

Utilisation d'un **FPGA** dans
l'expérience *MightyLaser*

Journées vlsi 2010 LAL Orsay

24 juin | 11:00 | 11:20
jehanno@lal.in2p3.fr



PLAN

- **Introduction (3)**

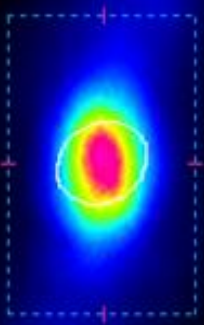
- Le système DAQ (4)

- Exemples d'implémentation dans le FPGA (10)

- Identification

- Filtres numériques

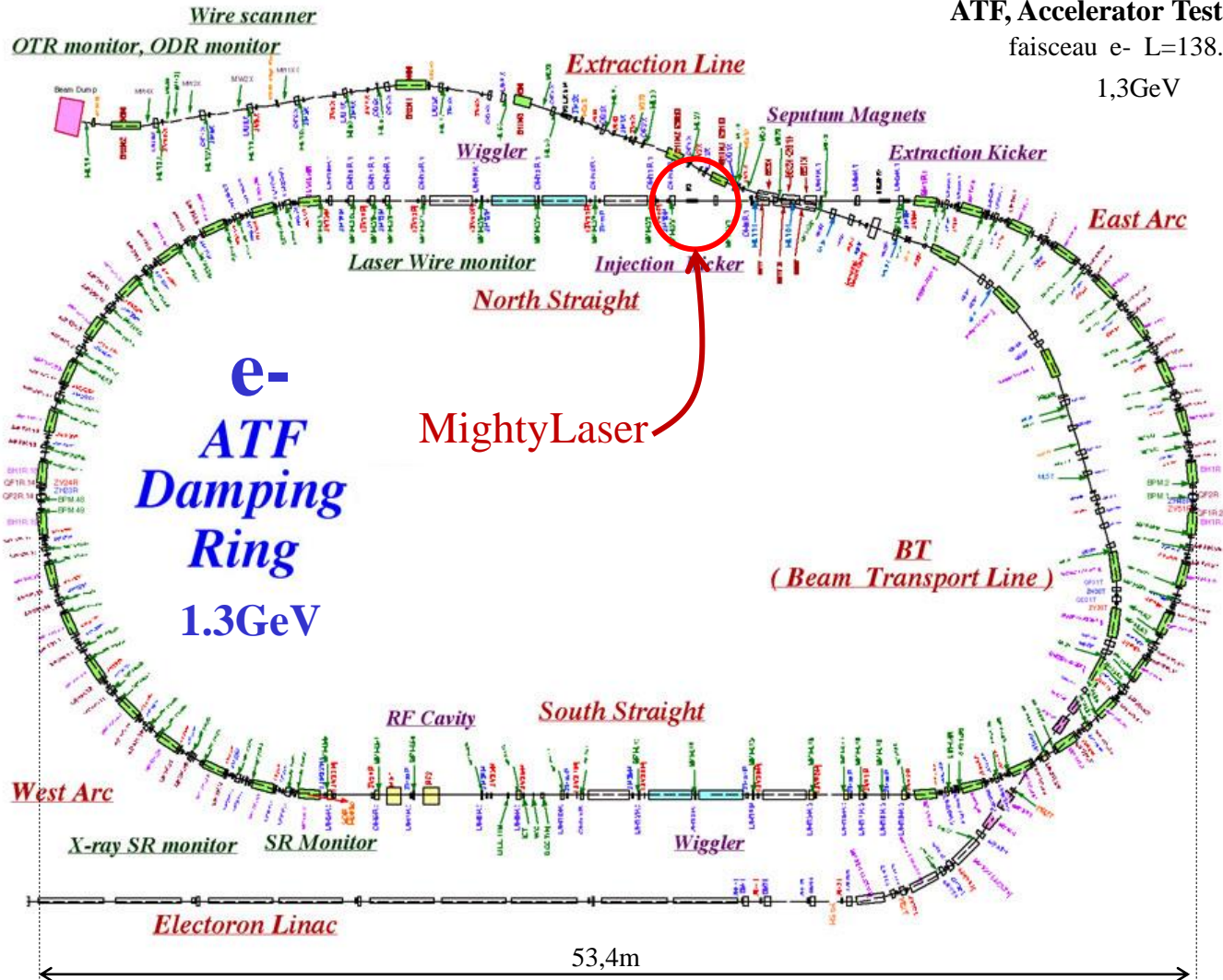
- La fonction Pound-Drever-Hall numérique



ANR-08-BLAN-0280-01

Amplification dans une **cavité Fabry-Perot** d'un **laser** à fibre picoseconde de très forte puissance moyenne • Application à la production de **rayons gamma** par **interaction compton**

KEK, Tsukuba, Japan
ATF, Accelerator Test Facility
 faisceau e- L=138.6m
 1,3GeV

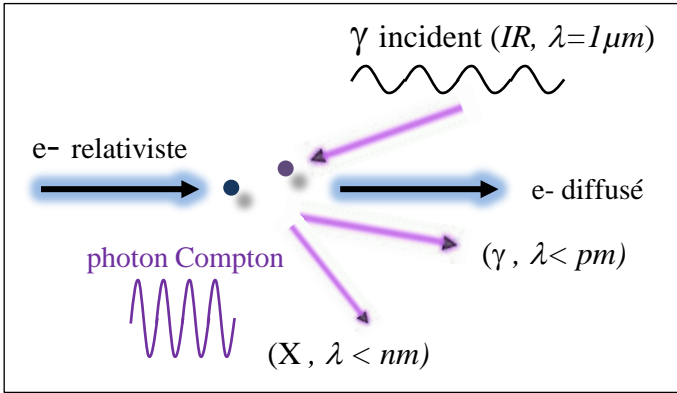


$$\frac{2}{17}$$

$$\lambda = \frac{c}{\nu}$$

$$E = h \cdot \nu$$

$$E = h \cdot \frac{c}{\lambda}$$

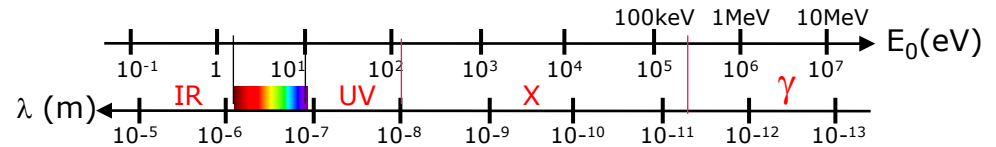


diffusion Compton inverse

interaction matière - lumière

diffusion d'un électron (e^-) sur un photon γ

les γ **absorbent** une partie de l'énergie des e^-



production de γ, X monochromatiques
sélection angulaire

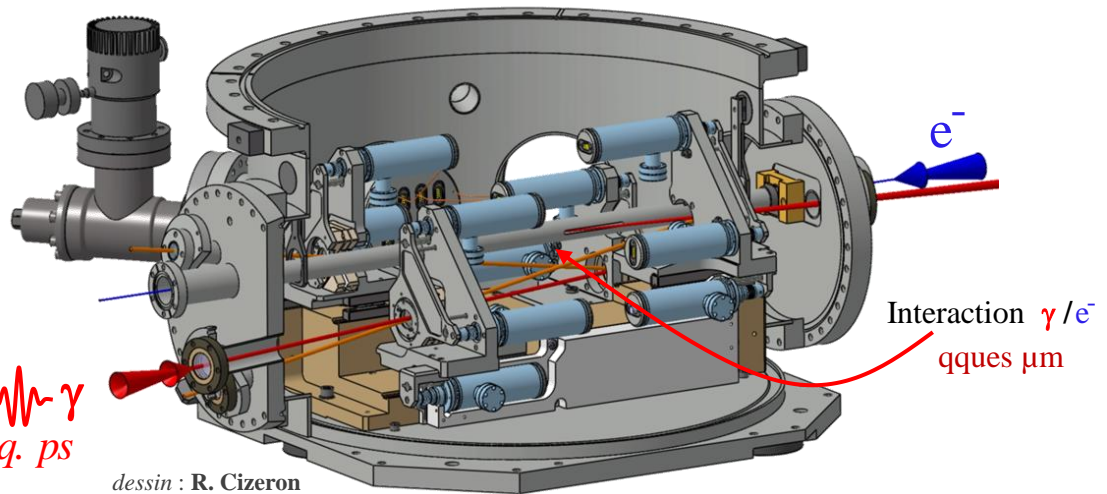
applications

- médecine (imagerie, radiothérapie)
- analyses (cristallographie)
- analyse non-destructive (oeuvres d'art)

cavité 4 miroirs Fabry-Perot
 $\varnothing=1m$ $V=1/3m^3$ $P\sim 10^{-9}mbar$

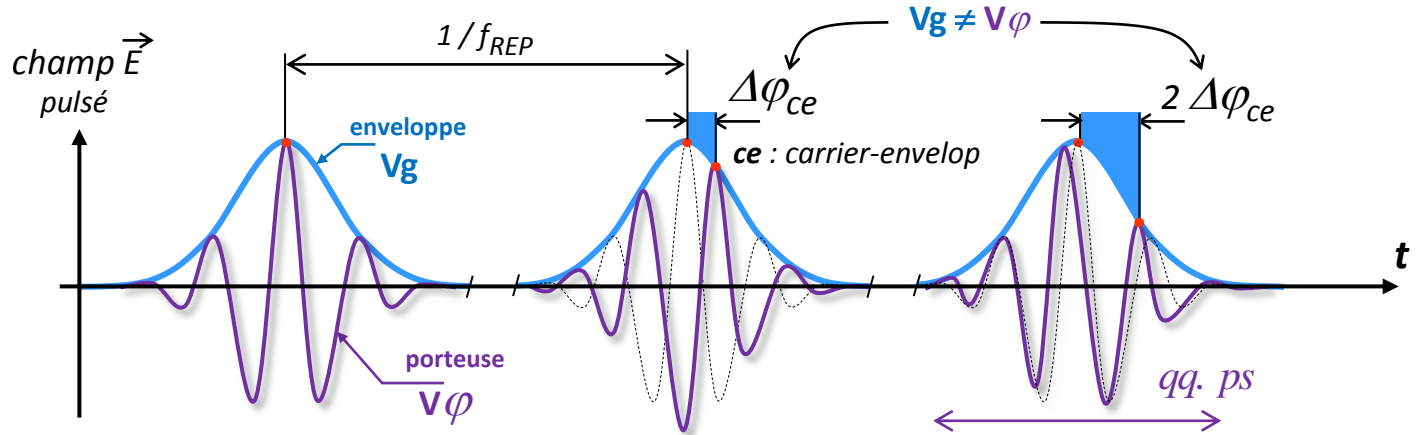
les photons sont issus d'un **laser pulsé**
($f_{REP}=178.5MHz$)

$qq. ps$



dessin : R. Cizeron

Amplification par **sommation** du champ laser **incident** avec le champ **intra-cavité** : **résonance** de la cavité
Phénomène **instable** (*bruits*) : la DAQ doit réaliser des **asservissements** de **longueurs**

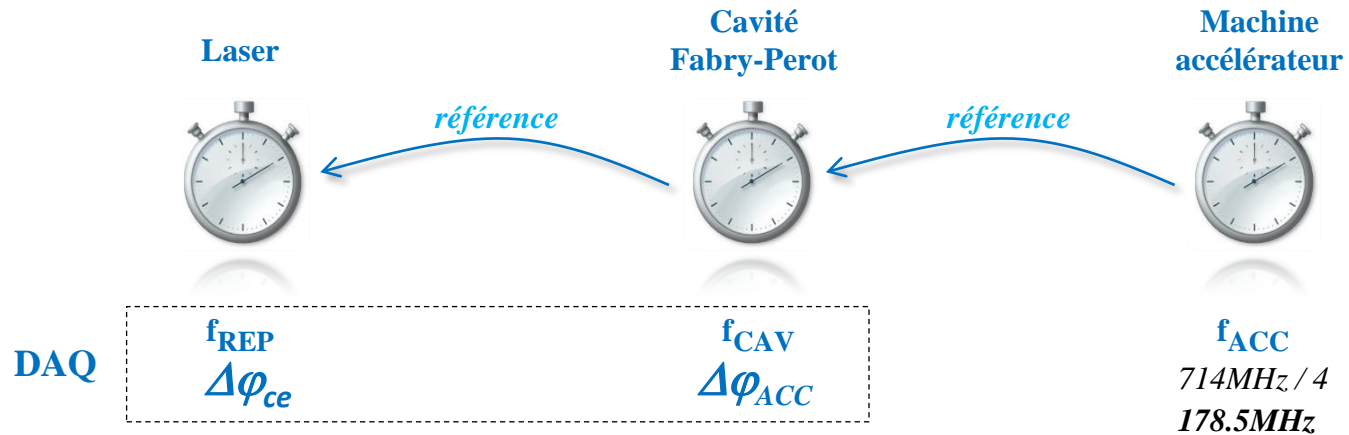


Système : Laser / Cavité Fabry-Perot

Amplification optique	
$f_{REP} = f_{CAV}$	$\Delta\phi_{ce} = 0$

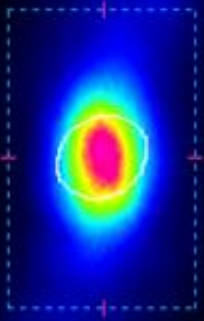
Système : Cavité Fabry-Perot / Accélérateur

Interaction Compton	
$f_{CAV} = f_{ACC}$	$\Delta\phi_{ACC} = 0$



PLAN

- Introduction (3)
- **Le système DAQ (4)**
- Exemples d'implémentation dans le FPGA (10)
 - Identification
 - Filtres numériques
 - La fonction Pound-Drever-Hall numérique



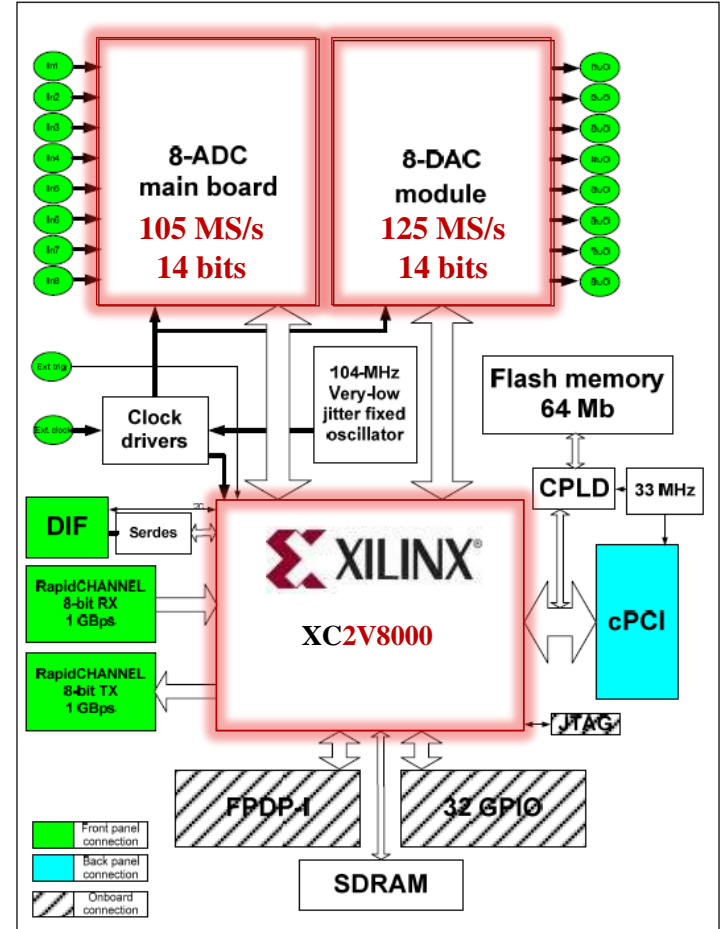
• Carte VHS ADAC V2 (LYRTECH)
dans un châssis cPCI



photo : LAL

- Pas de processeur
calcul en virgule fixe
- bruit de sortie BF non prévu (conception)
impose sorties différentielles

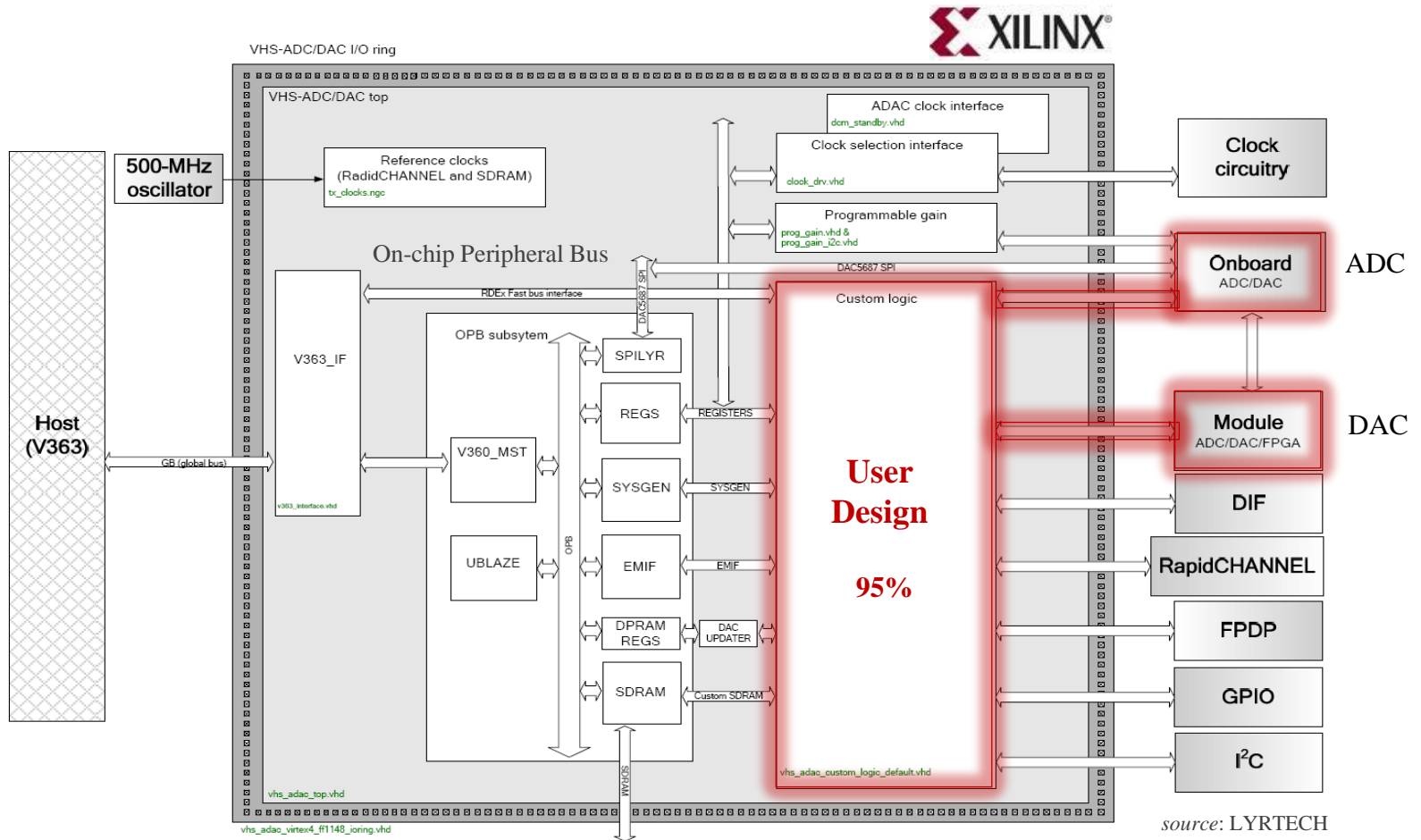
• Synoptique fonctionnel



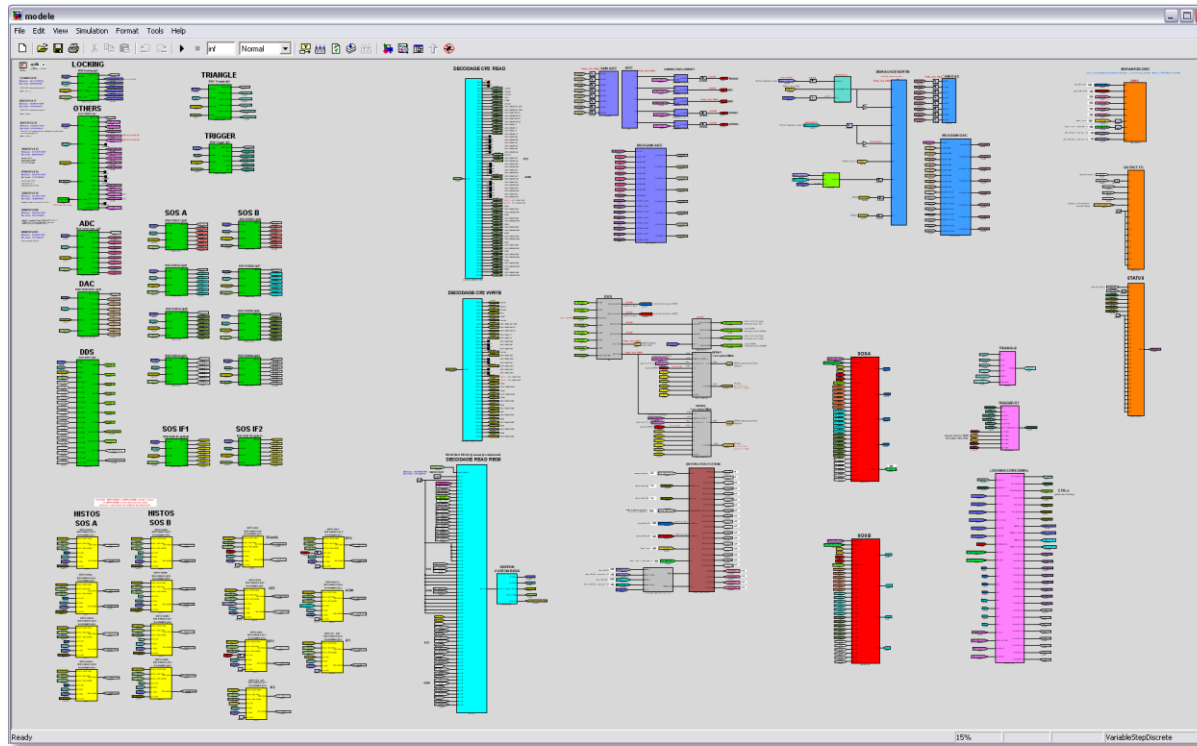
source: LYRTECH

- ++ 8 voies IN / 8 voies OUT
- ++ Temps in/out 100ns
important pour les marges de phase

- Le code du FPGA contient la **gestion des I/O** (cPCI gateway, GPIO, SDRAM...) et une partie **User Design**.
- L'ADC et le DAC sont connectés **directement** au FPGA (*fil à fil*)



- **Ecriture** d'un **modèle** (.mdl) sous **Simulink** : ensemble de **blocs fonctionnels** connectés entre eux
- **Blocs** : SIMULINK (*discarded*) , XILINX (*100, compiled*), LYRTECH propriétaires (*ADC, DAC*), VHDL user (*black box*)



Interface **Simulink**
modele.mdl

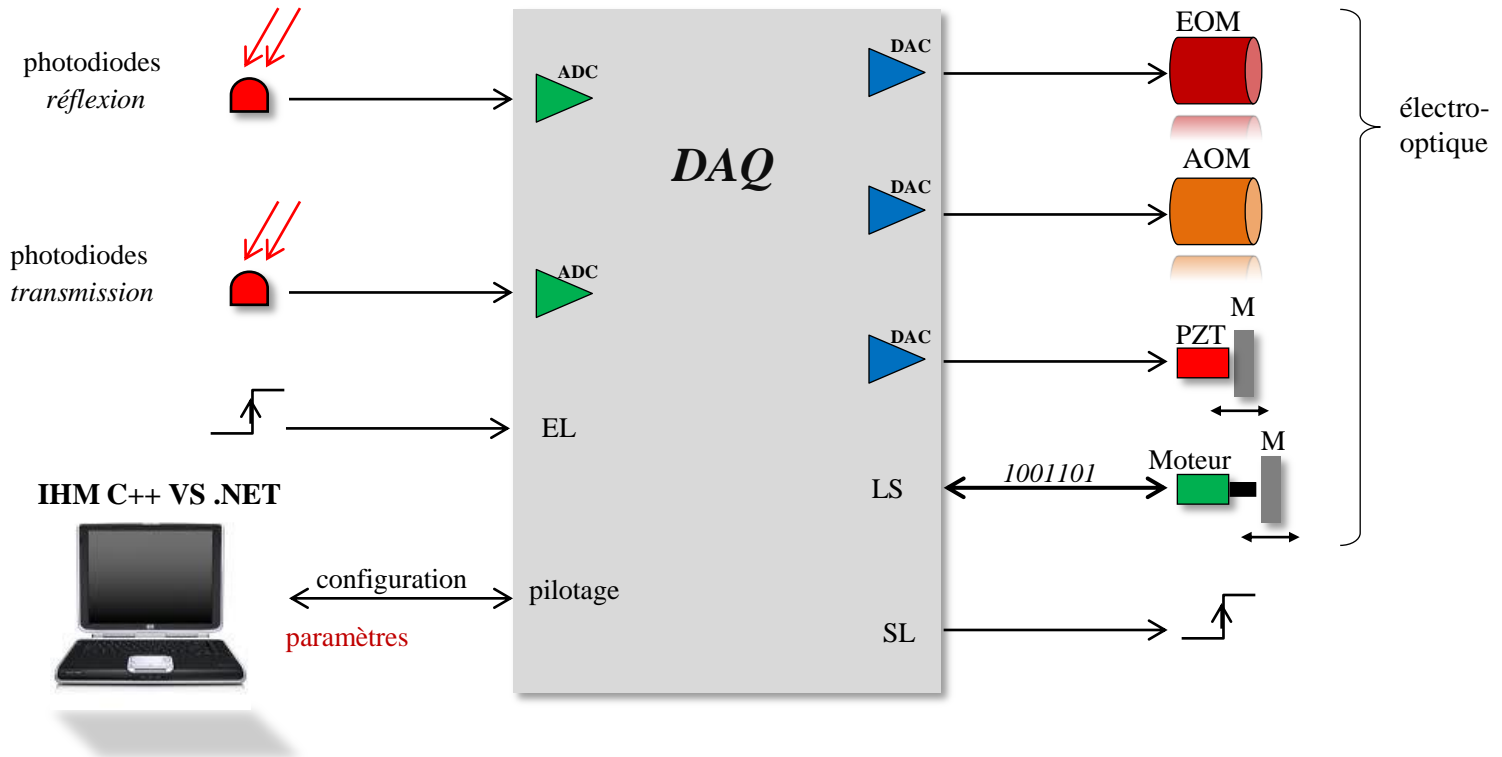
- **Outil System Generator** : concaténation / création d'un **modele_files.vhd** (**port map** automatique du top)
- **Appel de xflow.exe** : Synthèse / Placement / Routage / Generation bitstream / Chargement / Run



modele_files.vhd : 87000 lignes

T synthèse : 40 minutes

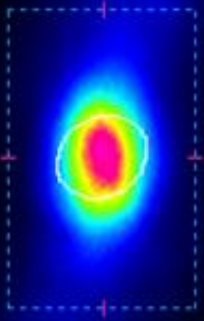
Slices utilisées @XC2V8000 : 33747 / 46592 (72%)

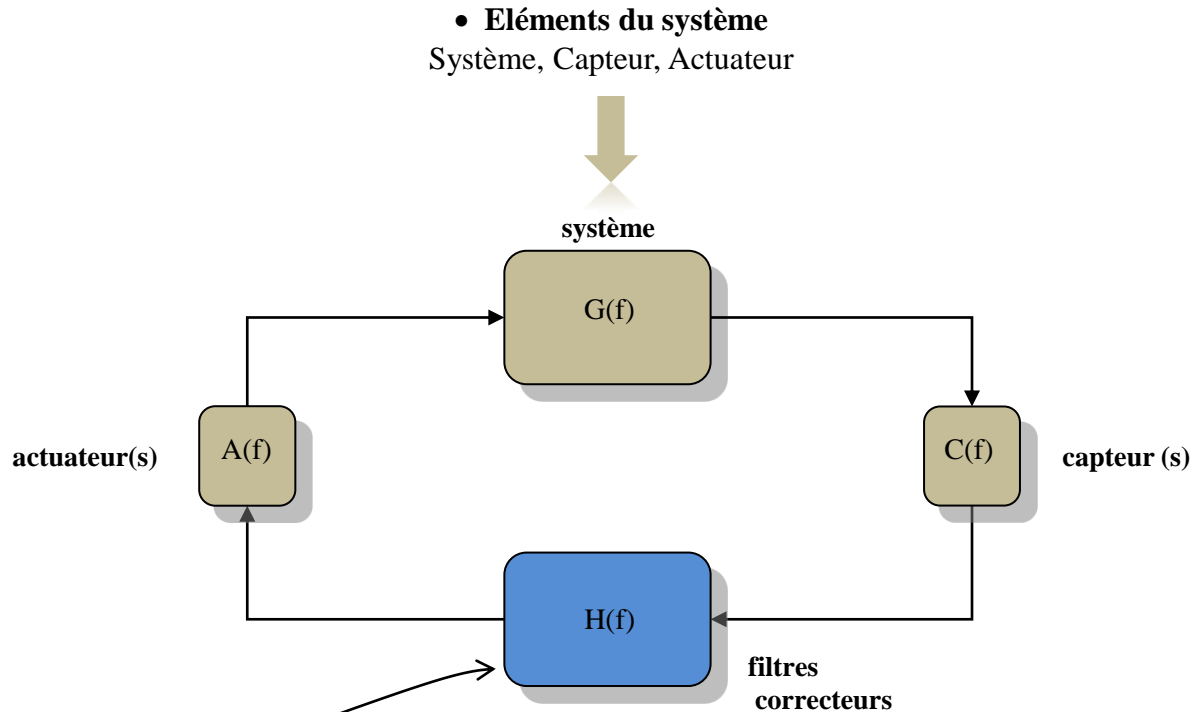


ENTREES	TRAITEMENTS réalisés par le FPGA		SORTIES
acquisition signaux analogiques	algorithmes (FSM)	calculs	produire des signaux analogiques
acquisition signaux logiques	synthèse de filtres	identification	produire des signaux logiques
acquisition liaison série RX	démultiplexage en bande de base	synthèse de fréquences DDS (Direct Digital Synthesis)	piloter liaison série TX
	synthèse de signaux (triangle...)	debug (traces, histos,...)	
	record	playback	

PLAN

- Introduction (3)
- Le système DAQ (4)
- **Exemples d'implémentation dans le FPGA (10)**
 - Identification
 - Filtres numériques
 - La fonction Pound-Drever-Hall numérique





• **BUT**

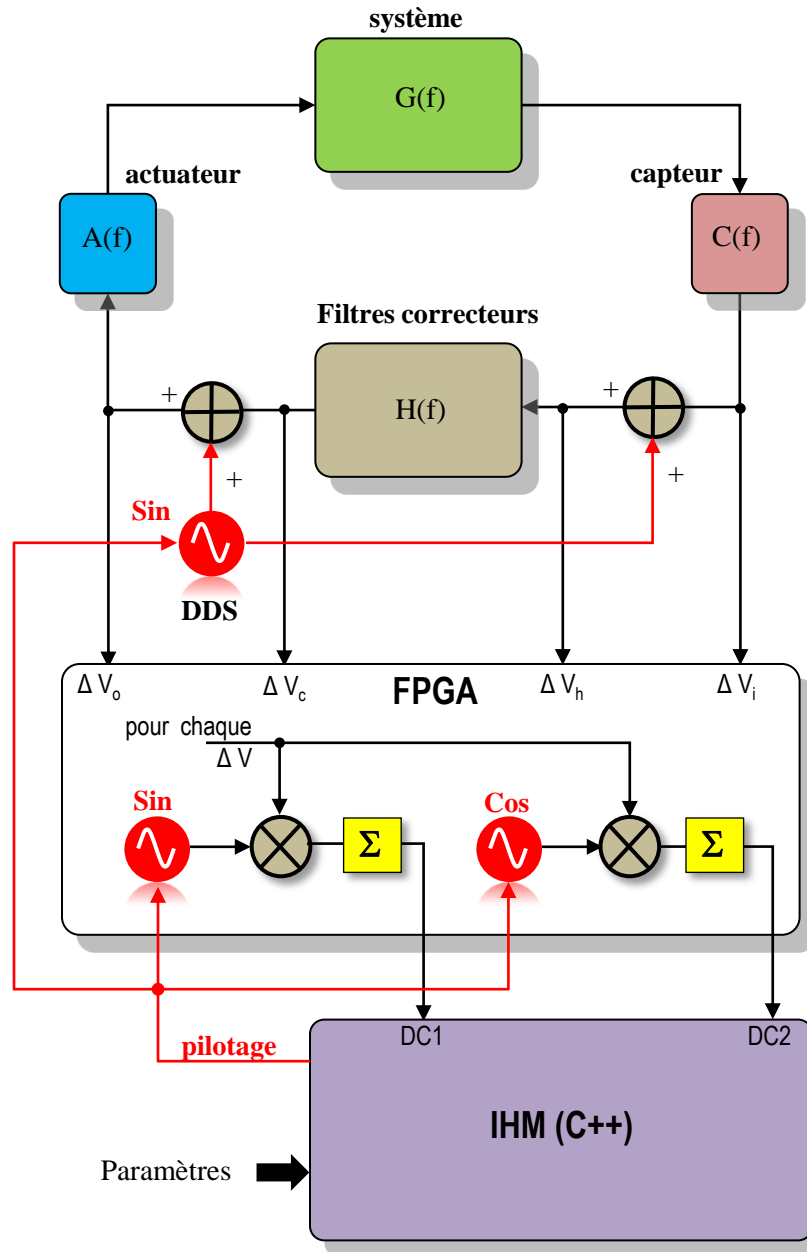
- asservir les Freq, φ Laser / Cavité
- asservir les Freq, φ Cavité / Accélérateur

• **Choix des filtres**

la structure et les coefficients des filtres correcteurs dépendent des **réponses en fréquence** des éléments A, G et C

• **Identification**

visé à déterminer ces **fonctions de transfert** (ie : 1Hz à 10MHz)



• Principe

Balayage de fréquence (DDS) piloté par l'IHM

- Intervalle F_{start} / F_{stop}
- Temps d'intégration
- Nb de points
- Equi-répartition (*Lin, Log*)
- Gabarit Sinus

Le FPGA est chargé de calculer les Σ sur des **nombre entiers** de périodes

- intégration pour obtenir le DC

Extraction **Gain/Phase** par l'IHM à partir du DC

Production en sortie de fichiers .dat

- compatibles **Matlab**
- plusieurs scans traçables sur la même courbe avec des résolutions différentes

- Utilisation**

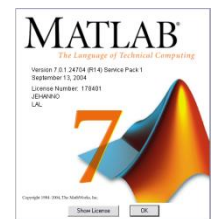
 - Auto-identification $H(f)$
 - validation du principe*
 - $A(f)$, $C(f)$, $G(f)$

Paramètres

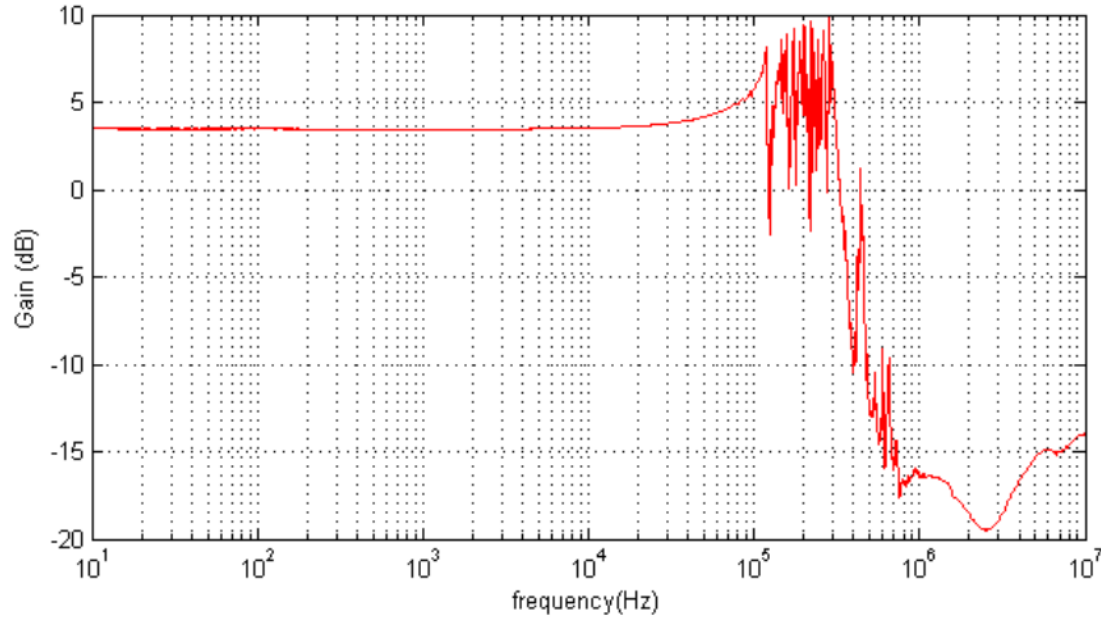
IHM (C++)



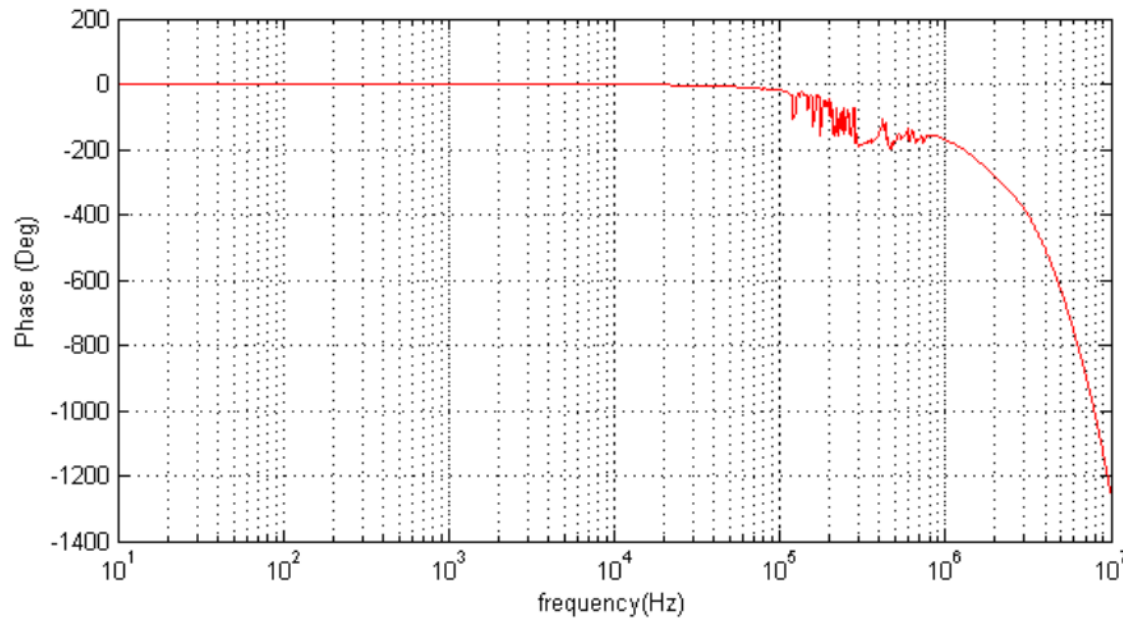
1 scan =
32 fichiers .dat
(compatible Matlab)



Identification d'un **actuateur PZT**



Gain



Phase

Filtre FIR Finite Impulse Response

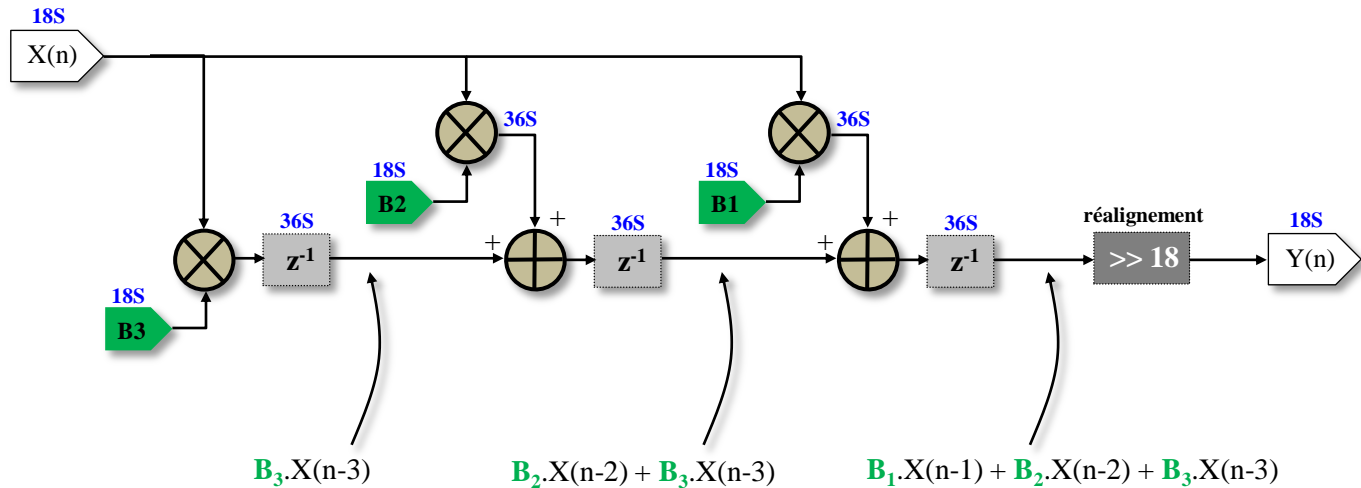
Certains filtres non réalisables (intégrateurs)

Consomme beaucoup de ressources FPGA

calcul virgule fixe

$$Y(n) = B_1 \cdot X(n-1) + B_2 \cdot X(n-2) + B_3 \cdot X(n-3)$$

$$y(n) = \sum_{k=1}^N B(k) x(n-k)$$



Filtre IIR Infinite Impulse Response

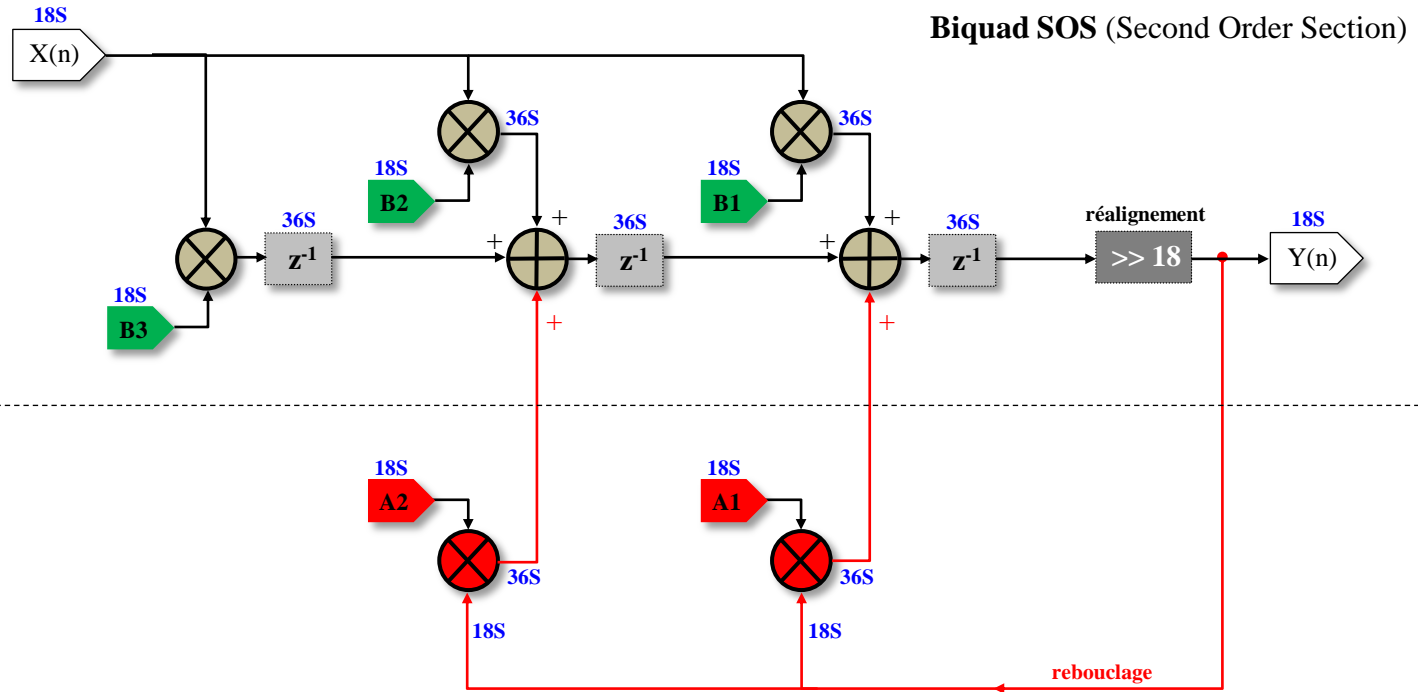
Simplicité de synthèse

Moins de coefficients pour un FIR équivalent

calcul virgule fixe

$$Y(n) = [B_1 \cdot X(n-1) + B_2 \cdot X(n-2) + B_3 \cdot X(n-3)] - [A_1 \cdot Y(n-1) + A_2 \cdot Y(n-2)]$$

$$y(n) = \sum_{k=1}^K B(k) x(n-k) - \sum_{m=1}^M A(m) y(n-m)$$



perte de précision due au réalignment

Troncature des résultats

Récursivité introduit une propagation des erreurs

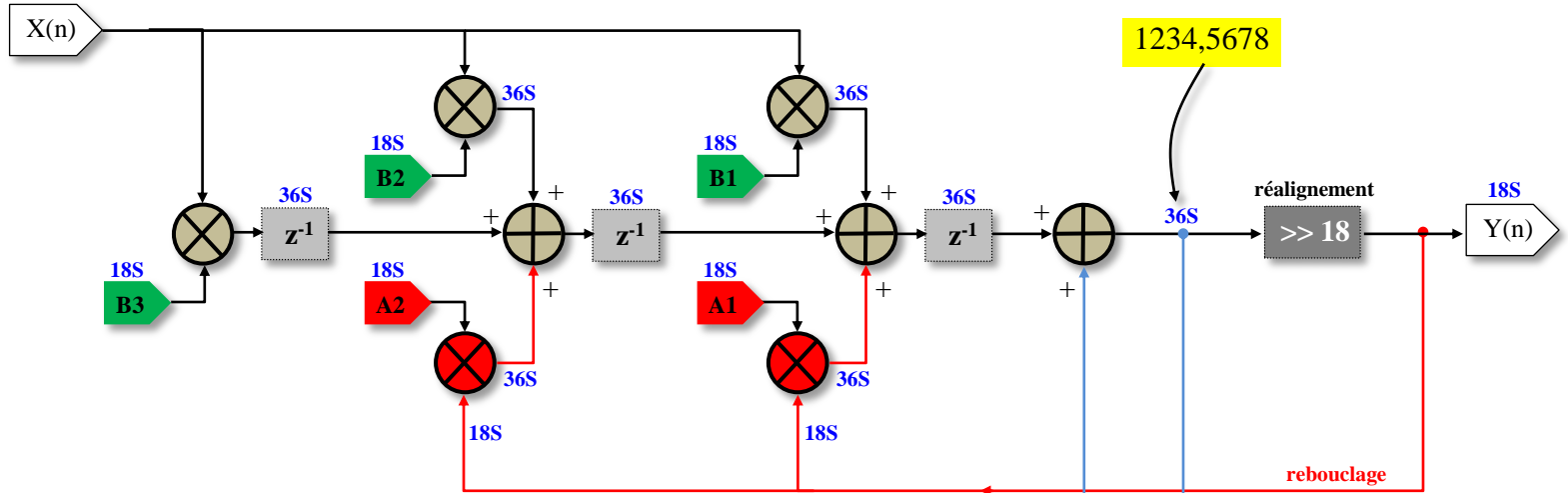
Instabilité

Structure SOS (Second Order Section) : limitation de ces effets

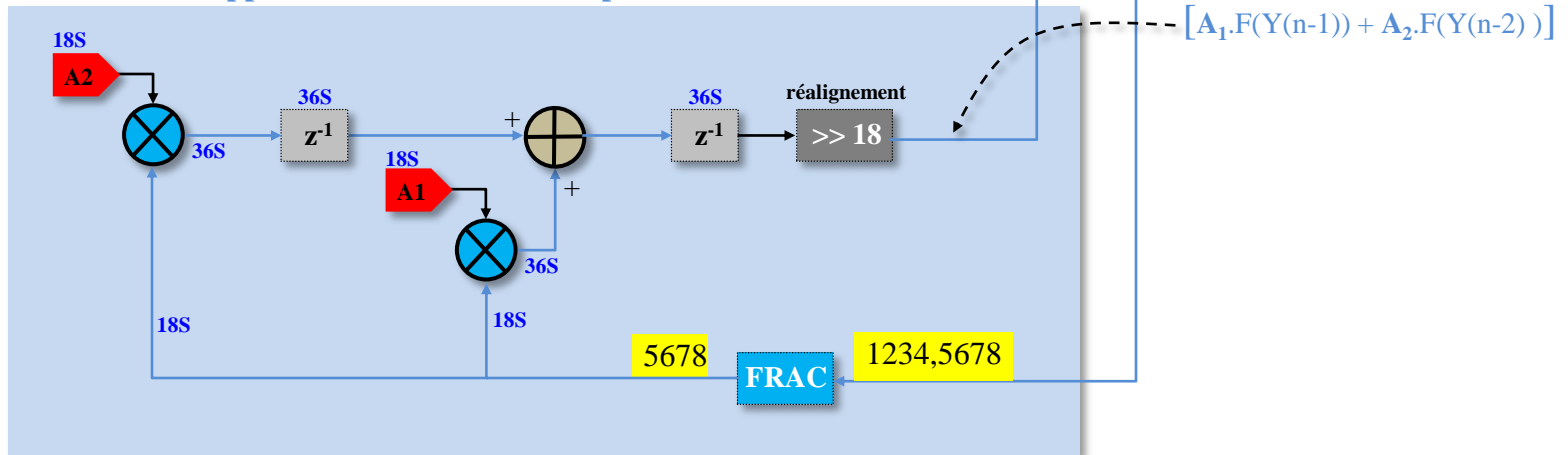
Mais ne suffit pas !

Enhanced IIR (Infinite Impulse Response)

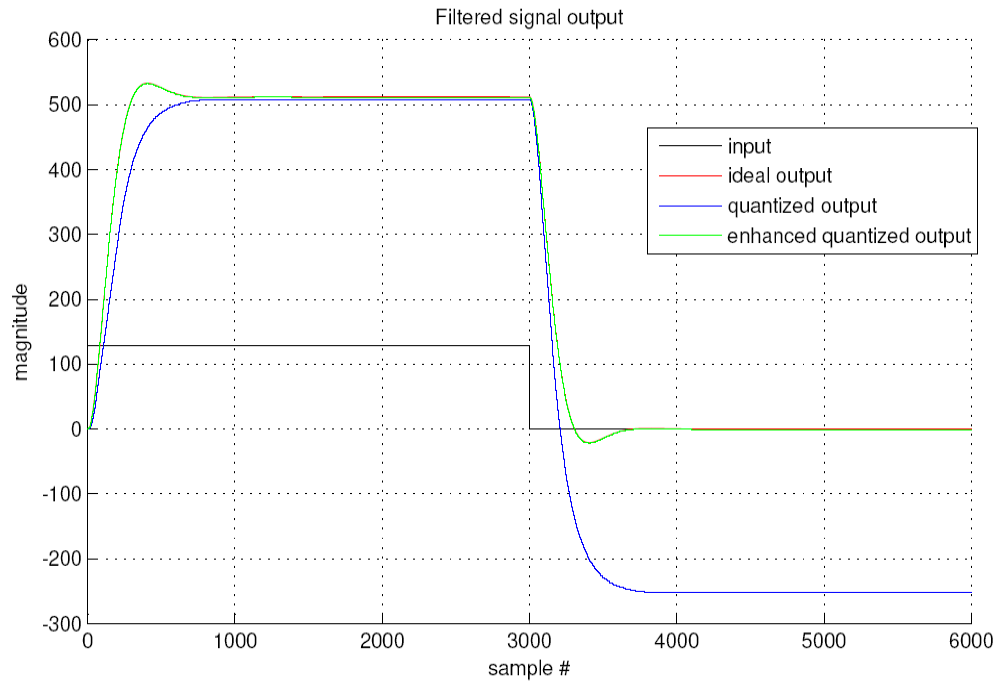
$$Y(n) = [B_1 \cdot X(n-1) + B_2 \cdot X(n-2) + B_3 \cdot X(n-3)] - [A_1 \cdot Y(n-1) + A_2 \cdot Y(n-2)]$$



Application des coefs Ax à la partie fractionnaire



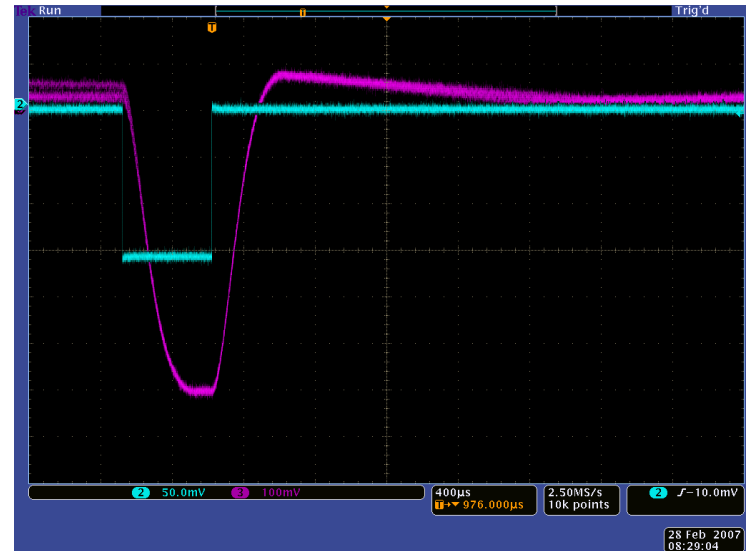
14
17



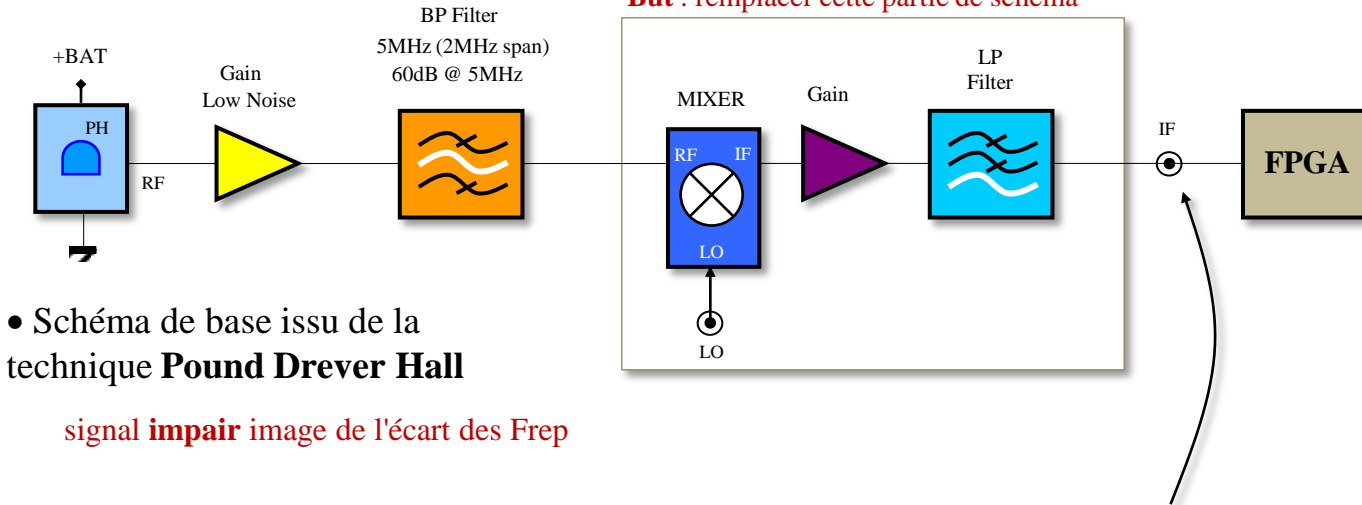
Plot 1 : Dirac



Plot 2 : échelons



Version 1 : tout analogique



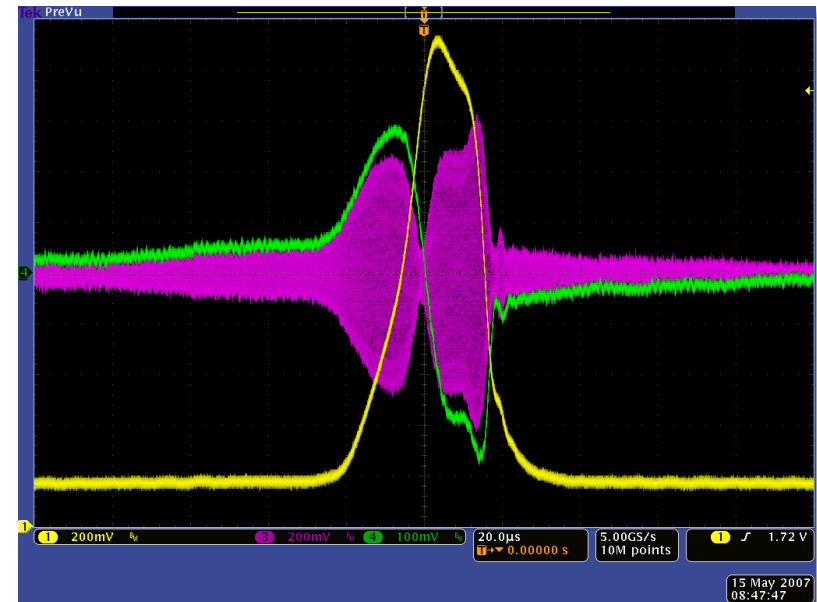
- Schéma de base issu de la technique **Pound Drever Hall**

signal **impair** image de l'écart des Freq

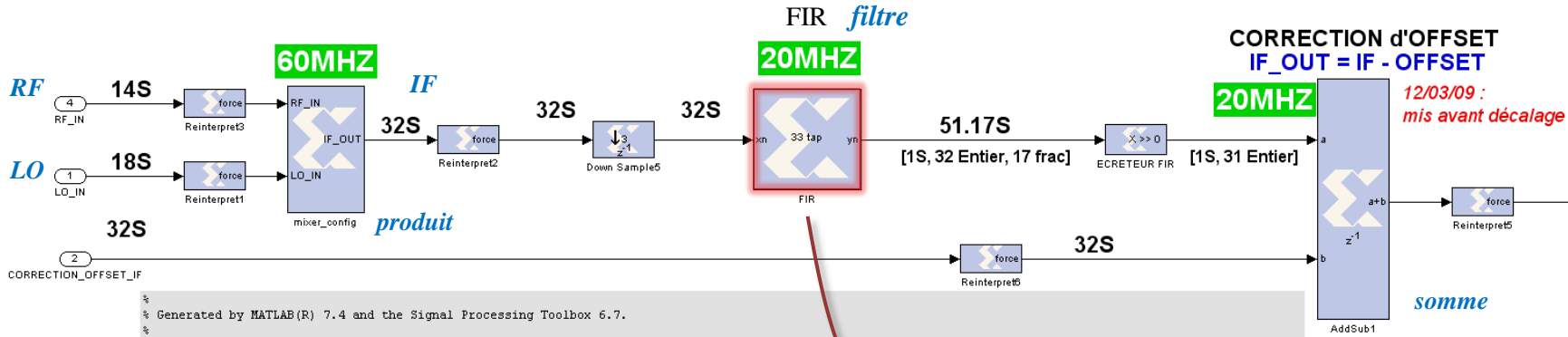
Version 2 : mixer et filtre dans le FPGA

- Simplification du front-end analogique
encombrement réduit sur la table optique
- Plus de fluctuations du DC de la démodulation
le mixer analogique laisse passer le DC gênant pour l'asservissement
- Plus de non linéarité du mixer
le mixer analogique est NL

Signal PDH démodulé (tout analogique)



16
17

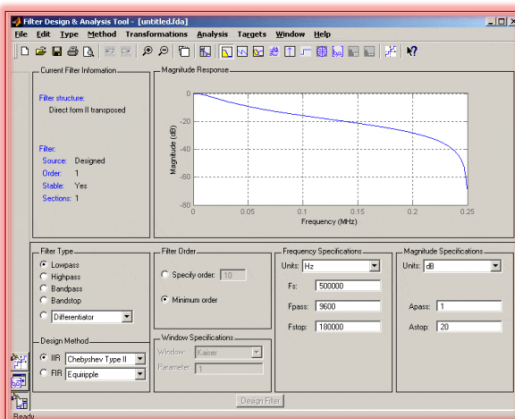


```

% Generated by MATLAB(R) 7.4 and the Signal Processing Toolbox 6.7.
% Generated on: 12-May-2009 10:31:r22
%
[0.00109100341796875 0.0065765380859375 0.02259063720703125 0.05594635009765625 0.108917236328125 0.17314910888671875 0.22786712646484375 0.2464752197265625
0.2105560302734375 0.12364959716796875 0.014617919921875 -0.0742645263671875 -0.109405517578125 -0.08454132080078125 -0.02356719970703125 0.03479766845703125
0.0602264404296875 0.0466156005859375 0.01117706298828125 -0.02048492431640625 -0.03151702880859375 -0.0217132568359375 -0.00315093994140625 0.0107574462890625
0.0137939453125 0.0083160400390625 0.000701904296875 -0.00405120849609375 -0.00479888916015625 -0.0031890069140625 -0.0013275146484375 -0.00028228759765625
0.0000228881835937]

% Coefficient Format: Decimal
% Discrete-Time FIR Filter (real)
%-----
% Filter Structure : Direct-Form FIR
% Filter Length : 33
% Stable : Yes
% Linear Phase : No
% Arithmetic : fixed
% Numerator : s16,17 -> [-2.500000e-001 2.500000e-001]
% Input : s32,0 -> [-2.147484e+009 2.147484e+009]
% Filter Internals : Full Precision
% Output : s50,17 -> [-4.294967e+009 4.294967e+009] (auto determined)
% Product : s47,17 -> [-536870912 536870912] (auto determined)
% Accumulator : s50,17 -> [-4.294967e+009 4.294967e+009] (auto determined)
% Round Mode : No rounding
% Overflow Mode : No overflow
    
```

configuration



Paramétrage du bloc FIR
Outil Matlab : FDA Tools

Recopie des coefficients

Function Block Parameters: FIR

Xilinx Finite Impulse Response Filter (mask) (link)

Finite impulse response (FIR) filter.

Hardware notes: Implemented using distributed arithmetic (DA). The hardware over sampling rate determines the degree of parallelism. A rate of one produces a fully parallel DA filter. A rate of n (resp. n+1) for an n-bit input signal produces a fully serial implementation for a non-symmetric (resp. symmetric) impulse response. Intermediate values produce implementations with intermediate levels of parallelism.

Parameters

Coefficients
[0.00109100341796875 0.0065765380859375 0.02259063720703125 0.05594635009765625 0.10891

Coefficient Structure | Inferred from Coefficients

Number of Bits per Coefficient (always treated as signed)
16

Binary Point for Coefficients
17

Number of Channels | 1

Polyphase Behavior | Single Rate: sample in - sample out

Latency
6

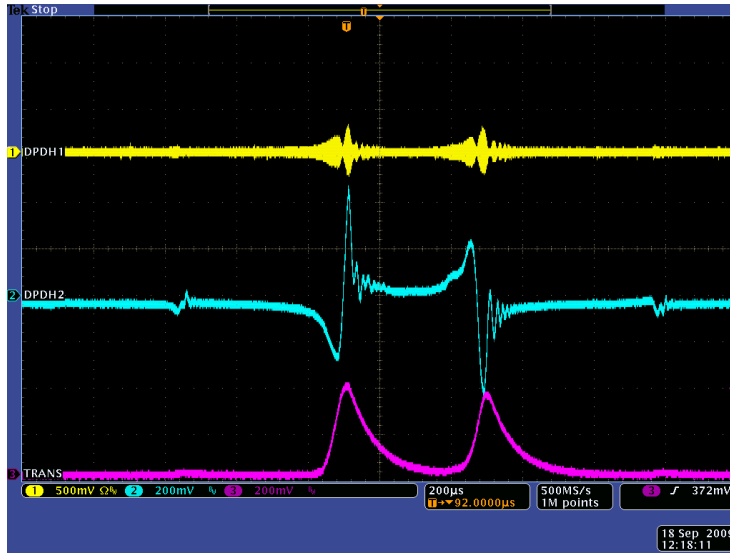
Hardware Over-Sampling Rate
1

Use Explicit Sample Period
 Provide Coefficient Reload Ports
 Provide Valid Ports
 Override with Doubles
 Show Implementation Parameters

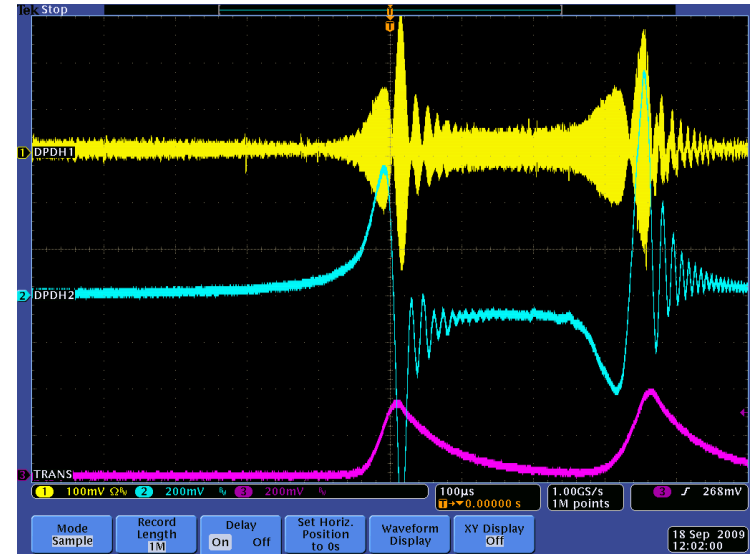
OK Cancel Help Apply

17
17

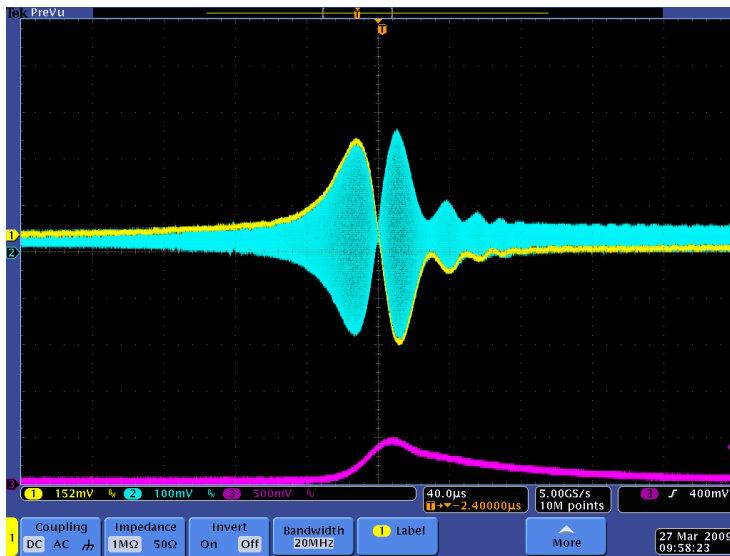
Violet : transmission (sortie de cavité)



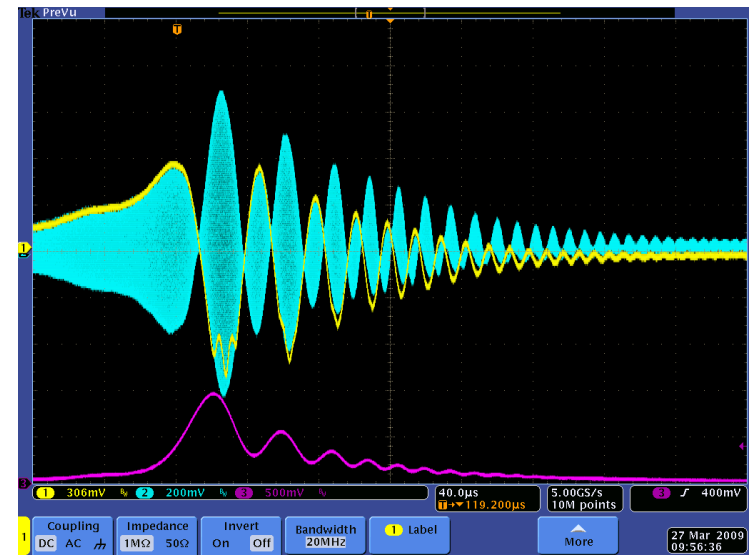
Jaune : signal PDH RF



Violet : transmission (sortie de cavité)



Bleu : signal PDH RF



Jaune : signal PDH démodulé (IF)

