

TESTABILITE DES MODULES ELECTRONIQUES

1 Introduction

Deux parties principales composent ce document :

- Une première partie - transversale - de description des motivations et des étapes franchies pour ce thème important de la conception électronique ;
- Une deuxième partie – plus technique – de présentation des parties importantes de la mise en œuvre et de l'élaboration des outils.

Cet ensemble correspond à la préparation de la partie « maintenance » du cycle en V de conception.

2 But

Obtenir des systèmes ou cartes électroniques de qualité élevée répondant aux exigences des clients finaux ou à ceux chargés de l'intégration ou de la maintenance.

3 Constat initial

- Lors des interventions de maintenance chez Siemens SA, j'ai perçu deux difficultés majeures :
 - Accessibilité des fonctions des différents modules difficiles ;
 - Risque de « sur » panne lors des interventions de maintenance.
- Chez Matra, au cours de la mise en œuvre des sous-ensembles embarqués, l'accessibilité des points de mesure et de vérification a posé des soucis d'accrochage mécanique des sondes d'oscilloscopes notamment et des risques de pannes.

4 Objectifs

- Prise en compte des besoins apparaissant lors de la maintenance du système ou de la carte ;
- Accessibilité matérielle des différentes fonctions d'une carte ou d'un module plus conséquent dans un système complet ;
- Création d'un ensemble d'informations annexes sur les constituants d'une carte ou d'un sous ensemble, y compris documentation sur les composants ;
- Constitution d'une « signalothèque », photographie du fonctionnement défini dans le cahier des charges idéal ou idéal ;
- Traitement ultérieur de signaux lors de la conception analogique ou génération de stimuli pour benches tests VHDL à partir des signaux numériques.

5 Description

5-1 Création des informations

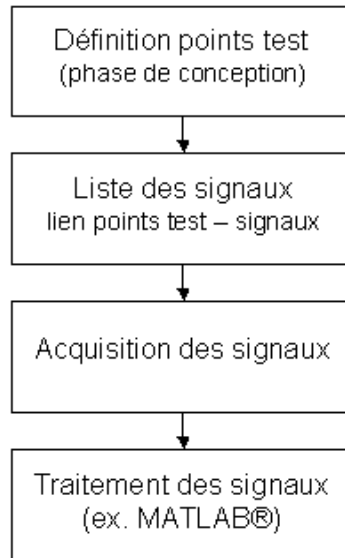
A partir de la génération de rapports du système de conception assistée par ordinateur CADSTAR®, plusieurs informations (voir [annexe 1](#)) peuvent être générées :

- La liste des signaux affectés aux broches de tous les composants d'une carte¹. Se basant sur ces premières informations, le programme *wirelist.exe* définit les listes des signaux, des points tests et des connecteurs avec les noms de signaux correspondants ;
- Les listes de points test, connecteurs, symboles schématiques et composants avec les coordonnées X et Y sur la carte. Un fichier Microsoft® Excel® regroupe alors ces listes.

¹ Netlist au format RINF.

5-2 Constitution de la « signalothèque »

Le diagramme ci-dessous illustre le déroulement de l'acquisition des signaux relatifs à une carte :



6 Outils

- Les outils matériels sont les oscilloscopes numériques avec lecteur de disquette ou de clef USB pour sauvegarder les configurations d'acquisitions et les signaux. En outre, pour des points tests sur les alimentations, le multimètre est préférable à l'oscilloscope. Lors de la conception et de l'intégration de systèmes complets, j'utilise de préférence des *connectiques standards*² que je teste avec des prises éclatées (voir [annexe 5](#)) en ayant des interconnexions les plus droites et directes possibles. Dans certains cas, les fréquencemètres donnent les informations de fréquence d'horloge. Les oscilloscopes numériques enregistrent les signaux dans les formats de fichier Mathcad, CSV³ pour le tableur Microsoft® Excel ® ou les formats propriétaires ISF. Lors des acquisitions, je choisis le format CSV ou le format ISF⁴ pour des raisons d'encombrement.
- Tant la lecture des fichiers ISF que celle des fichiers CSV ont fait l'objet de développements logiciels spécifiques (voir annexe) dont les fonctions principales sont :
 - Affichage d'une ou plusieurs courbes ;
 - Création de fichiers de données pour traitement de signal postérieur ;
 - Génération de fichiers d'images d'abord au format HPGL⁵ puis au format Post Script.
- L'acquisition se base sur les rapports générés lors de la phase de conception ([§ 5-1](#)).

² Il s'agit des connecteurs « Sub D » 9, 15, 25 ou 37 broches ou des connecteurs pour câbles en nappes

³ CSV : Comma Separated Values. Dans ces fichiers, les séparateurs de données sur les lignes sont des virgules ou des points virgules.

⁴ Le format de ces fichiers est spécifique à chaque type d'oscilloscope utilisé. Cependant, ces fichiers peuvent aussi être rechargés dans l'oscilloscope dont il est originaire.

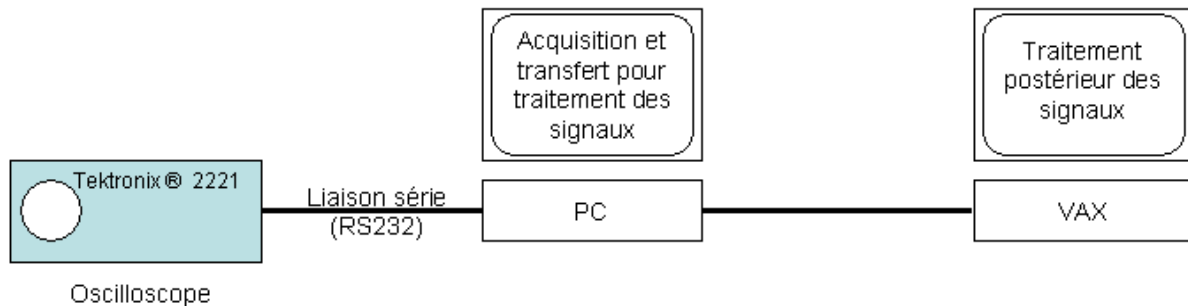
⁵ Ce format de fichier utilisé pour le pilotage des plotters Hewlett devient peu à peu obsolète.

7 Historique des connexions oscilloscopes – ordinateur de traitement de données

Les traitements des données provenant des oscilloscopes de ce paragraphe suivent l'algorithme global de l'[annexe 2](#).

7-1 Oscilloscope mixte Tektronix® 2221

Les développements logiciels autour de ce matériel utilisent la technologie des années 1990. L'oscilloscope numérique Tektronix® 2221 allie les caractéristiques numériques et analogiques⁶. Les acquisitions se déroulent selon le diagramme :



- Le PC émet vers l'oscilloscope les ordres nécessaires à l'acquisition (figeage de la session, transfert par la liaison série, sauvegarde sur le disque dur) ;
 - Un logiciel spécifique transforme les fichiers - signaux en fichiers compatibles pour un transfert vers l'ordinateur VAX (1990) ou un traitement dans MATLAB®⁷ (1995).
- Un exemple d'illustration est présenté au §2 de l'annexe 3.

7-2 Oscilloscope numérique Tektronix®TDS.3000

Les performances de ce matériel purement numérique constituent une avancée significative par rapport à l'oscilloscope 2221 (fréquence d'échantillonnage plus élevée, taille mémoire plus importante).

Les oscilloscopes de cette famille possèdent un lecteur de disquettes ou de clefs USB permettant l'enregistrement des signaux et des configurations d'acquisitions pour sauvegarde.

Les premiers développements dans MATLAB® (*tds3000.m*) remontent à 1999.

Le principal objectif est l'utilisation des fonctions évoluées de traitement de signaux.

Un exemple d'illustration est présenté au §3 de l'annexe 3.

7-3 Oscilloscope numérique Tektronix® TDS.1000

Ces oscilloscopes d'entrée de gamme sauvegardent les signaux dans le format CSV lisibles dans MATLAB pour effectuer des traitements de signaux similaires à ceux que propose le script *tds3000.m*

7-4 Oscilloscope numérique Tektronix® DPO 2000

J'ai utilisé cet oscilloscope de mars à juillet 2009 lors de la mission à l'Institut des Nanotechnologies de Lyon pour la mise au point de l'amplificateur audionumérique. Comme les fichiers au format spécifique de cet oscilloscope sont différents de ceux du TDS3000⁸, un développement spécifique dans MATLAB® de la partie lecture a donc été nécessaire (*dpo2000.m*). Cette application comprend donc les fonctions de l'oscilloscope TDS 3000.

En outre pour finaliser la mise au point du récepteur de trame I²S, j'ai développé l'algorithme de génération de stimuli ([§2 -annexe 2](#)) dont les résultats sont présentés dans l'[annexe 4](#).

⁶ Comme la fréquence d'échantillonnage de cet oscilloscope était basse (10 MHz), un algorithme stroboscopique permettait la visualisation des signaux de fréquence supérieure à la fréquence d'échantillonnage. Le risque de repliement des hautes fréquences dans les basses fréquences du au sous-échantillonnage était réel.

⁷ Le script *spec2221.m* effectue des traitements identiques à ceux implantés dans l'ordinateur VAX

⁸ les fréquences d'échantillonnages, les tailles mémoires de chaque voie et la définition des échantillons constituent les principales différences entre les deux matériels.

8 Conclusion

- L'alliance entre les définitions des points test matériels et les oscilloscopes est une première approche pour faciliter l'intégration et la maintenance des matériels ;
- Les besoins spécifiques des personnes chargées de l'intégration et de la maintenance doivent être pris en compte dès la conception, afin de faciliter les interventions aux différents stades du cycle de vie ;
- La poursuite de ces parties de projets me semble intéressante pour l'entreprise car elle permet une réduction de coûts pertinente et offre de larges perspectives, une vision large sur le métier de chacun et des possibilités d'échanges mutuels enrichissants entre les différents membres d'une équipe projet.

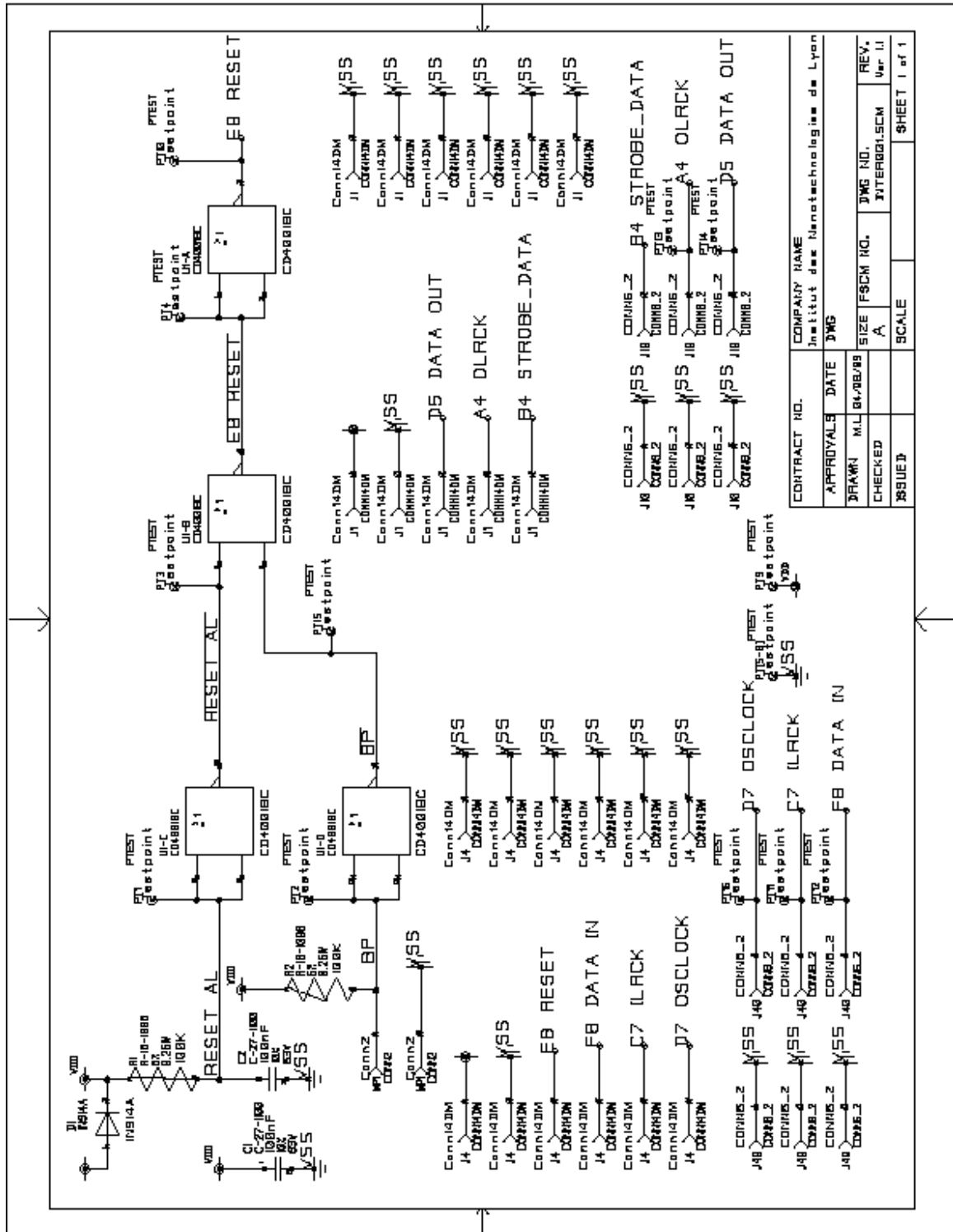
Contact pour la poursuite de la discussion sur ces thèmes : michel.lemoine60@sfr.fr

Projets de développement matériel : <http://michel.lemoine60.perso.sfr.fr/projets.html>.

ANNEXE 1 : RAPPORTS CRES LORS DE LA CONCEPTION

1 Introduction

Cet exemple est basé sur le schéma d'un module fournissant les signaux annexes de démarrage de la carte Xilinx SPARTAN 3E. Dans cet exemple, les points test sont présentés dans [§2](#).



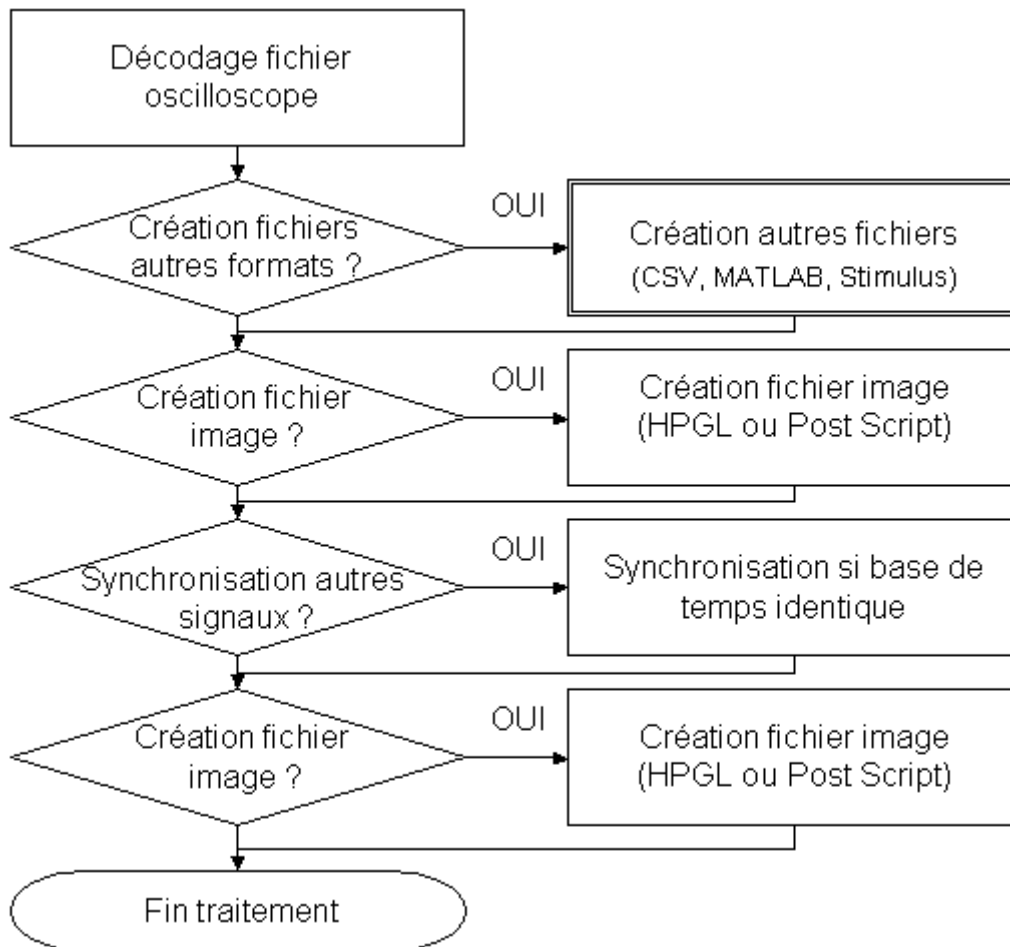
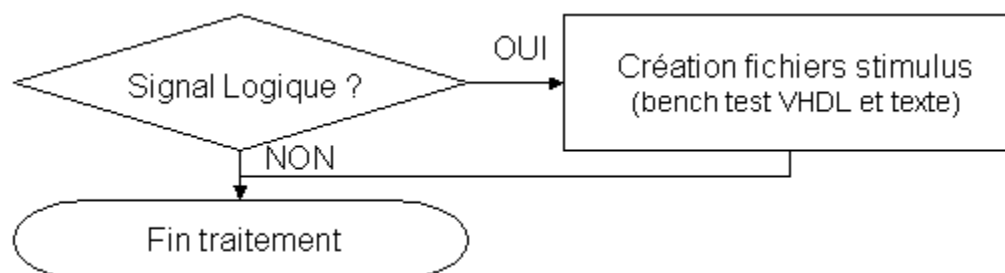
2 Points test2-1 Liste issue de wirelist.exe

Component	Pin	Signal
PT1	1	RESET AL
PT2	1	BP
PT3	1	\RESET AL\
PT5	1	VSS
PT6	1	VSS
PT7	1	VSS
PT8	1	VSS
PT9	1	VDD
PT4	1	\E8 RESET\
PT10	1	E8 RESET
PT12	1	F8 DATA IN
PT13	1	A4 OLRCK
PT14	1	D5 DATA OUT
PT15	1	\BP\
PT11	1	C7 ILRCK
PT16	1	D7 OSCLOCK

2-2 Rapport CADSTAR

Point test	Signal	Fichier ISF	HPGL	Setting	Feuille	Position X	Position Y
PT[5-8]					Interconnexion	6000	2650
PT9					Interconnexion	6800	2650
PT1					Interconnexion	4000	8200
PT3					Interconnexion	6800	8000
PT4					Interconnexion	9200	7950
PT10					Interconnexion	10600	8000
PT13					Interconnexion	9800	3550
PT14					Interconnexion	9800	3150
PT15					Interconnexion	6400	6600
PT2					Interconnexion	4000	6800
PT12					Interconnexion	4000	2150
PT11					Interconnexion	4000	2550
PT16					Interconnexion	4000	2950

ANNEXE 2 : ORGANIGRAMMES LOGICIELS

1 Algorithme global2 Génération de stimulus (application DPO2000)

ANNEXE 3 : EXEMPLES D'UTILISATION

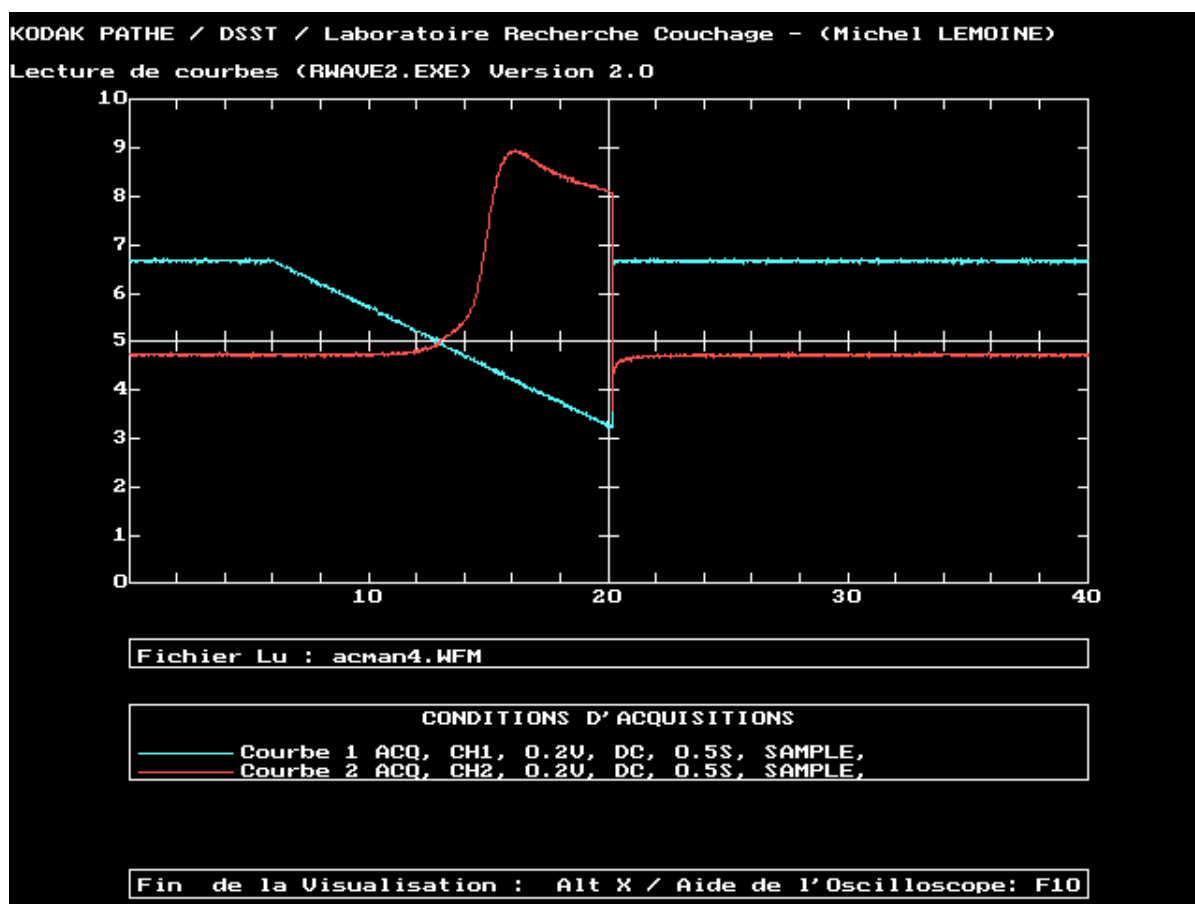
1 Introduction

Cette partie illustre l'historique des connexions oscilloscope – PC ([§7](#)). L'exemple de l'oscilloscope DPO 2000 fait l'objet d'une présentation spécifique à l'[annexe 4](#). Cette partie concerne les oscilloscopes [2221](#) et [TDS 3000](#).

2 Oscilloscope Tektronix® 2221

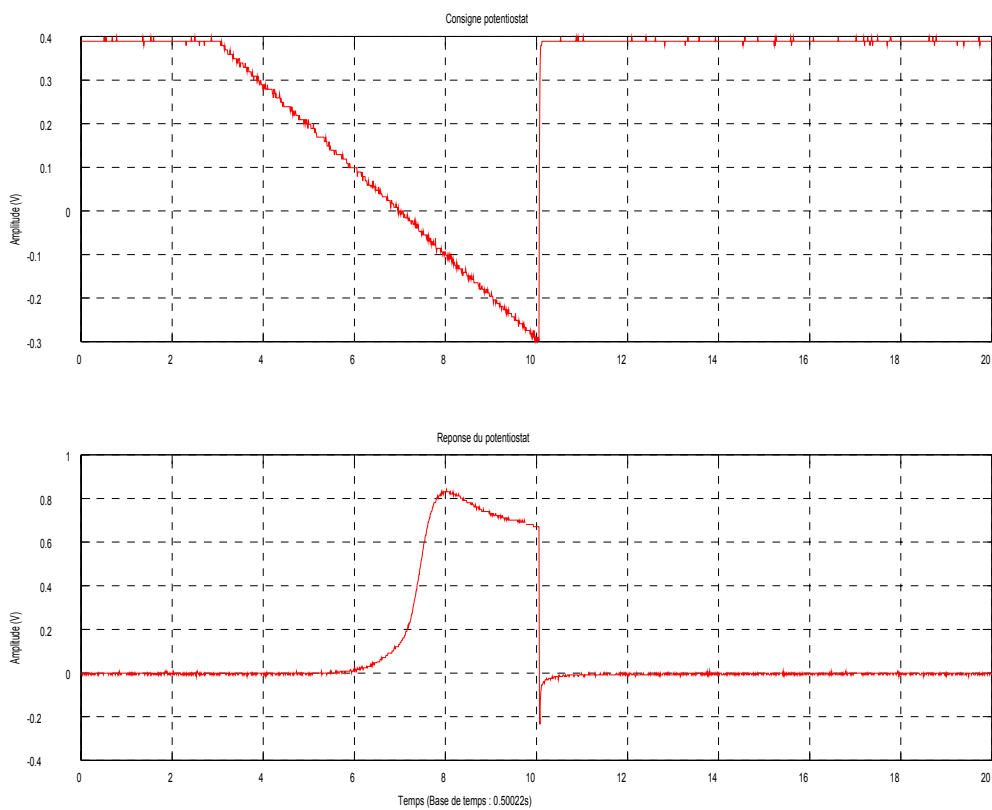
La connexion entre l'oscilloscope utilise deux logiciels principaux [rwave2.exe](#) et l'application MATLAB® [spec2221](#).

L'exemple présenté est tiré de l'appareil de mesure de concentration d'acide ascorbique. Un générateur de rampe fait varier le potentiel sur l'électrode de travail du potentiostat. La première image montre le résultat de la lecture dans [rwave2.exe](#).



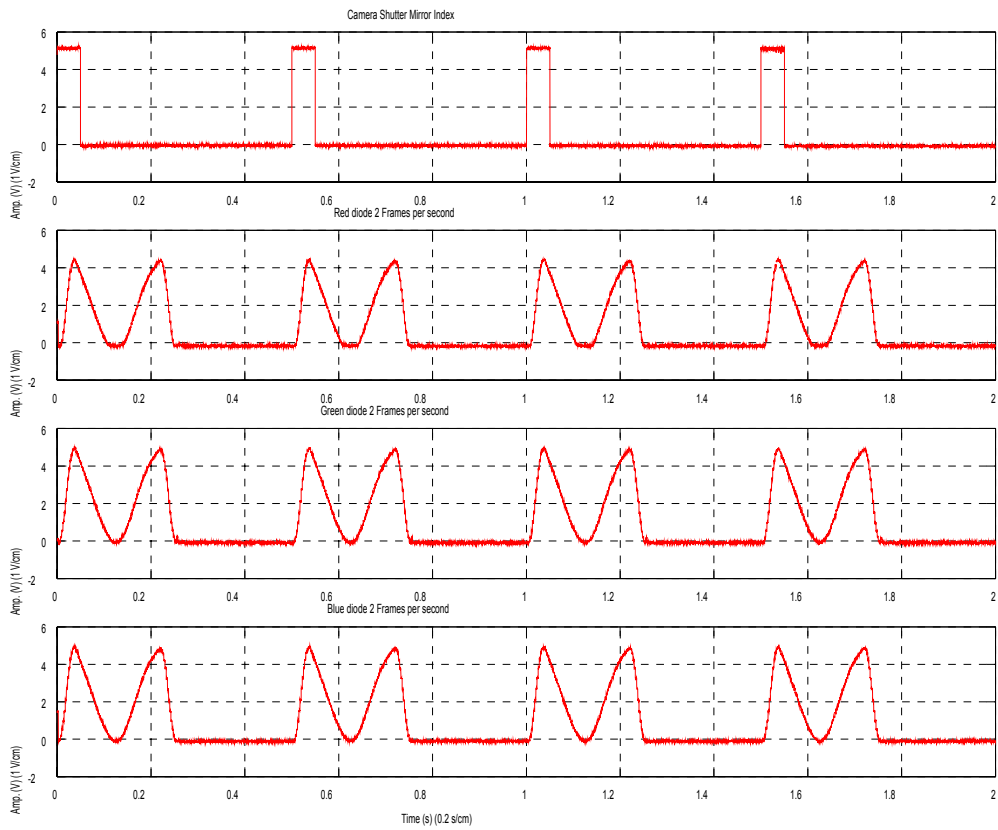
ANNEXE 3 : EXEMPLES UTILISATION

La deuxième image montre le résultat du traitement dans *spec2221.m* :



3 Oscilloscope Tektronix® TDS 3000

Cet exemple est la consigne des diodes électroluminescentes utilisées pour l'enregistrement de données sur film cinéma :



ANNEXE 4 : STIMULI VHDL

1 Introduction

Comme évoqué dans le §7-4, cette partie a servi lors de la mise au point de la description VHDL des émetteurs – récepteurs de trame I²S.

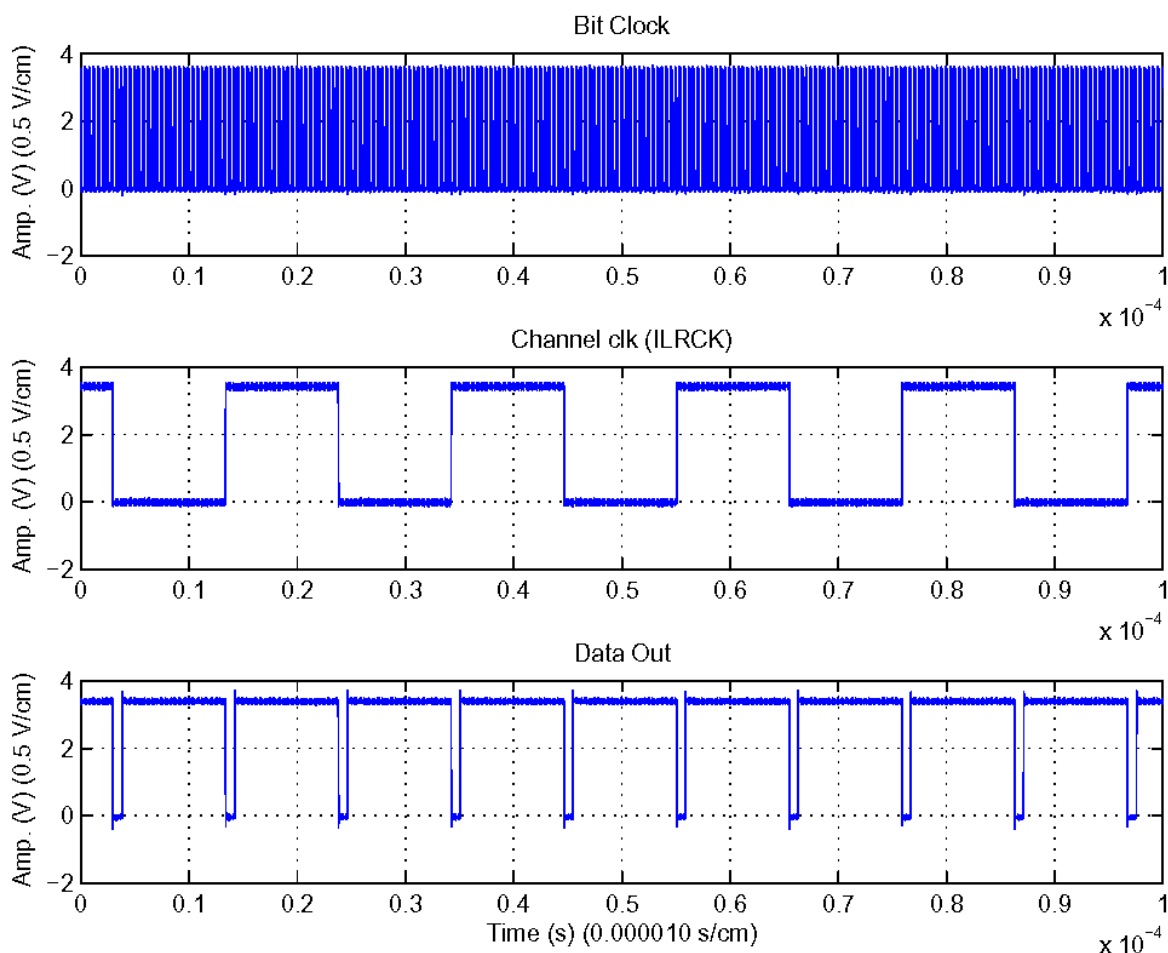
Comme le montre la figure de §2, trois signaux sont nécessaires à ce mode de transmission de signaux numériques pour le son stéréophonique :

- Le signal *bit_clock* pour la synchronisation des transmissions ;
- Le signal *ilrck*, indicateur de la voie gauche ou droite des données en cours de transmission ;
- Le signal *dataout* qui correspond au son numérisé.

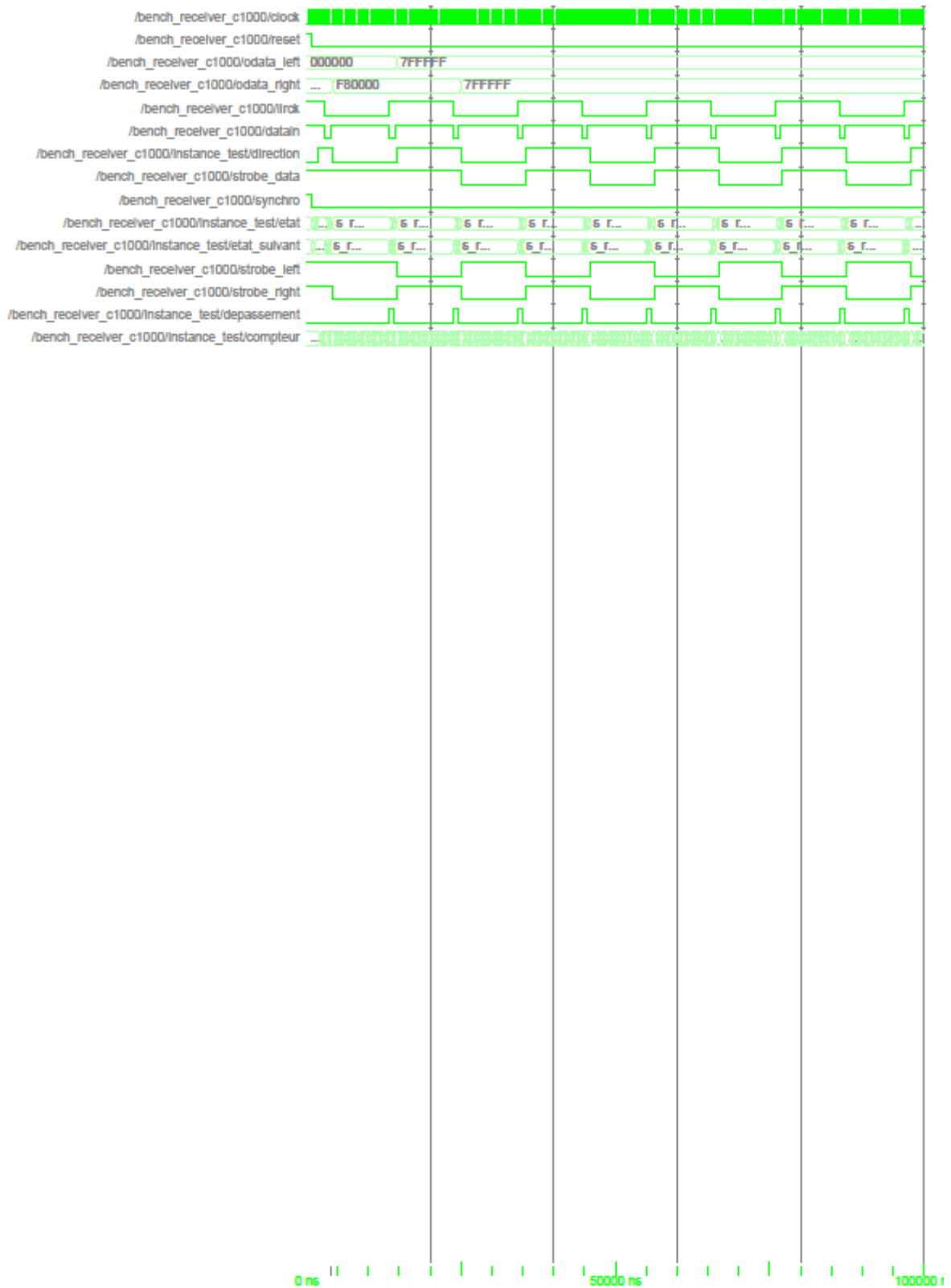
Dans notre cas, le son est codé sur 24 bits en virgule fixe correspondant à des valeurs numériques physiques comprises entre $1 \cdot 2^{-23}$ et -1 .

Le générateur interne de l'analyseur de son Audio Precision 2722 émet sur les voies droite et gauche une trame constante égale à $1 \cdot 2^{-23}$ soit la valeur hexadécimale 7FFFFF. Après acquisition sur l'oscilloscope DPO 2000, les signaux sont transformés en stimuli destinés à l'implantation dans un test bench VHDL (§4). Lors de la simulation, le FPGA Xilinx Spartan 3E décode cette valeur (§3).

2 Trames d'entrée

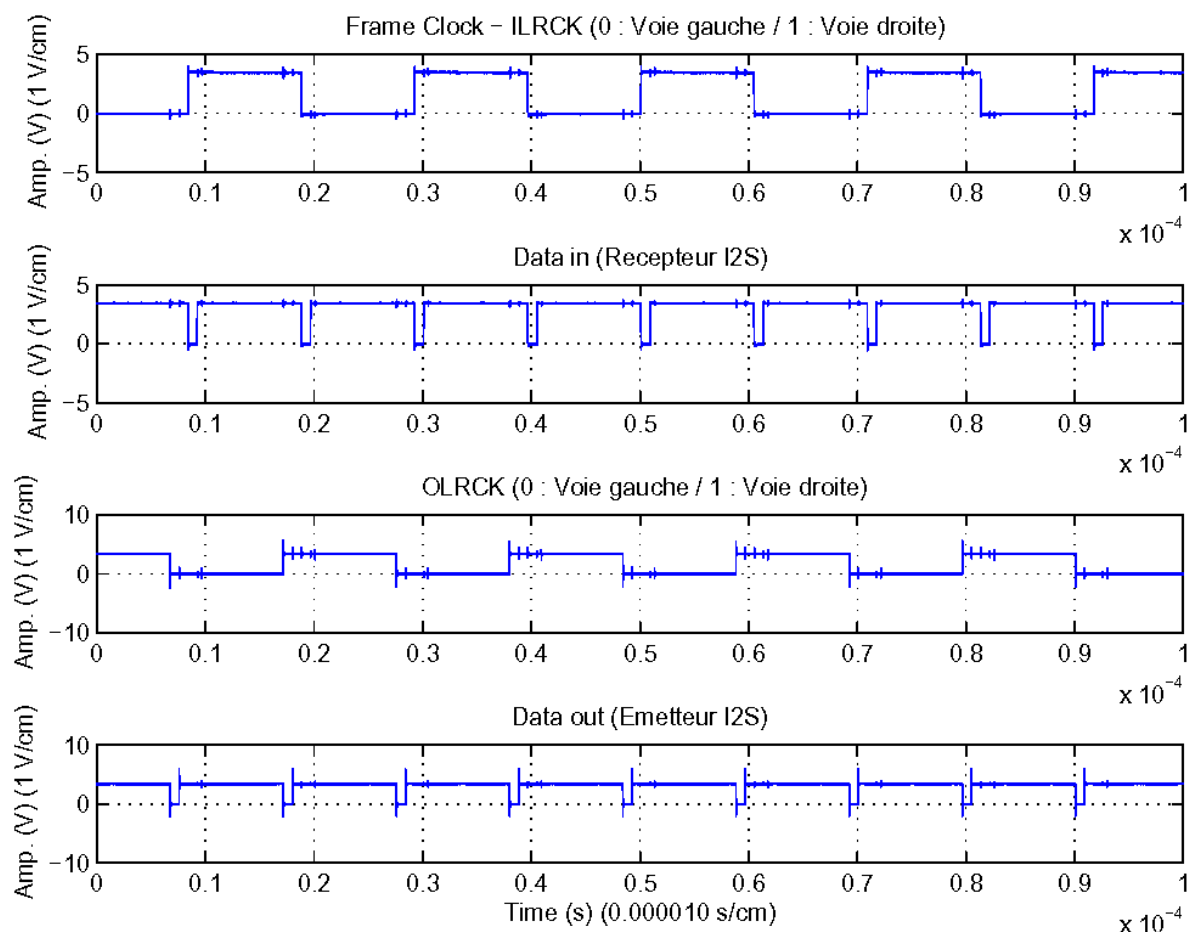


3 Résultat de simulation



Entity: bench_receiver_c1000 Architecture: tb_receiver Date: Tue Nov 30 21:18:10 Paris, Madrid 2010 Row: 1 Page: 1

Lors de la réception des signaux *ilrck* et *Data In*, le FPGA Xilinx Spartan 3E émet *olrck* et *Data Out* :



4 Exemple de fichier de stimulus (signal *ilrck*)

La routine MATLAB crée le signal (*stimulus*) et son inverse logique (*invstimulus*) :

```
--declaration du stimulus
signal stimulus : std_logic;
signal invstimulus : std_logic;
--Fin de la declaration stimulus
--Stimulus test bench
stimulus <= '1' after 0 ns, '0' after 2.966000e+000 us, '1' after 1.339300e+001 us,
'0' after 2.379900e+001 us, '1' after 3.422600e+001 us, '0' after 4.463300e+001 us,
'1' after 5.505900e+001 us, '0' after 6.546600e+001 us, '1' after 7.589300e+001 us,
'0' after 8.629900e+001 us, '1' after 9.672600e+001 us;
--Stimulus inverse test bench
invstimulus <= '0' after 0 ns, '1' after 2.966000e+000 us, '0' after 1.339300e+001
us, '1' after 2.379900e+001 us, '0' after 3.422600e+001 us, '1' after 4.463300e+001
us, '0' after 5.505900e+001 us, '1' after 6.546600e+001 us, '0' after 7.589300e+001
us, '1' after 8.629900e+001 us, '0' after 9.672600e+001 us;
--Temps maximal : 9.999900e-005
--Fin instruction
```

Cette partie de code doit ensuite être incluse dans le test bench VHDL regroupant tous les autres signaux de test.

ANNEXE 5 : CONNECTIQUES DE TEST

