulation du mode SINGLE

Ici (Figure 68), le seuil  $Tsgl$  est de 25%. Au niveau du curseur, on peut contrôler que  $(QU-QD)/QU = (5120-3820)/5120 > 25\%$  donc émission de  $BIC\_sgl$ . Une fois encore le signal  $BIC\_chk$  synchronisé sur  $CLKUc$ , horloge de la bascule de sortie, vérifie la fonction du module. Si l'on zoom, sur l'émission des signaux  $BIC\_sgl$  et  $BIC\_chk$ , il y a un retard de 13 ns dû au temps de propagation entre l'entrée et la sortie du monostable.

### 1.8 Le module SLICE MODE

Ce bloc est celui qui utilise le plus de ressource matériel. C'est aussi celui qui consomme le plus de temps de simulation. Pour un même banc de test, avec la version gratuite de Modelsim (MXE) il faut 15 mn alors qu'avec la version payante Modelsim PE il faut un peu moins d'une minute.



SLICE MODE PROCESSING

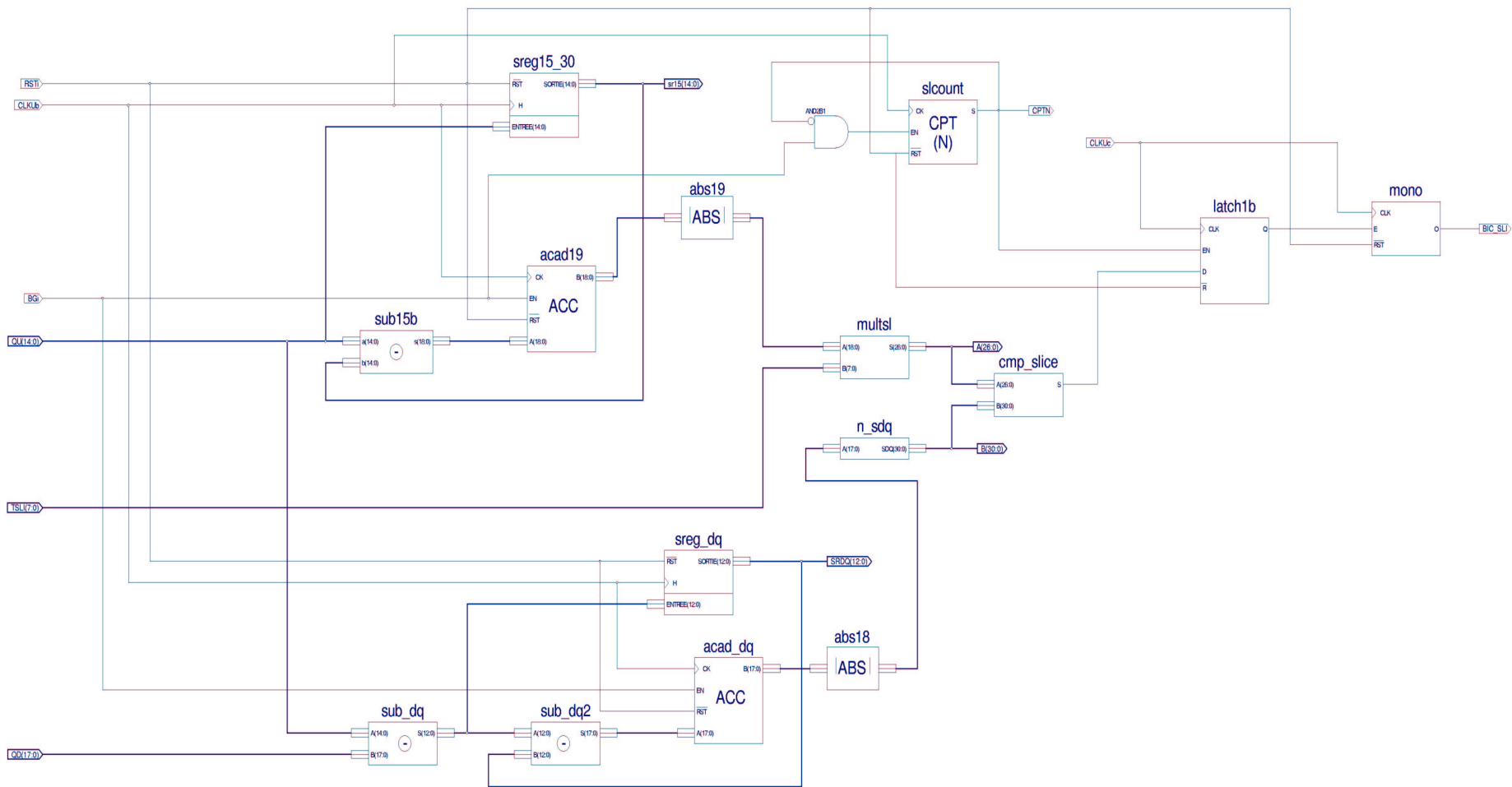


Figure 69 : Module SLICE MODE

De même que pour le mode single, la formule de calcul du mode slice est modifiée comme suit :

En faisant l'hypothèse  $|QD_i - QU_i| \ll QU_i$ , on peut dire :

$$\sum_{i=1}^N \left| \frac{QD_i - QU_i}{QU_i} \right| \approx \frac{1}{QU_i} \sum_{i=1}^N |QD_i - QU_i|$$

En développant la moyenne de  $QU_i$  sur  $N$  :

$$\sum_{i=1}^N \left| \frac{QD_i - QU_i}{QU_i} \right| \approx \frac{1}{\frac{\sum_{i=1}^N QU_i}{N}} \sum_{i=1}^N |QD_i - QU_i|$$

Par conséquent :

$$\sum_{i=1}^N \left| \frac{QD_i - QU_i}{QU_i} \right| \approx \frac{N \cdot \sum_{i=1}^N |QD_i - QU_i|}{\sum_{i=1}^N QU_i}$$

Enfin, on en déduit la condition de déclenche :

Équation 1 : Calcul réellement implémenté dans le FPGA

$$\frac{N \cdot \sum_{i=1}^N |QD_i - QU_i|}{\sum_{i=1}^N QU_i} > T_{slice} \Leftrightarrow 100 \cdot N \cdot \sum_{i=1}^N |QD_i - QU_i| > T_{slice} \times \sum_{i=1}^N QU_i$$

Dans notre architecture, nous implémentons la partie droite de l'équivalence (Équation 1). La sortie du module est inhibée tant que nous n'avons pas atteint la  $N^{\text{ième}}$  valeur sinon l'équivalence n'est plus vraie. Une autre possibilité est d'implémenter une moyenne variable selon le nombre de paquets intégrés, cela implique également un seuil variable.

Cet algorithme permet de moyennner sur les  $N$  dernières valeurs acquises. Dans le cas que nous modélisons ici, nous investiguons les 30 derniers paquets. Il y a deux zones de calcul, une pour chaque partie de l'inéquation. Par ailleurs, un compteur par 30 inhibe la sortie tant que les accumulateurs n'ont pas atteint la première tranche de 30 paquets.

L'adaptabilité des FPGA permet de n'utiliser que les formats dont nous avons besoin, des registres 15 ou 19 bits, un comparateur 29 bits par 38 bits. Ce ne sont pas des valeurs classiques de registre comme pour les microprocesseurs de 32 ou 64 bits.

Dans la simulation du module (Figure 70), Il y a des sorties du module que nous avons ajouté pendant l'étape de débogage et des signaux de vérifications. CPTN est un signal qui correspond à la validation de la bascule de sortie du module, Out\_ena est un signal du testbench pour vérifier CPTN. on mesure 3369 ns entre le signal BGi (Figure 56) et le signal CPTN, ce qui correspond à 30 échantillons de la tranche à moyennner ( $30 \cdot 110$  ns) plus une durée de 69 ns incluant l'avance du BGi sur l'horloge CLKUb et les retards de propagations du compteur 5 bits par 30. Pour les mêmes raisons que précédemment, c'est à dire les retards des portes logiques, (Figure 70) Out\_ena est un peu en avance sur CPTN.

Les signaux SR et SRDQ sont les sorties des registres à décalage ou pipeline de 30 cases. Ces registres permettent de retrancher la 31<sup>ème</sup> valeur précédente pour décaler la fenêtre de moyennage. Ainsi, notre fenêtre s'initialise jusqu'à atteindre 30 paquets et se décale d'un paquet pour parcourir toute la macro-impulsion. Le signal DQ\_CHK est un signal du banc de test pour connaître la différence QU – QD.

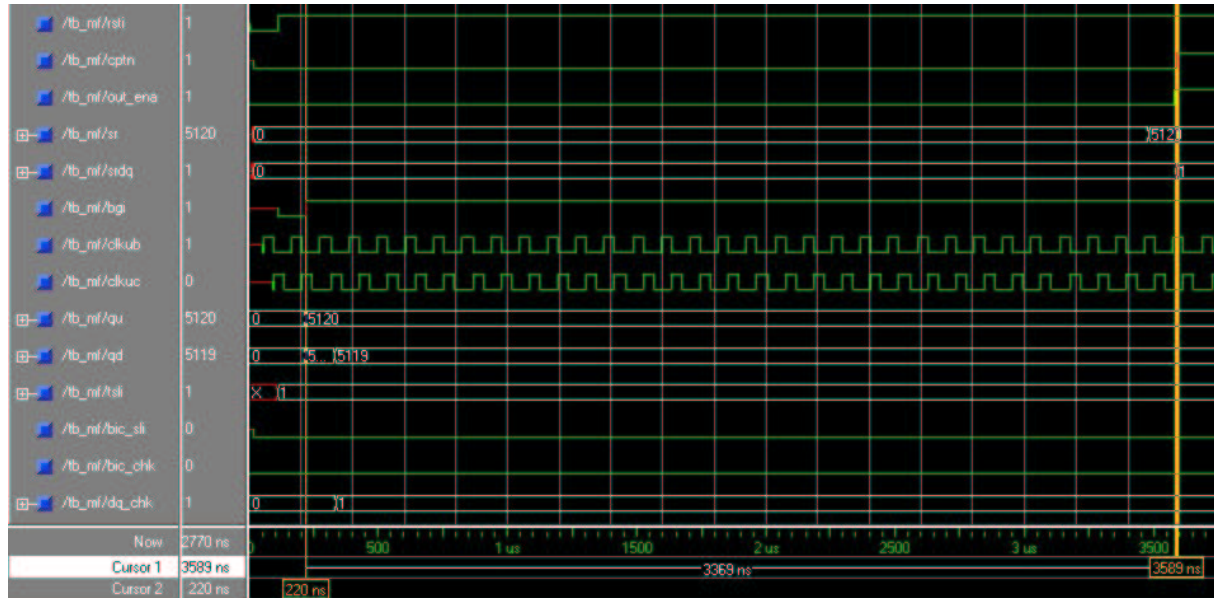


Figure 70 : Simulation du mode SLICE, zoom sur les signaux inhibiteurs de la sortie

Les données de vérification A\_CHK et B\_CHK correspondent aux opérandes du comparateur.

$$\left\{ A\_CHK = T_{slice} \times \sum_{i=1}^N QU_i \right\} < \left\{ B\_CHK = 100 \cdot N \cdot \sum_{i=1}^N |QD_i - QU_i| \right\}$$

Dès que B\_CHK est supérieur à A\_CHK, le signal de vérification BIC\_CHK est émis, pour être comparé avec la sortie BIC\_SLI du module.

Le curseur en surbrillance (Figure 71) indique les circonstances de déclenchement de l'alarme, les valeurs de cet instant sont celles écrites dans la marge après les noms des signaux. SQU\_CHK est la somme des charges du tore amont dans la fenêtre de moyennage. Le seuil TSLI est de 1% sur 30 paquets d'électrons de perte autorisé. C'est pourquoi nous avons : A\_CHK = SQU\_CHK. SDQ\_CHK est la somme des différences de charge entre tore amont et tore aval dans la fenêtre de moyennage. A et B sont des signaux intermédiaires à l'entrée du comparateur CMP\_SLICE du module, tel que A\_CHK et B\_CHK dans le banc de test VHDL.

La charge du tore amont est de 5120 (Figure 71) en décimal ou 2.56 nC. Lorsque les pertes simulées entre le tore amont et le tore aval atteignent 52 en décimal (> 1%) dans la fenêtre de 30 paquets, que l'on peut lire sur SDQ\_CHK, les signaux BIC\_SLI et BIC\_CHK passent au niveau logique haut pour indiquer une alarme. On remarque encore l'instabilité (glitch) des signaux A et B par rapports à leurs homologues du testbench.

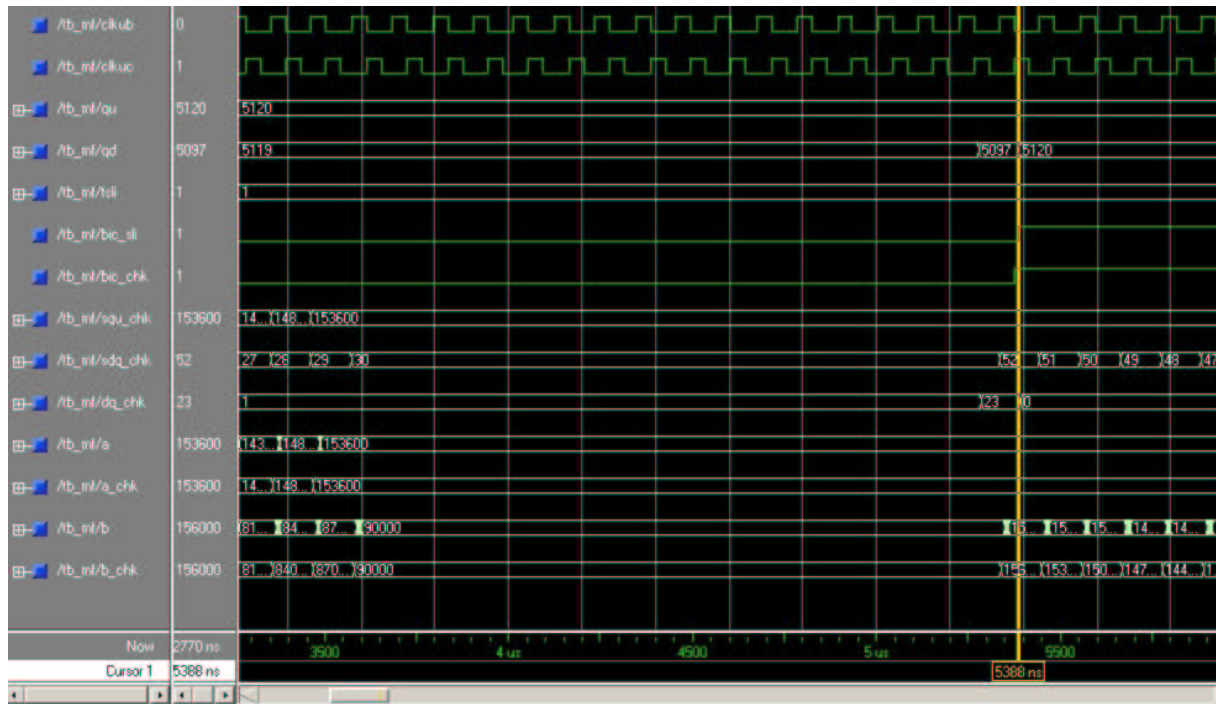


Figure 71 : Simulation du mode SLICE, zoom sur la sortie d'alerte

### 1.9 Le module INTEGRATION

Ce dernier module (Figure 72) examine la somme de la différence de charge de tous les paquets dans la macro-impulsion et déclenche une alarme dès que le seuil est franchi.

$$\sum_{i=0}^n |QD - QU|_i > T_{int}$$

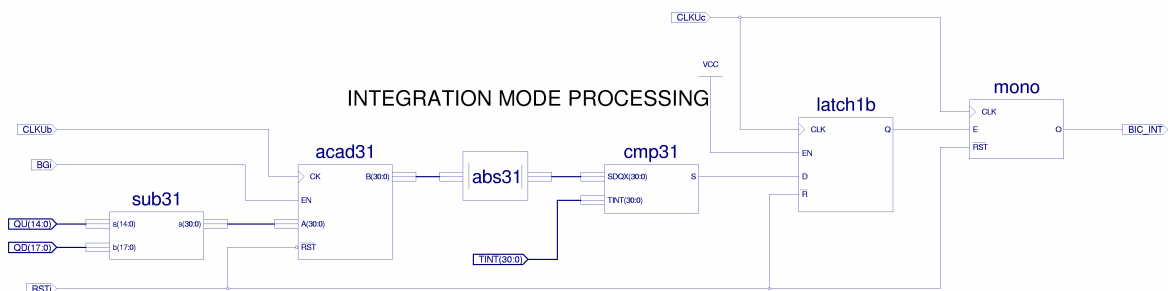


Figure 72 : Module INTEGRATION

Deux scénarios sont présentés (Figure 73). Dans un cas, la charge totale est supérieure au seuil Tint, dans l'autre elle ne l'est pas. De même que pour le testbench du module SLICE, un signal de vérification, SDQ\_CHK, donne la somme des différences (QU-QD). Il est comparé au seuil de ce mode Tint qui est une valeur donnée en nC, 16383 en décimale ou ~ 8.19 nC de pertes de faisceau autorisé sur toute la macro-impulsion. Selon le résultat, un signal BIC\_CHK permet de vérifier que la sortie BIC\_INT est correcte.

Dans tous les modules, la taille des registres a été calculé pour éviter des overflows pour toutes les valeurs possibles, par exemple pour une opération de soustraction entre deux mots de 14 bits, nous étendons la sortie sur 15 bits. Or, si l'on considère les valeurs mesurées en sortie des tores ( $< 2V$ ) et que nous tenons compte des seuils imposés pour chaque mode, nous pouvons économiser sur les tailles de registres de manière à ce qu'elle soit suffisante pour notre application. Ce sera fait dans la phase d'optimisation de l'architecture complète.

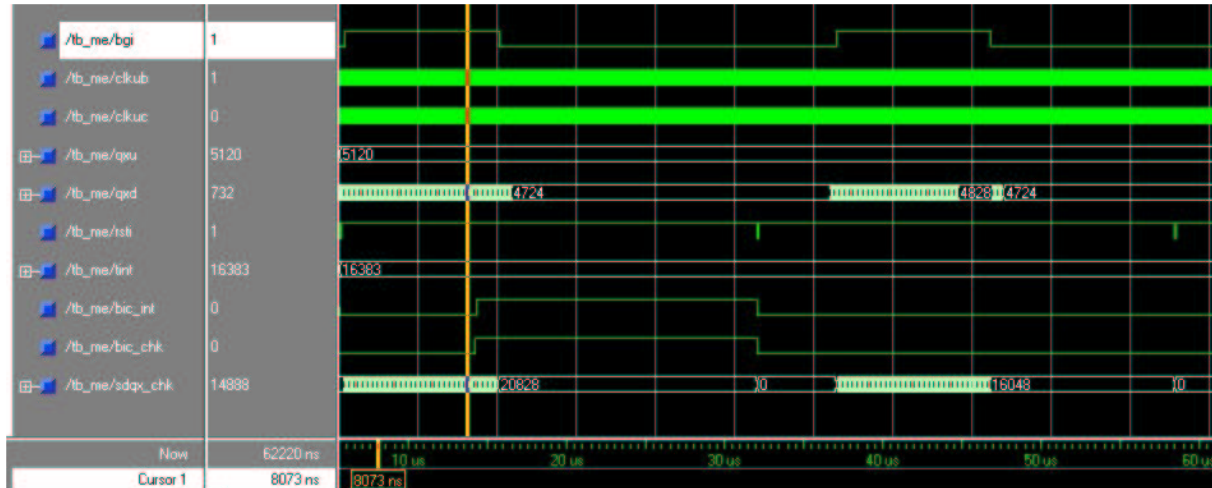


Figure 73 : Simulation du mode INTEGRATION

## 2 Essais de la carte TPS (Toroid Protection System)

### 2.1 Test de configuration du FPGA

Un premier test très simple consistait à mettre les sorties FLAGS[0:7] à l'état haut. Ces dernières étaient visualisées à l'aide d'un petit montage sur des LEDS. Nous avons repris exactement le même mode de configuration que sur la carte d'évaluation. A la première mise sous tension, contrairement au test de la carte d'évaluation, le téléchargement ne se faisait pas de la PROM vers le FPGA. Par contre, le téléchargement des données par le câble PCIV du PC vers la PROM était correct. En effet, la fonction de relecture des données téléchargée dans la PROM a permis de les comparer à ceux du fichier MCS contenant la configuration du FPGA. Après comparaison des fichiers, nous en avons déduit que ce n'était pas un problème de téléchargement par le PC vers la PROM. Après un certain nombre d'investigation, nous avons observé sur l'oscilloscope des overshoots sur les signaux DIN et CCLK. Il était donc question de couplage de ligne, car ces deux pistes de largeurs  $220 \mu\text{m}$  étaient trop proches l'une de l'autre. La solution fut d'ajouter une résistance en série pour allonger le temps de montée sur ces deux lignes et de déplacer la piste DIN à l'écart de l'horloge CCLK.

### 2.2 Signaux d'horloges

Nous avons vérifié que les échantillonnages des valeurs TOP et BOT étaient bien séparées de 80 ns (Figure 74). En bleu (Figure 75), nous visualisons les signaux d'horloges pour la synchronisation séquentielle du design FPGA. En noir, l'horloge d'échantillonnage de la valeur TOP sur le signal du tore amont, la sortie 14 bits de l'ADC\_TOP est présente 3 cycle + 8 ns après le front montant de CLKU\_T, nous nous laissons donc 10 ns supplémentaire avant de mémoriser les données dans un registre à l'aide de CLKUa. CLKUb permet de synchroniser les calculs intermédiaires : opération arithmétique, registre pipeline,

accumulateur alors que CLKc synchronise les signaux d'alarmes. Ces signaux ont profité des améliorations obtenues à l'aide des prototypes de modules ADC.

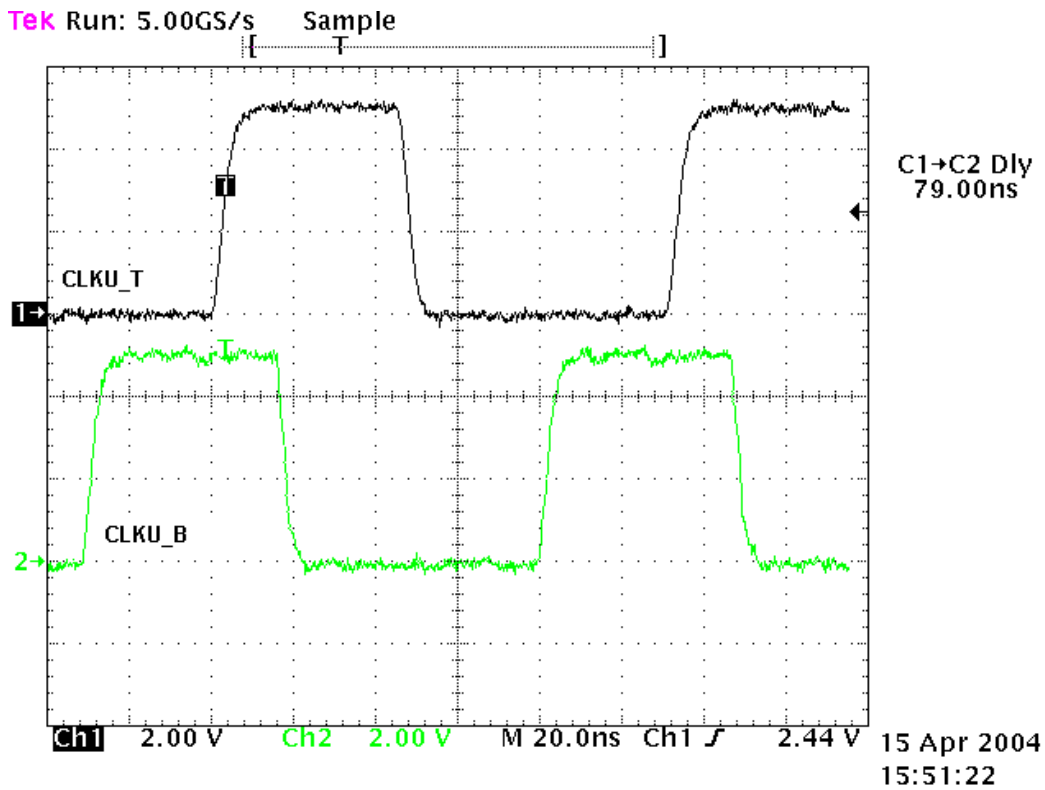


Figure 74 : Horloge des ADCs amonts de la carte TPS

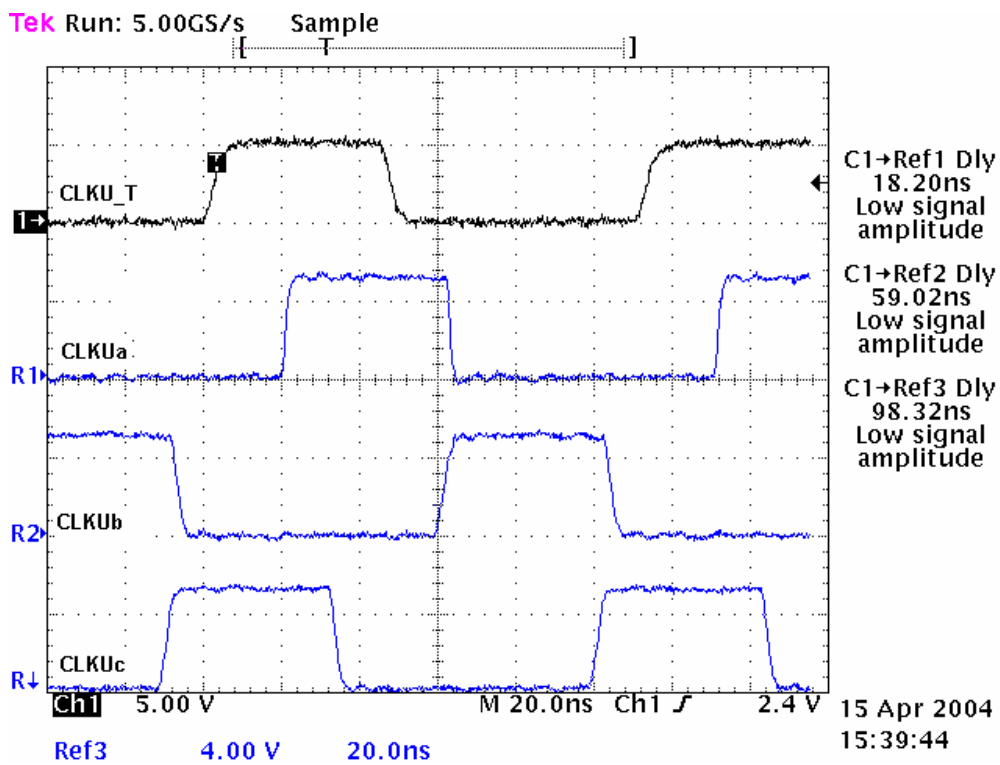


Figure 75 : Horloges pour la synchronisation interne des modules de l'architecture FPGA

## 2.3 Test des ADC et du DAC

Avant chaque configuration du FPGA, nous avons effectué la simulation post routage des modules avec l'assignement des broches correspondant à la carte. Nous configurons le FPGA pour permettre la sélection des PDG et des sorties ADC à transférer en entrée du DAC via le port 2 du microcontrôleur.

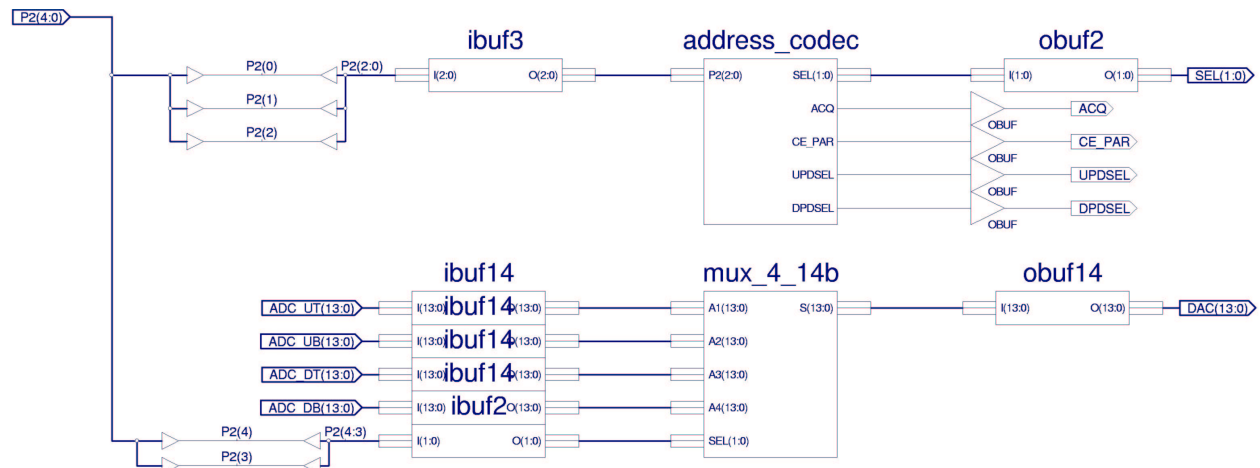


Figure 76 : Configuration FPGA pour les acquisitions ADC vers DAC

Pour la génération des signaux d'horloges et de stimuli, nous avons utilisé un générateur de signaux arbitraires 4 voies en mode intervoie pour la synchronisation.

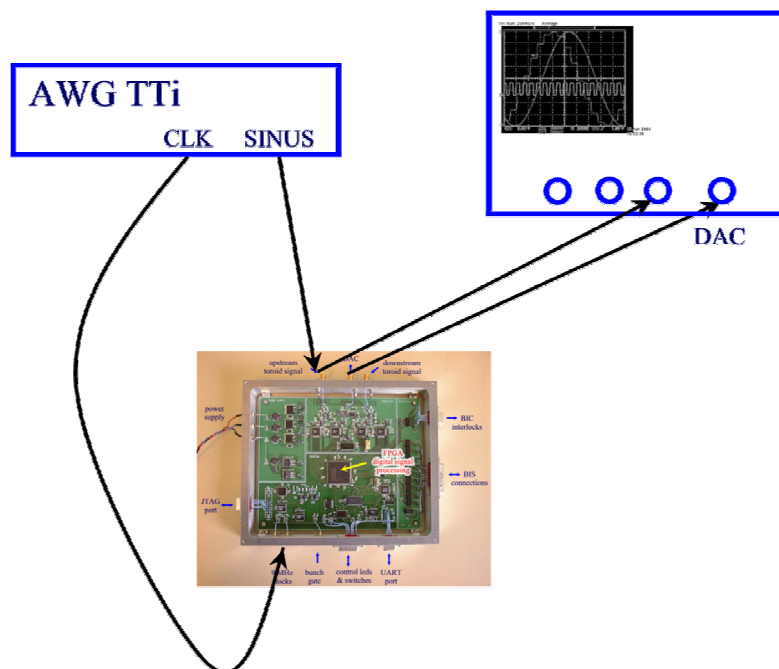


Figure 77 : Montage Acquisition d'un sinus

Une première vérification de l'ensemble ADC-FPGA-DAC est réalisée par échantillonnage d'un signal sinus que nous voulions synchroniser à l'horloge nominale de 9 MHz mais à une fréquence différente. Le générateur TTi ne peut être synchroniser qu'avec des fréquences multiples de deux. Pour cela, nous avons choisi  $f_{\text{sin}}=564 \text{ KHz}$ , ce qui donne une division par 16 de la fréquence d'horloge ou 16 échantillons par période.

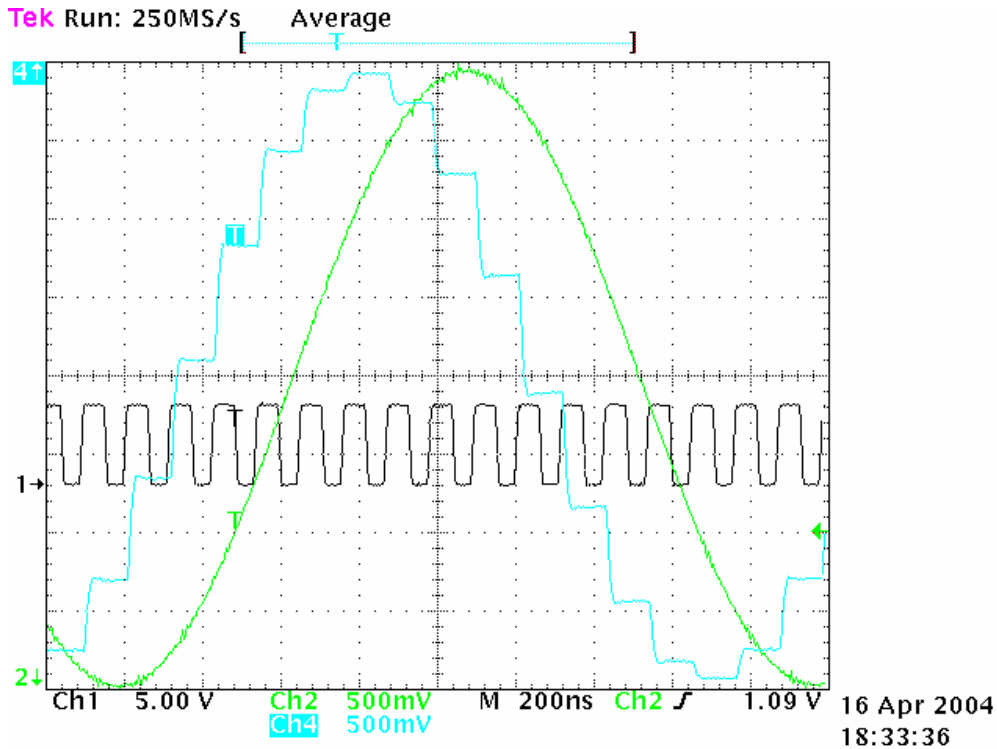


Figure 78 : Test échantillonnage

Le signal est échantillonné de la même manière sur les quatre ADCs. Le résultat est satisfaisant car à chaque front montant de l'horloge, nous avons un échantillon. Il y a bien 16 échantillons par période et l'amplitude en sortie du DAC correspond à celle du générateur.

## 2.4 Evaluation des ADC

Pour évaluer la précision de la mesure des ADC et savoir quelles sont les possibilités en termes de seuil que nous pourrions atteindre, nous mesurons une tension continue fournie par deux piles 1.5V reliée à un pont diviseur.

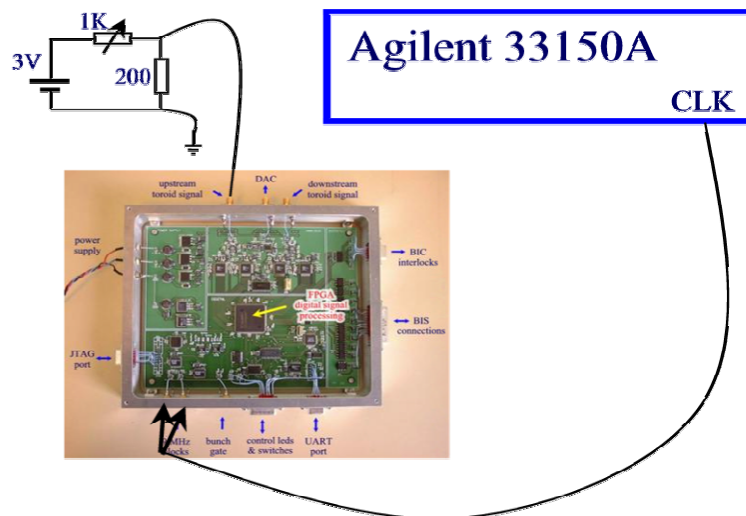


Figure 79 : Montage pour l'évaluation des ADC avec une tension continue



La mesure est faite sur une tension continue connectée à une seule entrée tore, pour cette raison nous avons programmé le FPGA pour transférer les acquisitions d'une seule entrée tore, donc deux sorties ADC.

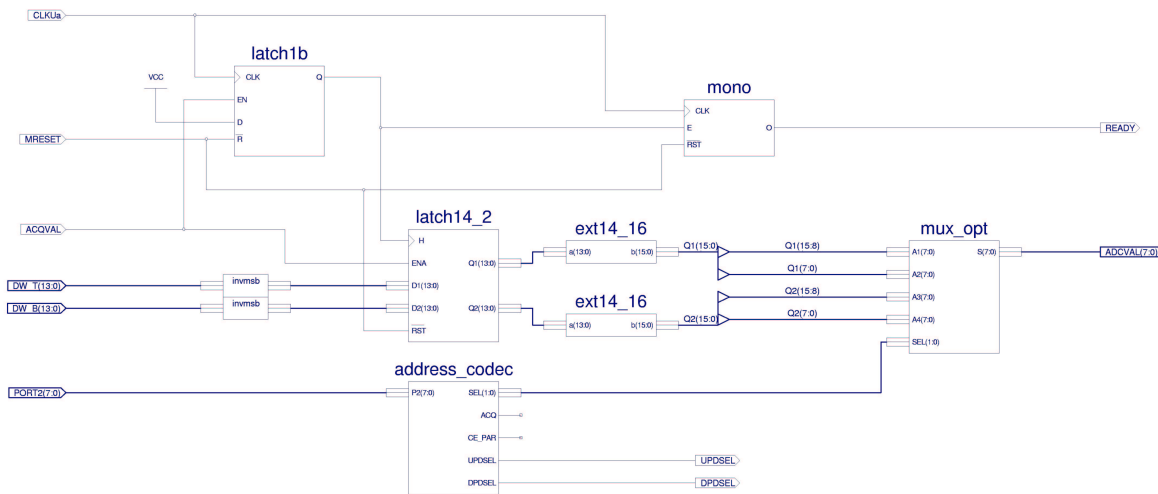


Figure 80 : Configuration FPGA pour l'évaluation des ADCs niveau 2

L'horloge d'échantillonnage est toujours à 9 MHz, les sélections par le port2 de UPDSEL et DPSEL sont disponibles pour valider les horloges. Pour chaque acquisition, le microcontrôleur fait une RAZ, active la commande ACQVAL et attends une interruption déclenchée par un front descendant de READY. Ci-dessous, la même configuration mais à un niveau hiérarchique supérieur (niveau 1).

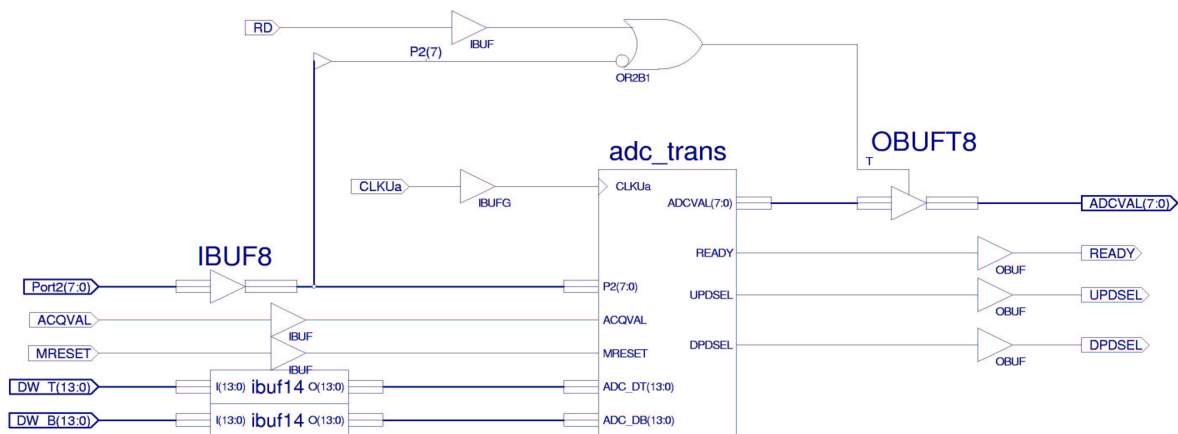


Figure 81 : Configuration FPGA pour l'évaluation des ADCs niveau 1

Dès l'interruption, l'ADUC812 adresse un octet et le lit sur le port P0 avant de le stocker dans la RAM. Ce qui a l'air simple mais si l'on n'y prend garde, il y a conflit sur P0 car il est déclaré en sortie pour le FPGA et permet d'adresser la RAM externe pour le microcontrôleur. La solution est d'utiliser la commande read RD ou la commande de sélection de la RAM P2(7) pour mettre P0 en haute impédance par le FPGA.

	0V	-1.5V	-1V	1V	1.5V
Max (V)	0.0155	-1.492	-0.9952	1.0192	1.519
Min (V)	0.0092	-1.483	-0.9862	1.0117	1.512
Val moy (V)	0.01226	-1.4886	-0.9916	1.015	1.515
$\sigma$ (mV)	1.045	1.225	1.15	1.136	1.11
$\sigma/V_{\text{moy}}(10^{-3})$		0.8229	1.1597	1.1192	0.7326

Tableau 8 Mesures de 10000 échantillons et calcul de l'écart type pour l'ADC\_UT

	0V	-1.5V	-1V	1V	1.5V
Max (V)	0.01025	-1.4905	-0.9912	0.998	1.4912
Min (V)	0.00525	-1.4830	-0.9847	0.9927	1.485
Val moy (V)	0.00722	-1.4872	-0.9887	0.9951	1.4877
$\sigma$ (mV)	0.869	1.02	0.939	0.882	0.931
$\sigma/V_{\text{moy}}(10^{-3})$		0.6858	0.9497	0.8863	0.6257

Tableau 9 Mesures de 10000 échantillons et calcul de l'écart type pour l'ADC\_UB

	0V	-1.5V	-1V	1V	1.5V
Max (V)	0.0117	-1.4932	-0.995	1.006	1.5
Min (V)	-0.00025	-1.482	-0.9827	0.994	1.49
Val moy (V)	0.00615	-1.4883	-0.989	1	1.4942
$\sigma$ (mV)	1.755	1.686	1.754	1.812	1.675
$\sigma/V_{\text{moy}}(10^{-3})$		1.1328	1.7735	1.812	1.121

Tableau 10 Mesures de 10000 échantillons et calcul de l'écart type pour l'ADC\_DT

	0V	-1.5V	-1V	1V	1.5V
Max (V)	0.00675	-1.488	-0.9927	0.9965	1.491
Min (V)	-0.00025	-1.479	-0.985	0.9887	1.4815
Val moy (V)	0.002568	-1.4842	-0.9894	0.9921	1.4863
$\sigma$ (mV)	1.287	1.33	1.264	1.352	1.507
$\sigma/V_{\text{moy}}(10^{-3})$		0.8961	1.2775	1.3627	1.0139

Tableau 11 Mesures de 10000 échantillons et calcul de l'écart type pour l'ADC\_DB

Les résultats des mesures d'écart type sur les ADCs du tore amont (Tableau 8, Tableau 9) sont meilleurs que pour les ADCs du tore aval (Tableau 10, Tableau 11). Les raisons de cette différence sont la tolérance des composants actifs et passifs du frontal analogique et les pistes des signaux d'horloge qui pour aller vers les ADC du tore aval doivent parcourir un plus long trajet.

Globalement, les résultats des mesures montrent que le travail sur les prototypes de module ADC a permis d'obtenir de bons résultats puisque nous atteignons une précision de la mesure des ADC de moins de 2 pour 1000. De très bon augure, s'il n'y a pas d'autres sources de bruit.

## 2.5 Test de la phase d'étalonnage

L'objectif est de chercher la valeur optimale sur 8 bits de chaque délai programmable pour ajuster l'échantillonnage de la valeur top des tores amont et aval. Puis, de calculer la valeur du facteur de correction et de le normaliser. Enfin, toutes ces valeurs doivent être stockées dans la mémoire flash du microcontrôleur.

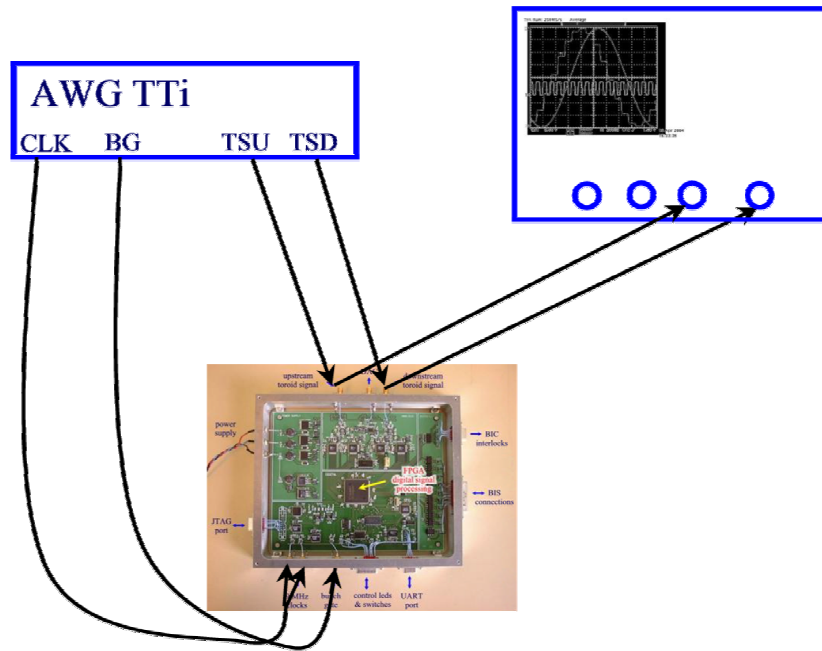


Figure 82 : Montage pour les tests d'étalonnage du système

La génération des signaux BG, CLK, TSU et TSD pour respectivement Bunch Gate, horloge 9 MHz, signal tore amont et signal tore aval est réalisée (Figure 82) pour avoir la configuration suivante :

- $F_{rep}$  macro-impulsion : 116  $\mu$ s
- Temps de vol (amont-aval) : 110.8 ns
- Un paquet/macro-impulsion
- BG centré sur signal tore amont.

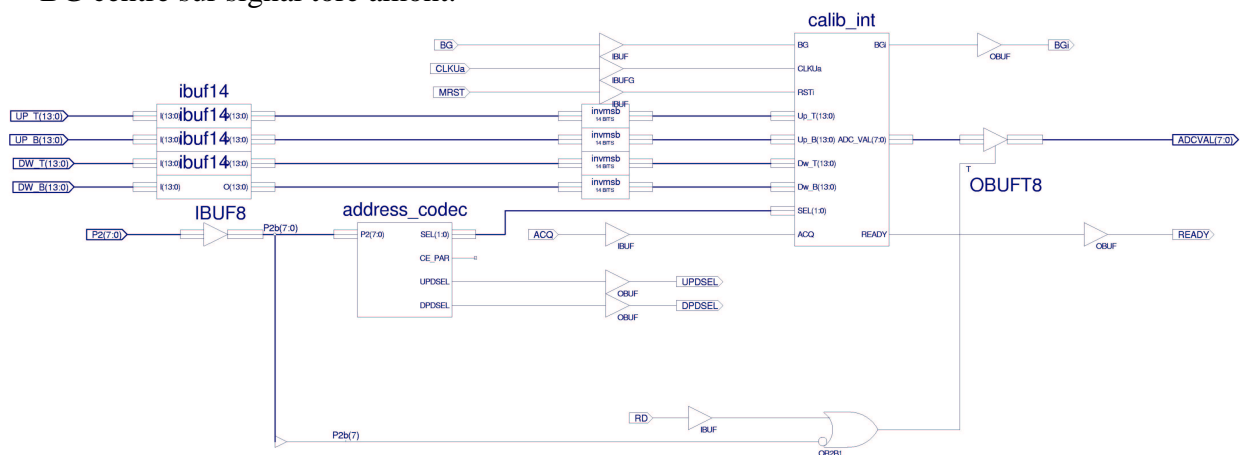


Figure 83 : Configuration FPGA pour l'étalonnage du système

Le microcontrôleur fixe une valeur sur le PDG amont et balaye 50 valeurs sur le PDG de l'horloge aval. Les 50 valeurs successives de retard permettent de couvrir une plage de 25ns, ce qui est suffisant pour atteindre le maximum du signal. Pour chaque retard, il acquiert 100 valeurs de QD, en fait la moyenne et la mémorise avec le retard. A la fin du balayage, il recherche la valeur max de QD et le retard correspondant, ce couple de données est rangé dans la RAM. La même acquisition est faite pour le PDG de l'horloge amont (QU). Enfin, le rapport QU/QD est calculé et mis en forme pour le traitement FPGA.

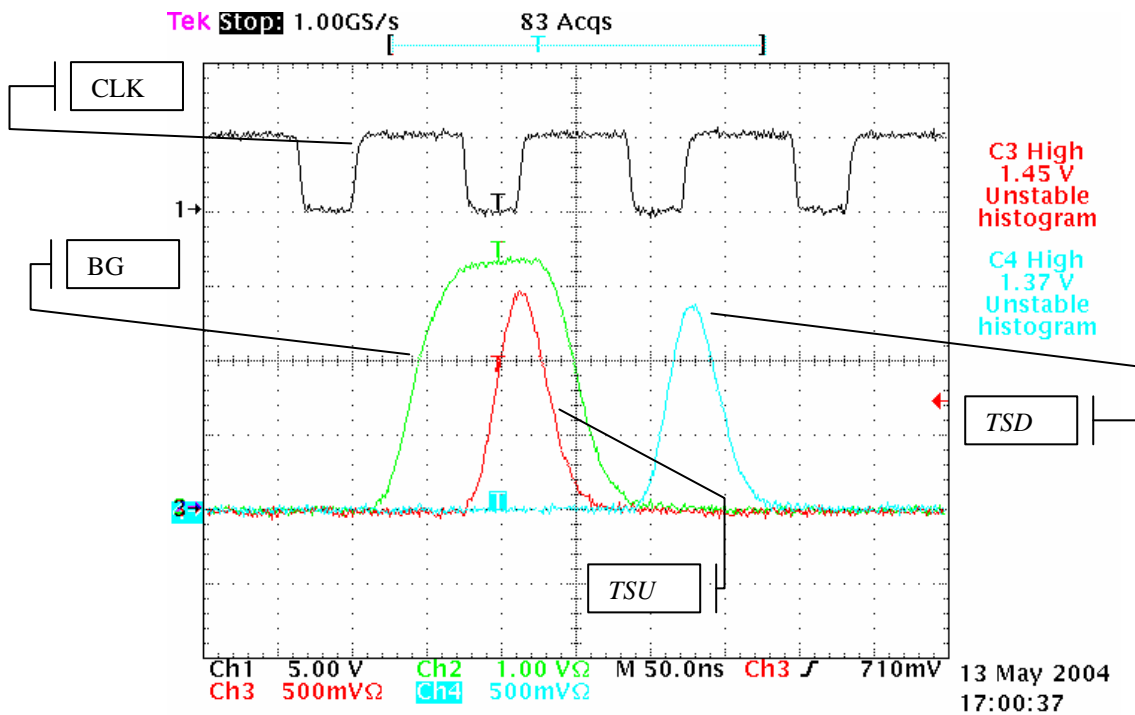


Figure 84 : Etalonnage, cas avec les amplitudes des signaux tores à peu près égales

### Résultats de l'acquisition pour les signaux de la Figure 84:

Charge amont (QU) :

la valeur max est = 1.422069V pour un retard de 17 ns

Charge aval (QD) :

la valeur max est = 1.373801V pour un retard de 17.5 ns

Le rapport QU/QD est de : 1.035 => soit un facteur de correction de  $(1059)_{dec}$ .

Les mesures ci-dessus sont conformes aux signaux injectés dans la carte TPS, les amplitudes maximums correspondent aux amplitudes des signaux du générateur.

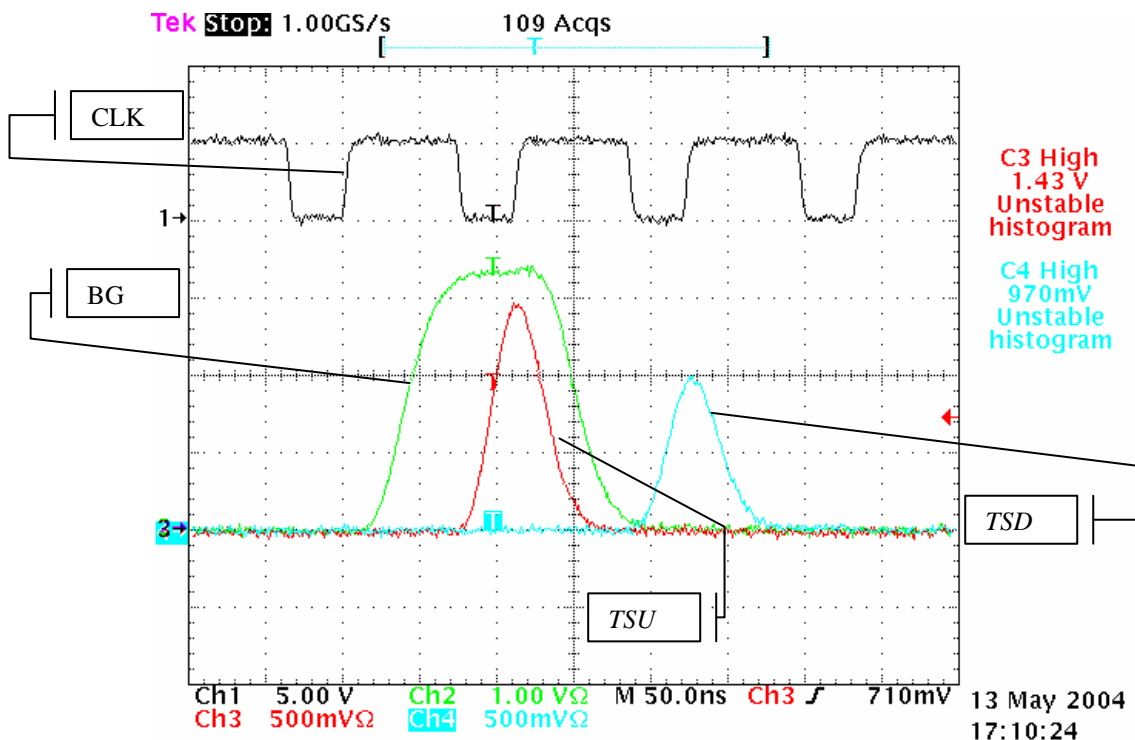


Figure 85 : Etalonnage, cas avec les amplitudes des signaux tores différentes

## Résultats de l'acquisition pour les signaux de la Figure 85 :

Charge amont (QU) :

la valeur max est = 1.409035V pour un retard de 16.5 ns

Charge aval (QD) :

la valeur max est = 0.962162V pour un retard de 17.5 ns

Le rapport QU/QD est de : 1.464 => soit un facteur de correction de (1499)<sub>decimale</sub>

Là aussi, les amplitudes mesurées sont semblables aux amplitudes des signaux du générateur de fonction. L'étalonnage du système fonctionne correctement, ce qui laisse à penser que les essais sur l'accélérateur ne poseront pas de problème.

### 3 Essais de la carte à DESY

Le câblage de la baie de sécurité de la machine n'étant pas fait, les essais ont été réalisés près du canon RF de l'accélérateur (Figure 86) pour utiliser les baies fournissant des diagnostics. Le canon RF se trouve juste derrière les blocs de bétons en rouge sur la Figure 86. Les premiers essais du canon RF, injecteur de paquet d'électrons, ont débuté au mois d'avril 2004, ces essais ont permis à nos collègues allemands de mesurer les tores T1 (sortie du canon RF) et T2 (sortie du 1<sup>er</sup> cryomodule). Notamment, ces tests leur ont aussi permis de régler les gains des amplificateurs 4 voies (Figure 12).

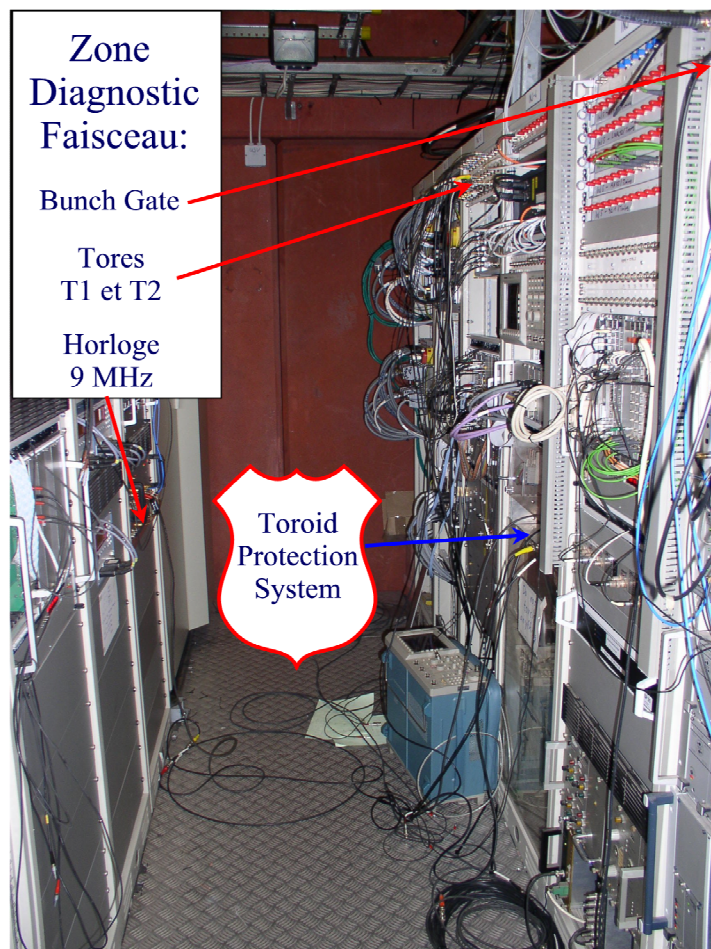


Figure 86 : Site de mesure à DESY pour les essais avec faisceaux du système TPS

Nous n'avions pas le droit à l'échec pour ces premiers essais car le tunnel où se trouve TTF2 devait fermer pour trois mois (été 2004) la semaine suivant nos essais, afin de finir les derniers montages de composants du linac. Nos collègues allemands nous ont fournis un montage (Figure 87) pour isoler galvaniquement notre carte des tiroirs générant les signaux de contrôle (horloge 9.0277 MHz et BG). Ce montage avait des sorties en haute impédance, il nécessitait des câbles les plus courts possibles entre la sortie du montage d'isolation et l'entrée de la carte TPS.

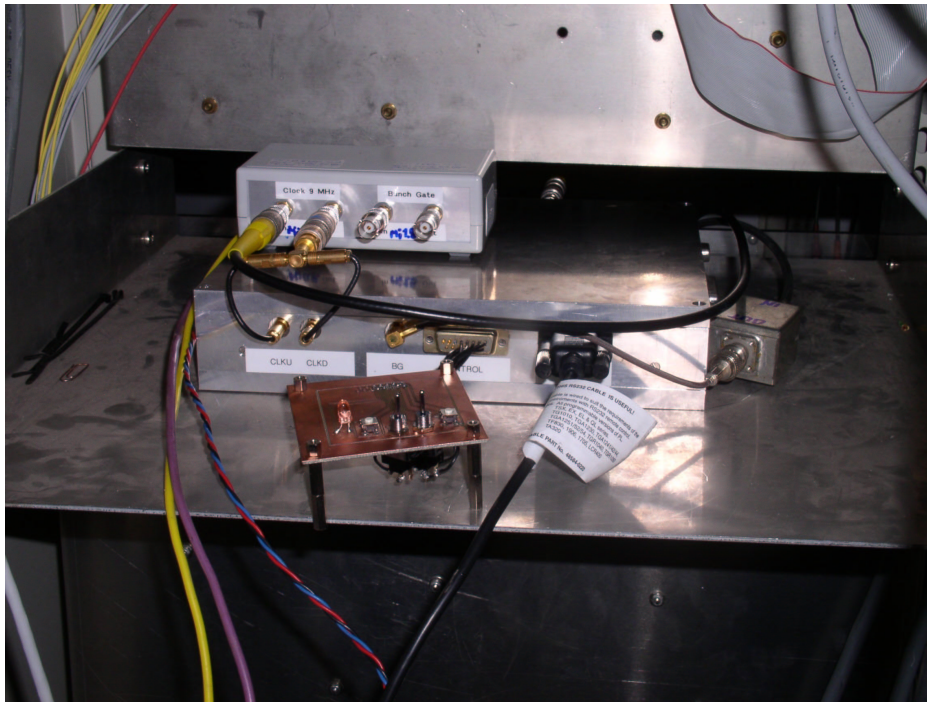


Figure 87 : Montage d'isolation galvanique des signaux de contrôle de la carte TPS

### 3.1 Essais de l'étalonnage du système

Dans un premier temps, nous avons réalisé des essais en laboratoire à DESY pour vérifier une dernière fois notre système. Lorsque nous étions prés, nous avons demandé au coordinateur en chef des opérations sur la machine TTF2 un faisceau avec un seul paquet d'électrons dans la macro-impulsion. Nous avons programmé le FPGA avec la configuration représentée par la Figure 83 comme pour les tests en laboratoire.

Nous n'avions pas exactement la mise en forme des signaux prévue pour le test, nous avons donc ajouté des délais à l'aide de câble pour pouvoir avoir la situation de la Figure 88. Puis, après la connexion de tous les signaux sur la carte, l'heure H du premier test avec faisceau arriva. Malgré une inquiétude due à la durée du traitement de l'étalonnage beaucoup plus longue, cette phase se passa aussi bien qu'en laboratoire. Effectivement, l'acquisition des signaux était plus longue car la fréquence de répétition des macro-impulsions générées pour les essais en laboratoire (116  $\mu$ s) est environ 1724 fois plus courte que celle de l'accélérateur (200 ms). Or le nombre de valeur par pas était de 100 échantillons en laboratoire, ce qui implique un temps de traitement de plus de 30 mn sur la machine !

Résultats des mesures d'étalonnage :

**Mesure 1 : (sans moyenne)**

**Condition :** => balayage sur 25 ns (50 pas)  
Nombre de valeur par pas : 1

Etalonnage QU (T1)

(T1) Tension maximale mesurée = 0.537250 V => **1.075 nC**  
delai\_QU = 43 (pas de décalage de l'horloge pour l'échantillonnage de la valeur maximum, c'est à dire 21.5 ns puisqu'un pas fait 0.5 ns)  
Écart-type = 6.5 mV

Etalonnage QD (T2)

(T2) Tension maximale mesurée = 0.446875 V => **0.89375 nC**  
delai\_QD = 43  
Écart-type = 3.625 mV  
Correction QU/QD = 1.202238

**Measure 2 :**

**Condition :** => balayage sur 25 ns (50 pas)  
Nombre de valeur par pas : 20

Etalonnage QU

(T1) Tension moyenne maximale mesurée = 0.529513 V => **1.059026 nC**  
delai\_QU = 44  
Écart-type = 9.494 mV

Etalonnage QD

(T2) Tension moyenne maximale mesurée = 0.414125 V => **0.82825 nC**  
delai\_QD = 46  
Écart-type = 51.851 mV  
Correction QU/QD = 1.278630

**Measure 3 :**

**Condition :** => balayage sur 7.5 ns (15 pas)  
Nombre de valeur par pas : 50

Etalonnage QU

(T1) Tension moyenne maximale mesurée = 0.509215 V => **1.01843 nC**  
delai\_QU = 46  
Écart-type = 9.451 mV

## Etalonnage QD

(T2) Tension moyenne maximale mesurée = 0.429975 V => **0.85995 nC**

delai\_QD = 41

Écart-type = 34.039 mV

Correction QU/QD=1.184290

Les résultats des mesures montrent des écarts types plus grand pour les mesures 2 et 3. Ce qui a été vérifié directement en sortie du tiroir (Figure 86) de sortie des tores. L'ondulation est bien de l'ordre d' ~ 30 mV en moyenne. Cette ondulation qui n'était pas visible lors des réglage sur les tores par nos collègues allemand, peut s'expliquer d'une part par la longueur des câbles pour amener les signaux de la machine jusqu'à la baie, d'autre part par l'environnement extrêmement bruyant du site de mesure.

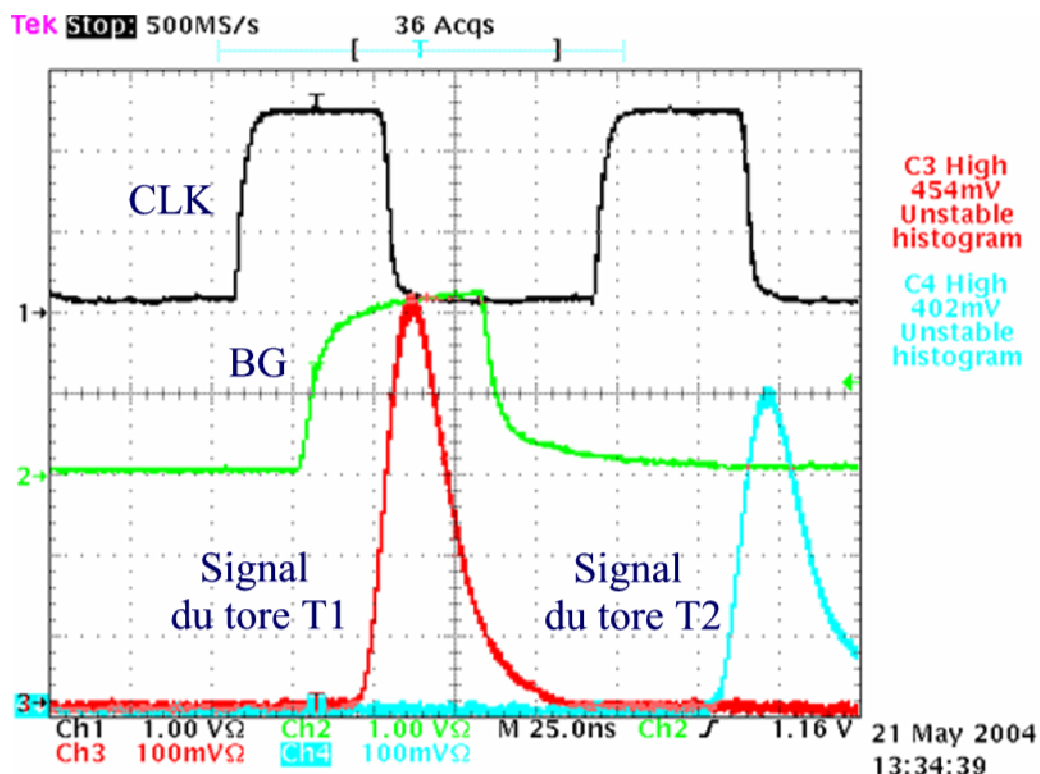


Figure 88 : Mesure des signaux BG, horloge 9 MHz et sortie des tores T1 et T2 sur l'accélérateur TTF2

### 3.2 Essais des modes Single Bunch et Charge Validation

Pour ce test, nous avons demandé au coordinateur des opérations de TTF2 deux formes de faisceaux, une pour la phase d'étalonnage (un paquet d'électrons par macro-impulsion) qui devra toujours précéder la première mise en route du système et une pour le test des modes absence de charge ou signal porte manquant et paquet par paquet (minimum de dix paquets par macro-impulsion).

La configuration du FPGA (Figure 90) consistait à intégrer sept modules sur les neuf modules de l'architecture complète simulés en laboratoire : les modules BTC, CALCQx&TRANSμC, AUTORST, ADDRESS CODEC, PARAMETERS INTERFACE, CHARGE VALIDATION et SINGLE MODE.



La vérification (Figure 89) des modes protection paquet par paquet et validation de charge est faite à l'aide de LEDs. Pour cela, j'ai modifié la configuration du FPGA pour pouvoir lire les sorties Interlocks à travers les opto-coupleurs du registre d'état de la carte. Ces derniers étaient reliés à un montage pour visualisation avec des LEDs.

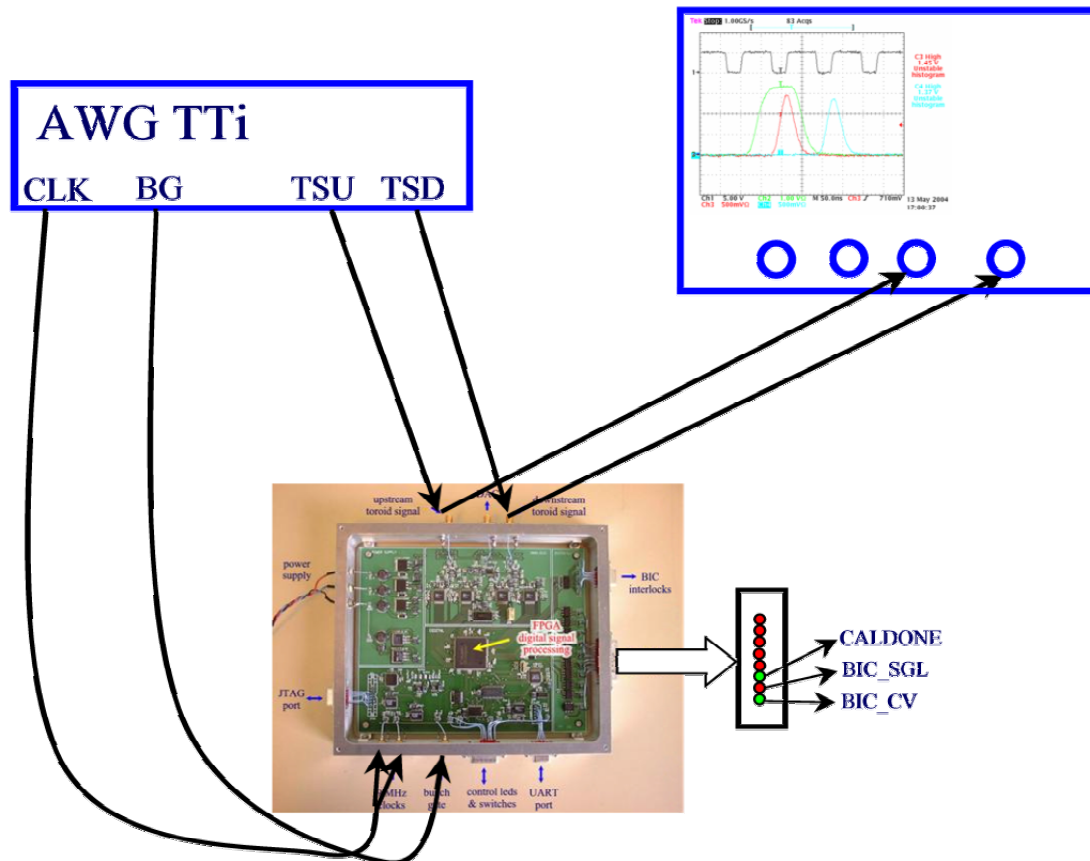


Figure 89 : Montage de test des modes Single et Charge Validation

Nous avons commencé par initier la procédure d'étalonnage qui incluait en plus le transfert des seuils pour les modules Charge Validation mode et Single mode ainsi que le facteur de correction calculé à la fin de l'étalonnage. Puis, après que la LED Caldone nous indiquait que l'étalonnage était terminé, nous demandions aux opérateurs de la machine de passer en transmission multi-paquets. Une fois que nous avons vérifié qu'il n'y avait pas de déclenche d'alarme. Nous contactons à nouveau la salle de commande pour demander aux opérateurs d'atténuer le faisceau afin que l'amplitude mesurée par le tore en aval (T2) soit suffisamment faible pour la déclenche des alarmes. Nous avons pu alors vérifier que l'alarme BIC\_SGL était émise. Le mode Charge validation a été contrôlé en débranchant le câble transportant le signal BG puis le signal TSU du tore en amont (T1).

Nous avons pu constater que les modes principaux de protection de la machine fonctionnaient correctement. Les modes Integration et Slice seront testés prochainement. D'abord, en laboratoire à l'aide d'une carte de test basée sur un FPGA en cours d'élaboration par nos collègues allemands. Puis à DESY en automne, pour notre deuxième campagne de test. Je pense que nous laisserons un système de protection sur place après ces essais.

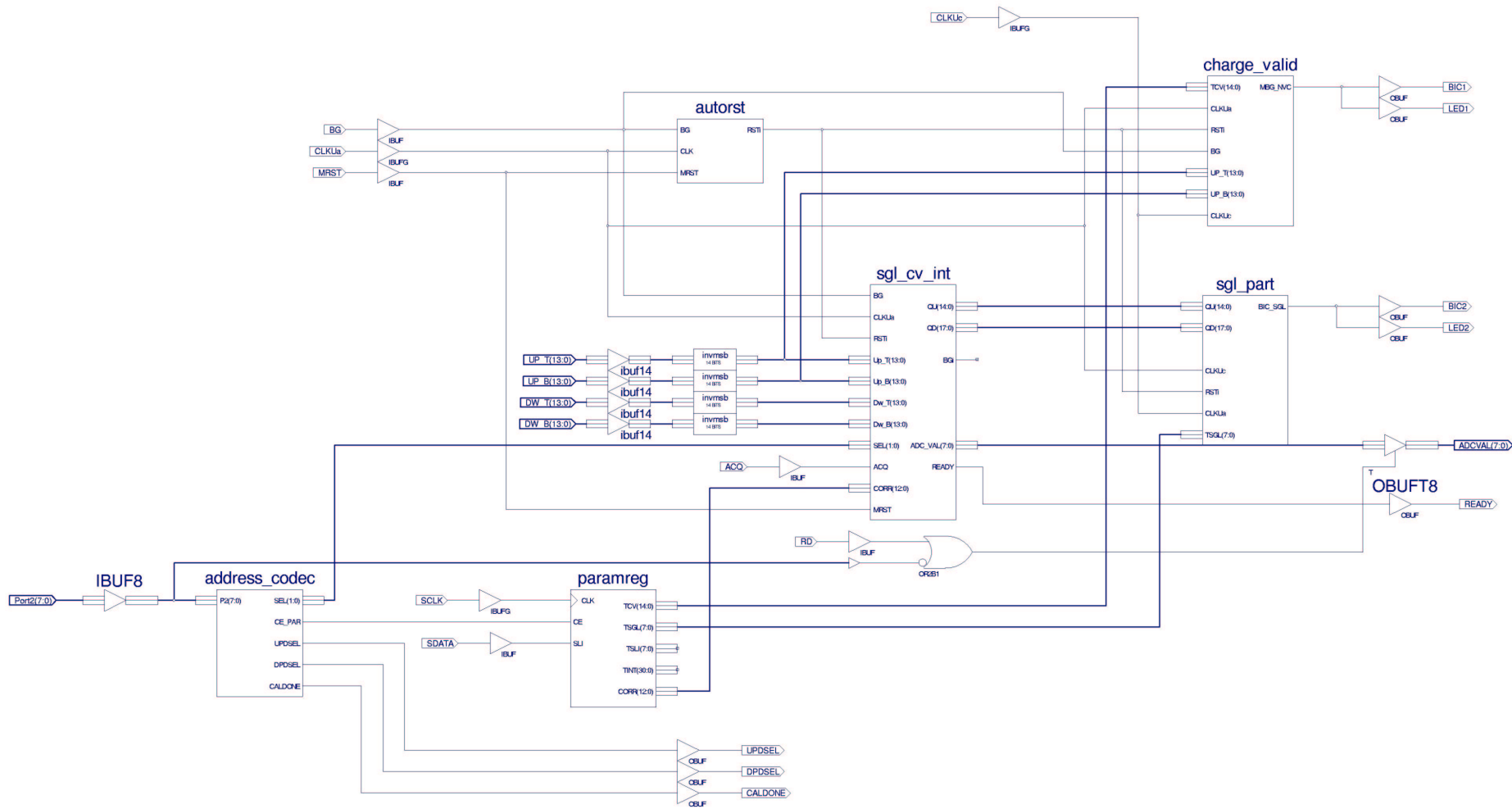


Figure 90 : Configuration du FPGA pour les essais des modes Single et Charge Validation

# Conclusion

Cette étude a été enrichissante à bien des égards, elle m'a permis de mettre en pratique des savoirs acquis pendant toutes ces longues soirées et week-end au CNAM. Jusqu'ici, seules mes compétences en hyperfréquence et en informatique industrielle avaient été utilisées. J'ai pu m'attaquer à l'électronique analogique et numérique pour un projet concret. C'est à la rencontre des problèmes telles que l'immunité au bruit et les couplages entre signaux pour la partie analogique et la réalisation de fonctions qui peuvent sembler simple tel qu'un monostable ou la synchronisation des signaux pour la partie numérique que l'on commence à comprendre réellement l'électronique.

La prochaine étape, à court terme, consiste en l'assemblage de l'ensemble des modules simulés en une seule architecture pour dans un premier temps une simulation tenant compte de l'assignement des broches conformément à la carte TPS. Dans un deuxième temps, nous ferons l'optimisation avec des contraintes de temps de propagation entre portes de manière à voir si les délais entre les horloges de synchronisation interne ne peuvent pas être un peu plus courts. Ce travail sera présenté sous forme de poster à l'EPAC (European Particle Accelerator Conference) 2004 en Suisse au mois de juillet 2004.

La réalisation d'un tiroir de distribution et de réglage automatique des signaux de contrôle ainsi qu'une seconde carte TPS doivent être réalisés pour la fin 2004. La prochaine carte TPS inclura des améliorations, notamment pour la configuration du FPGA et le remplacement des composants passifs dans la partie analogique par d'autres ayant des tolérances plus fine (0.1% au lieu de 5%). Ceci de façon à pourvoir l'installation TTF2 d'une sécurité rapide le plus tôt possible. Il faut savoir que les opérations sur le faisceau reprendront au mois de septembre 2004 et seront limités à des gradients en énergie de 100 MeV maximum en impulsion courte tant qu'il n'y aura pas de protection rapide.

Le système développé ici a hérité de l'expérience des premières sécurités différentielles réalisées dans un premier temps pour l'ALS (Accélérateur Linéaire de Saclay) puis pour TTF phase1. La dernière version : TVFP (Toroids based Very Fast Protection) était conçue pour protéger l'accélérateur en mode paquet par paquet à 1MHz, et en mode un paquet sur deux à 2.25MHz de fréquence de répétition des paquets d'électrons.

Pour des raisons de sécurité de la machine TTF2 (pour des dégâts causés sur une cavité supraconductrice, il faut compter environ 30k€ pour la remplacer) le système doit être très rapide pour détecter les pertes, néanmoins le temps de réaction du système sera en grande partie dû à la transmission de l'information en provenance de l'accélérateur et vers le système de contrôle du canon pour l'arrêt du faisceau. Malgré la rapidité du système de protection, le temps de réaction dû à la transmission des signaux d'alertes est de l'ordre de 3  $\mu$ s.

La sécurité différentielle TPS, permettra la génération de train d'impulsion ayant des formes arbitraires, répétition des paquets à fréquence variable au sein de la même macro-impulsion, fonctionnement paquet par paquet jusqu'à 10 MHz. La carte finale a coûté du point de vue matériel 4.5k€, la réalisation du splitter vaut environ 3k€, l'intégration de toutes les cartes dans des tiroirs avec leurs alimentations coûtera environ 1k€/carte, ce qui permet d'évaluer un coût total en matériel du système de protection pour quatre paire de tore d'environ 26 k€. Le système complet sera livré pour l'été 2005.

Un des problèmes qui a ralenti ce projet fut la modification des spécifications dans sa phase initiale. Cela concernait des parties non négligeables telle que l'existence ou non d'un signal de synchronisation sur chaque paquet d'électrons ou telle que les interfaces de communication avec le réseau de commande et de contrôle.

De plus, toutes les cartes (prototype module ADC : 4 couches, carte d'évaluation FPGA : 2 couches, carte TPS : 4 couches) ont été conçues du schéma au PCB par nos soins, ce qui demande une prise en main des outils de CAO. Toutes les réalisations faites autour de ce projet n'ont pas été décrites ici, notamment l'implémentation d'une carte prototype DAC 14 bits, d'un splitter 1 à 8 voies.

Dans une installation expérimentale du type de TTF2, il n'est pas aisé d'avoir accès au faisceau. Il y a des équipes d'opérateurs venant de tous les pays de la collaboration qui alternent jours et nuit pour le contrôle de la machine. Il y a aussi beaucoup d'utilisateurs avec des demandes de format de faisceau très divers. Le fait d'avoir pu utiliser la machine au moment et dans les conditions où nous le souhaitions montre une grande confiance entre nos partenaires à DESY et le laboratoire de Saclay. Je dois dire que travailler au sein d'une collaboration internationale dans l'objectif de relever un défi scientifique est très motivant en soi. Mais, la prise en charge d'un projet, son suivi, son développement a été une expérience nouvelle et riche pour moi.

# **Annexes**



## **Annexe A : Tores (DESY)**





## DESY TTF2 Toroid System

### Specifications:

Toroids	
Toroid core	Vitrovac 6030F, Vacuumschmelze Germany
Dimension 1	116 x 80 x 17,5 (mm)
AL1	1,4 $\mu$ H ... 1,6 $\mu$ H
Dimension 2	177 x 140 x 26,6 (mm)
AL2	1,56 $\mu$ H
Construction	half-cut for clamp over mounting, 4 single-turn windings with integrated impedance matching transformation, 2 single-turn windings for calibration signal input
Toroid core carrier	PEEK, Type NW34/ NW44, 150 x 20 mm, DESY Hamburg
Toroid housing	AlMg3, Type NW34/ NW44, 190 x 160 mm, DESY Hamburg
Bandwidth	12,5 kHz ... 150 MHz 12 kHz ... 200 MHz (typical)
Output Impedance	50 Ohms
Beam position dependency	nearly none
Sensitivity for 1nC Pulse	8 mVpp Output signal at 50 Ohms in configuration with 6dB Feedthrough and 15MHz LP-Gaussfilter
LP-Gaussian-Filter	
Construction	SMD components, DESY
Type	Z-matched LP-Filter, 12 <sup>th</sup> order
-3dB Cut-off Frequency	15 MHz (designed for 9 MHz Linac operation) Impedance 50 Ohms
Overshoot	0,5 %
Suppression	40 dB at 70 MHz 60 dB at 115 MHz
Passive Power Combiner	
Construction	brass, gold plated, DESY
Connectors	4x SMA Impedance 50 Ohms
<b>100 MHz Amplifier</b>	
Construction	SMD components, DESY
Bandwidth	1,8 kHz ... 100 MHz
Gain	40 dB (non-inverting) 36,5 dB setting for TTF2 Toroids
Adjustable Gain	$\pm 10\%$
Max. Output Power	$\pm 2$ Vpp for 4 single Outputs
Impedance In/Out	50 Ohms
Connectors	SMA
Supply Voltage	$\pm 10$ V ... 15V

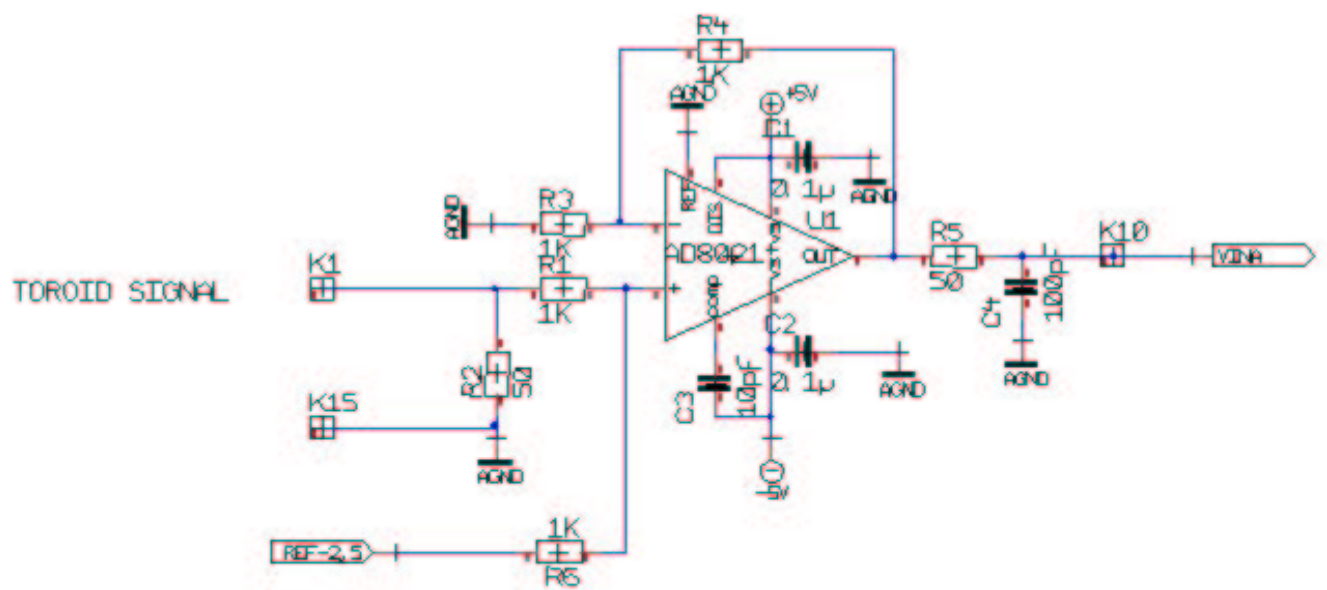


## **Annexe B : Schéma prototype ADC**

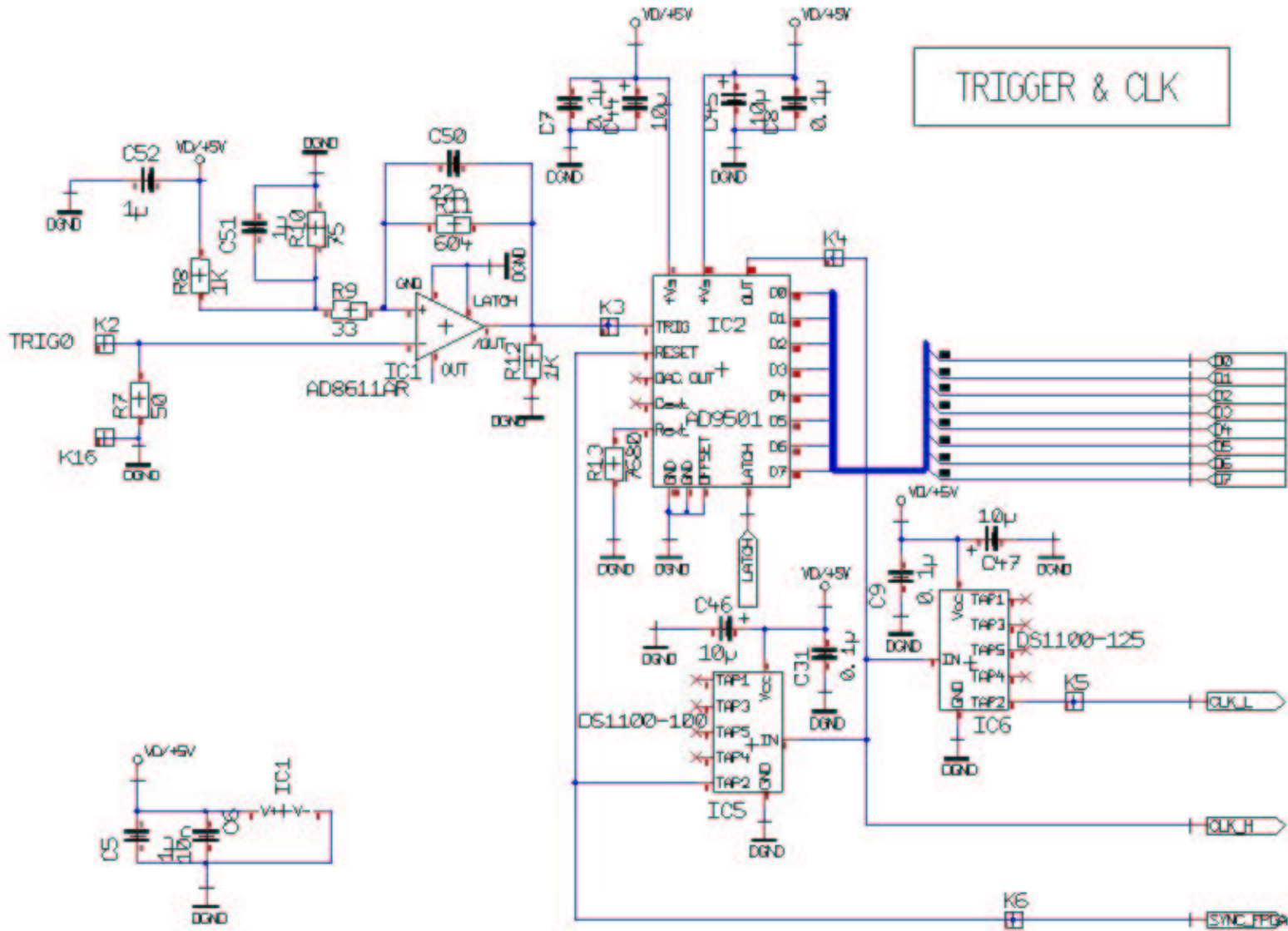


ADC module schematics (1/4)

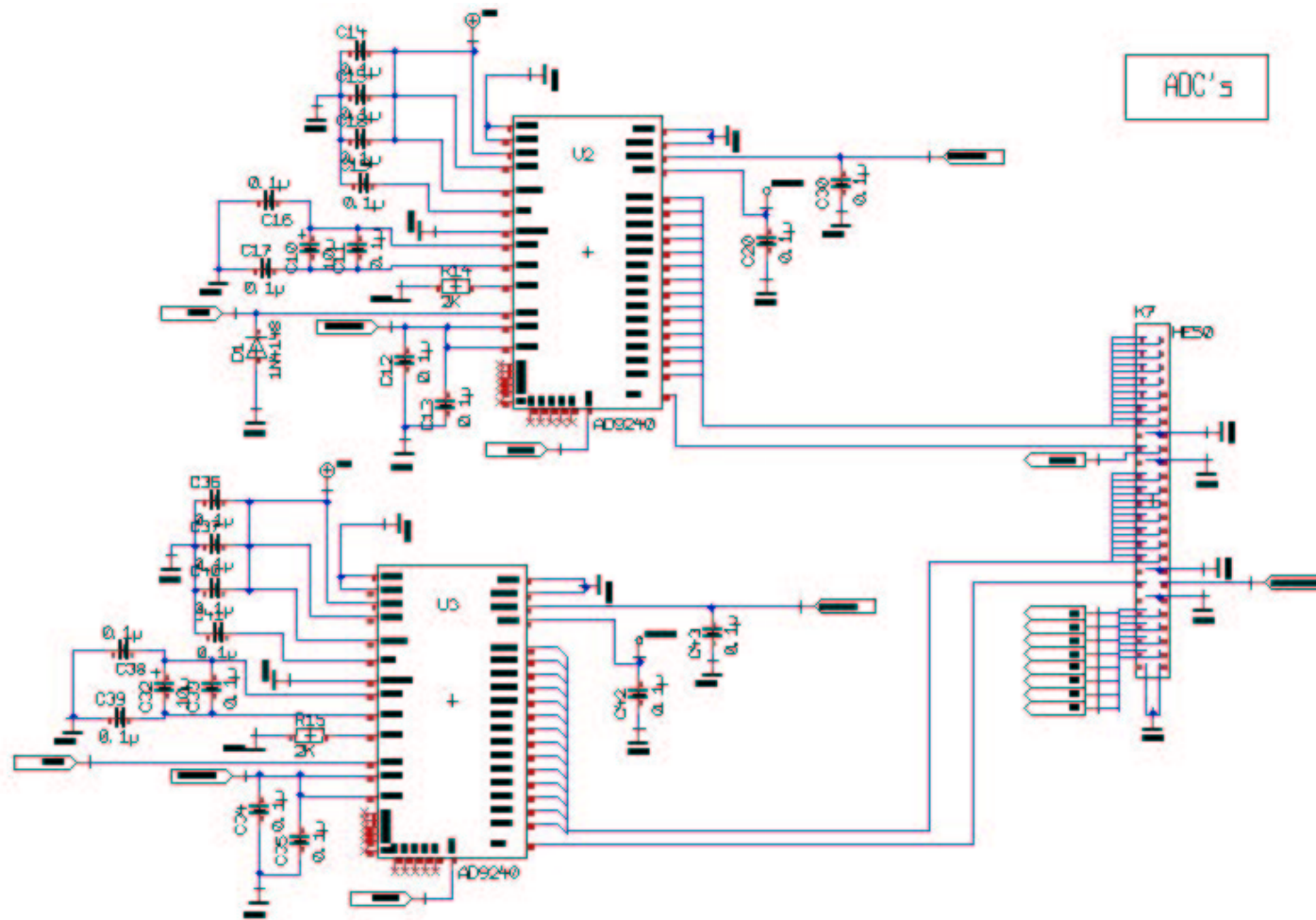
SHIFT LEVEL



**ADC module schematics (2/4)**

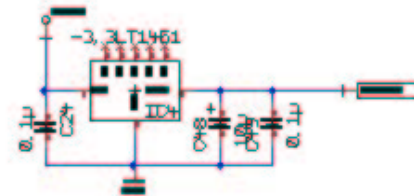
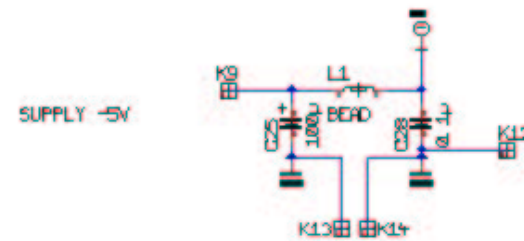
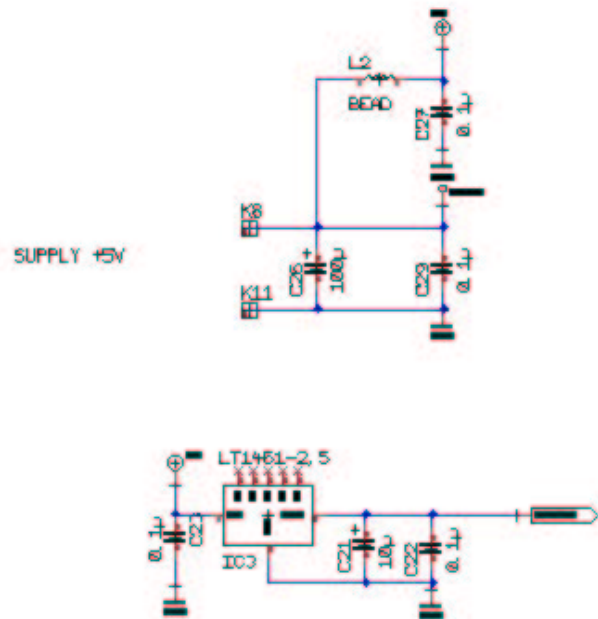


ADC module schematics (3/4)



ADC module schematics (4/4)

SUPPLY & VOLTAGE REF









## **Annexe C : Publication présentée à la Diagnostic particle accelerator conference 2003**



## The Beam Inhibit System For TTF II

D. Nölle, P. Göttlicher, R. Neumann, D. Pugachov, K. Wittenburg, M. Wendt, M. Werner, M. Staack,  
 DESY, Hamburg, Germany  
 M. Desmons, A. Hamdi, M. Jablonka, M. Luong, CEA, DAPNIA, Saclay, France  
 H. Schlarb, SLAC, Stanford, USA  
 e-mail: [dirk.noelle@desy.de](mailto:dirk.noelle@desy.de)

### Abstract

The new generation of light sources based on SASE Free-Electron-Lasers driven by LINACs operate with electron beams with high beam currents and duty cycles. This is especially true for the superconducting machines like TTF 2 and the X-RAY FEL, under construction or planning at DESY. Elaborate fast protections systems are required not only to protect the machine from electron beams hitting and destroying the vacuum chamber, but also to prevent the machine from running at high loss levels, dangerous for components like the FEL undulator.

This paper will give an overview over the different protection systems currently under construction for TTF 2. The very fast systems, based on transmission measurements and distributed loss detection monitors, will be described in detail. This description will include the fast electronics to collect and to transmit the different interlock signals.

### INTRODUCTION

The TESLA Test Facility phase 2 (**TTF 2**) is currently under construction at DESY in Hamburg. This machine has two main objectives [1]:

- a) serve as a test facility for accelerator components for the future TESLA and X-RAY FEL [2].
- b) Operate as a 4<sup>th</sup> Generation Light Source to provide SASE FEL radiation in the range between 100 and 6 nm.

In order to demonstrate the requirements for the large machines for high luminosity in case of the collider and high peak and average brightness in case of the SASE light source, the TTF 2 is capable to run a 800  $\mu$ s long beam pulse at 10 Hz rep. Rate with a 9 MHz bunch frequency. With the design charge of 1 nC this yields currents of 9 mA averaged over the bunch train, or a total average current of 72  $\mu$ A.

Although the average current of TTF II is rather small compared to a typical conventional 3<sup>rd</sup> generation light source, the intrinsic energy of this system is much higher. (Table 1).

Table 1: Comparison of power and energy stored in the beam for LINAC and storage ring driven light sources

	Storage Ring	TTF 2
Average Current	200 mA	72 $\mu$ A
Circumference/Length	200 m	250 m
Energy	2 GeV	1 GeV
Avg. Beam Power	400 MW	72 kW
Energy in the Beam/in the Macropulse	0.13 kJ	7.2 kJ

Therefore, effective protection systems are required to prevent the machine from serious damage due to the operation. These systems have to protect the machine from different damage mechanisms resulting in different damage levels at different time scales, e.g.:

- Beam hitting a part of the vacuum chamber. The beam has to be stopped as fast as possible. In the worst case less than 10 bunches can cause damage. Therefore, the reaction time of the system has to be as short as possible, and is mainly determined by signal transmission times.
- Beam halo or dark current is (partially) lost in the machine. Such losses would result in increased activation of components. In the undulator accumulated losses can degrade the performance of the magnets substantially. Therefore, losses have to be observed down to a level of  $10^{-7}$ .
- A screen or another obstacle has been inserted into the beam pipe. Such an event has to be recognized by a protection system, to restrict the number of bunches within a bunch train to a number given by the sensitivity of the inserted object.

In order to allow safe machine operation, TTF 2 will have monitoring as well as interlock systems. There will be rather slow monitoring system based on Dosimetrie, either using conventional TLD crystals or optical fibres. The fibre-based system will provide information about the radiation profile in the machine with a time resolution of some minutes [3].

The active part of the protection system is taken by the Beam Inhibit System (**BIS**). This system was redesigned based on the experience with a predecessor system at TTF 1 [4].

### THE BEAM INHIBIT SYSTEM

In order to deal with different machine settings a number of operation and beam modes are defined.

Operation modes define the path the electrons have to take, like gun mode (beam stopped by a faraday cup before the first module), undulator and bypass mode.

The beam modes determine, whether short, long or even only single bunch operation is allowed. These modes depend on the machine setting, e.g. if a screen is inserted only single bunch mode is allowed. Furthermore, they depend on the performance of the machine, i.e. the beam mode is switched back from long to short pulse or even to single bunch in the case the losses get too high or transmission gets too bad.

These actions are taken by the BIS system automatically depending on the settings of the machine.

The BIS is a PLC based system to collect all interlock relevant information and to use this information to block risky or dangerous operation modes. The interlock signals taken by the BIS are divided into two categories. Slow signals like screens, power supplies or valves are connected directly to the BIS. The reaction to this information is taken on a shot to shot basis.

Fast signals, i.e. signals that require switching off the beam immediately within a bunch train, are collected by the **Very Fast Beam Inhibit System**. Signals connected to this system are the transmission based interlock system as well as the beam loss monitors. Furthermore, the fast acting valves and some fast RF signals are connected.

### THE VERY FAST BEAM INHIBIT SYSTEM

The very fast beam inhibit system is a kind of distributed logical **OR**. It is built from beam concentrator units (**BIC**) with well-specified I/O channels, concentrating 16 input channels to 1 interlock output. The units can be cascaded and will be used to set up a tree like topology to concentrate the interlock signals (Fig. 1) to one signal acting on the drive laser of the photoinjector. The reaction time of the system of about 3  $\mu$ s is dominated by the signal delay in the cables together with the time of flight of the electron bunches from the gun to the detectors.

In order to deal with different modes and also special situations during the operation, the parameters of this very fast system are controlled by the BIS. It is possible to mask individual interlock channels by software to allow special machine operations, like wire-scans. The interlock status of the system is provided to the control system on a shot to shot basis.

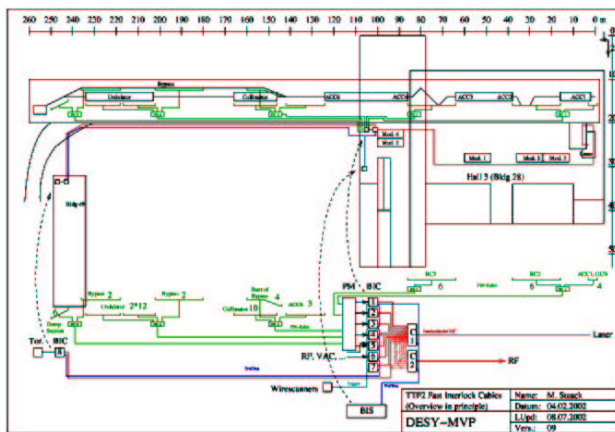


Figure 1: Topology of the BIC System

### The Beam Interlock Concentrator

The BIC modules (Fig. 2) are the central hardware to concentrate all fast interlock signals. 16 input channels

are reduced to 1 output using a logical OR comparison with a processing time of 100 ns. In addition to the fast interlock output, there is a second output line to the BIS. As this system is a central safety system apart from speed redundancy is an essential requirement. Therefore, the hardware operates with 2 independent circuits, one based on discrete logics, the other on digital signal processing implemented in a FPGA. In addition to the basic processing, the FPGA provides additional features like a Profibus interface to the BIS with the possibility to

- Read back the status of the interlock inputs
- To set different kinds of masks via the BIS; so called hard-masks in order to disable interlock channels permanently and soft-masks to disable them for a single bunch train only.

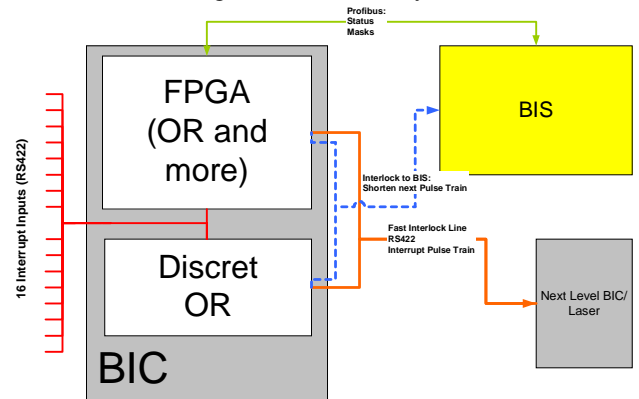


Fig. 2: Block Diagram of a BIC module and its interfaces

In order to avoid problems due to ground loops all I/O channels have galvanic isolation. A differential communication line (RS422) is used for both input and output. A logical high represents OK, so that the interlocks are safe, even in case of cable breaks or shorts.

### The Transmission Based Interlock System

In order to run the machine with high charge and high duty cycle, the transmission from gun to dump has to be close to 100%. The transmission is measured as a charge difference using pairs of charge monitors (toroids), as listed in table 2. Each toroid provides a signal with a bandwidth of about 100 MHz (single bunch resolution) with an accuracy of about 1% [5]. Two pairs have been defined for the two operation modes: undulator and bypass operation. As the path length of an upstream and a downstream toroid can be up to 250 m, the minimum reaction time of this system cannot be faster than about 1 $\mu$ s, plus the signal processing and transfer time of the very fast interlock system to switch off the laser.

The toroid signals are transmitted to the protection system, located in a building at about 1/2 of the length of the machine. The electronics consists of fast ADCs and a digital signal processor per pair, to calculate the charge difference ( $Q_d - Q_u$ ) between downstream and upstream monitor and to compare it on different time scales and with different alarm thresholds:

- Charge Validation:

If  $Q_u < T_Q$ ;  $T_Q \approx 0.05$  nC; Input charge at upstream toroid not valid

- Single Bunch Transmission Threshold:

$$\left( \frac{|Q_u - Q_d|}{Q_u} \right)_i > T_{sgl}; T_{sgl} \cong 25\%$$

The single bunch mode detects single bunch losses, if the threshold  $T_{sgl}$  is exceeded an interlock is released.

- Slice Averaged Transmission Threshold:

$$\sum_{j=i-l}^i \left| \frac{Q_u - Q_d}{Q_u} \right|_j > T_{Slice}; T_{Slice} \cong O(1\%)$$

In this mode the transmission in the bunch train is measured over a slice of bunches ( $i=O(100)$ ) “moving” over the train. As soon as the averaged loss reaches the threshold  $T_{Slice}$  an interlock is released. Due to the averaging the resolution of the system will improve compared to the single bunch resolution. Slice resolutions down to  $10^{-3}$  are expected.

- Pulse Integrated Transmission Threshold:

$$\sum_{i=0}^n |Q_u - Q_d|_i > T_{int}; T_{int} \cong O(10 \text{ nC})$$

The integration mode sums the losses over the whole bunch train. An interlock is released as soon as the threshold is reached. A threshold of 10 nC corresponds to  $1 \cdot 10^{-4}$  of the nominal charge within a bunch train.

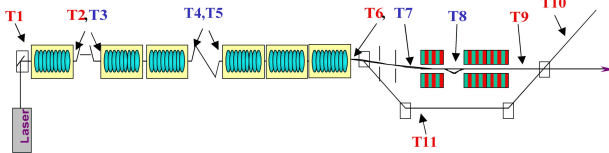


Figure 3: Sketch of TTF 1, showing of charge monitors along the machine. Devices indicated in red serve as input for the protection system.

Table 2: Toroid Pairs for the Protection System

Toroid 1	Toroid 2	Purpose
T <sub>1</sub>	T <sub>9</sub>	Undulator Mode
T <sub>1</sub>	T <sub>11</sub>	Bypass Mode
T <sub>2</sub>	T <sub>10</sub>	Make sure Beam reaches the dump (undulator beam path)
T <sub>2</sub>	T <sub>10</sub>	Make sure Beam reaches the dump (bypass beam path)

### The Beam Loss Monitor System

The beam loss monitor (BLM) system is based on measurements of radiation due to the electromagnetic showers produced by lost electrons. The monitors are located at positions that are rather radiation sensitive. Compared to the global acting transmission based protection system, it has much higher, but localized sensitivity  $O(10^{-7})$  [6]. Thus both systems complement

each other, providing high redundancy and safety for the machine.

Beam losses will be measured by 49 photomultipliers (PMs) equipped with scintillator and 18 secondary emission multipliers (SEMs) equipped with Al cathode. The multipliers will deliver a 20 ns long pulse with an amplitude of up to 100 mA. The reaction time of the system will be about 2 μs depending mainly on the length of the cables. Fig. 3 gives a schematic view of the BLM distribution along the accelerator. Scintillator plates (40cm x 40 cm and 10 x 10 cm) are foreseen to be installed along the linac, long scintillator rods (220cm x 5 cm) will be placed parallel to the undulator vacuum chamber. SEMs will be placed in locations with relaxed radiation sensitivity like bypass, collimator and dump sections.

The readout of the BLMs will be done by a special VME based electronics. These VME-systems will also control and monitor the high voltage (HV) of the multiplier units and are able to generate test pulses. Each loss monitor will release alarms for immediate beam pulse interruption to the BIC in case of exceeding thresholds due to high losses from the beam or dark current (distinguishing between these two loss sources) or if the HV is missing.

For recording of the data the losses are integrated over a time period of 9 bunches and the time of the following bunch is used to digitise and reset the integrator. The loss profile of course is the main tool to optimise the performance of the machine.

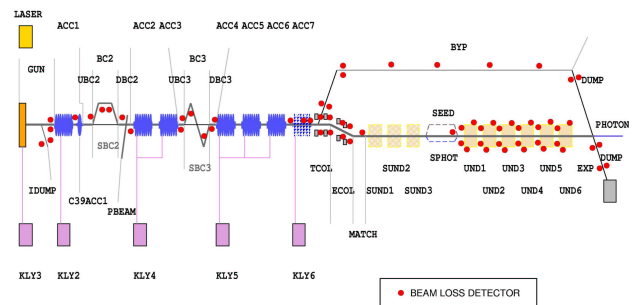


Figure 4: Sketch of TTF 2; the red dots indicate the position of the BLMs along the machine.

### References

- [1] TTF 2 Design Dokument
- [2] TESLA; Technical Design Report, DESY 2002/2003
- [3] Optical Fibre Reference
- [4] TTF 1 Bis
- [5] Torids
- [6] PM as beam loss monitors





**Annexe D : Poster présenté à l'I.T.R.P.  
(International Technology Recommendation Panel)**



# Machine Protection Systems

M. Körfer, D. Noelle, H. Schlarb (DESY)  
 A. Hamdi, M. Jablonka, M. Luong (CEA/DAPNIA)

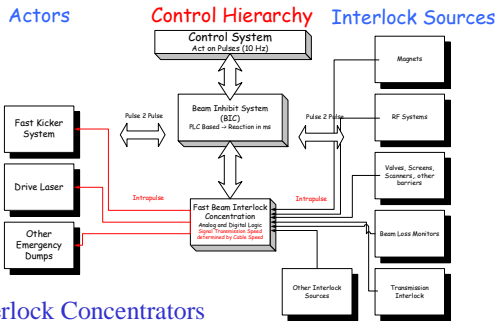
ITRP Visit to DESY, 5<sup>th</sup>/6<sup>th</sup> April 2004



The LC cold option

## Motivation

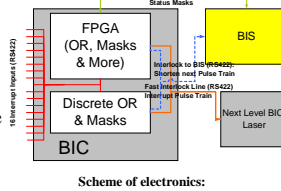
- Warm machine parts in the TESLA linac are critical regions, like :
  - Injector
  - Injection-/Ejection area of the damping rings and the bunch compressor
  - Collimation and Final Focus Systems up to the IP
  - Superconducting wiggler for positron production
  - Dump beam line
- The time structure of the macro pulse train requires a combination of a slow (pulse to pulse) and fast (intra-pulse) beam loss protection systems
- The slow system inhibits the beam injection into such a sub-system in case a critical device fails (magnets, RF, etc.)
- The fast systems fire emergency kickers that can withstand the unavoidable amount of already stored beam power in the subsystem (Inj., DR, main Linac)
- For TTF1 & TTF2 fast systems with high accuracy have been developed to detect losses and to react with high reliability within the bunch train



## Beam Interlock Concentrators

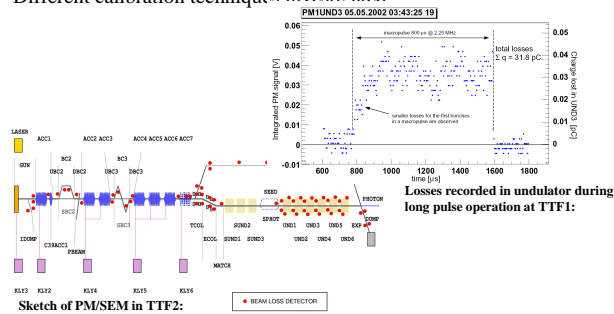
Collects the fast interlocks and distributes to the emergency system procedure

- Fast logical OR Gate
- 16 input/output
- Configurable from shot to shot
- Controlled by the Beam Inhibit System
- Collects all events of the entire machine
- Allows to remove remotely channels
- Full history recording!



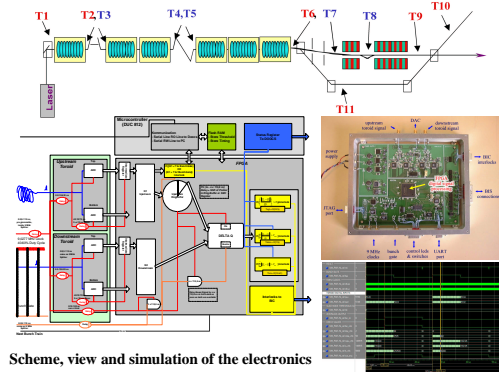
## Beam Loss Monitor System (localized losses)

- Redundant network of
  - Photomultiplier (for ultra sensitive measurements)
  - SEM tubes (withstand occasionally high radiation level)
- Sensitivity to losses  $\Delta Q/Q < 10^{-6}$
- Fast, reaction time determined by cable length
- New electronics with safety checks before pulses
- Different calibration techniques investigated

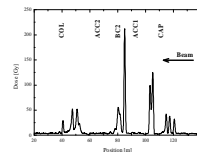
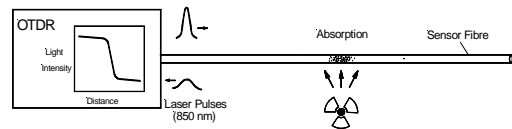


## Transmission based systems

- Beam transport check between two points in the machine
- Complementary to radiation based loss detection
- Uses charge measurement by pairs of Toroids
- Electronics based on FPGA
- Resolution aimed for TTF2:
  - Single Bunch  $< 10^{-2}$
  - Slice of 100 Bunches  $< 10^{-3}$
  - Bunch Train  $< 10^{-4}$



## Distributed dosimeter system using an Optical Time Domain Reflectometer (OTDR).



Few segments of optical fibers along the LINAC allow the localization of losses. These segments could be as long a cryogenic sections.



## **Annexe E : Architecture logique du système**



**Annexe F : Exemple de listing de banc de test VHDL  
pour la simulation des modules  
de traitement numérique**





```
-- E:\DESIGN_FPGA\SECDIFF\MODULES\MODULE_D
-- TESTBENCH FOR THE PARAMETERS TRANSMISSION OF THE TOROID PROTECTION SYSTEM
```

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.NUMERIC_STD.ALL;
USE IEEE.STD_LOGIC_TEXTIO.ALL;
USE IEEE.STD_LOGIC_ARITH.ALL;
USE STD.TEXTIO.ALL;
```

```
ENTITY TB_MD IS
END TB_MD;
```

```
ARCHITECTURE testbench_arch OF TB_MD IS
```

```
-----
-- COMPONENT UNDER TEST
-----
```

```
FILE RESULTS: TEXT OPEN WRITE_MODE IS "results.txt";
  COMPONENT inst_md
    PORT (
      CE_PAR : In std_logic;
      SCLK : In std_logic;
      SDATA : In std_logic;
      CORR : Out std_logic_vector (12 DOWNT0 0);
      TCV : Out std_logic_vector (14 DOWNT0 0);
      TINT : Out std_logic_vector (30 DOWNT0 0);
      TSGL : Out std_logic_vector (7 DOWNT0 0);
      TSLI : Out std_logic_vector (7 DOWNT0 0)
    );
  END COMPONENT;
```

```
-----
-- STIMULATION SIGNALS
-----
```

```
SIGNAL CE_PAR : std_logic;
SIGNAL SCLK : std_logic;
SIGNAL SDATA : std_logic;
SIGNAL CORR : std_logic_vector (12 DOWNT0 0);
SIGNAL TCV : std_logic_vector (14 DOWNT0 0);
SIGNAL TINT : std_logic_vector (30 DOWNT0 0);
SIGNAL TSGL : std_logic_vector (7 DOWNT0 0);
SIGNAL TSLI : std_logic_vector (7 DOWNT0 0);
SIGNAL CORR_chk : std_logic_vector (12 DOWNT0 0) := (others => '0');
```

```
-----
-- CHECK SIGNALS
-----
```

```
SIGNAL TCV_chk : std_logic_vector (14 DOWNT0 0) := (others => '0');
SIGNAL TINT_chk : std_logic_vector (30 DOWNT0 0) := (others => '0');
SIGNAL TSGL_chk : std_logic_vector (7 DOWNT0 0) := (others => '0');
SIGNAL TSLI_chk : std_logic_vector (7 DOWNT0 0) := (others => '0');
```

```

BEGIN
  UUT : inst_md  -- MAPPING OF STIMULATION SIGNALS
                -- TO THE I/O OF THE COMPONENT INSTANCIATED

  PORT MAP (
    CE_PAR => CE_PAR,
    SCLK => SCLK,
    SDATA => SDATA,
    CORR => CORR,
    TCV => TCV,
    TINT => TINT,
    TSGL => TSGL,
    TSLI => TSLI
  );

  PROCESS -- Process for SCLK

    VARIABLE TX_OUT : LINE;
    VARIABLE TX_ERROR : INTEGER := 0;
    VARIABLE j : integer ;

    BEGIN
    -----
    -- CHECK VALUES INITIALISATION
    -----
      CORR_chk <= conv_std_logic_vector(1024,13);
      TINT_chk <= conv_std_logic_vector(500,31);
      TSLI_chk <= conv_std_logic_vector(1,8);
      TSGL_chk <= conv_std_logic_vector(25,8);
      TCV_chk <= conv_std_logic_vector(10000,15);

      -----
      CE_PAR <= transport '0';           -- INPUTS INITIALISATION
      SDATA <= transport '0';
      SCLK <= '0';
      -----
      WAIT FOR 1 us;
      CE_PAR <= transport '1';         -- MODULE SELECTION
      -----
      -- WRITE LOOP FOR CHARGE VALIDATION THRESHOLD
      -----
      j := 14;                          -- INIT j WITH NB OF BIT_VECTOR
      For i in 1 to 15 loop
        SCLK <= '0';
        SDATA <= TCV_chk(j);           -- WRITE BIT j
        j := j-1;                       -- DEC j
        WAIT FOR 1 us;
        SCLK <= '1';                   -- SHIFT BIT VALUE
        WAIT FOR 1.3 us;
      end loop;
    
```

```

-----
-- WRITE LOOP FOR SINGLE MODE THRESHOLD
-----
j := 7;
For i in 1 to 8 loop
SCLK <= '0';
SDATA <= TSGL_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
end loop;
-----
-- WRITE LOOP FOR SLICE MODE THRESHOLD
-----
j := 7;
For i in 1 to 8 loop
SCLK <= '0';
SDATA <= TSLI_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
end loop;

-----
-- WRITE LOOP FOR INTEGRATION MODE THRESHOLD
-----
j := 30;
For i in 1 to 31 loop
SCLK <= '0';
SDATA <= TINT_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
end loop;

-----
-- WRITE LOOP FOR CORRECTION FACTOR
-----
j := 12;
For i in 1 to 13 loop
SCLK <= '0';
SDATA <= CORR_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
if j = 0 then
SCLK <= '0';
SDATA <= CORR_chk(j);
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
CE_PAR <= '0';
end if;
end loop;

```

```

-----
SCLK <= '0';
WAIT FOR 7 us;
SCLK <= '1';
WAIT FOR 1.3 us;
SCLK <= '0';
-----

WAIT FOR 10 us; -- Time=9356 ns
-----
CORR_chk <= conv_std_logic_vector(1126,13);
TINT_chk <= conv_std_logic_vector(60,31);
TSLI_chk <= conv_std_logic_vector(5,8);
TSGL_chk <= conv_std_logic_vector(15,8);
TCV_chk <= conv_std_logic_vector(12500,15);

-----
CE_PAR <= transport '0';
SDATA <= transport '0';
SCLK <= '0';
-----
WAIT FOR 1 us; -- Time=2 us
CE_PAR <= transport '1';
-----
j := 14;
For i in 1 to 15 loop
SCLK <= '0';
SDATA <= TCV_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
end loop;
-----
j := 7;
For i in 1 to 8 loop
SCLK <= '0';
SDATA <= TSGL_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
end loop;
-----
j := 7;
For i in 1 to 8 loop
SCLK <= '0';
SDATA <= TSLI_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
end loop;

```

-- 2nd SCENARIO

```

-----
j := 30;
For i in 1 to 31 loop
SCLK <= '0';
SDATA <= TINT_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
end loop;

-----

j := 12;
For i in 1 to 13 loop
SCLK <= '0';
SDATA <= CORR_chk(j);
j := j-1;
WAIT FOR 1 us;
SCLK <= '1';
WAIT FOR 1.3 us;
if j = 0 then
  SCLK <= '0';
  SDATA <= CORR_chk(j);
  WAIT FOR 1 us;
  SCLK <= '1';
  WAIT FOR 1.3 us;
  CE_PAR <= '0';
end if;
end loop;

-----

SCLK <= '0';
WAIT FOR 7 us;
SCLK <= '1';
WAIT FOR 1.3 us;
SCLK <= '0';

-----

WAIT FOR 3 us;

-----

SCLK <= '1';
WAIT FOR 1.3 us;
SCLK <= '0';

-----

WAIT FOR 10 us;

IF (TX_ERROR = 0) THEN
  STD.TEXTIO.write(TX_OUT,string("No errors or warnings"));
  STD.TEXTIO.writeline(results, TX_OUT);
  ASSERT (FALSE) REPORT
    "Simulation successful (not a failure). No problems detected. "
    SEVERITY FAILURE;
ELSE
  STD.TEXTIO.write(TX_OUT, TX_ERROR);
  STD.TEXTIO.write(TX_OUT, string'(
    " errors found in simulation"));
  STD.TEXTIO.writeline(results, TX_OUT);
  ASSERT (FALSE) REPORT
    "Errors found during simulation"
    SEVERITY FAILURE;
END IF;
END PROCESS;

```

```
END testbench_arch;

CONFIGURATION inst_md_cfg OF TB_MD IS
    FOR testbench_arch
    END FOR;
END inst_md_cfg;
```

**Annexe G : Listing du programme en langage C du  
microcontrôleur ADUC812 pour l'étalonnage de la  
carte TPS**





```

#include <stdio.h>
#include <aduc812.h>
#include <math.h>
#include <intrins.h>

//-----

#define TRUE          0x01          // Value representing TRUE
#define FALSE        0x00          // Value representing FALSE
#define ON           0x01          // Value representing ON
#define OFF          0x00          // Value representing OFF
#define HIGH         0x01          // Value representing HIGH
#define LOW          0x00          // Value representing LOW
#define YES          0x01          // Value representing YES
#define NO           0x00          // Value representing NO
#define DELAY_WRITE  1000         // Value for delay write time
#define period       100          // value of I2c clock period

//-----

#define DS1020_1     0x81          // Address select of Upstream PDG
#define DS1020_2     0x82          // Address select of Downstream PDG
#define ACQVAL       P3_5         // Calibration command
#define MRESET       P3_4         // Master Reset of the system
#define nb_val       20           // Number of values to acquire/Toroid/delay
#define delay_Q      15           // Delay initialisation
#define delay_start  35           // value for delay start

sbit      P3_5 = 0xB5;
sbit      P3_4 = 0xB4;

//-----

void write_byte (unsigned long data_out,int nb);
void i2c_init (void);                // Initialize I2C
bit i2c_write (unsigned long input_data,int nb); // Writes data over the I2C bus
void initialize_transmission(void); // Initializes MCU, except I2C
void delay_time (unsigned int time_end); // pause execution for pre-determined time

//-----

unsigned int xdata timeout_counter; // Variable used for delay routines
unsigned long xdata TINT,TSGL,TSLI,TCV,CORR; // Variable used for transmission
// Parameters

//-----

bit inter_spi,inter;
int xdata table[nb_val];
unsigned char adresse, adresse1, adresse2;
unsigned char res;
unsigned int delay, adc;
int retard;
int i,j,e;
int valeur_finale;
float val_moy,val_ecart,erreur;
float xdata max_QU,ecart_QU;
unsigned char QU_delay_optimun;
float xdata max_QD,ecart_QD;

```

```

unsigned char QD_delay_optimun;
float rapport;
unsigned long idata toto;

//***** extern interruption 1 *****//

void IE1_interrupt () interrupt 2
{
inter = 1;
}

//*****interruption SPI *****//

void SPI_interrupt () interrupt 7
{
inter_spi=1;
}

//***** init UART *****//

void initialize_UART (void)
{
    SCON = 0x50;                // 0101,0000 (Mode 1 and RxD enable)

    TMOD = 0x20;                // Timer #1 in autoreload 8 bit mode
    TCON = 0x40;                // Set Timer #1 to run mode
    TH1 = 0xFD;                 // Baud rate is determined by
                                // Timer #1 overflow rate
                                // Baud Rate = (Fcpu / 384) / (256 - TH1)
                                // Fcpu = 11.0592 MHz
                                // TH1 = 253
    TR1 = 1;                    // Turn on Timer 1
    TI = 1;                     // Set UART to send first cha
}

//-----

int acquisition(unsigned char adresse)
{
int mesure;
unsigned char *p;

    adresse1 = adresse;        // adresse1=> 8bits MSB
    adresse2 = adresse+1;      // adresse2=> 8bits LSB

    EX1 = 1;                    // externe interruption
    IT1 =1;                     // on the falling edge transition
    inter =0;

    //***** ask interruption

    ACQVAL = 0;
    MRESET = 1;
    MRESET = 0;
    MRESET = 1;
    ACQVAL = 1;

    while(inter!=1){}

```

```

    ACQVAL = 0;
    inter=0;

    p = &mesure;

    //*****LSB Acquisition

    P2 = adresse1;
    RD =0;
    *p = P0;
    RD = 1;
    p=p+1;

    //***** MSB Acquisition

    P2 = adresse2;
    RD = 0;
    *p = P0 ;
    RD = 1;
    p=p+1;

    return(mesure);
}

//-----

float average (void)
{
    float resultat;
    int n, i;

    for(i=0;i<nb_val;i++)
    {
        valeur_finale = acquisition(adresse);
        table[i] = valeur_finale;
    };

    resultat=0;
    for(n=0;n<nb_val;n++)
    {
        resultat=table[n]+resultat;
    };

    return (resultat);
}

//-----

float standard_deviation (float moyenne)
{
    float ecart;
    int i;
    ecart=0;

    for (i=0;i<nb_val;i++)
    {
        ecart = ecart + pow(table[i]*0.00025 - moyenne,2);
    }
}

```

```

    ecart = sqrt(ecart/nb_val);

    return(ecart);
}

//-----

void DS1020_prog(unsigned char num_DS1020,unsigned char res)
{
    SPICON = 0x33;                //init SPI
    IE2=0x01;                    // SPI interruption
    inter_spi=0;

    P2 = num_DS1020;
    SPIDAT = res ;               //transmitting value
    while(inter_spi!=1){}
    inter_spi=0;

    for (delay = 0; delay < 100; delay++) {} ;           //timer 50us
    P2 = 0xFF;

    SPICON = 0X00;                //switch on external interrupt
    SPE = 0;
    IE2=0x00;
}

//-----

void main(void)
{

    EA = 1;                       // switch on all interrupts
    initialize_UART();            // init UART

    //////////////////////////////////////
    ////////////////////////////////////// MEASURE QU //////////////////////////////////////
    //////////////////////////////////////

    res = delay_start;
    max_QU = -32768;
    QU_delay_optimum = 0;
    adresse = 0x84;

    printf("\fcalibration QU\n");

    for(e=0;e<delay_Q;e++)
    {
        printf(".");

        DS1020_prog(DS1020_1,res);           // load the first delay

        val_moy = average ();                // calculation of the middle value

        if(val_moy > max_QU)
        {
            max_QU = val_moy;
            QU_delay_optimum = res;
            ecart_QU = standard_deviation (val_moy);
        }
    }
}

```

```
res = res+1;}
```

```
////////////////////////////////////
//////////////////////////////////// MEASURE QD ///////////////////////////////////
////////////////////////////////////
```

```
res = delay_start;
max_QD = -32768;
QD_delay_optimun = 0;
adresse = 0x86;
```

```
printf("\fcalibration QD\n");
```

```
for(e=0;e<delay_Q;e++)
{
printf(".");
```

```
DS1020_prog(DS1020_2,res); //load the second load
```

```
val_moy = average ();
```

```
if(val_moy > max_QD)
{
max_QD=val_moy;
QD_delay_optimun=res;
ecart_QD= standard_deviation (val_moy);
}
```

```
res=res+1;
}
```

```
printf("\fcalibration QU\n");
printf("\nQU max (averaged) =%f V",max_QU);
printf("\nstandard deviation=%f V",ecart_QU);
printf("\nQU delay =%.1f ns\n",(float)((QU_delay_optimun*0.5)+10));
```

```
printf("\ncalibration QD\n");
printf("\nvaleur moyenne_QD_optimun =%f V",max_QD);
printf("\nstandard deviation=%f V",ecart_QD);
printf("\nQD delay =%.1f ns\n\n",(float)((QD_delay_optimun*0.5)+10));
```

```
rapport=max_QU/max_QD;
printf("\nratio QU/QD=%5f\n",rapport);
```

```
DS1020_prog(DS1020_1,QU_delay_optimun);
DS1020_prog(DS1020_2,QD_delay_optimun);
```

```
initialize_transmission();
i2c_init();
```

```
printf("\n\ntransmitting parameters\n");
```

```
CORR=(unsigned long)(rapport*1024);
TCV=1000;
TSGL=0x1E;
TSLI=0x00;
TINT=0;
```

```

        P2= 0x80;
        write_byte(TCV,15);
        write_byte(TSGL,8);
        write_byte(TSSL,8);
        write_byte(TINT,31);
        write_byte(CORR,13);

        printf("\nmode single bunch actif" );

        P2= 0x88;

        MRESET = 1;
        MRESET = 0;
        MRESET = 1;

        while(1){ //fin while
} // end main

//-----

void write_byte (unsigned long data_out,int nb)
{
    i2c_write(data_out,nb);
    delay_time(DELAY_WRITE);           // Delay a period of time to write
}

void i2c_init (void)
{
    I2CM = HIGH;           // Set master mode
    MDO = HIGH;           // Set the data bit to HIGH
    MDE = FALSE;         // Not using the pin yet
    MCO = HIGH;           // Set the clock bit to HIGH
}

bit i2c_write (unsigned long output_data,int nb)
{
    int index;
    unsigned long output;

    index = 32-nb;
    output = _lrol_(output_data,index);

    MDE = TRUE;           // Master Mode Data Output Enable: ON

    for(index = 0; index < nb; index++)           // Send 8 bits out the port
    {
        if((output & 0x80000000)==0){MDO=0;}
        else{MDO=1;}
        MCO = HIGH;           // Clock the data into the EEPROM
        delay_time(period);   // Repeat for delay
        MCO = LOW;
        delay_time(period);   // Repeat for delay
        output = _lrol_(output,1);
        printf(".");
    }
    return FALSE;
}

```

```
}

void initialize_transmission (void)
{
// Initialize the serial port (9600, 8, N, 1) [see page 32 of ADUC812 data sheet]

    PCON &= 0x7F;           // Clear bit 7 of the PCON register (SMOD1 = 0)

    TMOD |= 0x22;          // Timer #0 and #1 in autoreload 8 bit mode
    TCON = 0x40;           // Set Timer #1 to run mode
    TR0 = 1;               // Turn on Timer 0
    TI = 1;                // Set UART to send first char

    IE = 0x82;             // Turn on Timer #0 interrupts (bit 1 and 7)
}

void delay_time (unsigned int time_end)
{
    timeout_counter = 0x0000;
    while (timeout_counter < time_end);
}

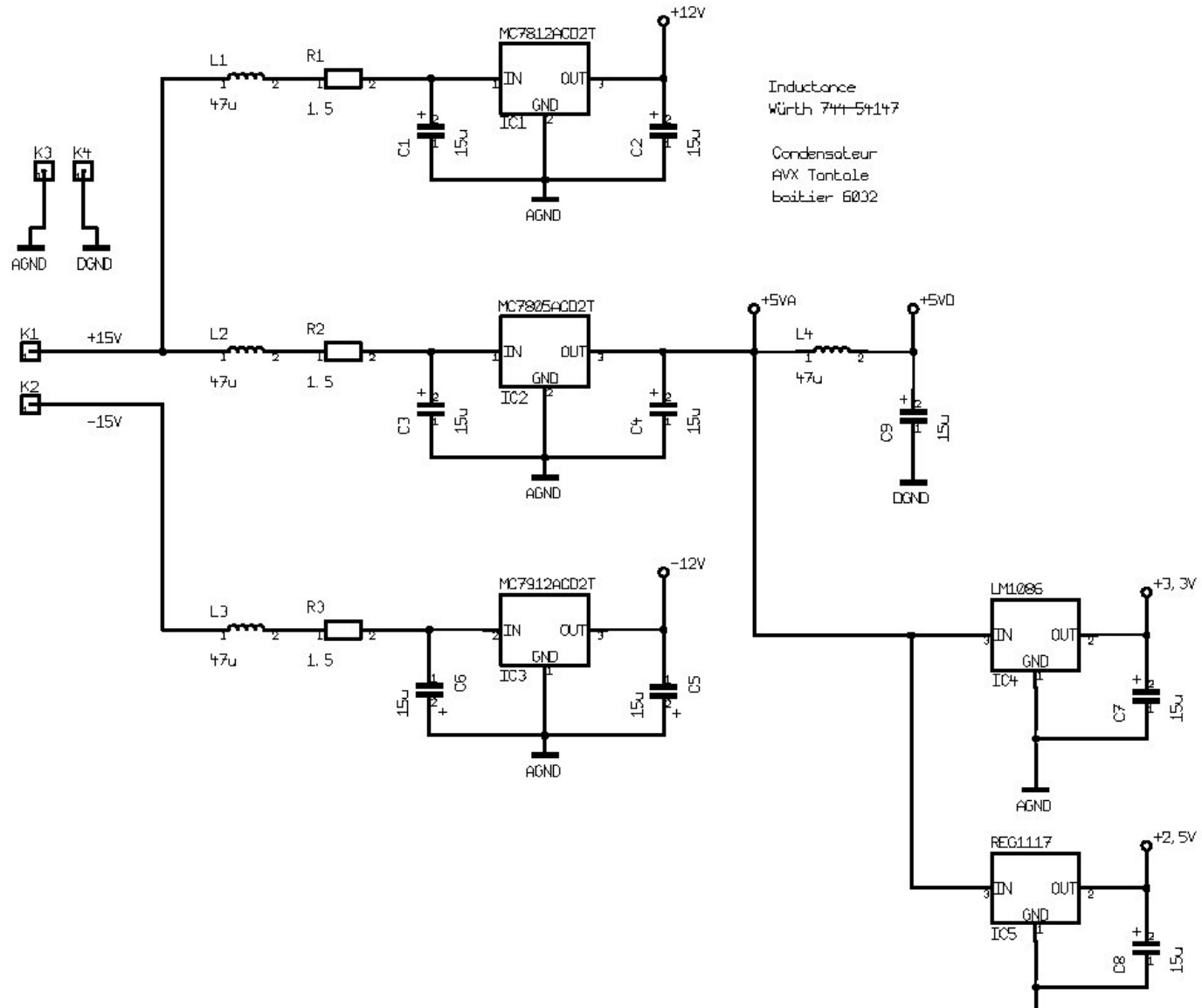
void timer_0_isr (void) interrupt 1
{
    timeout_counter++;      // Increment time-delay counter
    if (timeout_counter > 64000) timeout_counter = 64000;
}
}
```

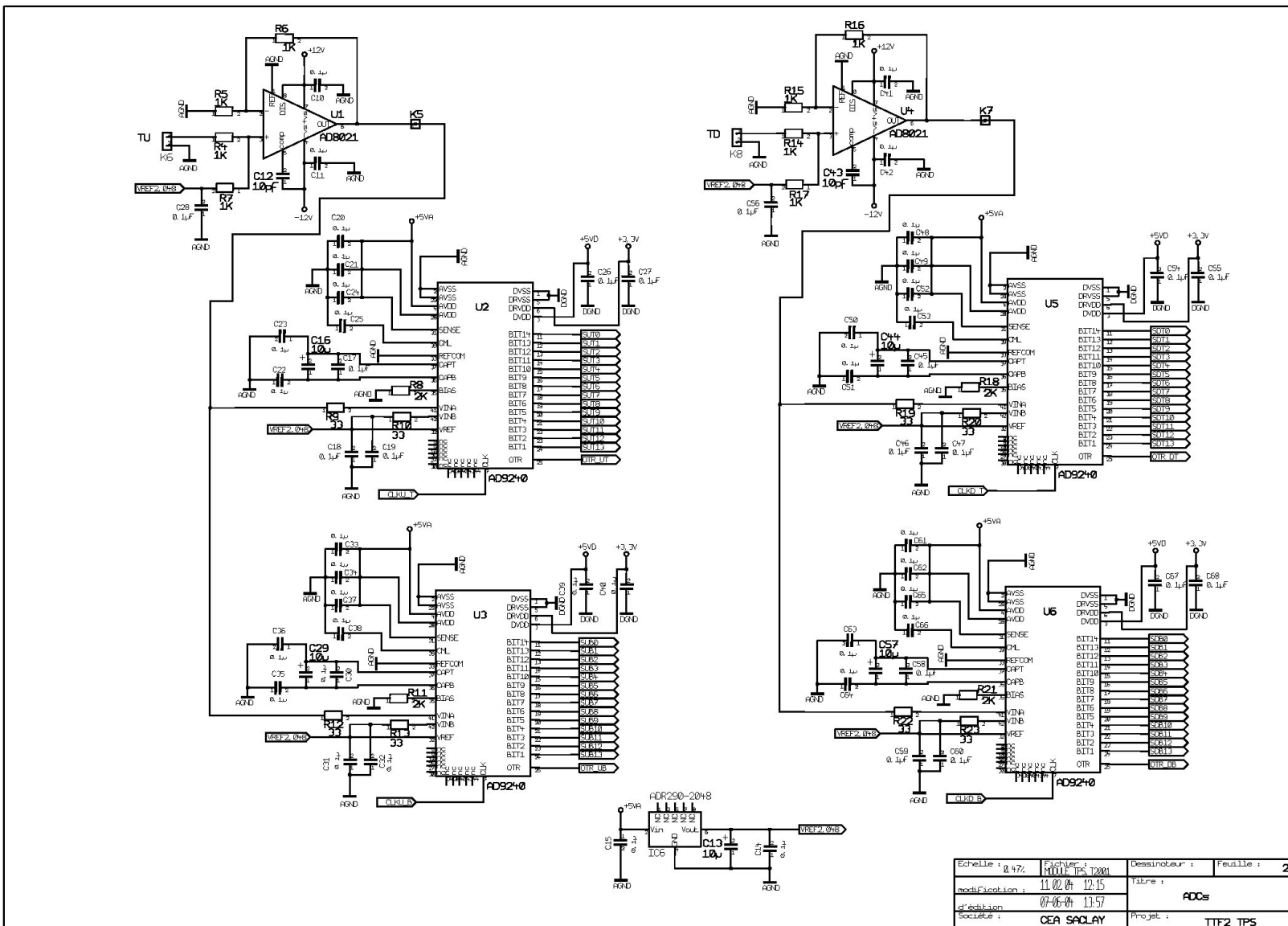




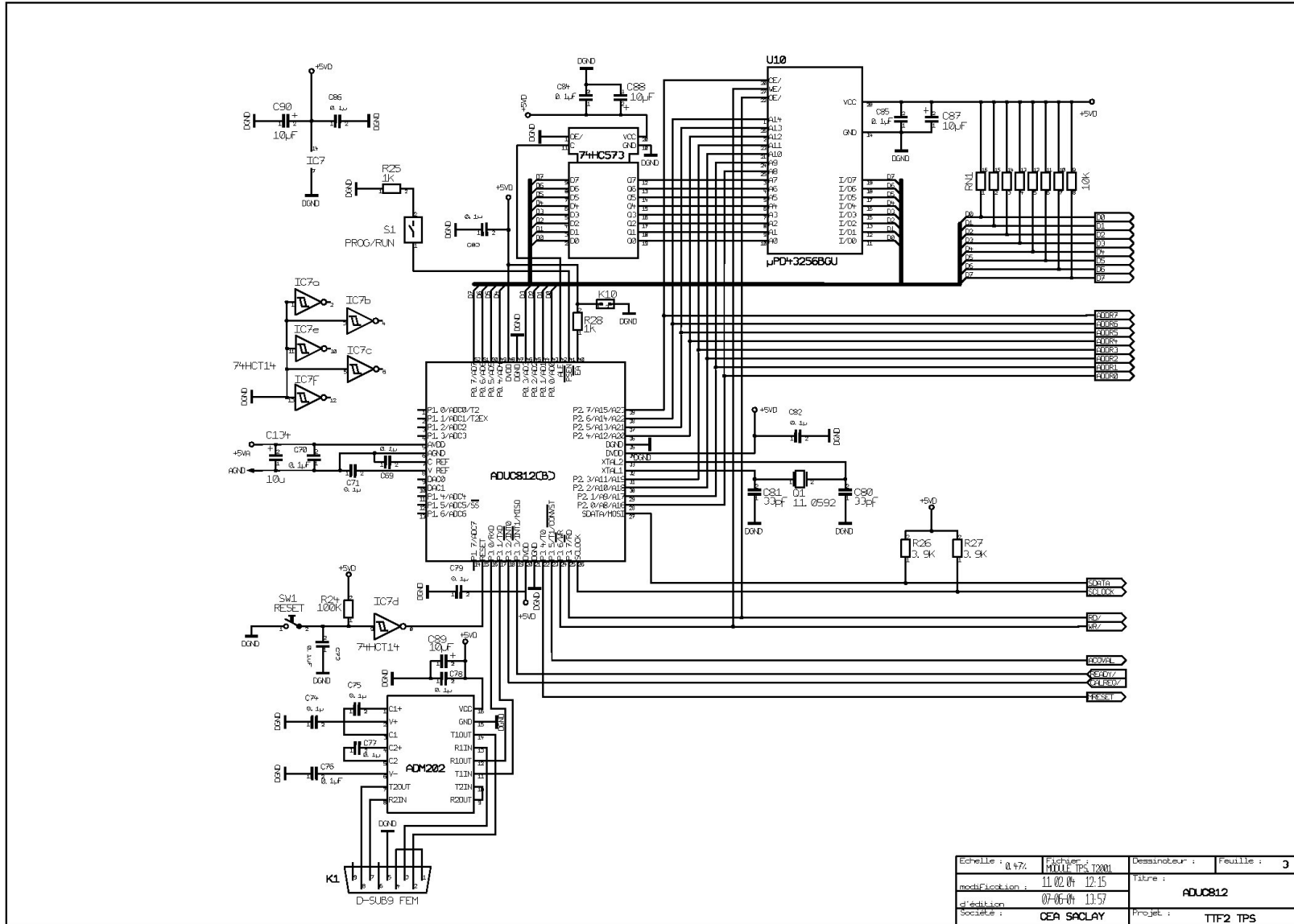
## **Annexe H : Schéma de la carte TPS**



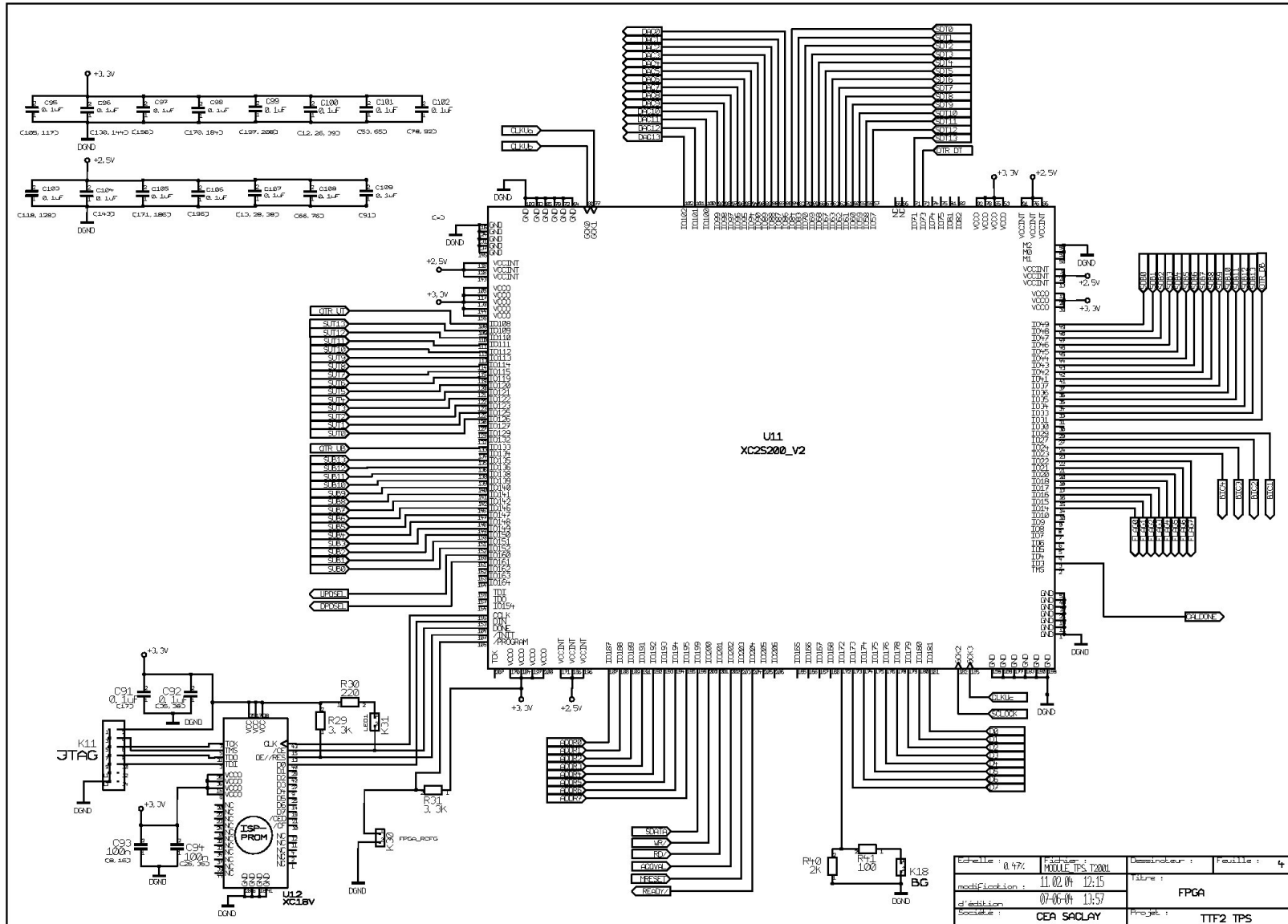




Echelle : 1/47	Échelle : 1/1011 TPS 12001	Dessinateur :	Feuille : 2
Modification : 11.02.04	12.15	Titre : ADCs	
date d'édition : 07-06-04	13.57	Projet : TTF2 TPS	
Société : CEA SACLAY			



Echelle : 1/47	Fichier : M11.F TPS 1201	Destinateur :	Feuille : 3
modification : 11.02.04 12:15		Titre : ADUC812	
date : 07-06-04 13:57		Projet : TIF2 TPS	
Societe : CEA SACLAY			



Echelle : 0.4%	Fichier : M01LE_TPS_T2001	Destinateur :	Feuille : 4
modification : 11.02.04 12:15		Titre :	FPGA
date edition : 07-06-04 13:57		Projet :	TTF2 TPS
Societe : CEA SACLAY			







## **Annexe I : Synoptique de la carte TPS**



# Bibliographie

---

- [1] The Tesla Linear Collider and X-ray FEL K. Floettmann, Proceedings of LINAC2002, Gyeongju, Korea
- [2] Tesla – Technical Design Report 2001 – DESY (Deutsches Elektronen-Synchrotron)
- [3] Electronique International Hebdo – N° 530 – 09/2003
- [4] The DSP handbook – A. Bateman, I. Paterson-Stephens
- [5] Spartan-II 2.5V FPGA Family: Complete Data Sheet – Xilinx Product Specification
- [6] Development System Reference Guide – Xilinx ISE
- [7] VHDL : méthodologie de design et techniques avancées – Thierry Schneider
- [8] Configuration Quick Start Guidelines – Stephanie Tapp
- [9] VHDL langage, modélisation, synthèse – R Airiau, J-M Bergé, V. Olive, J. Rouillard