



HAL
open science

Modélisation multidisciplinaire VHDL-AMS de systèmes complexes : vers le Prototypage Virtuel

Sébsatien Snaidero

► **To cite this version:**

Sébsatien Snaidero. Modélisation multidisciplinaire VHDL-AMS de systèmes complexes : vers le Prototypage Virtuel. Micro et nanotechnologies/Microélectronique. Université Louis Pasteur - Strasbourg I, 2004. Français. NNT : . tel-00009325v2

HAL Id: tel-00009325

<https://theses.hal.science/tel-00009325v2>

Submitted on 29 Jun 2005

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

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



**Thèse présentée afin d'obtenir le grade de
Docteur de l'Université Louis Pasteur
Strasbourg I**



Discipline :
Electronique, électrotechnique, automatique
Spécialité :
Microélectronique ■

Modélisation multidisciplinaire VHDL-AMS de systèmes complexes : vers le Prototypage Virtuel

Soutenue publiquement à l'ENSPS le 03/12/2004,
par **Sébastien SNAIDERO** ■

Membres du jury : ■

Rapporteur Externe : M. Pascal FOUILLAT, Professeur, Université Bordeaux 1
Rapporteur Externe : M. Denis FLANDRE, Professeur, UCL Louvain-la-Neuve (Belgique)
Rapporteur Interne : M. Christophe LALLEMENT, Professeur, ULP Strasbourg
Examineur : Mme Patricia DESGREYS, MdC, ENST Paris
Examineur : M. Daniel MATHIOT, Professeur, ULP Strasbourg
Directeur de Thèse : M. Yannick HERVE, MdC (HDR), ULP Strasbourg

Remerciements

Je souhaiterais tout d'abord exprimer ma gratitude à mon directeur de thèse, Yannick HERVE, pour m'avoir donné la possibilité d'effectuer cette thèse dans le domaine de la modélisation des systèmes complexes dont il est un des pionniers. Je le remercie également pour ses encouragements et ses remarques savamment distillés au cours de la thèse à des moments où l'espoir d'y arriver s'amenuisait quelque peu.

Je suis tout particulièrement reconnaissant à Messieurs Pascal FOUILLAT, Denis FLANDRE et Christophe LALLEMENT d'avoir accepté d'être rapporteurs de ma thèse, ainsi qu'à Madame Patricia DESGREYS et Monsieur Daniel MATHIOT d'en avoir été les examinateurs.

J'adresse aussi un chaleureux merci à toute l'équipe du projet SHAMAN qui a permis par sa bonne humeur, son sérieux et son entrain de faire avancer ce projet de manière intéressante.

Je dis également un grand merci à mes collègues du laboratoire PHASE, et spécialement à Yann LEROY qui, tant comme ami que collègue, m'a apporté son soutien moral et technique quand j'en ai eu besoin.

Enfin, j'adresse, à mes parents, des remerciements particulièrement chaleureux pour m'avoir encouragé, compris, soutenu et réconforté durant ces deux années de thèse et toutes celles qui les ont précédées.

Table des Matières

Publications auxquelles je suis associé	9
En relation avec le projet SHAMAN	9
Extérieures au projet SHAMAN	9
Introduction	10
Chapitre 1 - Objectifs et contexte de la Thèse [SHA1]	13
1.1 - Les motivations du projet SHAMAN	13
1.2 - Les objectifs du projet	14
1.3 - Le démonstrateur	15
1.3.1 - Structure du démonstrateur	15
1.3.2 - Caractéristiques du démonstrateur	18
1.4 - Le consortium	18
1.4.1 - Les partenaires	18
1.4.2 - L'organisation du projet	20
1.5 - L'analyse de la problématique	21
1.5.1 - Le modèle industriel descriptif	25
1.5.2 - Le modèle prédictif de développement	25
Chapitre 2 - Langages de modélisation et simulateurs	30
2.1 - Le choix du langage de modélisation	30
2.1.1 - Les langages de programmation orientés objet	30
2.1.2 - Les langages de modélisation numérique	30
2.1.3 - Les langages mathématiques formels explicites	31
2.1.4 - Les langages de modélisation implicites dédiés à l'électronique	31
2.1.5 - Les langages de modélisation analogique multi-domaines	31
2.1.6 - Les langages de modélisation mixte multi-domaines	32
2.2 - VHDL-AMS [HERI, CHRI, PETI]	34
2.2.1 - Histoire de VHDL-AMS	34
2.2.2 - Les avantages de VHDL-AMS	34
2.2.3 - Les limites de VHDL-AMS	35
2.2.4 - Les évolutions liées à VHDL-AMS [LRM1]	36
2.3 - Le choix du logiciel de simulation	37
2.3.1 - Les simulateurs de première génération à code apparent	37
2.3.2 - L'évolution des utilisateurs	38
2.3.3 - L'évolution des environnements de travail	39
2.3.4 - Les possibilités offertes par l'interfaçage avec VHDL-AMS	41

2.3.5 - Les nouveaux simulateurs disposant d'une GUI	41
2.3.6 - Conclusion	43
2.4 - Simplorer [ANSI]	44
2.4.1 - Introduction	44
2.4.2 - L'interface graphique	45
2.4.3 - La création de modèles en mode texte	47
2.4.4 - L'implémentation de la norme VHDL-AMS	48
2.4.5 - Les possibilités d'interfaçage	49
2.5 - Conclusion	52
Chapitre 3 - Les modèles élaborés	53
3.1 - La fibre optique	53
3.1.1 - La dispersion modale	54
3.1.2 - La dispersion chromatique	54
3.1.3 - La dispersion totale [SHAI]	55
3.1.4 - Les simulations d'une fibre multimodes	56
3.2 - Le couplage VCSEL-Fibre optique	59
3.2.1 - Le couplage modélisé et simulé avec CodeV	62
3.2.2 - Le couplage modélisé avec d'autres langages	65
3.3 - Le couplage Fibre optique-Photodiode	70
3.4 - Le générateur pseudo-aléatoire de bits	74
3.5 - La thermique	74
3.6 - La thermomécanique	77
3.7 - Le module complet	77
3.7.1 - La simulation avec Comsis d'un canal de transmission	77
3.7.2 - La simulation VHDL-AMS d'un canal de transmission	79
3.7.3 - La simulation VHDL-AMS du POM complet	88
3.8 - Conclusion	90
Chapitre 4 - Méthodologie de modélisation des systèmes complexes	91
4.1 - Le cycle de conception traditionnel en V	92
4.2 - Le Prototypage Virtuel	94
4.2.1 - Les objectifs du Prototypage Virtuel	94
4.2.2 - L'analyse et la conception "Top-down" des modèles	96
4.2.3 - La vérification "Bottom-up" des modèles	99
4.3 - L'intégration des collaborations dans le PV	101
4.4 - Conclusion	103
Conclusion et Perspectives	104
Conclusion	104
Perspectives	105
Bibliographie	106
Glossaire	110
Annexes	111
Annexe I : Modèle comportemental du POM	112

Annexe II : Modèles de couplage VCSEL-Fibre optique	114
Scripts CodeV	114
Modélisation Mathcad	125
Modélisation C/C++	127
Second modèle – Modélisation Mathcad	131
Annexe III : Modèle de Fibre optique multimode	134
Annexe IV : Modèle de connexion inter Fibre optique	137
Annexe V : Modèle de couplage Fibre optique-Photodiode	139
Annexe VI : Modèles de générateur pseudo-aléatoire	140
Annexe VII : Implémentation Simplorer du modèle de VCSEL	141
Annexe VIII : Implémentation Simplorer de la Photodiode	144
Annexe IX : Diagramme de l'oeil	146

Table des Figures

Fig. 01 - Evolution de la densité d'intégration des transistors dans les microprocesseurs [INT1] _____	10
Fig. 02 - Le démonstrateur SHAMAN [SHA1] _____	16
Fig. 03 - Coupe du module d'émission du démonstrateur [SHA1] _____	17
Fig. 04 - Coupe du module de réception du démonstrateur [SHA1] _____	17
Fig. 05 - Domaines d'actions et collaborations entre les partenaires du projet [SHA1] _____	20
Fig. 06 - Code VHDL-AMS et représentation graphique associée d'une résistance (Simplorer 6.0) _____	23
Fig. 07 - Code VHDL-AMS et représentation graphique d'un circuit RLC série (Simplorer 6.0) _____	23
Fig. 08 - Résultats de simulation VHDL-AMS du modèle comportemental (a), et de mesures (b) _____	25
Fig. 09 - Bloc diagramme type d'un modèle de composant [SHA1] _____	27
Fig. 10 - Bloc diagramme du module optoélectronique complet [SHA1] _____	27
Fig. 11 - Méthodologie SHAMAN [SHA1] _____	28
Fig. 12 - Représentation des domaines de validité de différents langages de modélisation [HER5] _____	33
Fig. 13 - Vue schématique d'un circuit multi-abstractions, multi-domaines dans Simplorer _____	42
Fig. 14 - Vue schématique d'un circuit multi-abstractions, multi-domaines dans SystemVision _____	43
Fig. 15 - Quelques aspects de l'interface graphique utilisateur _____	45
Fig. 16 - Exemples de symboles extraits des bibliothèques standard de Simplorer et natures ou domaines associés _____	46
Fig. 17 - Interface de conception textuelle des modèles _____	47
Fig. 18 - Schéma de fonctionnement de l'intégration de MATHCAD dans Simplorer [ANS4] _____	50
Fig. 19 - Communication entre VHDL-AMS et les différentes composantes d'un modèle C/C++ sous Simplorer _____	51
Fig. 20 - Interfaçage VHDL-AMS – Matlab/Simulink sous Simplorer _____	51
Fig. 21 - Transmission d'un signal de débit 1, 2.5, 5, 10 GBits/s dans une fibre GI de 300 m _____	56
Fig. 22 - Représentation graphique choisie pour la fibre optique et affichage des paramètres modifiables directement sur le graphique sous Simplorer _____	57
Fig. 23 - Interface de choix de l'architecture d'un composant à simuler _____	57
Fig. 24 - Transmission d'un signal de débit 1, 2.5, 5, 10GBits/s dans une fibre GI de 300 m _____	58
Fig. 25 - Diagramme bloc de la fibre multimodes _____	59
Fig. 26 - Vue tridimensionnelle du dispositif de couplage VCSEL - fibre optique _____	60
Fig. 27 - Couplages mesurés par THALES _____	61
Fig. 28 - Découpage du dispositif VCSEL-Fibre optique avec CodeV _____	63
Fig. 29 - Graphique Matlab de l'évolution du taux de couplage en fonction des variations suivant x et y à z fixé. Résultats issus des simulations CodeV _____	64
Fig. 30 - Comparaison des mesures expérimentales (a) de couplage à la simulation CodeV (b) _____	64
Fig. 31 - Représentation du plan de comptage des rayons transmis (cercle vert et cercle violet) par rapport à la VCSEL (cercle noir) et à la surface réfléchissante du biseau de la fibre (ellipse orange) _____	66
Fig. 32 - Représentation du Taux de couplage en fonction du défaut d'alignement transversal (μm) par simulation avec Mathcad _____	66
Fig. 33 - Résultat de simulation du couplage VCSEL-fibre en fonction de x et y à z fixé _____	68
Fig. 34 - Possibilités de devenir du signal lumineux (a. et b. sortent, c. entre s'il est réfléchi) _____	69
Fig. 35 - Résultats de ce modèle _____	69
Fig. 36 - Deux vues d'un diagramme de rayonnement de VCSEL expérimental (THALES) _____	72

Fig. 37 - Profil du faisceau lumineux après 5 m et 205 m de fibre optique GI [SHA1]	72
Fig. 38 - Illustration de l'évolution du diagramme de rayonnement en sortie de fibre optique	73
Fig. 39 - Evolution du spectre d'une VCSEL en fonction de la température (Supaero) [SHA1]	75
Fig. 40 - Influence de la température sur les niveaux de sortie du driver (ENST) [SHA1]	75
Fig. 41 - Modèle thermique du VCSEL [SHA1]	76
Fig. 42 - Modèle thermique du Driver [SHA1]	76
Fig. 43 - Schéma d'un canal de transmission du module dans Comsis	78
Fig. 44 - Résultats de simulation Comsis pour un canal (Courant en sortie du driver, Puissance optique en sortie du VCSEL, Puissance optique en sortie de la fibre, Courant en sortie du bloc de la photodiode) [SHA1]	78
Fig. 45 - Diagramme de l'œil en sortie de la chaîne [SHA1]	79
Fig. 46 - Schéma Simplorer du VCSEL et simulation non thermique avec bruit	81
Fig. 47 - Popt(I) mesurée par THALES (1) et simulée avec Simplorer (2) pour différentes températures	81
Fig. 48 - Schéma Simplorer et résultats de simulation du Driver	82
Fig. 49 - Schéma de la photodiode, sources de bruit associées et résultats de simulation	83
Fig. 50 - Circuit de test et résultats de simulation associés au TIA	84
Fig. 51 - Feuille de schéma Simplorer assemblant les composants d'un canal de transmission	85
Fig. 52 - Résultats de simulation du canal de transmission	87
Fig. 53 - Exemple de diagramme de l'œil du canal de transmission	88
Fig. 54 - Transformation d'un schéma en élément d'un système plus important	89
Fig. 55 - Feuille de schéma du module complet à trois canaux avec circuit thermique des VCSELS	89
Fig. 56 - Cycle de conception en V	92
Fig. 57 - Niveaux d'abstraction pouvant intervenir dans le processus de modélisation d'un système	95
Fig. 58 - Cycle idéal de conception par Prototypage Virtuel	96
Fig. 59 - Cycle récursif réel de conception par Prototypage Virtuel	100
Fig. 60 - Données fournies par le commanditaire (en bleu), dans le cadre du développement d'un modèle complet	102
Fig. 61 - Données fournies par le commanditaire (en bleu), dans le cadre du développement d'un composant manquant pour une modélisation complète (en rouge)	103
Fig. 62 - Dimensions caractéristiques et masques pour les diagrammes de l'œil	147

Table des Tableaux

<i>Tab. 01 -</i>	<i>Caractéristiques techniques du démonstrateur [SHA1]</i>	<i>18</i>
<i>Tab. 02 -</i>	<i>Dispersion modale dans les fibres multimodes SI et GI [SHA1]</i>	<i>54</i>
<i>Tab. 03 -</i>	<i>Bande passante de deux fibres de paramètres identiques, mais de profils différents</i>	<i>55</i>
<i>Tab. 04 -</i>	<i>Les différentes erreurs de placement de la barrette de VCSEL</i>	<i>60</i>
<i>Tab. 05 -</i>	<i>Diagramme d'intensité pour différentes longueurs de fibre</i>	<i>71</i>
<i>Tab. 06 -</i>	<i>Equations de couplage permettant de calculer les pertes</i>	<i>138</i>

Publications auxquelles je suis associé

En relation avec le projet SHAMAN

- [DES1] "SoC modelling for virtual prototyping with VHDL-AMS"
Patricia DESGREYS et al, FDL 2004, 13-17 septembre, Lille
- [KAR1] "Modèle VHDL-AMS d'un module optoélectronique : photorécepteur"
Mohamed KARRAY et al, JSF 2003, 20-22 décembre, Tozeur, Tunisie
- [SNA1] "Final User Oriented SOC Modeling"
Sébastien SNAIDERO et Yanick HERVE, DCIS 2004, 24-26 novembre,
Bordeaux
- [TOF1] "Multilevel Behavioral Simulation of VCSEL based Optoelectronic Modules"
Zeno TOFFANO et al, IEEE J. Sel. Top. Quant., Vol. 9 No. 3, mai/juin 2003

Extérieures au projet SHAMAN

- [HER5] "Sampling Rate and Quantification Density Control in VHDL-AMS"
Yannick HERVE et Sébastien SNAIDERO, BMAS 2002, 6-8 octobre, Santa
Rosa, Californie
- [MIL1] "Behavioural Library Development: Models Documentation and Qualification"
Noëlle MILET-LEWIS et al, FDL 2002, 24-27 septembre, Marseille

Introduction

Dans notre époque technologique et nomade, nous sommes perpétuellement à la recherche de systèmes électroniques de plus en plus puissants, de plus en plus miniaturisés et disposant de l'autonomie la plus importante possible. La complexité croissante des circuits intégrés (Integrated Circuit – IC) qui s'est opérée en conséquence sur les 30 dernières années a suivi avec plus ou moins de fidélité la "loi de Moore" (*figure 01*) qui veut que la densité d'intégration des composants double tous les 18 à 24 mois.

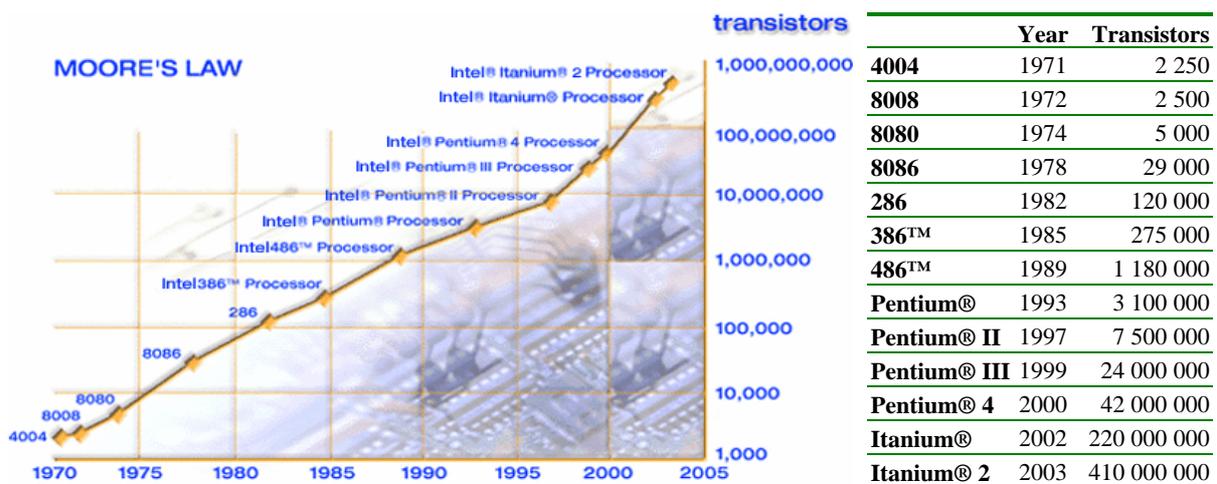


Fig. 01 - Evolution de la densité d'intégration des transistors dans les microprocesseurs [INT1]

Aujourd'hui, les systèmes ne sont plus exclusivement électroniques, mais font intervenir d'autres domaines technologiques comme l'optique ou la mécanique au sein d'une même puce (System on a Chip – SoC) ou dans un même boîtier (System in Package – SiP).

Hormis le fait que des circuits d'une telle densité – regroupés sous l'appellation VLSI (Very Large Scale Integration) – et d'une telle complexité ne sont plus réalisables "à la main", les contraintes des marchés actuels poussent à des temps de conception et de mise

sur le marché (Time to Market) toujours plus réduits. Les concepteurs de systèmes ont donc un grand besoin d'outils permettant un développement rapide et à moindre coût. Cette attente devrait trouver une réponse dans le "Prototypage Virtuel" (PV) des systèmes, en utilisant des langages de description de matériel (Hardware Description Language – HDL), qui permettrait de réduire à leurs minima les phases de test sur des prototypes physiques, très coûteuses en temps et en argent.

Ces langages de modélisation, dont les précurseurs sont apparus il y a plus de 30 ans, s'implantent aujourd'hui profondément dans le paysage de la conception industrielle des systèmes microélectroniques complexes et touchent de ce fait un public plus large qui ne dispose pas forcément des connaissances, du temps ou de l'envie, lui permettant de décortiquer les lignes de code hermétiques qui composent les modèles.

Il résulte de ces deux aspects, de nouveaux besoins quant à la manipulation des modèles dans les processus de maquettage virtuel des systèmes complexes. Il faut en effet assurer la formalisation, la capitalisation et la réutilisation des modèles développés. Capitalisation et formalisation des modèles, pour être capable de stocker dans des bibliothèques des composants généralistes et bien documentés autorisant la réutilisation intensive de ces derniers, même par des non spécialistes des HDLs, afin d'économiser du temps de développement en n'ayant pas à "réinventer la roue" à chaque nouveau système, mais à l'améliorer tout au plus.

Projet de recherche induit :

- **Dans cette perspective de productivité accrue et de vulgarisation d'emploi, on peut se demander si le langage VHDL-AMS et les outils qui lui sont associés permettent de mener à bien la tâche du prototypage virtuel des systèmes complexes ?**
- **Quelles sont alors la méthodologie et les compétences à mettre en oeuvre pour y aboutir ?**
- **Et quelles sont les briques manquantes dans le flot de conception qui peuvent l'entraver ou le ralentir ?**

Pour tenter de répondre à ces questions, nous avons utilisé le projet SHAMAN (Simulation Hybride pour Applications Multi-domaines Analogiques ou Numériques), commandité par la société THALES, comme support expérimental de la problématique développée. Ce projet, qui a également servi de support financier à la thèse, consistait en l'étude et la modélisation d'un SiP de communication optoélectronique à la pointe de la technologie.

En effet, ce module de communication optique sans dissipateur thermique [PEZ1] trouve naturellement sa place dans les dispositifs qui atteignent les limites de la connectique métallique et pour lesquels il est intéressant d'envisager les solutions de transport de données optiques. Malgré un surcoût lié à l'utilisation de la fibre par rapport à un câble, cela permet de s'affranchir de bon nombre de problèmes liés à la propagation d'ondes à hautes fréquences dans les milieux conducteurs [PEZ2].

L'étude d'un système comme celui-ci a nécessité le développement de modèles complexes, fortement multidisciplinaires, en utilisant la dernière génération d'outils de simulation dont la commercialisation a correspondu avec le début du projet. Cette démarche s'inscrit dans le cadre d'objectifs industriels concrets qui nécessitent une modélisation physique permettant l'étude et l'amélioration du dispositif, mais également une modélisation comportementale du fonctionnement "normal" du système, destinée à être insérée dans les systèmes plus conséquents des clients de l'entreprise.

Dans le cadre de cette étude, nous allons tout d'abord présenter de manière plus approfondie les tenants et les aboutissants du projet SHAMAN support principal de cette thèse. Vient ensuite la justification technologique et stratégique de l'emploi de VHDL-AMS accompagnée d'une courte présentation du langage et des outils associés, tant dans leurs possibilités que leurs limites. Dans un troisième volet, nous présentons les différents objets modélisés dans le cadre des objectifs du projet SHAMAN, pour conclure dans une quatrième partie concernant les aspects méthodologiques du développement de modèles complets et complexes comme ce fut le cas ici.

Objectifs et contexte de la Thèse [SHA1]

1.1 - Les motivations du projet SHAMAN

L'introduction des nouvelles technologies de traitement numérique et le développement des réseaux de transmission de l'information tels qu'Internet ou les LANs, dans des applications comme l'aéronautique et le spatial/défense, entraîne une forte augmentation des débits de communications entre les différents éléments et fonctions du système. Cette croissance des besoins en communication dans les environnements fortement contraints : vibrations, température, sensibilité aux rayonnements..., implique l'utilisation de la fibre optique comme support de transmission. La grande sensibilité des composants optoélectroniques du marché aux précédentes contraintes a, jusqu'à présent, considérablement ralenti l'introduction de ces technologies de communications dans ces secteurs industriels particulièrement conservateurs.

Un positionnement le plus rapide possible sur ce nouveau marché de la télécommunication très haut débit par fibre optique est donc un enjeu majeur des entreprises concernées. C'est le cas de la société D-LIGHT SYS issue du groupe THALES, productrice d'un module optique parallèle (Parallel Optical Module – POM) de télécommunication très courte portée (Very Short Reach – VSR) à haut débit.

Ces systèmes micro-optoélectromécaniques (Micro-Opto-Electro-Mechanical Systems – MOEMS) sont des modules technologiquement complexes dans lesquels cohabitent différentes familles de composants (optiques, mécaniques et électroniques) au sein d'un microsystème encapsulé dans un boîtier.

Il n'existait, jusqu'à récemment, aucun outil de simulation suffisamment fiable et complet pour permettre de simuler ces systèmes en prenant en compte les différents domaines technologiques utilisés, mais surtout les interactions entre ces domaines. L'utilisation des langages de modélisation modernes doit permettre d'intégrer au plus tôt les différentes contraintes environnementales et de fabrication, dans le flot de conception. Par ailleurs les dernières générations de logiciels de modélisation permettent d'envisager l'utilisation par un plus large public de ces modèles complexes.

Le projet SHAMAN a donc été motivé par le besoin de THALES (puis D-LIGHT SYS), et plus généralement des fabricants de MOEMS, de disposer de modèles permettant de prendre en compte les multiples effets pouvant interférer sur les performances de l'ensemble d'un système très tôt dans le cycle de conception et par la même d'en connaître et d'en réduire l'impact.

1.2 - Les objectifs du projet

Le but du projet SHAMAN était de développer une bibliothèque de modèles de composants optoélectroniques en VHDL-AMS *{IEEE 1076.1-1999}*, langage particulièrement bien adapté à la simulation de problèmes multi-domaines et qui n'avait encore été que très peu exploré pour les applications de transfert de données (Datacom). Ceci a fait de SHAMAN un projet novateur et ambitieux qui a donné lieu à un grand nombre de publications.

Les différents composants ainsi modélisés ont été associés dans un modèle global du système afin de permettre la simulation du module optoélectronique complet en y intégrant au maximum les contraintes inhérentes aux différents domaines, ainsi que les interactions entre ceux-ci.

Les modèles que nous devons concevoir avaient trois objectifs :

- tout d'abord, correspondre aux besoins des concepteurs et développeurs afin de leur permettre de mettre en évidence les forces et faiblesses des modules, mais également celles des techniques de fabrication (process) pour, notamment, identifier ceux où les tolérances de fabrications pourraient être relâchées dans un soucis principalement économique ;

- également, s'adresser aux industriels utilisant le module afin de leur permettre de simuler facilement l'intégration de ce dernier dans leurs systèmes ;
- enfin, être conçus de manière à être réutilisables au maximum pour envisager l'élaboration de la génération suivante de module de communication qui vise un débit 10 fois supérieur à celui du module actuel.

Pour ce faire, nous avons dû penser une nouvelle manière d'élaborer les modèles.

1.3 - Le démonstrateur

Le travail de modélisation du système n'a pas été un travail de conception partant de rien puisque le prototype du module aujourd'hui commercialisé existait déjà au début du projet. Notre travail a consisté à l'étudier et à le modéliser le plus finement possible.

1.3.1 - Structure du démonstrateur

Le démonstrateur SHAMAN (*figure 02*) fourni par THALES est un POM, comportant 8 canaux, destiné à des communications VSR (jusqu'à 300 m) à haut débit (1.25 Gb/s par canal) qui comprend une partie émission et une partie réception d'empaquetages identiques. L'originalité de ce dispositif repose sur le fait qu'il est le seul sur le marché actuel à proposer un système sans dissipateur thermique, réduisant ainsi, notamment, ses dimensions.

La partie émission est constituée d'une barrette de lasers à cavité verticale émettant par la surface (Vertical-Cavity Surface-Emitting Lasers – VCSELs) qui sont chargés de la conversion des signaux électroniques en puissances optiques. La barrette est associée à un circuit intégré dédié (Application-Specific Integrated Circuit – ASIC) qui assure l'alimentation des diodes lasers en courant. Le module d'émission comprend également un sous-ensemble optique et un ruban de fibres optiques multimodes, associé à son connecteur, servant de canal de propagation de l'information.

La partie réception comprend également un ruban de fibres multimodes avec son connecteur, une barrette de photodiodes PIN (Positive-Intrinsic-Negative) qui sont

chargées de la conversion des puissances optiques en signaux électroniques, un sous-ensemble optique ainsi qu'une électronique de réception (amplificateur transimpédance et amplificateur limiteur) chargée de l'amplification et de la remise en forme du signal.

Le sous-ensemble optique assure le couplage passif des fibres optiques multimodes avec les composants optoélectroniques d'émission et de réception (VCSELS et photodiodes).

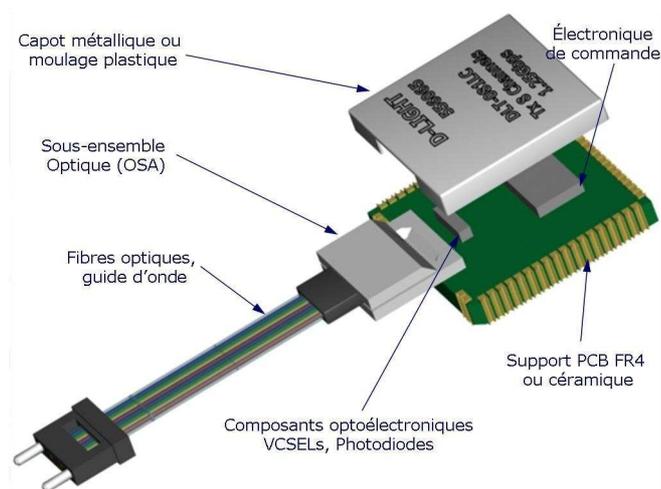


Fig. 02 - Le démonstrateur SHAMAN [SHA1]

Les domaines à prendre en compte dans la modélisation d'un tel composant sont :

- **L'électronique**

Avec, pour la partie émission, les drivers qui alimentent les VCSELS en courant et, pour la partie réception, l'amplificateur transimpédance et l'amplificateur limiteur chargés respectivement de l'amplification du signal après détection et de sa remise en forme.

- **L'optoélectronique**

Avec les VCSELS pour la partie émission, et les photodiodes pour la partie réception.

- **Les couplages optiques**

Même si les couplages optiques sont dus à la forme géométrique et au traitement de la fibre optique en ses extrémités, ces aspects font l'objet d'une modélisation indépendante de celle du transport de l'information dans la fibre optique elle-même. En effet, les conditions d'injection et d'extraction particulières de ce système nécessitent un traitement spécifique

plus complexe que l'on se propose d'étudier à part, de manière à donner au modèle de fibre optique une "réutilisabilité" maximale et augmenter son niveau d'abstraction.

- **L'optique**

Avec les fibres multimodes qui constituent le milieu de transmission de l'information.

- **La thermique et la mécanique**

Ces deux domaines traduisent les contraintes et les interactions qui existent entre les différents éléments qui constituent le module.

- **Les technologies de report et d'assemblage et les substrats utilisés**

Du fait que le module ne comporte pas de dissipateur thermique, ces aspects jouent un rôle important dans les transferts de la chaleur et les dilatations thermiques.

Les figures ci-dessous (*figures 03* et *04*) représentent des vues en coupe des modules d'émission et de réception et situent les contraintes qu'il sera impératif de modéliser :

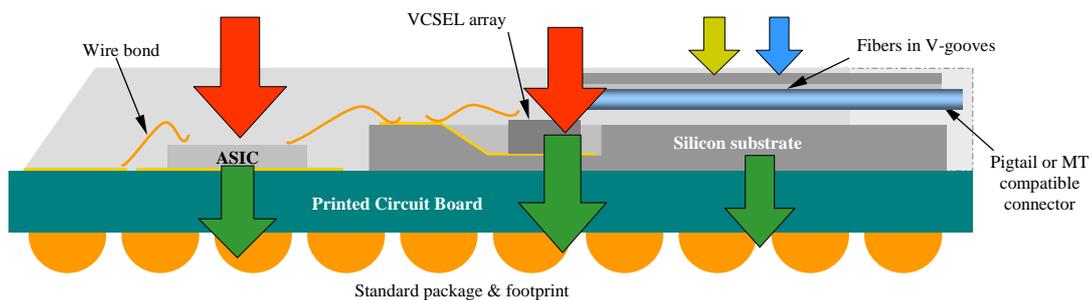


Fig. 03 - Coupe du module d'émission du démonstrateur [SHA1]

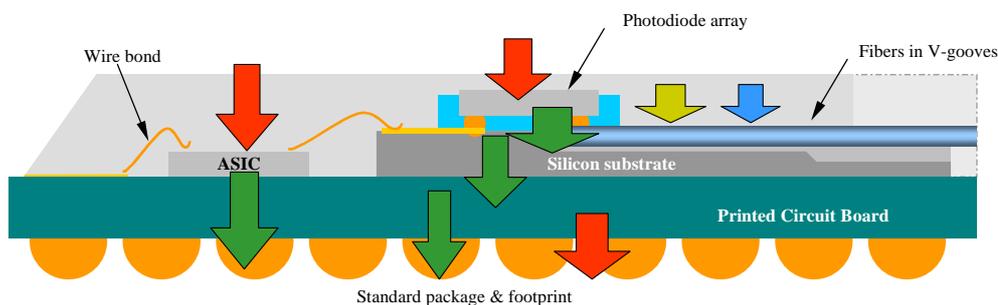


Fig. 04 - Coupe du module de réception du démonstrateur [SHA1]

➡ Electric ➡ Mécanique ➡ Thermique ➡ Optique

1.3.2 - Caractéristiques du démonstrateur

Le tableau suivant répertorie les grandeurs caractéristiques d'un module d'interconnexions optiques ainsi que les valeurs des standards optiques *{IEEE 802.3ae, ANSI SONET OC24/STM8}* appliquées au démonstrateur étudié dans le cadre de SHAMAN :

Paramètres	Symbole	Unité	Valeur
Nombre de canaux	N_{ch}	-	12
Longueur d'onde centrale	λ	Nm	840
Tension appliquée	V_{CC}	V	3.3
Température d'utilisation	T	°C	-40/85
Débit	D	Gbits/s	1.25
Taux d'erreur binaire	BER	-	$<10^{-12}$

Tab. 01 - Caractéristiques techniques du démonstrateur [SHA1]

1.4 - Le consortium

La modélisation des différents domaines et de leurs interactions au sein d'un MOEMS nécessitent des compétences multiples, c'est pourquoi SHAMAN a réuni des spécialistes des différentes disciplines représentées, à savoir, des physiciens, des spécialistes de la modélisation, de la simulation de systèmes optiques ainsi que du VHDL-AMS. Cette synergie entre les partenaires à la pointe dans leur domaine a permis de jeter les passerelles manquantes entre des disciplines relativement disjointes. Nous avons été trois doctorants ainsi qu'un post-doctorant à travailler sur ce projet dans le cadre de nos thèses, encadrés par une équipe de 7 spécialistes.

1.4.1 - Les partenaires

Comme précisé ci-dessus, le développement d'une bibliothèque de modèles VHDL-AMS pour la conception de MOEMS nécessite des compétences variées. Cette diversité se retrouve dans la complémentarité des partenaires impliqués dans le projet :

- **Le Laboratoire THALES Recherche et Technologie (THALES TRT)**

La section "Interconnexions optiques" de THALES TRT s'est attachée durant de nombreuses années à démontrer la faisabilité des systèmes communicants optiques pour les

applications embarquées civiles ou militaires, en proposant, sur la base de composants commerciaux, des systèmes adaptés.

La participation de THALES TRT a été primordiale en tant qu'utilisateur final et concepteur du démonstrateur. En effet, la mise à disposition des technologies et procédés de fabrication, ainsi que la fourniture d'un grand nombre de mesures et d'échantillons ou composants aux différents partenaires impliqués dans la modélisation physique, ont permis le développement de modèles physiques comportementaux simplifiés, représentatifs du fonctionnement des composants dans leurs environnements.

C'est aujourd'hui la société D-LIGHT SYS issue de THALES qui commercialise et fait évoluer le module sur lequel nous avons travaillé.

- **L'Ecole Supérieure d'électricité (Supelec) et l'Ecole Nationale Supérieure de l'Aéronautique et de l'Espace (Supaero)**

Ces deux partenaires spécialisés dans la modélisation physique de composants optoélectroniques ont fait intervenir leurs compétences dans les domaines de l'optoélectronique et de la thermique.

Supelec a travaillé sur les normes des composants optiques hauts débits, l'adaptation de leur modèle de propagation thermique à la barrette de VCSELs, ainsi que de mesures spectrales sur les VCSELs et la modélisation de leur comportement multimodes.

Supaéro a réalisé une étude sur la caractérisation et la modélisation optoélectronique des VCSELs et des photodiodes.

- **La société IPSIS**

La société IPSIS, coordinatrice du projet, est spécialisée dans la conception de logiciels de simulation aussi bien de traitement du signal que de communication. Elle a pris en charge l'intégration et la validation, au sein de son logiciel Comsis, des modèles physiques élaborés par les partenaires universitaires, avant leur implémentation en VHDL-AMS.

- **Le Laboratoire de PHysique et Applications des SEMi-conducteurs (PHASE) du CNRS et L'Ecole Nationale Supérieure des Télécommunications (ENST)**

Ces deux entités, spécialisées dans la modélisation VHDL-AMS et les méthodologies associées, ont pris en charge la modélisation multidomaines des composants et du module.

L'ENST s'est occupé de la modélisation des composants électroniques, tandis que le laboratoire PHASE a travaillé sur les composants optiques et les modélisations de l'ensemble du dispositif.

- **La société Mentor Graphics**

Elle a mis à disposition du projet son outil ADVance MS [MEN1]. Ce logiciel n'apportant, pour l'instant, pas de solution satisfaisante pour les modélisations globales et l'accessibilité aux non initiés, nous avons préféré utiliser un logiciel concurrent de la génération suivante: Simplorer 6.0 (ANSOFT) [ANS1].

Le graphique ci-après (*figure 05*) présente les relations de collaboration entre les différents partenaires du projet.

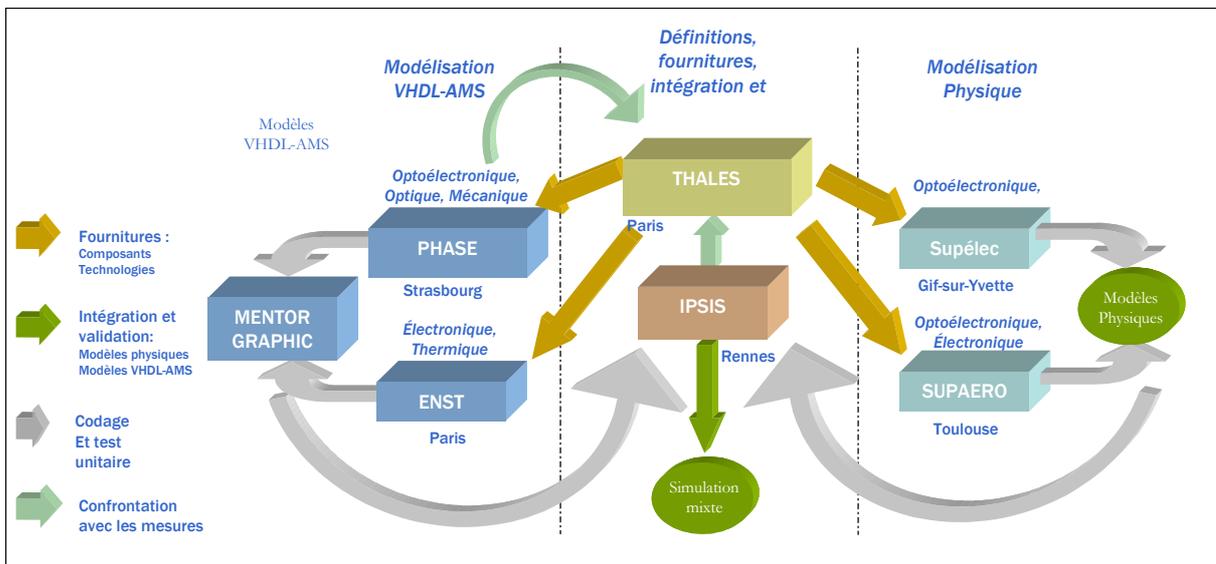


Fig. 05 - Domaines d'actions et collaborations entre les partenaires du projet [SHA1]

1.4.2 - L'organisation du projet

Le projet SHAMAN s'est déroulé sur deux ans et a été globalement coordonné par IPSIS. Des réunions entre tous les partenaires ont été organisées régulièrement tout au long du projet afin de confronter les idées, les mesures et les résultats de chacun.

Au démarrage du projet, il a été constaté que les partenaires venaient d'origines diverses, donc de cultures différentes, ce qui entraînait quelques soucis de divergence dans l'interprétation du vocabulaire. Au cours des six premiers mois, des réunions mensuelles ont donc été organisées afin de bien définir les concepts utilisés et de partir sur des idées communes. Par la suite, une réunion tous les deux mois s'est avérée suffisante. C'est à cette époque que je suis entré dans le projet.

La première phase du projet a consisté à définir les besoins, en particulier du point de vue de THALES qui représente l'utilisateur final et qui est donc le plus à même d'exprimer les attentes d'un concepteur de micro-système en terme de simulation de MOEMS. Tous les partenaires ont pris part à cette phase importante du projet. En effet, il était primordial de pouvoir identifier, dès le début du projet, les problèmes physiques dus aux contraintes sur le démonstrateur et de définir les grandeurs caractéristiques devant être obtenues à partir de la modélisation.

Une fois les composants à modéliser identifiés au sein du démonstrateur, les premières études physiques ont pu démarrer dans le but d'étudier l'influence des différentes contraintes sur les grandeurs de sortie des composants. Parallèlement à ces études physiques, un grand nombre de mesures sur les composants du module ont été réalisées par les différents laboratoires afin, non seulement de caractériser les composants, mais également de quantifier certaines contraintes et de valider les modèles développés par la suite.

Ces phases achevées, nous avons pu implémenter les modèles dans Comsis puis les valider avant de les implémenter en VHDL-AMS. La phase finale du projet a consisté à assembler les différents modèles ainsi développés pour effectuer une simulation du module complet.

1.5 - L'analyse de la problématique

Le projet SHAMAN a été amené par THALES pour répondre à un besoin de modélisation de son dispositif expérimental afin d'en optimiser le fonctionnement et les techniques de fabrication, mais également dans la perspective de fournir à ses clients un modèle simple et rapide du POM qu'ils puissent inclure dans les simulations de leurs systèmes.

Au-delà du projet SHAMAN, cette étude et la démarche qui l'accompagne sont susceptibles de répondre aux questions et aux besoins de beaucoup d'entreprises. Cette application industrielle de la modélisation des systèmes multidisciplinaires tenant sur une puce (System on a Chip – SoC) a permis de réfléchir et de réaliser des tests sur les techniques et les besoins de la modélisation adaptée aux nouveaux utilisateurs et aux nouvelles possibilités des logiciels [SNAI].

Effectivement, la pénétration dans l'industrie des langages de modélisation en a démocratisé l'utilisation. De ce fait, de nouveaux utilisateurs de la modélisation sont apparus. Ce ne sont pas des spécialistes des langages et ils n'ont probablement ni le temps ni l'envie de se plonger dans l'apprentissage de langages tels VHDL-AMS alors que des experts peuvent créer, si ce n'est déjà fait, les modèles des composants dont ils ont besoin. On voit ici tout l'intérêt que peuvent revêtir les bibliothèques de modèles que tentent de créer et de démocratiser des groupes comme l'association BEAMS [MILI].

Le partage des connaissances au travers de modèles de composants disponibles dans une bibliothèque implique que leur conception soit adaptée à cette optique. En effet, il ne s'agit plus ici, pour les concepteurs, de créer un modèle dédié à une application spécifique, mais d'essayer dès sa genèse de penser à son ouverture et à sa "réutilisabilité".

Ceci passe tout d'abord par une écriture documentée des modèles (notamment au moyen d'un document texte pouvant être inséré dans le modèle ou l'accompagner et par l'utilisation systématique de commentaires), afin de les rendre compréhensibles et modifiables aisément, mais aussi et surtout par une conception ayant recours à des paramètres accessibles aux utilisateurs, le plus souvent issus des fiches techniques (datasheet), par opposition aux paramètres physiques souvent connus des seuls fabricants.

Cette approche n'est possible que dans le cadre de logiciels adaptés. Quoi de plus rebutant pour un utilisateur non spécialiste qu'une page remplie de lignes de code dont il n'est pas en mesure de saisir le sens. Dans un souci de convivialité et de simplicité et au vu des environnements de travail informatique actuels, la solution la plus appropriée apparaît comme une interface graphique cachant, derrière une représentation de type "boîte noire", le code qui réalisera sa fonction.

La **figure 06** illustre le contraste entre l'aspect hermétique du code représentant une simple résistance et celui, beaucoup plus avenant, du symbole illustrant sa fonction. Ce dernier est une représentation standardisée qui est familière à tout électronicien.

```

-----
-- Copyright (c) 2002 by ANSOFT Corp. All rights reserved. --
-----

LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;

ENTITY R IS
  GENERIC (R : REAL := 1.0e+3);
  PORT    (TERMINAL p,m : ELECTRICAL);
END ENTITY;

ARCHITECTURE behav OF R IS
  QUANTITY v ACROSS i THROUGH p TO m;
BEGIN
  v == i*R;
END behav;

```

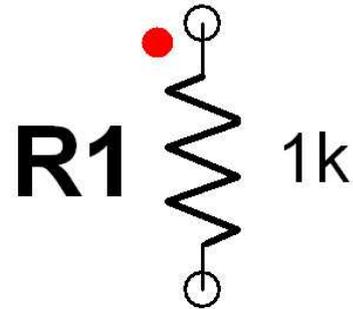


Fig. 06 - Code VHDL-AMS et représentation graphique associée d'une résistance (Simplorer 6.0)

De cette manière, l'utilisateur non spécialiste des langages pourra simplement agencer des boîtes graphiques les unes avec les autres de manière à représenter au mieux la fonction du système qu'il souhaite simuler sans avoir à établir le code correspondant (**figure 07**).

```

-----
-- Exemple d'instanciation de composants - circuit RLC --
-- On suppose que les composants existent déjà dans la --
-- bibliothèque de travail "work" --
-----

LIBRARY IEEE;
USE IEEE.ELECTRICAL_SYSTEMS.ALL;

ENTITY RLC IS
  GENERIC (R_val : REAL := 1.0e+3;
          L_val : REAL := 1.0e-6;
          C_val : REAL := 1.0e-6);
  PORT    (TERMINAL Tp,Tm : ELECTRICAL);
END ENTITY;

ARCHITECTURE behav OF RLC IS
  TERMINAL Ti1,Ti2 : ELECTRICAL;
BEGIN
  ENTITY work.resistor(beh)
    GENERIC MAP (R_val)
    PORT MAP (Tp, Ti1);

  ENTITY work.inductance(beh)
    GENERIC MAP (L_val)
    PORT MAP (Ti1, Ti2);

  ENTITY work.capacitor(beh)
    GENERIC MAP (C_val)
    PORT MAP (Ti2, Tm);

END behav;

```

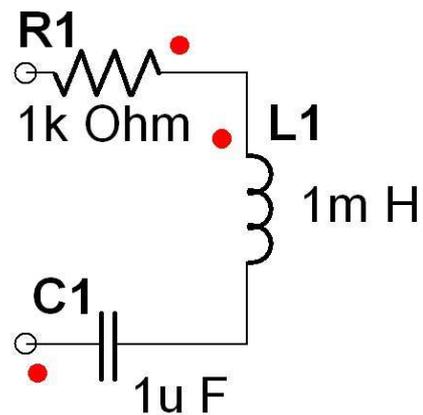


Fig. 07 - Code VHDL-AMS et représentation graphique d'un circuit RLC série (Simplorer 6.0)

La plupart des langages étant multidisciplinaire, il est important que cette diversité apparaisse d'une manière graphique et que l'intégrité des connexions puisse être vérifiée par le logiciel de simulation. Une autre difficulté de ce travail de vulgarisation repose sur la non-unicité des langages de modélisation dont l'utilisateur n'est pas supposé avoir connaissance. L'interface idéale devrait être en mesure de connecter ces différents langages de manière transparente pour l'utilisateur. Devant la complexité extrême de la tâche, ceci n'est que partiellement réalisé dans les simulateurs actuels.

Si ce type de représentation sied à la construction d'un système complet par boîtes noires, il est très difficile de l'utiliser pour la conception des composants eux-mêmes. En effet, tout élément n'est pas forcément décomposable en entités déjà conçues : enrober un composant par une couche graphique est quasiment toujours possible, mais rendre toute la richesse des langages par le graphique apparaît impossible. Comme dans beaucoup de domaines, le niveau d'abstraction supérieur offert par la conception graphique ôte une part de la richesse et de la souplesse des langages complexes. C'est pourquoi il est primordial de conserver une interface de type texte qui permette la conception des primitives ou des modules les plus complexes au sein du logiciel de simulation et leur association à des représentations graphiques.

Afin d'exploiter toute la richesse des HDLs, il est également très important que le jeu d'instruction des langages soit implémenté de la manière la plus complète possible. Cette dernière n'est que partielle dans les logiciels actuels où certaines instructions sont manquantes. Suivant les moteurs de simulation, chaque logiciel a ses propres lacunes, et aucun n'implémente, par exemple, la totalité d'un langage comme VHDL-AMS, pourtant normalisé. Il est probable qu'il faille attendre une nouvelle génération de moteurs de simulation pour voir les méta-simulateurs être capables de traiter et d'interconnecter l'ensemble des jeux d'instruction des principaux langages de modélisation disponibles.

Dans le projet SHAMAN, deux types de modèles ont été demandés par THALES. Chacun d'eux possède un niveau d'abstraction et une conception différente. En effet, pour répondre aux besoins de simulation rapide des industriels, nous avons besoin d'un modèle *descriptif* de haut niveau d'abstraction alors que les modèles de développement requièrent une conception *prédictive* s'associant sur plusieurs niveaux d'abstraction selon les objets [HER3].

1.5.1 - Le modèle industriel descriptif

Le tout premier travail de modélisation dans SHAMAN, alors que je n'étais pas encore intégré au projet, a été réalisé par le Dr Yannick HERVE (PHASE) et a consisté à établir un modèle purement comportemental basé sur la seule norme du module.

Il s'agit d'une modélisation de très haut niveau, puisque les spécifications du module ont été écrites sous forme d'équations afin de modéliser le comportement du module complet sans tenir compte de la réalité physique de l'implantation réelle dans les équations. Par essence, ce modèle est purement descriptif [DESI], puisqu'il utilise les données et mesures d'entrées/sorties ainsi que les spécifications des fiches techniques afin de créer un modèle du composant comme s'il s'agissait d'une "boîte noire" au contenu inconnu.

Il est important de noter qu'à ce stade de la modélisation aucune étude structurelle du module n'a été réalisée, et qu'il s'agit d'un modèle comportemental global. Ce modèle constitue le modèle simulable de référence pouvant être distribué pour les simulations systèmes aux utilisateurs du module.

Ce modèle permet de tracer le diagramme de l'œil du système, comme sur la *figure 08* :

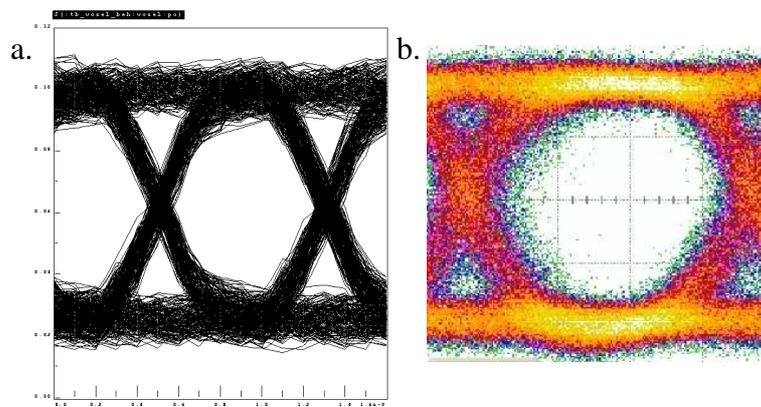


Fig. 08 - Résultats de simulation VHDL-AMS du modèle comportemental (a), et de mesures (b)

Le code VHDL-AMS associé au modèle comportemental du module complet est donné en *Annexe I*.

1.5.2 - Le modèle prédictif de développement

Cette modélisation destinée au personnel responsable de la conception et du développement du module a pour objectif principal d'obtenir des modèles suffisamment

fins et complets pour permettre la simulation précise de l'influence de nombreux paramètres sur les performances du module. Cette démarche peut s'inscrire dans le cadre du développement d'une bibliothèque de modèles de composants optoélectroniques pour la simulation de MOEMS.

Le besoin de prévoir avec précision les performances des composants du système nécessite une approche physique de la modélisation et la mise au point de modèles prédictifs [DESI]. En effet les modèles développés ici sont basés sur les équations de la physique des composants et utilisent leurs paramètres internes au niveau d'abstraction que l'on peut implémenter.

Pour ce faire, nous avons besoin de réaliser des études préliminaires du module, afin d'en extraire les blocs fonctionnels associés aux composants du module dans le but de planifier et d'attribuer les tâches au sein des groupes les plus qualifiés.

- **Etude structurelle**

Il s'agit ici d'étudier la structure du module existant afin d'identifier les composants, les grandeurs et les interactions à modéliser.

Les composants identifiés dans le module, et donc modélisés dans SHAMAN, sont :

- Les VCSELS et les photodiodes pour les composants optoélectroniques
- Les fibres multimodes et les couplages pour les composants optiques
- Les drivers, les amplificateurs transimpédance et limiteur pour les composants électroniques.

Les principales grandeurs modélisées sont les grandeurs électriques et optiques qui traduisent le fonctionnement du MOEMS, mais aussi les grandeurs thermiques et mécaniques, qui traduisent les contraintes et interactions qui existent entre les différents éléments constitutifs du MOEMS.

Il a été également important de prendre en compte les technologies de report et d'assemblage (flip chip, brasure ou collage) et les différents substrats (céramique, verre époxy, silicium) notamment pour l'étude de la thermique.

Le modèle de chaque composant du MOEMS disposera de grandeurs d'entrée appartenant à un ou plusieurs des domaines précédemment cités et d'un certain nombre de paramètres

intrinsèques au composant pour simuler les grandeurs de sortie du modèle pouvant appartenir elles aussi aux différents domaines.

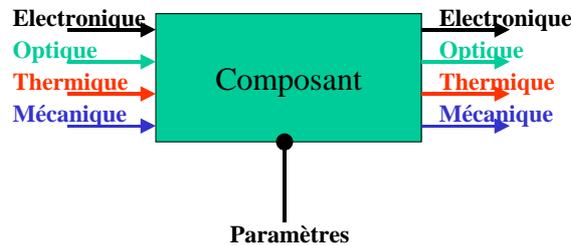


Fig. 09 - Bloc diagramme type d'un modèle de composant [SHA1]

Ainsi, chaque composant a pu être représenté par un bloc diagramme faisant apparaître les différentes informations entrant en jeu dans la modélisation (*figure 09*):

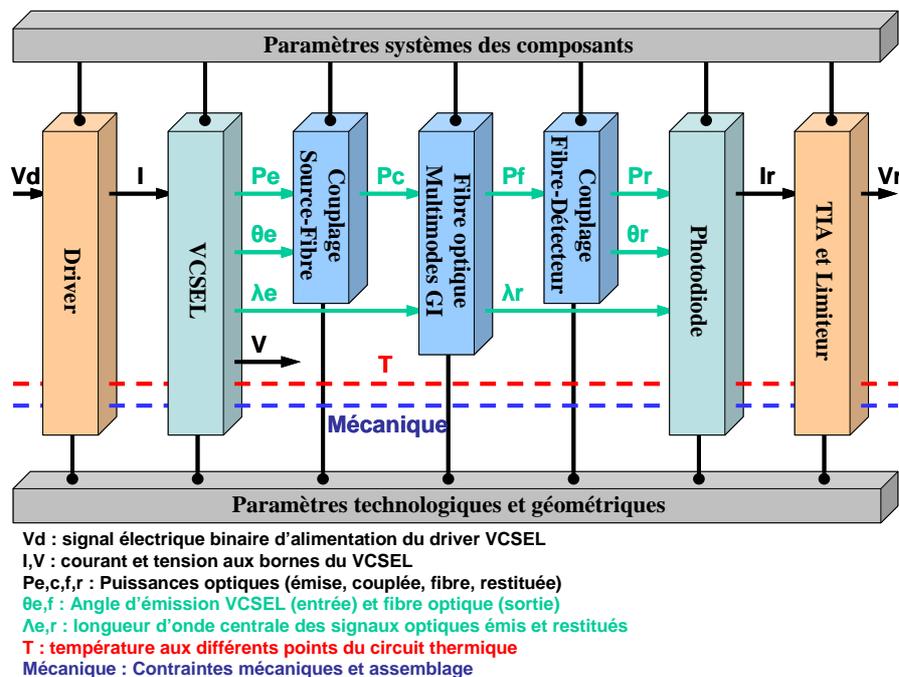


Fig. 10 - Bloc diagramme du module optoélectronique complet [SHA1]

Nous pouvons, selon ce principe, représenter l'ensemble du module SHAMAN (*figure 10*). Cette représentation permet de mettre en évidence non seulement les différents composants à modéliser (Drivers, VCSELs, fibres optiques, photodiodes, couplages et amplificateurs)

mais également leurs grandeurs d'entrées/sorties et donc les domaines à prendre en compte à chaque niveau du module.

● Etude Physique

Un certain nombre d'études physiques préliminaires, menées de façon théorique ou avec l'appui d'outils informatiques spécialisés (CodeV, Matcad, ADS) ont ensuite été nécessaires pour permettre d'établir les modèles de bas niveaux basés sur les équations fondamentales des composants ou les modèles plus abstraits ayant des performances suffisantes pour l'objectif visé.

La *figure 11* expose de façon schématique la méthodologie adoptée au cours du projet SHAMAN :

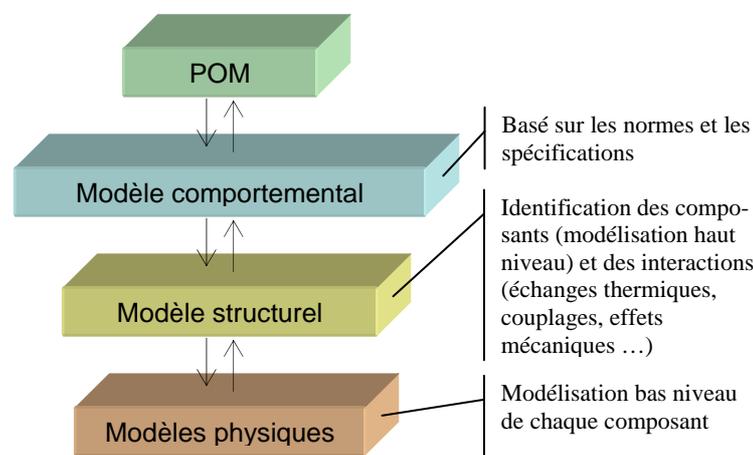


Fig. 11 - Méthodologie SHAMAN [SHA1]

Cette méthodologie présente plusieurs avantages : le module peut être simulé entièrement à plusieurs niveaux d'abstraction. Ainsi, en fonction du phénomène étudié, tous les composants n'ont pas besoin d'être instanciés au plus bas niveau, permettant ainsi de réduire les temps de simulation.

Une fois les résultats de simulation du module au plus bas niveau d'abstraction cohérents avec le cahier des charges, il est possible de modifier les caractéristiques d'un composant, une technologie ou un couplage afin d'en étudier les répercussions sur le module complet. La nouvelle structure étant fixée et les technologies choisies, les résultats de simulation

globaux du module permettent de réécrire un nouveau modèle de haut niveau purement descriptif étant à même de servir pour les simulations au niveau du système complet.

Langages de modélisation et simulateurs

2.1 - Le choix du langage de modélisation

Après avoir déterminé, grâce aux études du POM, les étapes et les niveaux de modélisation nécessaires, il nous a fallu déterminer le langage de modélisation le mieux adapté à nos besoins. Les langages susceptibles d'apporter une réponse aux problèmes de modélisation sont nombreux, mais peu répondent à nos attentes.

2.1.1 - Les langages de programmation orientés objet

Ces langages tels C++, VisualBasic ou Java sont des langages évolués aux possibilités quasi-infinies du fait même qu'ils sont les briques de base pour la construction des applications. Les simulateurs en faisant partie, il est tout à fait possible d'utiliser ces langages à des fins de simulation.

Cependant, puisque des simulateurs performants existent déjà, et étant donné qu'il est souvent possible de faire interagir les HDL avec le langage C++, il n'apparaît pas nécessaire d'en faire usage en tant que simulateur. De plus, notre objectif étant la rationalisation de l'utilisation des simulateurs et non leur optimisation il n'est, d'une part, pas nécessaire et, d'autre part, pas dans nos capacités de créer un nouveau simulateur.

2.1.2 - Les langages de modélisation numérique

On retrouve dans cette catégorie les langages comme VHDL *{IEEE 1076-2000}*, Verilog *{IEEE 1364-2001}*, SystemC *[LRM4]*, SpecC *[LRM5]* et SystemVerilog *[LRM6]*. Ces

langages très efficaces sont aujourd'hui associés à des outils qui permettent, moyennant une utilisation appropriée, la synthèse numérique de dispositifs complexes et le co-design pour les plus évolués. Ces langages sont inadaptés à notre projet qui a besoin de simulation analogique et multidisciplinaire.

2.1.3 - Les langages mathématiques formels explicites

Un langage comme Matlab [*MAT1*] associé à Simulink permet de proposer aux utilisateurs une représentation graphique sous forme de boîtes noires pouvant être connectées les unes aux autres et pouvant avoir des fonctions de transfert très complexes grâce aux possibilités mathématiques très poussées de Matlab. Dans cette conception de la modélisation, la gestion des équations implicites au sein d'un circuit (lois de Kirchhoff aux nœuds du circuit) n'est pas assurée. De ce fait, la conception d'un système optoélectronique complet à l'aide de Matlab/Simulink n'est pas envisageable simplement.

2.1.4 - Les langages de modélisation implicites dédiés à l'électronique

Dans cette famille incluant les langages comme SPICE, il n'est question que de modélisation électrique analogique. Il s'agit alors pour les concepteurs de modèles multi-domaines de transférer l'ensemble de leur dispositif dans le domaine électrique par analogie. Cette démarche est courante, mais peu pratique lorsque l'on sait que d'autres logiciels permettent de travailler directement dans les différents domaines de la physique. De plus, ce langage n'est pas capable d'assurer seul la modélisation des systèmes mixtes, ce qui n'est pas le cas de certains de ses concurrents. Par ailleurs, les composants instanciables dans SPICE doivent se trouver dans le noyau, cela implique une recompilation du noyau à chaque nouveau modèle ce qui est peu pratique et en complète contradiction avec les objectifs de "réutilisabilité" des modèles développés.

2.1.5 - Les langages de modélisation analogique multi-domaines

Cette famille contient deux langages aujourd'hui supplantés par leurs "successeurs" : HDL-A (qui s'est trouvé redirigé vers VHDL-AMS [*LRM1*]) et Verilog-A (qui a dû également s'orienter vers une technologie de simulation mixte sous la concurrence de VHDL-AMS avec Verilog-AMS [*LRM3*]).

Il existe cependant dans cette famille de langages le MAST [SYNI]. A l'origine propriétaire de Analog, passé depuis dans le domaine public, c'est un langage aux bases solides. En effet, depuis sa création en 1986, ce langage de simulation multi-domaines, mais uniquement analogique, a su conquérir les industries n'ayant pas recours au numérique, comme en mécanique ou en électronique de puissance.

Cependant, l'avènement des technologies numériques provoque de plus en plus d'incursions dans le domaine analogique (pour des structures de contrôle par exemple). De ce fait, les logiciels de simulation associés au langage MAST comme SABER incluent le support et l'interfaçage avec des langages comme VHDL [SYNI].

Notre projet reposant sur un système multi-domaines mixte, il ne semble pas très intéressant de devoir utiliser ce genre de langage propriétaire alors que certains sont conçus et normalisés pour les systèmes mixtes.

2.1.6 - Les langages de modélisation mixte multi-domaines

On retrouve ici les langages Verilog-AMS [LRM2] et VHDL-AMS [LRM1]. Ces langages dont les bases numériques se trouvent environ 20 ans en arrière, ont bénéficié récemment (1998 pour Verilog-AMS et 1999 pour VHDL-AMS) d'extensions pour les signaux analogiques et mixtes (Analog and Mixed-Signal – AMS). Comme leurs noms l'indiquent, ces langages permettent de traiter indifféremment des modélisations logiques, analogiques ou mixtes au sein d'un même composant ou système. Par ailleurs, la philosophie de conception de ces langages et leurs jeux d'instructions en font des langages intrinsèquement multi-domaines qui gèrent les équations implicites liées au fonctionnement d'un circuit.

Enfin, la richesse de ces deux langages et les méthodes d'instanciation de composants en font des outils permettant d'aborder la modélisation sous plusieurs angles et d'atteindre plusieurs niveaux d'abstraction, jusqu'au sein d'un même modèle avec VHDL-AMS, en fonction des besoins des concepteurs.

La *figure 12* présente les domaines pour lesquels plusieurs langages sont utilisables.

	Level/Domain	Solid mechanic	Point mechanic	Multiphysic e.g optic, thermal	Analog electronic	Digital electronic	Control	Software	
Continuous & discrete time simulation	Specification level			VHDL-AMS					
	Behavioral level				SPICE	VHDL	MATLAB Simulink	SystemC	
	Architectural level								
	Component level	CAD Mesh simulation							
CAD	Physical model	CAD/CFM (e.g. Catia)			Electronic CAD tools (e.g. PCB, IC masks)				
	Components examples	Motor CAD, transformer, electro-magnetic field CAD	Airbag sensor, CAR movement	Coil, Magnetic head, heat-pressure sensor	Antenna, Microphone, transistor, power amplifier	Processor, Digital IC, DSP	Finite state machine, signal filters	Embedded software, Hardware-software co design	
	Applications	Car, Computer Hard Disk			Mobile phone				
		Smart label, integrated sensors					Network		
		CAR servo-electronic suspension, ABS							
		Solid state gyroscope			Cockpit simulation				

Fig. 12 - Représentation des domaines de validité de différents langages de modélisation [HER5]

Nous avons finalement opté pour VHDL-AMS au détriment de Verilog-AMS pour différentes raisons :

- Tout d'abord, Verilog-AMS permet d'optimiser les codes à un niveau d'abstraction plus bas, par l'utilisation de primitives plus proches des circuits, alors que VHDL-AMS autorise une meilleure vision au niveau du système [PECI]. Etant donné que nous avons besoin d'un niveau d'abstraction élevé pour le projet SHAMAN, afin de pouvoir modéliser le POM complet, VHDL-AMS semble plus adapté.
- Ensuite, l'offre logicielle en outils graphiques de dernière génération se porte principalement sur VHDL-AMS avec les outils développés par Ansoft (Simplorer 6.0 et 7.0) et Mentor Graphics (SystemVision).

- Finalement, il y a également une notion d'habitude, de formation et de convenances personnelles qui font que les concepteurs de modèles associés au projet SHAMAN sont des spécialistes de VHDL-AMS et non de Verilog-AMS.

2.2 - VHDL-AMS [HER1, CHR1, PET1]

2.2.1 - Histoire de VHDL-AMS

Né en 1987 du langage ADA (1979) [LRM3], VHDL {IEEE 1076-1987} est à l'origine un langage de modélisation de systèmes numériques. Complété et enrichi en 1993 {IEEE 1076-1993 et IEEE 1164-1993} et en 2000 {IEEE 1076-2000}, ce langage est basé sur une simulation événementielle, et non temporelle, des systèmes.

Avec les extensions {IEEE 1076.3-1997, IEEE 1076.4-1995 et 1076.6-1999} de VHDL et les outils qui y sont progressivement associés, les concepteurs de modèles arrivent jusqu'à la synthèse logique de circuits numériques à partir de simples morceaux de code, écrits de manière adéquate, décrivant les fonctions d'un système.

Alors que dans le même temps, d'autres langages, tels MAST [SYN1] (1986) et SPICE (1969), permettent de gérer des modélisations principalement analogiques, notamment dans les domaines de la mécanique et de l'électronique de puissance, il n'existe pas de langage permettant de modéliser les systèmes mixtes (numériques et analogiques) qui émergent avec l'apparition de l'électronique numérique de commande.

Après une réflexion d'une dizaine d'années associant chercheurs et industriels, une extension dans le domaine analogique et mixte de VHDL est présentée en 1999 : VHDL-AMS [LRM1].

2.2.2 - Les avantages de VHDL-AMS

Lorsque VHDL-AMS a été créé, il existait de nombreux langages de conception propriétaires pour chaque fondeur ou fournisseur. Ceci était un obstacle à la communication entre domaines scientifiques et posait de graves problèmes aux sociétés lorsqu'un intervenant de la chaîne venait à disparaître ou à être remplacé, car le portage des

modèles était alors très délicat et nécessitait de nombreuses heures de travail supplémentaires.

VHDL-AMS est quant à lui un produit non propriétaire et normalisé par l'IEEE qui tend à être reconnu par le plus grand nombre. L'utilisation généralisée de ce langage facilite la communication entre les différents domaines scientifiques grâce à son approche multi-domaines native qui permet aussi bien à un électronicien qu'à un mécanicien ou même un chimiste [PECI] de modéliser la partie d'un dispositif qui le concerne directement sans problèmes de dialogue avec les autres parties.

La normalisation non unilatérale du langage, basée sur une étude de plusieurs années alliant industriels et chercheurs, a permis de généraliser son utilisation et de rendre les projets indépendants des fournisseurs de logiciels choisis, ce qui est primordial pour les entreprises qui étaient extrêmement dépendantes de ce lien auparavant.

La grande force de ce langage est de permettre la simulation mixte en autorisant aussi bien les modélisations à temps continu (analogiques) qu'à événements discrets (logiques) ou mélangeant les deux. A cette flexibilité d'emploi s'ajoute la possibilité pour les concepteurs d'aborder leurs modèles à différents niveaux d'abstraction. En effet, VHDL-AMS propose des mécanismes permettant de gérer aussi bien les abstractions comportementales (c'est la fonction réalisée par le système qui est modélisée et non sa physique), que les abstractions structurelles (le système est divisé en sous-ensembles qui peuvent eux-mêmes être modélisés au moyen de différentes abstractions, ...) ou bien de type work-flow (enchaînement de blocs fonctionnels dont les entrées n'ont pas d'influence sur les sorties des blocs précédents). Les modèles créés avec VHDL-AMS peuvent donc aussi bien être descriptifs que prédictifs.

Deux autres caractéristiques de VHDL-AMS, dont chacune est partagée avec quelques autres langages, sont le traitement des équations implicites (c'est-à-dire des équations où l'inconnue ne se trouve pas forcément dans le membre de gauche) et à travers celles-ci, l'utilisation des lois de Kirchhoff généralisées, fondement des relations implicites entre les différents noeuds d'un système.

2.2.3 - Les limites de VHDL-AMS

VHDL-AMS n'est cependant pas à même de proposer des instructions répondant à tous les besoins des concepteurs de modèles. Par exemple, l'utilisation des dérivations spatiales

n'est pas prévue par le langage, ce qui rend délicat les modélisations géométriques. Seules les dérivations temporelles sont acceptées par VHDL-AMS.

Par ailleurs, même si le langage est à même de supporter des palliatifs à ses manques grâce à ses possibilités d'interfaçage avec d'autres langages (notamment le C/C++), la forme de ces interfaces n'est pas standardisée. De ce fait, les modèles ayant recours à des langages extérieurs à VHDL-AMS ne sont généralement pas portables.

Même si VHDL-AMS laisse à l'utilisateur la possibilité de définir ses propres natures, il n'offre pas d'alternative possible à la sémantique de connexion faisant intervenir les lois de Kirchhoff généralisées. Cela devient un handicap lorsque l'on veut traiter d'autres systèmes de relations physiques. Il n'est, par exemple, pas possible de traiter la propagation des ondes électromagnétiques au moyen des terminaux, car les règles associées à cette propagation ne vérifient pas les lois de Kirchhoff.

Enfin, le fait que les simulateurs actuels soient basés sur des extensions et des modifications d'anciens simulateurs, et pas encore sur de nouvelles techniques de simulation spécifiques, implique des limitations dans les possibilités de simulation qui empêchent l'implémentation de certaines instructions du langage. C'est le cas de l'instruction PROCEDURAL qui pourrait nous permettre de simuler un bloc d'instructions séquentielles à chaque ASP par exemple. La méthode de résolution matricielle des simulateurs empêche quant à elle d'implémenter une instruction comme DISCONNECT qui permettrait de retirer un élément du système, ce qui modifierait dans le même temps la structure de la matrice de calcul. Or cette opération ne peut être refaite en cours de simulation.

2.2.4 - Les évolutions liées à VHDL-AMS [LRM1]

Avec la réaffirmation de la normalisation VHDL-AMS prévue en 2004, et son évolution future, une partie des problèmes présentés ci-dessus devrait trouver une solution. Par ailleurs, le langage verra ses possibilités prochainement étendues grâce à l'extension Radio Fréquence/Micro Ondes (Radio Frequency/ Micro Waves - RF/MW) en cours de développement. Des pistes complémentaires sont également étudiées pour élever le niveau d'abstraction des connexions, afin d'autoriser l'utilisation de différentes classes de signaux sur un même dispositif de connexion en fonction du niveau d'abstraction du modèle. Il est

également question d'incorporer des primitives SPICE à VHDL-AMS et d'y adjoindre des mécanismes automatiques de spécification et de vérification des modèles.

Par ailleurs, avec les possibilités grandissantes offertes par la nouvelle génération de simulateurs VHDL-AMS, présentée ci-après, il est légitime de se demander si les efforts de recherche pour développer des simulateurs spécialement dédiés à la simulation analogique supportant le jeu complet d'instructions VHDL-AMS seront encouragés. En effet, des logiciels comme Simplorer permettent de pallier à beaucoup de problèmes, comme l'absence du PROCEDURAL, par des moyens détournés comme l'utilisation de modèles C/C++ ou celle de composants utilisant le "méta-langage" SML de Simplorer qui peut communiquer avec le noyau de simulation VHDL-AMS et réagir sur ses ASP. Ceci a cependant l'inconvénient d'ôter la portabilité du système et il est peu probable qu'un palliatif puisse être trouvé aux problèmes du type DISCONNECT qui relèvent directement du fonctionnement du noyau de simulation. Les concepteurs font sans ces instructions pour l'instant, devront-ils continuer ainsi, avec à terme l'abandon de celles-ci dans la norme, ou bien les noyaux de simulation VHDL-AMS vont-ils évoluer ?

2.3 - Le choix du logiciel de simulation

Comme énoncé précédemment, notre choix s'est porté sur VHDL-AMS, en partie à cause de la diversité des simulateurs disponibles pour ce langage. En effet, l'offre logicielle dans ce domaine est très importante, mais également très variée. Outre la concurrence qui fait naître des outils similaires, l'évolution dans l'utilisation du langage lui-même a fait émerger plusieurs philosophies de conception. Nous avons donc dû choisir parmi tout cela le simulateur le plus adapté à nos besoins.

2.3.1 - Les simulateurs de première génération à code apparent

Ces outils (hAMSter, ADVance MS, SMASH, ...) appartiennent pour la plupart à une génération de simulateurs qui tend à devenir obsolète. En effet, ils ne s'inscrivent pas dans le cadre de l'accessibilité aux non spécialistes du langage. Il n'en reste pas moins que ce sont des outils de conception très puissants et intéressants pour les spécialistes à condition de pouvoir ensuite porter les modèles d'une plateforme de simulation à une autre.

Dans cette catégorie de simulateurs, ADVance MS (Mentor Graphics) est probablement le meilleur grâce à son implémentation la plus fournie de la norme VHDL-AMS et son moteur de simulation ELDO qui est un des plus rapide et dont les simulations convergent le mieux. C'est notre ancien outil de travail, mais également celui qu'a utilisé l'ENST pour concevoir ses modèles dans le cadre du projet SHAMAN. Un avantage et une faiblesse d'ELDO est sa capacité à utiliser une bibliothèque modèles C propriétaire très performante qui permet de simuler très efficacement et très finement (avec les effets thermiques par exemple) des composants comme une diode ou un transistor. Ceci nous a posé, et pose encore, de gros problèmes de portage pour les modèles développés avec ADVance MS vers Simplorer, et vice-versa.

Le logiciel de hAMSter [*HAMI*], qui n'est plus distribué aujourd'hui (racheté par Ansoft pour l'intégrer à Simplorer), a quant à lui une très grande simplicité d'utilisation, mais une implémentation plus incomplète de la norme et un moteur de simulation moins rapide. Il est cependant gratuit, ce qui en fait un concurrent redoutable et une base très intéressante pour l'enseignement. Le noyau de simulation de hAMSter a servi de base de développement pour celui du simulateur dernière génération d'Ansoft : Simplorer.

2.3.2 - L'évolution des utilisateurs

Devant les impératifs de compétitivité et de productivité de plus en plus sévères sur le marché des microcomposants à technologies mixtes (SoC, SiP, MEMS, MOEMS, ...), la nécessité et l'intérêt de la modélisation devient une évidence pour les industriels qui ont, de ce fait, recours de plus en plus fréquemment à VHDL-AMS et ses concurrents.

L'utilisation des modèles de composants ou de systèmes n'est donc plus l'apanage des concepteurs, spécialistes des langages capables de comprendre le code du modèle et de le modifier le cas échéant.

Ces derniers sont bien sûr toujours à la base des modèles, car il ne peut en être autrement devant la complexité et l'étendue des possibilités et subtilités des langages, mais ils ne sont désormais plus leurs seuls utilisateurs. En effet, des personnes non-spécialistes du langage doivent actuellement être en mesure de se servir des composants modélisés par les spécialistes et de les assembler selon leurs besoins tel des briques de *LEGO*.

Cette vulgarisation de l'utilisation des modèles doit s'accompagner d'une évolution des philosophies de leur conception pour être viable. Il est, en effet, essentiel qu'un maximum

de cas de figures aient été envisagés par les concepteurs des modèles. La conception des modèles doit dépasser le cadre de l'application pour laquelle elle a été nécessaire et permettre une réutilisation dans d'autres contextes par d'autres personnes. Pour ce faire, il est indispensable de faire un effort de généralisation et de documentation des modèles [MIL1].

L'utilisation généralisée et rapide des composants existants à la conception de systèmes plus complexes nécessite la disponibilité de ces derniers. En effet, il est intéressant, voire nécessaire, pour les industriels de disposer de bibliothèques de composants. Il peut alors s'agir de ceux développés par leurs experts, mais également de bibliothèques de composants fournies par des tiers. De fait, la standardisation des langages de modélisation a permis l'avènement de la propriété intellectuelle (Intellectual Property – IP) dans la conception des modèles, ouvrant la voie à un métier de service qui consiste à fournir des modèles.

L'utilisation par le plus grand nombre signifie également que ces tiers n'ont probablement pas accès à d'autres informations sur un produit que celles contenues dans les feuilles de caractéristiques. Il est alors indispensable que le concepteur de modèles génériques s'appuie sur ces paramètres et non sur les données physiques des composants qui sont le plus souvent jalousement gardées par les fabricants.

2.3.3 - L'évolution des environnements de travail

Afin de rendre plus aisées la manipulation et l'assemblage de ces composants par des non spécialistes des HDLs, il est nécessaire d'automatiser les processus d'instanciation et de connexion des modèles, ainsi que les paramètres de simulation et la représentation des résultats.

Dans notre époque informatique bercée par des environnements graphiques gérés à l'aide de la souris, quoi de plus naturel et intuitif pour les utilisateurs non spécialistes des langages que l'assemblage d'icônes représentant les fonctions des systèmes.

Cette interface graphique utilisateur (*Graphical User Interface* – GUI) est un des aspects développés par la nouvelle génération de simulateurs. Elle permet de cacher un code HDL derrière l'image de sa fonction, comme l'illustre la **figure 06**. De cette manière, l'utilisateur non spécialiste se réfère dans un premier temps à la fonction qu'il veut réaliser et non à la manière dont elle est implémentée.

Pour ce faire, l'interface graphique n'a besoin de connaître que les informations contenues dans la déclaration des modèles (ENTITY) qui spécifient les entrées/sorties et les paramètres. L'implémentation des modèles (ARCHITECTURE) n'intervient pas dans cette représentation, mais permet ensuite à l'utilisateur de choisir un niveau de modélisation parmi ceux éventuellement prévus par le concepteur du modèle. Par des interfaces relativement simples et conviviales, l'utilisateur peut ainsi construire un modèle en "instanciant" graphiquement ses composants, en les connectant entre eux et en traitant les résultats souhaités. En ce qui concerne les connexions, qui peuvent être de différentes classes (chaque classe transportant plusieurs types d'informations) dans VHDL-AMS, leur intégrité est vérifiée par l'environnement de simulation au moment où elles sont dessinées par l'utilisateur, et une couleur peut leur être attribuée en fonction du domaine de la physique auquel elles font référence.

Un outil qui réalise les fonctions ci-dessus décharge l'utilisateur de la nécessité de connaître le langage, mais pas de la compréhension de certains mécanismes de base comme les types de connexion. Car, même si les nouveaux simulateurs doivent répondre aux besoins des nouveaux utilisateurs, ils ne doivent pas pour autant limiter les possibilités des développeurs.

En effet, l'outil graphique, qui est une abstraction du code VHDL-AMS, représente une limitation pour les développeurs comme toute abstraction. A titre d'exemple, il est effectivement tout à fait impossible de rendre compte précisément de tous les comportements possibles d'un système à l'aide d'un modèle comportemental. Il aura cependant l'intérêt d'être rapide, le plus souvent simple comparé au modèle physique, il ne nécessitera pas de connaissances spécifiques dans un domaine particulier de la physique et sera facilement intégrable dans un système plus important. Sa fonction est de rendre compte du fonctionnement "normal" du dispositif. Il en va de même pour le GUI du logiciel qui représente une simplicité d'utilisation, mais fait perdre une partie de la souplesse et des possibilités du langage. Ceci rend impossible une conception entièrement graphique des composants élémentaires. Il est donc indispensable pour les développeurs de modèles de continuer à bénéficier d'une interface texte leur permettant d'utiliser l'intégralité des possibilités offertes par les langages afin d'offrir les modèles les plus complets et les plus polyvalents possibles aux utilisateurs non spécialistes.

Les nouveaux environnements de travail pour les HDLs doivent donc inclure en leur sein une représentation symbolique simple des composants et de leurs connectiques, tout en conservant la conception de modèles à l'aide des outils textes classiques.

2.3.4 - Les possibilités offertes par l'interfaçage avec VHDL-AMS

Nous l'avons spécifié précédemment, VHDL-AMS possède des limites. Les systèmes faisant appels à des domaines tels que le traitement géométrique et les dérivées spatiales ne sont pas aisément modélisables au seul moyen de ce langage. Il en existe cependant d'autres comme le C/C++ qui permettent de développer ces algorithmes. Il est donc important de pouvoir concevoir des modèles qui fassent cohabiter des composants de différents langages. Pour cela, il faut pouvoir les interfacier.

La plupart des logiciels du marché permettent ces interactions, mais elles sont le fruit de démarches propriétaires. En effet, la norme VHDL-AMS ne donne aucune information sur la façon dont le langage devrait interagir avec d'autres. De ce fait, la portabilité des modèles complexes est extrêmement réduite.

Il serait intéressant que la norme définisse une façon d'interagir avec l'extérieur (un prototype d'appel de fonction ou une instruction propre par exemple), de manière à donner une direction aux développeurs de simulateurs. C'est cependant une tâche très ardue si l'on veut essayer de mettre au point un mécanisme simple qui permette d'accéder à n'importe quel langage étranger sans pour autant en limiter les possibilités.

2.3.5 - Les nouveaux simulateurs disposant d'une GUI

Ces outils (Simplorer 6/7 [ANSI], SystemVision [MEN3], ...) représentent la dernière génération de simulateurs. Nés des besoins exposés ci-dessus, ils incluent la prise en charge et la cohabitation de ces différents langages afin de couvrir au mieux le spectre des niveaux et des types de modélisation existant.

Le pionnier dans ce domaine, apparu peu avant le début de ma thèse, est Simplorer 6.0 (Ansoft [ANSI]).

Cet environnement de simulation révolutionnaire, basé, pour sa partie VHDL-AMS, sur le noyau de simulation de hAMSter, permet d'allier une représentation graphique conviviale et intuitive à des modèles aussi complexes qu'on le souhaite (*figure 13*). De plus, dans sa version complète, le logiciel permet de créer des modèles C/C++ et de les interfacier,

comme Matlab/Simulink ou Mathcad, avec VHDL-AMS. Ce logiciel inclut également la compatibilité avec d'autres outils de simulation du groupe Ansoft, comme "Maxwell" qui permet de modéliser des comportements avec une méthode de type éléments finis et de les intégrer dans un modèle système.

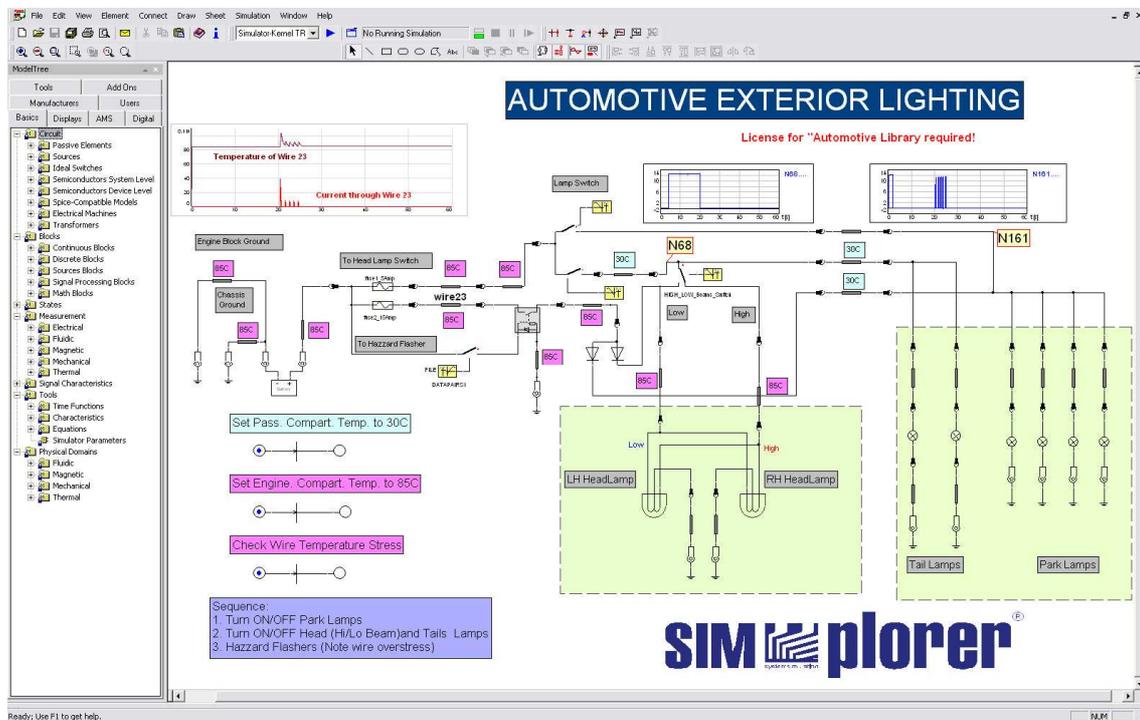


Fig. 13 - Vue schématique d'un circuit multi-abstractions, multi-domaines dans Simplorer

La réponse de Mentor Graphics à Ansoft a pour nom SystemVision [MEN3]. Il est fondamentalement basé sur ADVance MS, ce qui en fait un outil puissant qui s'interface plus naturellement avec les modèles en C/C++. Un effort particulier a été fourni pour le portage de SystemVision sous Windows, alors qu'ADVance MS n'était utilisable que sous Unix, le rendant accessible à un plus grand nombre. Il possède en outre les forces et faiblesses de son aïeul.

On peut cependant regretter les lacunes de l'interface graphique de ce logiciel (*figure 14*). En effet, avec une grande austérité, un manque de convivialité et des fonctions graphiques communément attendues absentes, SystemVision ne semble pas capable, dans sa forme actuelle, de satisfaire aux besoins des utilisateurs non spécialistes.

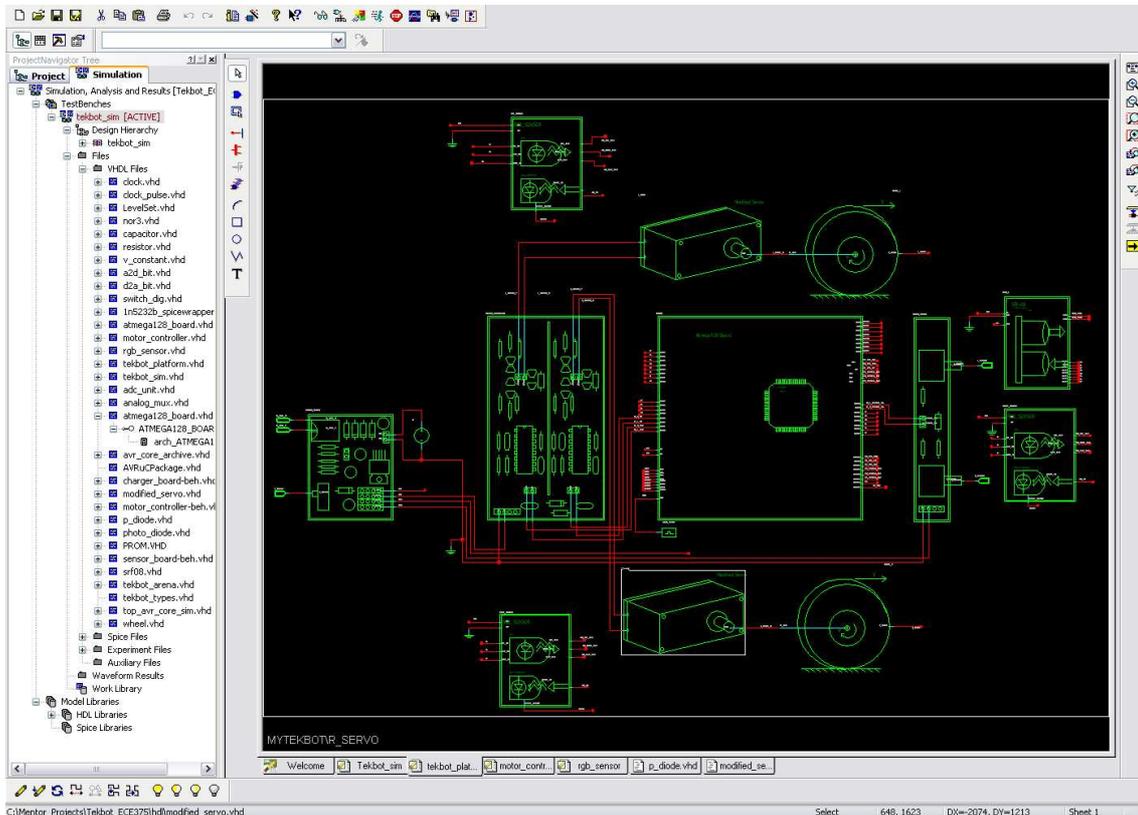


Fig. 14 - Vue schématique d'un circuit multi-abstractions, multi-domaines dans SystemVision

Ansoft et Mentor Graphics proposent tous deux des versions d'évaluation gratuites mais incomplètes de leurs logiciels pour les étudiants (surtout en ce qui concerne l'interfaçage, mais pas dans l'implémentation et la simulation de VHDL-AMS lui-même).

2.3.6 - Conclusion

Il existe actuellement trois produits suffisamment puissants pour prétendre à une utilisation industrielle :

- **SMASH (Dolphin)**
- **SystemVision (Mentor Graphics)**

- **Simplorer (Ansoft)**

Le premier, bien que n'appartenant pas à la dernière génération de simulateurs offre un très bon support de VHDL-AMS et une polyvalence très importante quant aux différents langages qu'il supporte et permet d'interfacer. De par l'absence d'une interface graphique indispensable pour les utilisateurs finals du projet SHAMAN, nous n'avons pas retenu ce candidat pour nos simulations.

A la convivialité de Simplorer s'ajoute, notamment, l'indisponibilité du produit Mentor Graphics au début du projet et de la thèse, ce qui nous a conduit à utiliser Simplorer pour réaliser nos simulations. Nous avons, par là même, dû faire face aux problèmes de portage des modèles, tous fonctionnels, conçus avec ADVance MS par nos collègues de l'ENST vers Simplorer pour l'intégration dans le modèle global. Effectivement, ces modèles font appel à des primitives propriétaires en C/C++ de ADVance MS, surtout au niveau des transistors, qui n'existent pas dans Simplorer.

Cet aspect demeure le point noir du portage entre simulateurs VHDL-AMS. Car, même si les problèmes de portage liés à l'implémentation incomplète ou incompatible de la norme VHDL-AMS elle-même vont s'estomper avec les prochaines versions des logiciels, les appels de primitives, les interfaces graphiques et les moyens d'interfaçage avec les autres langages resteront des systèmes propriétaires ne pouvant être facilement portés si aucune normalisation au niveau de ces problèmes n'intervient prochainement.

2.4 - Simplorer [ANS1]

2.4.1 - Introduction

Ce simulateur appartenant à la dernière génération utilise un noyau de simulation VHDL-AMS issu du simulateur hAMster [*HAMI*], disponible jusqu'alors gratuitement sur Internet. Ce logiciel supporte l'interfaçage entre les langages et logiciels tels que VHDL, VHDL-AMS, C, Simulink, Mathcad, et la suite de logiciels dédiés à la conception et à la simulation des systèmes électromagnétiques d'Ansoft. De plus, Simplorer utilise un langage propriétaire (SML) et qui lui permet d'utiliser une bibliothèque de primitives

propres faisant intervenir des clones des modèles SPICE les plus utilisés et d'autoriser certains fonctionnements comme les réseaux de Petri.

Son interface intuitive et riche en couleurs laisse une grande marge de manœuvre tant au concepteur pour l'écriture des modèles et le dessin des symboles associés, qu'à l'utilisateur qui peut mettre facilement en avant les différents blocs de son schéma et y incorporer directement les résultats de simulation qu'il souhaite afficher, rendant ainsi la feuille de schéma utilisable directement pour un rapport écrit.

Comme nous l'avons déjà écrit, c'est principalement à cause de sa sortie plus précoce que nous avons opté pour Simplorer, mais c'est également parce qu'il nous a apporté satisfaction que nous l'avons conservé.

2.4.2 - L'interface graphique

L'interface graphique de Simplorer comporte plusieurs parties telles que les gestionnaires de symboles et de bibliothèques, ainsi que les interfaces de pré/post-traitements. Nous ne présenterons cependant ici (*figure 15*) que l'interface des feuilles de schémas permettant de connecter et de faire interagir les différents éléments d'un système :

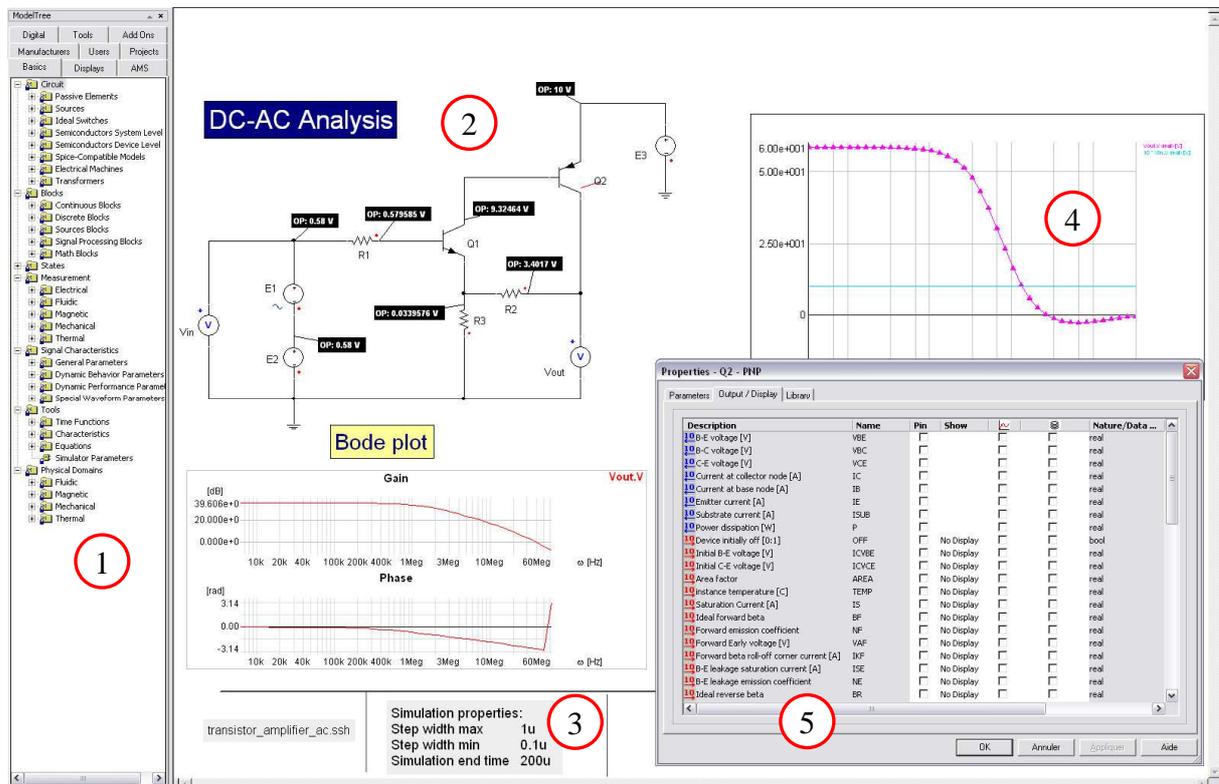


Fig. 15 - Quelques aspects de l'interface graphique utilisateur

Cette GUI met à portée de souris l'ensemble des composants disponibles dans les bibliothèques standard et importées, ainsi que dans celle propre au projet (work) (*figure 15.①*). L'ensemble de ces éléments est instanciable par simple "drag & drop" sur la feuille de schéma (*figure 15.②*). Ils peuvent ensuite être paramétrés selon différents aspects (couleurs, entrées/sorties, sélection des données pour le post-traitement, ...) (*figure 15.③*). Le logiciel offre également la possibilité d'afficher, au sein même de la feuille de schéma, différentes informations ou résultats de simulation comme les paramètres passés au simulateur (*figure 15.③*) ou des résultats de simulation bruts tels que les évolutions temporelles ou fréquentielles des signaux (*figure 15.④*).

Dans un environnement graphique comme celui de Simplorer, la représentation des composants devient un enjeu majeur de la compréhension par le plus grand nombre d'utilisateurs possible. Les gestionnaires de modèles et de symboles permettent en cela de leur associer des représentations graphiques, statiques, symboliques et explicites.

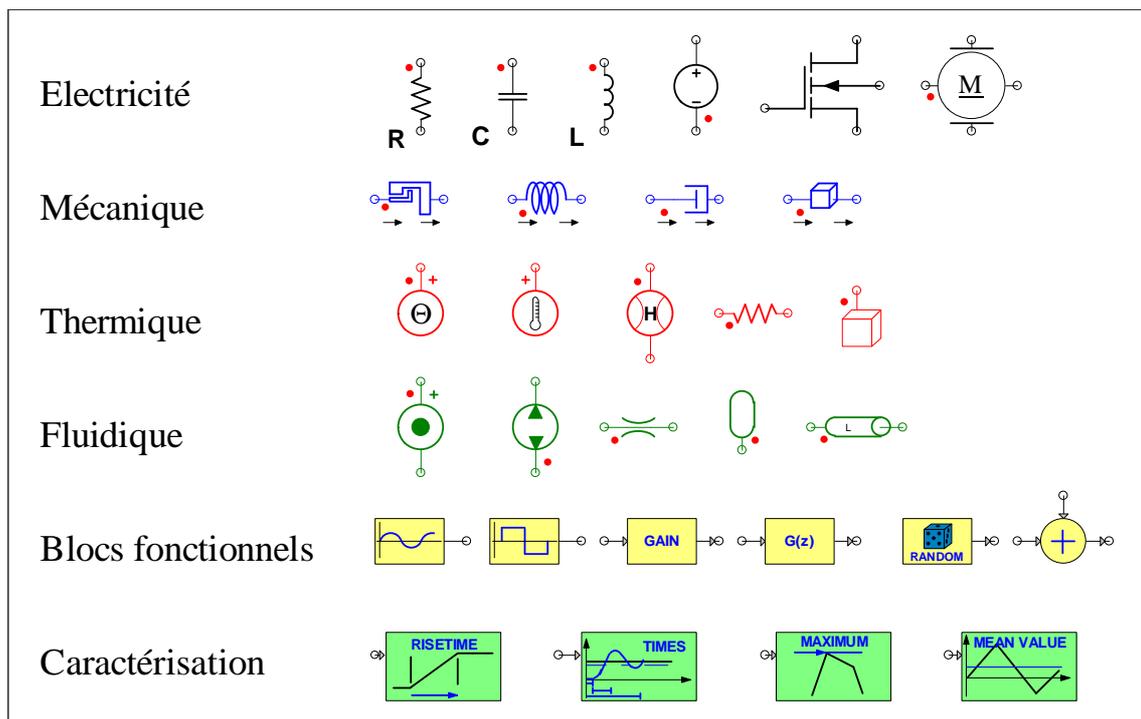


Fig. 16 - Exemples de symboles extraits des bibliothèques standard de Simplorer et natures ou domaines associés

Dans le cadre d'un langage comme VHDL-AMS qui permet de simuler différents champs de la physique en interdisant les connexions abusives entre domaines, il peut être intéressant d'avoir recours à un code de couleurs dépendant de la "NATURE" des ports utilisée. Cette possibilité est offerte par défaut dans Simplorer (*figure 16*), tout en offrant à l'utilisateur la faculté de personnaliser complètement ce jeu de couleurs, mais également celles de tout élément du schéma. Cette option est très intéressante lorsque l'on veut faire ressortir des portions de circuit qui ne peuvent être traitées en utilisant les terminaux. C'est le cas de la partie optique du projet SHAMAN sur lequel nous avons travaillé, et qui ne comporte que des modèles échangeant des quantités.

2.4.3 - La création de modèles en mode texte

Bien que la conception graphique apporte des solutions à un grand nombre de cas de figures ou permette d'avoir recours aux bibliothèques de composants adaptées, sans être confronté à leur code VHDL-AMS, elle ne résout pas tout.

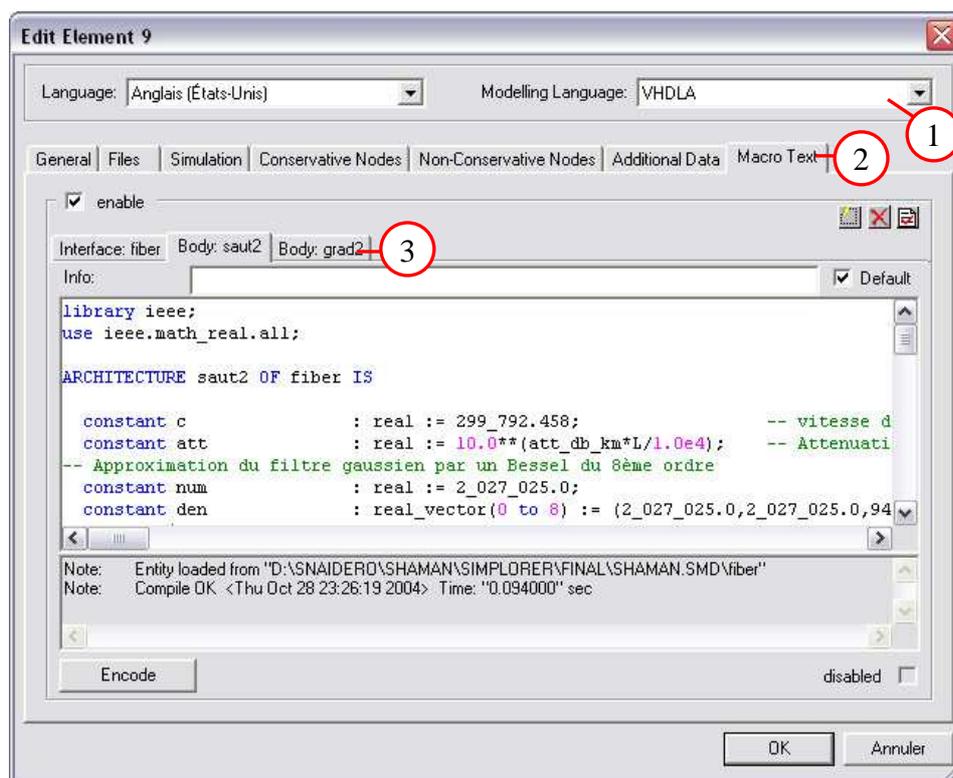


Fig. 17 - Interface de conception textuelle des modèles

En effet, il faut bien créer les modèles utilisés dans les bibliothèques et donner aux spécialistes des langages le moyen de concevoir les objets qui ne peuvent pas être construits par l'assemblage d'éléments existants.

A cette fin, Simplorer 6.0 dispose d'un éditeur de modèles en mode texte qui permet de définir les composants avec une certaine facilité. Il offre la possibilité de créer les différents aspects d'un composant au sein d'une même interface (*figure 17*). On y trouve notamment la possibilité d'écrire des modèles en VHDL-AMS ou SML (*figure 17.①*). Les onglets (*figure 17.②*) permettent quant à eux de gérer les différents aspects d'un modèle (nom, symbole, fichiers impliqués, entrées/sortie, code). En ce qui concerne la fenêtre de code (*figure 17.③*), elle comporte différents boutons pour créer/supprimer l'"ENTITY" et des "ARCHITECTURES" qui sont organisées de manière ingénieuse en onglets, et vérifier que le code écrit peut être compilé par Simplorer. La fenêtre comporte également un bouton "ENCODE" pour compiler le modèle au sein de la bibliothèque Simplorer. Ceci permet d'empêcher l'accès au code source du modèle et donc d'en protéger l'IP, mais au détriment de sa portabilité.

2.4.4 - L'implémentation de la norme VHDL-AMS

La norme VHDL-AMS [LRMI] est un standard complet et complexe qu'aucun simulateur n'implémente complètement aujourd'hui. Simplorer n'échappe pas à la règle et les points suivants sont décrits comme non supportés par la documentation de la version 6.0 :

- **Instructions, attributs, types de données et structures non supportés par Simplorer**
 - GENERATE, ALIAS, GROUP, fonction de RESOLUTION, groupes de LIMIT/TOLERANCE et les attributs définis par l'utilisateur.
 - Q'tolerance, DT'range et les attributs de terminaux
 - Pointeurs, natures composites et les tableaux de quantités
 - Le "port map" et le "generic map" (implicitement géré par Simplorer dans les connexions des schémas)

- **Différences liées à l'interface graphique**
 - Certains types de paramètres sont absents
 - Les signaux sont considérés comme des quantités
 - Les quantités génèrent un évènement à chaque changement de valeur
 - Les ports et les génériques sont traités de la même manière
 - Les unités de base sont différentes et ne sont pas affichées dans les graphiques
 - Il n'y a pas d'interface avec le C/C++ à l'intérieur des modèles

Même si SML, le langage propriétaire d'ANSOFT, permet de contourner certaines de ces impossibilités et si l'interfaçage avec le C/C++ est possible au travers de l'emploi de DLL au sein de Windows, ces aspects ne sont, bien sûr, pas portables vers les autres simulateurs. Par ailleurs, cette description n'est pas exhaustive, car nombre d'instructions sont implémentées de manière incomplète ou "incorrecte". En effet, des portions de la grammaire VHDL-AMS ne sont pas supportées sans être mentionnées (c'est le cas de l'instruction "BREAK ... FOR" par exemple) et, malgré les années de travail de mise au point, la norme contient des non-dits laissés à l'appréciation des concepteurs de simulateurs qui peuvent compromettre la portabilité des modèles. Même si Simplorer 7.0 améliore le support de la norme VHDL-AMS et en fait un peu plus le pilier central du logiciel, il reste trop incomplet pour une utilisation dans le cadre d'un processus de développement entièrement en VHDL-AMS et complètement portable.

2.4.5 - Les possibilités d'interfaçage

Comme nous y avons fait allusion ci-dessus, l'interfaçage entre VHDL-AMS et les autres langages n'est pas une fonction inhérente à VHDL-AMS, mais développée au sein du simulateur par le biais de composants spécifiques accessibles seulement depuis l'interface graphique. Simplorer offre la possibilité de faire cohabiter des composants VHDL-AMS avec plusieurs autres langages ou logiciels au sein d'une même feuille de schéma :

- **SML**

Ce langage, propriétaire, utilisé par Ansoft, était la base des anciennes versions de Simplorer. Les deux dernières versions (Simplorer 6.0 et 7.0) permettent d'écrire des modèles dans ce langage qui est encore au centre du logiciel, mais tend probablement à être relégué, par VHDL-AMS, au rang d'outil optionnel dans un avenir proche. De plus, la documentation de Simplorer ne propose pas d'aide à la création de modèles en SML, ce qui limite son utilisation à la bibliothèque fournie par Ansoft et à la gestion de la simulation.

- **Mathcad**

Mathcad est un outil de calcul mathématique puissant et largement répandu. Il peut sembler intéressant de disposer de la possibilité de communication avec ce logiciel qu'offre Simplorer.

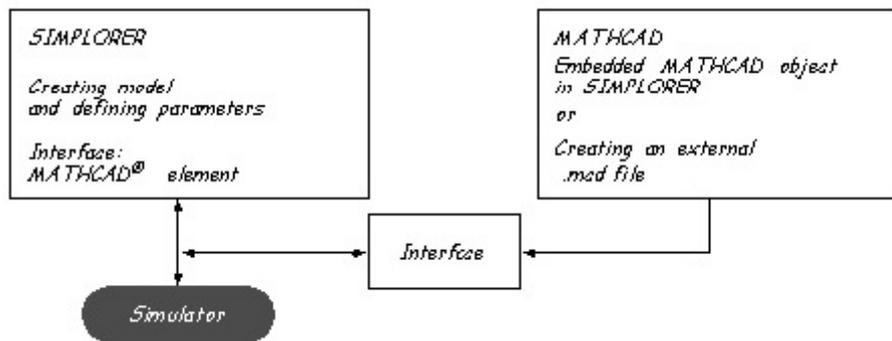


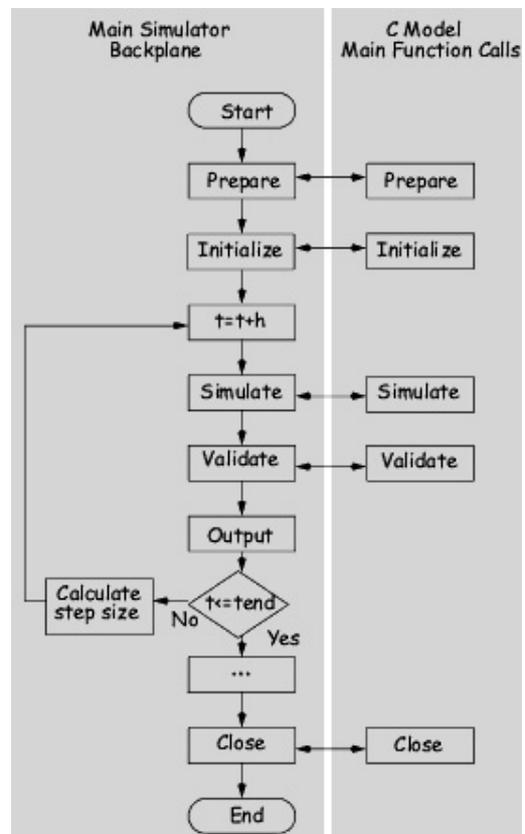
Fig. 18 - Schéma de fonctionnement de l'intégration de MATHCAD dans Simplorer [ANS4]

Après quelques essais infructueux (*Chapitre 3.2.2*), nous en sommes cependant arrivés à la conclusion que ce langage est plus utile en post-traitement des données de simulation que comme composant dans les simulations elles-mêmes.

- **C/C++**

Le langage C/C++ pourrait occuper une place à part dans l'interfaçage de modèles, puisqu'il est présent dans les logiciels des deux principaux acteurs du marché et qu'il constitue un bon complément de VHDL-AMS permettant, par exemple, de s'intéresser à des modèles géométriques. Malheureusement, les mécanismes d'interfaçage n'étant pas définis dans la norme VHDL-AMS, Simplorer et SystemVision proposent des approches totalement différentes interdisant la portabilité des composants développés en C/C++.

En effet, Mentor Graphics, de culture Unix, permet d'insérer le code C directement dans le code VHDL-AMS et utilise un compilateur intégré au système d'exploitation sous-jacent pour simuler le modèle. Dans le monde Windows d'Ansoft, il est nécessaire d'avoir recours à l'outil "Visual Studio" de Microsoft pour créer des fichiers bibliothèques dynamiques (DLL), utilisés ensuite dans les composants, à partir de fichiers d'exemples à la structure très complexe. L'introduction d'un assistant à la création de modèles dans la version 7.0 pourrait simplifier la mise en œuvre de ces éléments, mais ne résoudrait en rien l'aspect non portable de ces modèles. La *figure 19* présente le fonctionnement de la communication entre le simulateur et les appels de fonction en C/C++ dans Simplorer 6.0.

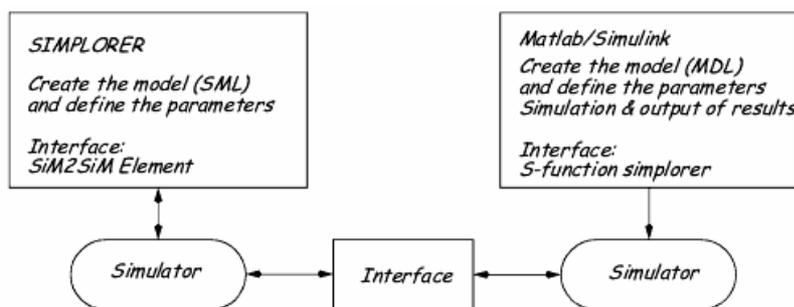


[ANS4]

Fig. 19 - Communication entre VHDL-AMS et les différentes composantes d'un modèle C/C++ sous Simplorer

● **Matlab/Simulink**

L'association de Matlab/Simulink avec VHDL-AMS semble être une autre voie d'investigation dans la recherche d'un accroissement des possibilités de simulation.



[ANS4]

Fig. 20 - Interfaçage VHDL-AMS – Matlab/Simulink sous Simplorer

Contrairement aux modules C/C++ auxquels le simulateur a directement accès, Matlab/Simulink fonctionne en co-simulation avec le simulateur VHDL-AMS. C'est-à-dire que deux simulateurs fonctionnent en parallèle et échangent des informations pour tenter de se synchroniser et s'apporter des réponses mutuelles (*figure 20*).

- **Autres logiciels ANSOFT (Maxwell, ...)**

Il existe d'autres interfaces possibles selon les logiciels de simulation utilisés, mais elles portent principalement sur une compatibilité avec les autres produits de simulation du fabricant.

Même si elles peuvent apporter des solutions intéressantes à des problèmes ponctuels, ces interfaces restent propriétaires et perturbent la portabilité naturelle attendue avec VHDL-AMS, ce qui, dans le cadre d'un développement réutilisable, est à éviter si l'on peut faire autrement.

2.5 - Conclusion

Etant donné la position que semble prendre VHDL-AMS au sein des entreprises, l'évolution des simulateurs dans le sens de la dernière génération devrait se voir renforcée avec une prise en charge plus complète et plus centrale de VHDL-AMS. Cette tendance s'observe déjà au sein de Simplorer 7.0 qui propose une meilleure implémentation de la norme et qui a uniformisé ses interfaces de composants quel que soit le langage utilisé.

Pour la portabilité des modèles, nous sommes en droit d'espérer une évolution qui conduise vers une gestion des projets entièrement en VHDL-AMS, mais ceci nécessitera probablement une révision en profondeur de la norme permettant de standardiser les points importants tels que le format de la feuille de schéma ou le formalisme des interfaces.

Heureusement, VHDL est une norme active qui a su et saura probablement encore s'adapter aux besoins de son marché.

3.1 - La fibre optique

Les fibres utilisées dans le POM SHAMAN sont de type multimodes à gradient d'indice (GI) ou à saut d'indice (SI). La modélisation de ce composant a pour objectif de rendre compte de l'influence de ses paramètres opto-géométriques sur les caractéristiques spectrales et temporelles du signal que l'on propage. Etant donné le niveau d'abstraction souhaité pour décrire le comportement de cet objet, une modélisation de l'évolution du champ optique par la méthode de propagation de rayon (Beam Propagation Method – BPM) n'était pas nécessaire.

Les effets qu'il était important de prendre en compte dans ce modèle sont :

- L'atténuation ;
- La dispersion modale, qui traduit les différents chemins optiques pris par la lumière lors de la propagation le long de la fibre ;
- La dispersion chromatique, qui traduit la différence de vitesse de propagation des différentes longueurs d'ondes transmises dans la fibre.

La valeur linéique de l'atténuation est disponible dans les fiches des constructeurs. Elle a été intégrée dans notre modèle comme une atténuation sur la puissance optique transmise par la fibre en fonction de sa longueur.

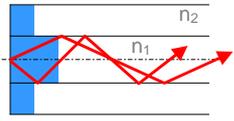
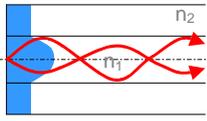
Les spécialistes en optique du projet ont déterminé les dispersions chromatiques et modales grâce à une approche géométrique qui ont ensuite été utilisées pour caractériser la bande passante d'un filtre modélisant le phénomène de manière globale.

Il est important de remarquer que les équations décrites ci-après ne sont valables que si la fibre optique est parcourue par un faisceau lumineux de profil gaussienne.

3.1.1 - La dispersion modale

La dispersion modale D_{mod} traduit la diversité des chemins optiques pris par la lumière lors de sa propagation le long d'un guide (fibre ou autre), elle est donc dépendante des conditions d'injection dans ce dernier. Dans le cas d'une injection gaussienne ou uniforme, tous les modes de la fibre sont excités, les trajets extrêmes ont donc un retard à la propagation plus important que les trajets situés sur l'axe optique. De plus les courbures de la fibre vont modifier la répartition de ces trajets tout au long de la propagation.

Le retard temporel imputable à la dispersion modale peut être calculé en comparant les chemins optiques maximaux et minimaux des rayons dans la fibre (fibres SI) ou en comparant les vitesses de groupe des modes de la fibre (fibres GI).

Fibre à saut d'indice (SI)	Fibre à gradient d'indice (GI)	Paramètres
 $D_{mod} = \frac{(O.N)^2}{2n_1c}$	 $D_{mod} = \frac{(O.N)^4}{8n_1^3c}$	<p>O.N. : Ouverture Numérique de la fibre n₁ : indice optique du cœur de fibre L : longueur de la fibre optique</p>
$\Delta\tau_{mod} = L \times D_{mod}$		

Tab. 02 - Dispersion modale dans les fibres multimodes SI et GI [SHA1]

3.1.2 - La dispersion chromatique

La dispersion chromatique D_λ traduit le fait que toutes les longueurs d'ondes transmises dans la fibre ne se propagent pas à la même vitesse. Ce phénomène est d'autant plus important que le laser est multimodes. Le détecteur, n'ayant pas une résolution spectrale suffisante pour séparer les différentes longueurs d'ondes, effectue la sommation de ces dernières. Il en résulte une impulsion élargie et donc une limitation de la bande passante.

Le retard de propagation dû à la dispersion chromatique peut s'exprimer comme suit:

$$\Delta\tau_c = -L \times D_\lambda \times \Delta\lambda \text{ [SHA1]}$$

3.1.3 - La dispersion totale [SHA1]

De par leurs effets statistiques sur les trajets, les dispersions modale et chromatique s'additionnent de façon quadratique :

$$\Delta\tau_T = \sqrt{\Delta\tau_c^2 + \Delta\tau_{\text{mod}}^2}$$

Si l'on considère la réponse d'une fibre optique à une impulsion gaussienne, d'écart type σ_T de formule:

$$\sigma_T^2 = \frac{1}{3} \left(\frac{\Delta\tau_T}{2} \right)^2$$

On en déduit la bande passante du filtre qui modélisera la fibre optique multimodes par l'expression:

$$BP = \frac{\sqrt{2 \times \ln(2)}}{2 \times \pi \times \sigma_T} \approx \frac{0.187}{\sigma_T} \text{ soit } BP \# \frac{0.187 \times \sqrt{12}}{\Delta\tau_T}$$

Cette formule nous permet d'estimer la bande passante d'une fibre SI et d'une fibre GI ayant les paramètres suivants:

Paramètre	Valeur	Fibre SI	Fibre GI
Ouverture numérique	0.275	$\Delta\tau_{\text{mod}} = 25,21ns$	$\Delta\tau_{\text{mod}} = 211,97ps$
Longueur de fibre	300 m	$\Delta\tau_c = 13,8ps$	$\Delta\tau_c = 13,8ps$
Atténuation	2 dB.Km ⁻¹	$\Delta\tau_T \approx \Delta\tau_{\text{mod}}$	$\Delta\tau_T = \sqrt{\Delta\tau_{\text{mod}}^2 + \Delta\tau_c^2}$
Indice de cœur	1.5	$\Delta\tau_T \# 25,21ns$	$\Delta\tau_T \# 212,42ps$
Dispersion	-46 ps/nm/km	$BP \# 25,8MHz$	$BP \# 3,1GHz$
Largeur spectrale du laser	1 nm		

Tab. 03 - Bande passante de deux fibres de paramètres identiques, mais de profils différents

On trouve dans ces calculs un rapport de grandeur de l'ordre de 120 entre les dispersions des deux types de fibres. Cette différence est un peu inférieure à celle présentée dans [TOF1] (environ 200), mais on conserve toutefois deux ordres de grandeur d'écart entre les deux types de fibres, ce qui suffit à justifier l'usage des fibres GI dans les communications

à haut débit, puisqu'à 1GHz, les données ne peuvent être dissociées du bruit en sortie de fibre.

3.1.4 - Les simulations d'une fibre multimodes

Deux modèles de fibre multimodes ont été développés, l'un par IPSIS dans le logiciel Comsis [IPSI], l'autre en VHDL-AMS par PHASE (Le code du modèle VHDL-AMS est donné en *Annexe III*). Les deux modélisations sont similaires et partent de l'hypothèse que la fibre optique peut être modélisée par un filtre dont la bande passante est déterminée par la méthode décrite précédemment.

- **Simulation sous Comsis**

Le logiciel Comsis offre une double définition des paramètres de la fibre multimodes selon ceux dont l'utilisateur dispose. L'une est appelée définition système, l'autre définition physique. Dans ce dernier cas, la bande passante du filtre modélisant la fibre optique est calculée à partir des paramètres saisis par l'utilisateur.

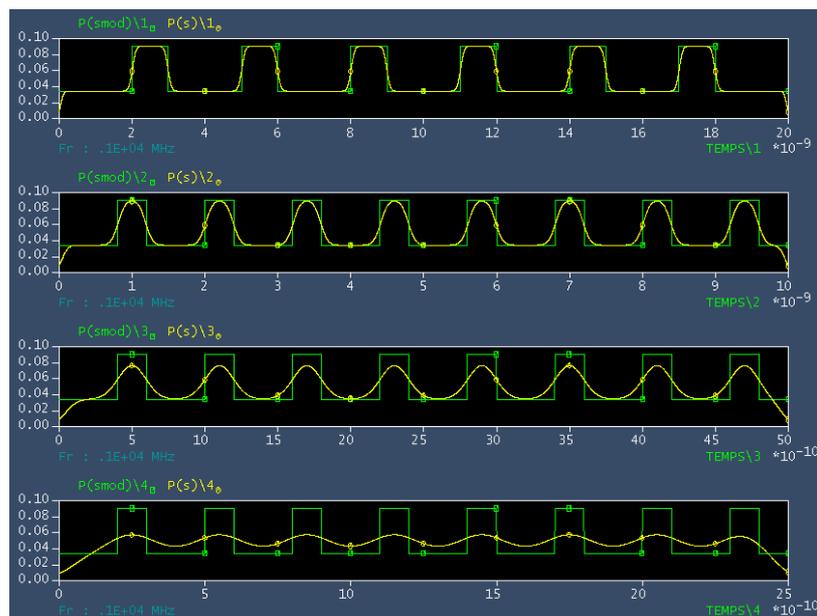


Fig. 21 - Transmission d'un signal de débit 1, 2.5, 5, 10 GBits/s dans une fibre GI de 300 m

La **figure 21** présente les résultats Comsis de simulation de la propagation de signaux à 1Gbit/s, 2.5Gbits/s, 5Gbits/s puis 10Gbits/s sur 300m de fibre optique à gradient d'indice.

- **Simulation VHDL-AMS**

La **figure 22** montre l'aspect graphique du composant représentant la fibre optique sous Simplorer.

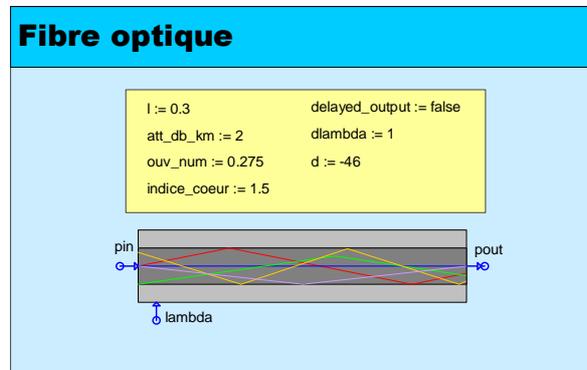


Fig. 22 - Représentation graphique choisie pour la fibre optique et affichage des paramètres modifiables directement sur le graphique sous Simplorer

Dans un soucis de généralisation et de "réutilisabilité" du modèle, nous avons élaboré deux architectures possibles selon le type de fibre (SI et GI). De cette manière, il devient extrêmement simple pour l'utilisateur de sélectionner la fibre à simuler (**figure 23**), puisqu'il n'a qu'à choisir l'architecture dont il a besoin dans l'interface graphique de Simplorer.

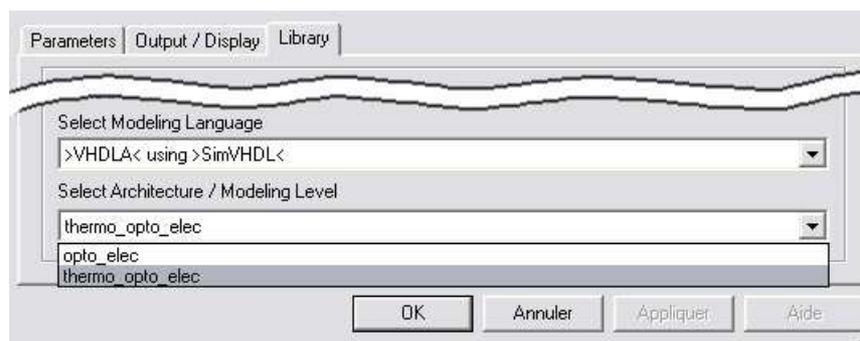


Fig. 23 - Interface de choix de l'architecture d'un composant à simuler

La **figure 24** présente la même simulation que celle effectuée par nos collègues de IPSIS (**figure 21**) en utilisant notre modèle VHDL-AMS. Pour des questions de représentation, nous avons désactivé la prise en compte du décalage temporel entre les signaux d'entrée et

de sortie de la fibre dû au temps de parcours dans celle-ci. Les résultats de simulation obtenus avec VHDL-AMS et Comsis sont tout à fait comparables dans les effets des dispersions.

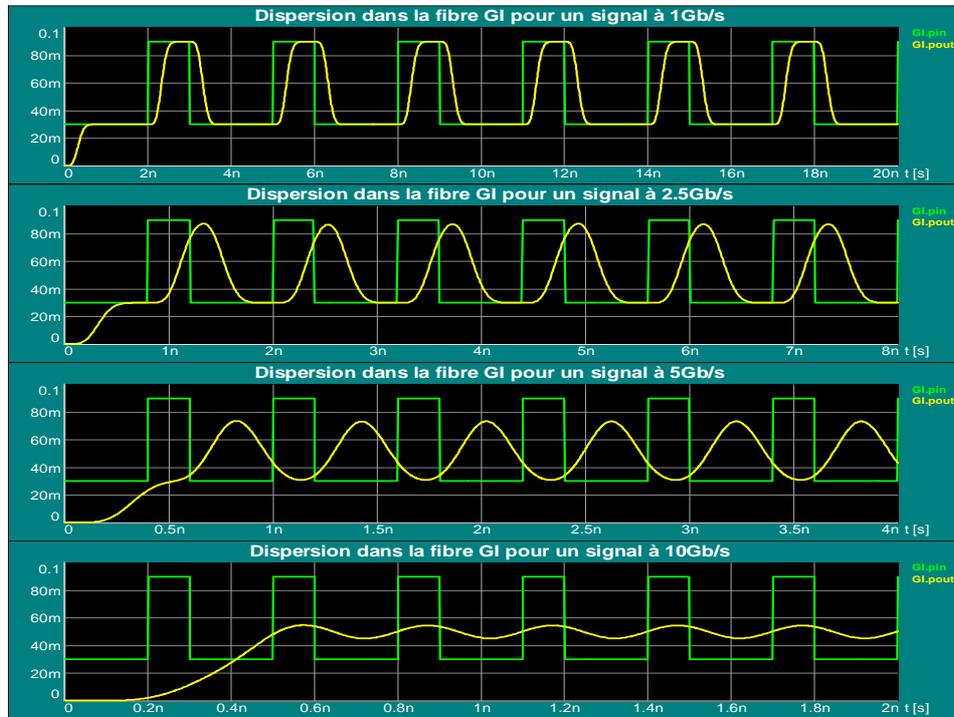


Fig. 24 - Transmission d'un signal de débit 1, 2.5, 5, 10Gbits/s dans une fibre GI de 300 m

On observe que, pour des débits supérieurs à 5Gb.s⁻¹, le signal devient très déformé et difficilement exploitable sur des distances de cet ordre (300 m). Par ailleurs, on peut relever une différence d'affichage entre les deux simulateurs. En effet, la causalité des signaux n'apparaît pas dans le graphique issu de Comsis, mais cette présentation permet en revanche de comparer plus aisément la largeur des impulsions de sortie à celles injectées.

Lors de la connexion d'un ruban de fibre avec un autre, il peut y avoir des défauts d'alignement des fibres optiques entre elles. Nous avons donc mis au point un modèle de connecteur qui permet de simuler la liaison entre deux fibres optiques et les pertes de signal qui y sont associées (les types de défauts d'alignement et le code du modèle associé se trouvent en *Annexe IV*).

Notre modèle VHDL-AMS de fibre optique prend en compte les effets intrinsèques à la fibre multimodes (atténuation, dispersion modale et chromatique) de façon globale sous la forme d'une limitation de bande passante et d'une atténuation linéique. Du fait que nous ne travaillons pas en utilisant une méthode de propagation de rayon (Beam Propagation Method – BPM), les conditions d'injection et d'extraction de la lumière dans la fibre ne sont pas prises en compte. C'est pourquoi il est nécessaire d'associer à la fibre optique des modèles remplissant les fonctions de couplage avec les VCSELs et les photodiodes afin de rendre compte des conditions de couplage originales de ce POM.

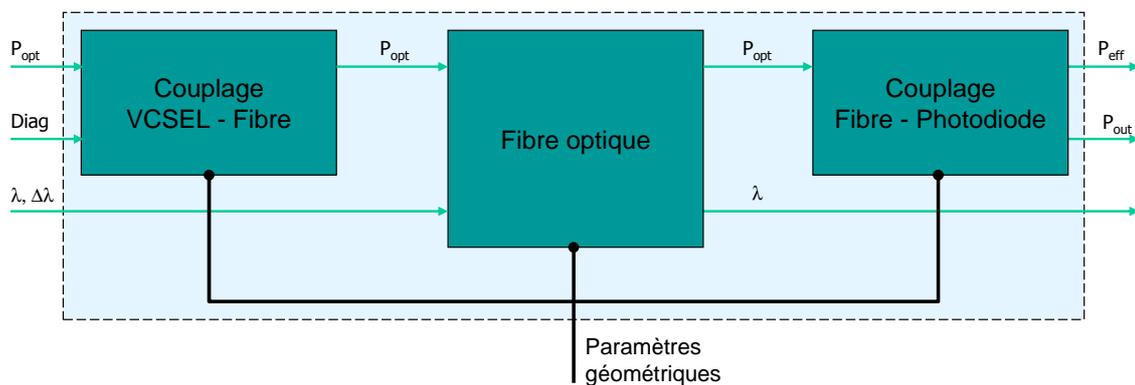


Fig. 25 - Diagramme bloc de la fibre multimodes

3.2 - Le couplage VCSEL-Fibre optique

La méthode utilisée pour transmettre la puissance optique issue d'une diode VCSEL à la fibre optique du canal associé est un principe novateur et breveté par D-LIGHT SYS (*figure 26*). En effet, la lumière entre dans la fibre optique perpendiculairement à celle-ci en traversant sa surface latérale cylindrique. La lumière est alors réfléchiée dans l'axe de la fibre par un biseau de la fibre à 45° poli et métallisé à l'or.

A cette condition d'injection assez peu usuelle, vient s'ajouter le diagramme de rayonnement spécifique du VCSEL. Les études menées par [DAVI] et Supelec ont montré que la VCSEL fonctionne selon différents modes d'émission en fonction du courant qui l'alimente. Etant donnée la plage d'alimentation des diodes laser pour le projet SHAMAN,

nous avons pu limiter à deux les modes de fonctionnement du VCSEL pris en compte pour les simulations.

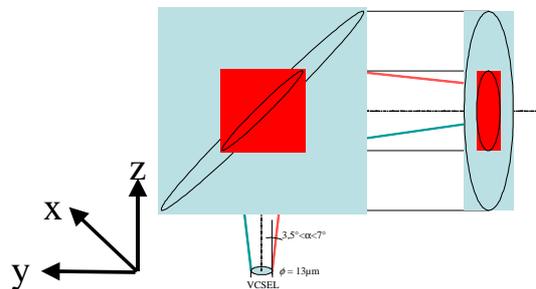
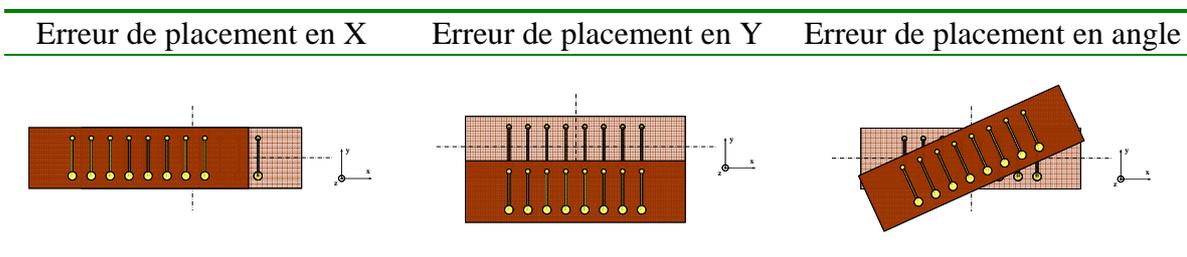


Fig. 26 - Vue tridimensionnelle du dispositif de couplage VCSEL - fibre optique

Ces modes ne font cependant pas apparaître des régimes de fonctionnement indépendants et l'on passe progressivement de l'un à l'autre. Cette évolution a pu être modélisée [DAVI] par la soustraction d'une fonction gaussienne à une autre. Ce modèle utilise malheureusement des paramètres n'apparaissant pas dans les fiches de documentation des composants et qui devront être déterminés expérimentalement. Enfin, l'ouverture numérique (ON) du VCSEL évolue elle aussi avec l'alimentation, ce qui complique encore un peu la modélisation du dispositif de couplage.

A ce problème de modélisation du diagramme de rayonnement des diodes laser, vient s'ajouter une donnée de modélisation supplémentaire d'ordre mécanique. En effet, les marges d'erreur liées aux processus d'assemblage des barrettes de VCSELs sur les supports silicium induisent des décalages par rapport à la position idéale du centre du VCSEL par rapport à la fibre.



Tab. 04 - Les différentes erreurs de placement de la barrette de VCSEL

A cela vient s'ajouter la marge d'erreur de placement suivant l'axe longitudinal (en Y) de la fibre optique. De fait, le placement transversal (en X) de la fibre optique est garanti par le "V-groove" (canal en forme de V creusé chimiquement dans la silicium du support), mais il n'existe pas de dispositif qui garantisse le placement longitudinal exacte de la fibre dans le guide.

L'étude de ces facteurs est primordiale, car la puissance optique couplée dans la fibre optique ne doit pas descendre en dessous d'un certain seuil afin de garantir la réception d'un signal ayant suffisamment d'amplitude et assurer l'intégrité des données transmises. Ceci conditionne le taux d'erreur binaire (Bit Error Rate – BER) qui doit répondre aux normes *{IEEE 802.3ae, ANSI SONET OC24/STM8}*. Ceci induit le durcissement ou le relâchement des contraintes de fabrication, et donc des coûts associés. Etant donné que D-LIGHT SYS avait connaissance du bon fonctionnement de son module, il s'agissait surtout pour eux de savoir s'il était possible de réduire les contraintes de fabrication pour réduire le coût de production des modules.

L'étude réalisée dans [DAVI] montre des mesures du taux de couplage d'une VCSEL avec une fibre optique en fonction du déplacement en X et Y du VCSEL par rapport à la fibre effectuées par THALES.

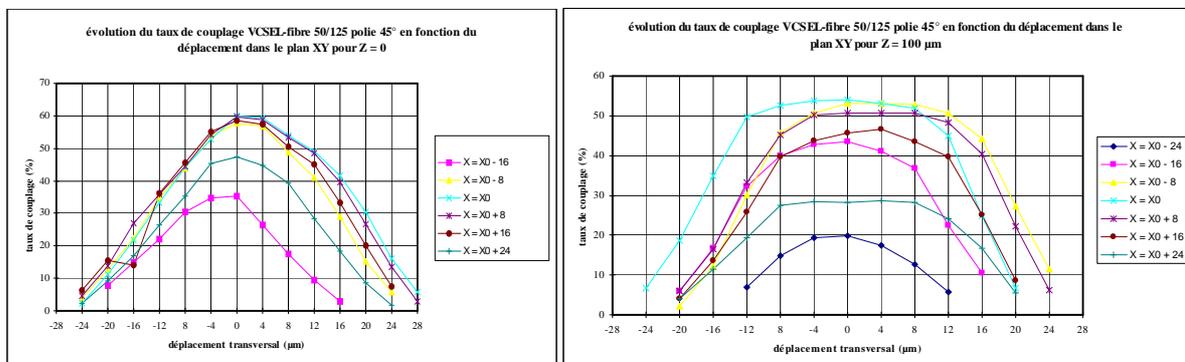


Fig. 27 - Couplages mesurés par THALES

Pour réaliser la modélisation du couplage entre la diode laser et la fibre optique, notre première idée fut d'en établir une tabulation tridimensionnelle à l'aide du logiciel d'optique CodeV (ORA – Optical Research Associates), puis d'utiliser les fonctionnalités d'accès

fichier de VHDL-AMS ou du C/C++ pour effectuer une interpolation de ces données au point mécanique de simulation qui nous aurait intéressé.

3.2.1 - Le couplage modélisé et simulé avec CodeV

Le logiciel CodeV est un outil utilisé par nos collègues du laboratoire de photonique qui permet de modéliser géométriquement des dispositifs optiques et d'en simuler le fonctionnement, entre autre, par lancé de rayons.

- **Le modèle de couplage VCSEL – fibre optique**

Pour établir la modélisation du couplage, nous avons choisi d'intégrer les éléments suivants au modèle:

- La surface d'émission d'un VCSEL ;
- Le milieu optique traversé par la lumière entre le VCSEL et la fibre (dans le cas du démonstrateur, il pouvait s'agir d'air ou de résine, dont l'indice, très proche de celui de la gaine, avait pour effet d'annuler la focalisation des rayons incidents lors de leur passage à l'interface entre la résine et la gaine) ;
- Une fibre optique multimodes (à l'origine SI ou GI) ;
- Un miroir placé à 45° en tête de fibre optique, simulant le biseau poli et doré de la fibre.

La modélisation au sein de CodeV nécessite de positionner les éléments constituant le dispositif dans l'espace. Par ailleurs, l'originalité de la configuration géométrique du couplage (injection latérale de la lumière) ne nous a pas permis d'utiliser les modèles de fibres optiques préexistants. La modélisation de la fibre optique SI a pu être réalisée à partir des éléments suivants:

- 4 quarts de tube formant l'interface air-gaine
- 4 quarts de tube formant l'interface gaine-cœur
- 2 disques évidés fermant la gaine
- 2 disques fermant le cœur

- 1 miroir elliptique évidé dans la gaine
- 1 miroir elliptique dans le cœur

La VCSEL, source de lumière, a été modélisée au moyen des éléments suivants:

- 1 source ponctuelle
- 1 diaphragme matérialisant sa surface

La *figure 28* expose une représentation tridimensionnelle filaire de la source lumineuse, de la fibre optique et de la surface réfléchissante :

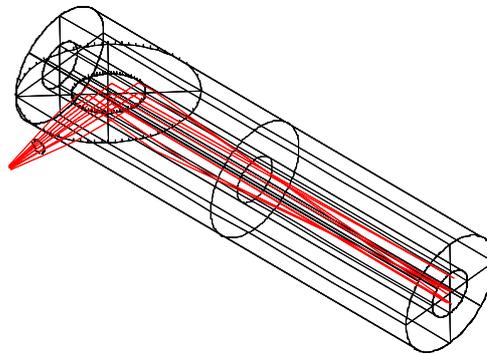


Fig. 28 - Découpage du dispositif VCSEL-Fibre optique avec CodeV

Le fait que les rayons lumineux rencontrent plusieurs fois une même interface, dans un ordre différent et un nombre de fois variable pour chaque rayon tiré, place le dispositif dans les systèmes appelés "non-séquentiels" par CodeV. Ceci entraîne une description beaucoup plus lourde du dispositif et une forte augmentation des temps de simulation.

Les scripts de conception et de simulation du dispositif sont répertoriés dans l'*Annexe II*

● **Les résultats**

A partir de notre système avec fibre SI, nous avons commencé à tabuler l'espace des couplages possibles et, travaillant par plans successifs, on fixe la distance VCSEL-fibre et on fait varier la position du VCSEL par rapport à la fibre dans le plan XY.

Les résultats sont obtenus sous forme de tableaux tridimensionnels de points extraits des fichiers de résultats de CodeV et traités avec Matlab. Les exemples ci-dessous donnent

l'évolution du coefficient de transmission pour un z (distance VCSEL-fibre) fixé et une variation de $\pm 50 \mu\text{m}$ suivant x et $\pm 100 \mu\text{m}$ suivant y :

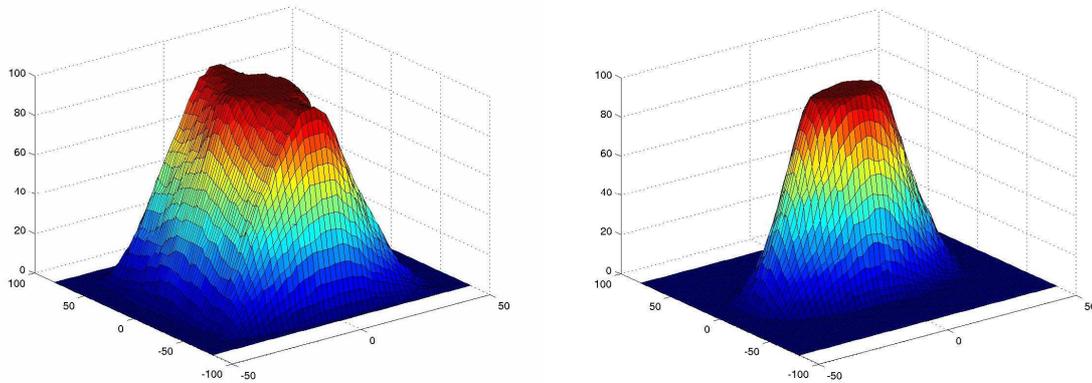


Fig. 29 - Graphique Matlab de l'évolution du taux de couplage en fonction des variations suivant x et y à z fixé. Résultats issus des simulations CodeV

La comparaison des courbes expérimentales obtenues par THALES à celles issues de la simulation montre des différences :

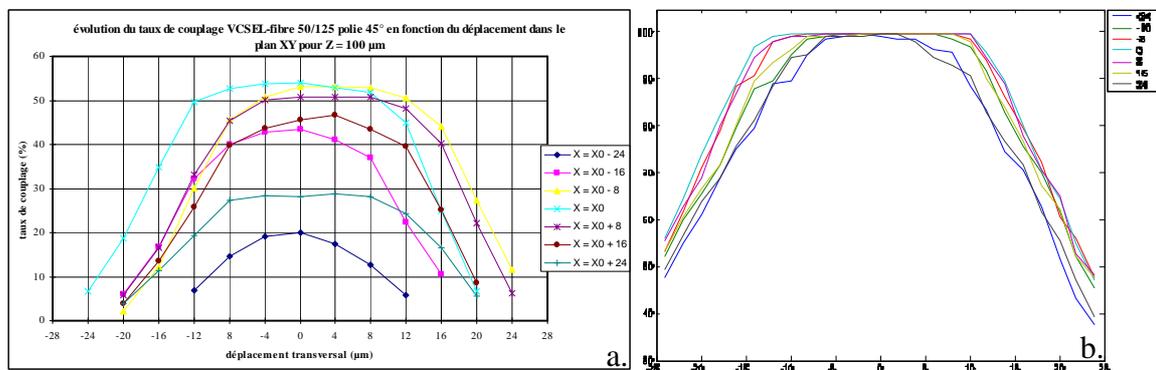


Fig. 30 - Comparaison des mesures expérimentales (a) de couplage à la simulation CodeV (b)

On peut remarquer que les allures générales des courbes de taux de couplage ne sont pas trop éloignées l'une de l'autre. Cependant, la simulation fait apparaître un taux de couplage maximum de 100%, valeur jamais atteinte d'après les mesures. Les spécialistes ont supposé que cette différence est due aux pertes liées au passage des interfaces optiques (4% par interface) et peut-être à la réflexion non totale des rayons lumineux par le biseau

de la fibre. On peut enfin supposer que les différences sont liées au fait que la fibre modélisée est une fibre SI, alors que les mesures ont été réalisées sur des fibres GI.

En raison de l'impossibilité technique de transformer la fibre optique SI en fibre optique GI au sein du logiciel, nous n'avons pas pu vérifier la dernière hypothèse et n'avons pas poussé plus loin les investigations avec le logiciel d'optique.

3.2.2 - Le couplage modélisé avec d'autres langages

Suite à la remise en cause de la tabulation tridimensionnelle du couplage VCSEL-fibre par simulation optique, nous avons dû trouver une alternative.

Nous avons alors envisagé d'utiliser directement l'étude mathématique théorique présentée dans [DAVI].

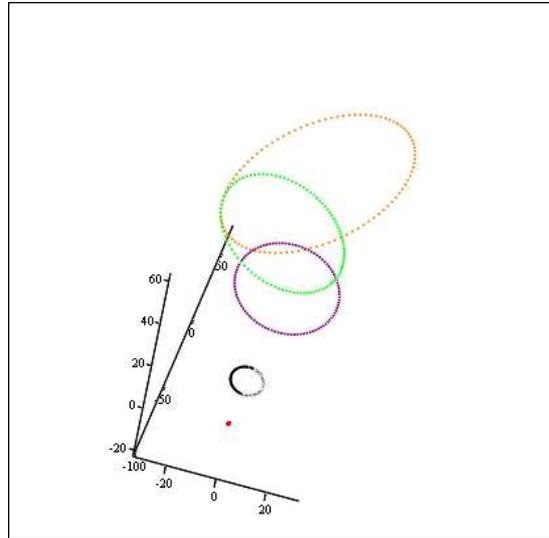
En ce qui concerne la modélisation du diagramme d'émission par une double gaussienne, les paramètres utilisés dans cette modélisation ne sont pas physiques et nous ne connaissons pas l'évolution de la combinaison linéaire des deux gaussiennes avec les variations du courant d'alimentation du VCSEL. Nous avons cependant pris cela comme base, car l'équipe de Supelec travaille encore aujourd'hui sur le sujet sous la direction du Pr. Zeno Toffano.

Pour ce qui est du calcul du taux de couplage à proprement parler, nous étions dans l'impossibilité de réaliser les intégrales spatiales. En effet, des instructions de ce type sont totalement exclues du langage VHDL-AMS qui ne traite que les intégrations et dérivations temporelles. La réalisation par un programme d'une fonction d'intégration simulée à chaque pas de calcul analogique (Analog Simulation Point – ASP) nous a semblé trop complexe et nous a fait craindre une explosion du temps de calcul. Nous avons préféré opter pour une approximation stochastique de l'intégration par la méthode de Monte-Carlo. Il s'agit alors d'observer le devenir d'un certain nombre de rayons lumineux, tirés aléatoirement, issus d'une VCSEL, puis d'en déduire le taux de couplage du dispositif.

Du fait que nous travaillions initialement sur la version d'évaluation de Simplorer, qui ne comprend pas les capacités d'interfaçage C/C++, nous avons tout d'abord pensé à utiliser l'interfaçage avec Mathcad.

● La modélisation avec Mathcad

Afin de tester la faisabilité de l'entreprise, nous étions parti sur un modèle de couplage assez simple (script en *Annexe II*).



$(X, Y, Z), (X_v, Y_v, Z_v), (X_s, Y_s, Z_s), (X_f, Y_f, Z_f), (X_{ve}, Y_{ve}, Z_{ve})$

Fig. 31 - Représentation du plan de comptage des rayons transmis (cercle vert et cercle violet) par rapport à la VCSEL (cercle noir) et à la surface réfléchissante du biseau de la fibre (ellipse orange)

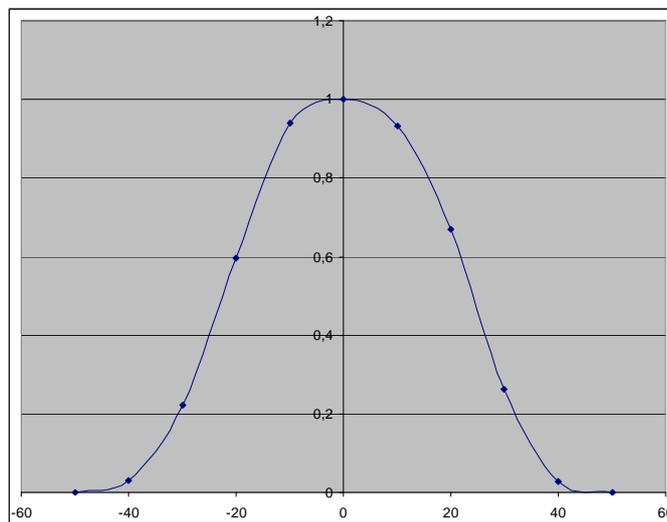


Fig. 32 - Représentation du Taux de couplage en fonction du défaut d'alignement transversal (μm) par simulation avec Mathcad

En effet, nous avons choisi de ne pas considérer la forme cylindrique de la fibre optique par laquelle les rayons passent, mais de l'assimiler au plan parallèle au plan XY, tangent à la fibre optique (plan contenant les *cercles vert et violet* sur la **figure 31**).

Il s'est ensuite agi de déduire la zone lumineuse éclairée par la fibre (*cercle violet*) en utilisant la VCSEL (*cercle noir*) et son angle d'émission. En projetant la forme de la surface réfléchissante (*ellipse orange*) sur le plan de travail (*cercle vert*), nous avons une approximation de la zone d'intersection des cercles vert et violet. En réalisant un tirage aléatoire de rayons dans le cercle violet pondéré avec le diagramme de rayonnement du VCSEL et en testant la présence dans le cercle vert, nous avons une approximation du taux de couplage VCSEL-Fibre.

Même si le modèle semblait vraisemblable (**figure 32**), nous ne pouvions pas le tester, car nous n'avions pas le lien entre les paramètres du modèle et la physique du VCSEL.

Nous avons cependant voulu intégrer notre modèle à Simplorer pour en tester l'interfaçage et les temps de calcul. Nous nous sommes retrouvé dans l'incapacité de faire simuler la feuille de calcul Mathcad à chaque ASP. Effectivement, cette opération était réalisée une fois pour toute à l'initialisation de la simulation et ne se répétait plus par la suite. Il s'avère que Mathcad est un outil très adapté à Simplorer dans le cadre du post-traitement des données, mais pas dans la modélisation en elle-même.

Suite à ce problème d'interfaçage, nous avons acquis la version complète de Simplorer, ce qui nous a permis d'envisager le développement de modèles en C/C++.

- **La modélisation en C/C++**

Le langage C nous offre plusieurs avantages. La rapidité d'exécution d'un modèle compilé en C est bien supérieure à celle d'un modèle MathCad interprété.

Nous avons tout d'abord envisagé un modèle (code en **Annexe II**) qui reprenne le principe du modèle Mathcad. Le travail "d'étude" ayant déjà été réalisé précédemment, il a d'abord fallu comprendre comment l'interface fonctionnait. Il existe des fichiers d'exemples et des structures de bases qui permettent de créer son propre modèle, mais la documentation est limitée. Contrairement à ADVance MS (Mentor Graphics), qui est mieux documenté et fonctionne sous Linux, ce qui lui permet d'utiliser de manière transparente les compilateurs C/C++ libres fournis avec, Simplorer nécessite l'utilisation de Visual C++ et la création d'un fichier DLL. Par ailleurs, les instructions et la structure du fichier nécessaires pour

créer la DLL ne sont pas du tout portable (il n'y a pas de DLL sous Linux). Au delà de cela, la partie du code qui réalise le traitement est, elle, portable (sauf peut-être les noms des entrées/sorties, mais il suffit alors de concevoir son modèle avec des variables intermédiaires qui rendront le modèle plus portable).

Le modèle implémenté comprend ainsi une atténuation artificielle supplémentaire, d'un facteur 0.6, par rapport au modèle Mathcad de manière à essayer de se rapprocher des mesures réalisées par THALES.

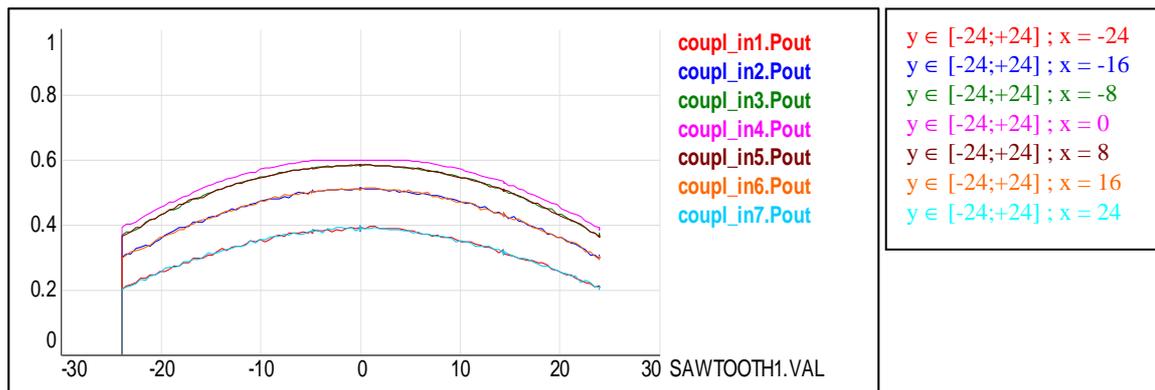


Fig. 33 - Résultat de simulation du couplage VCSEL-fibre en fonction de x et y à z fixé

Ce modèle permet d'obtenir une évolution du taux de couplage en fonction des erreurs d'alignement, mais il semble difficile de définir une distribution associée à la surface de projection de la surface biseautée (correspondant au cercle vert de la **figure 31**) qui rende compte du couplage complexe et sans symétrie de révolution du système (**figures 29 et 30**).

Nous avons alors essayé d'approcher le problème sous un autre angle en reprenant le développement avec Mathcad. Nous avons conservé le profil du diagramme de rayonnement et le principe stochastique, mais le traitement des rayons diffère (**figure 34**).

On estime l'angle d'incidence du rayon issu du VCSEL par rapport au plan tangent à la gaine au point d'impact du rayon, puis par rapport au plan tangent au cœur si le rayon arrive jusqu'à ce dernier. Si le cœur de fibre est atteint, on vérifie finalement que le rayon arrive bien dans la surface couverte par le miroir biseauté de la fibre. Nous avons considéré que la lumière dans la gaine était perdue et nous avons introduit une perte de 4% du signal

au passage de chaque interface optique, en accord avec les informations qui nous ont été transmises par les spécialistes de l'optique.

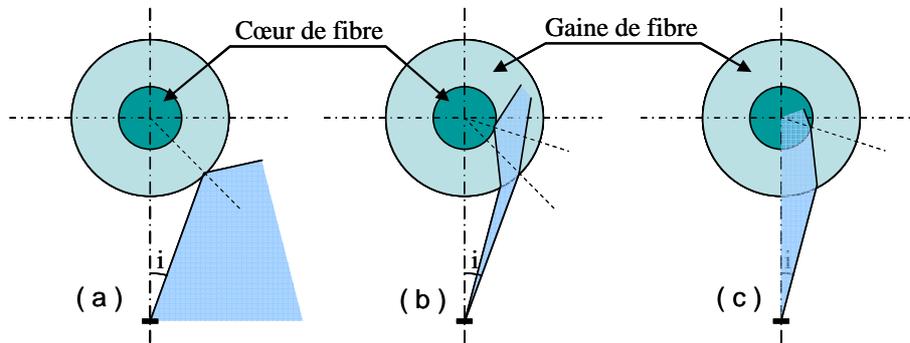


Fig. 34 - Possibilités de devenir du signal lumineux (a. et b. sortent, c. entre s'il est réfléchi)

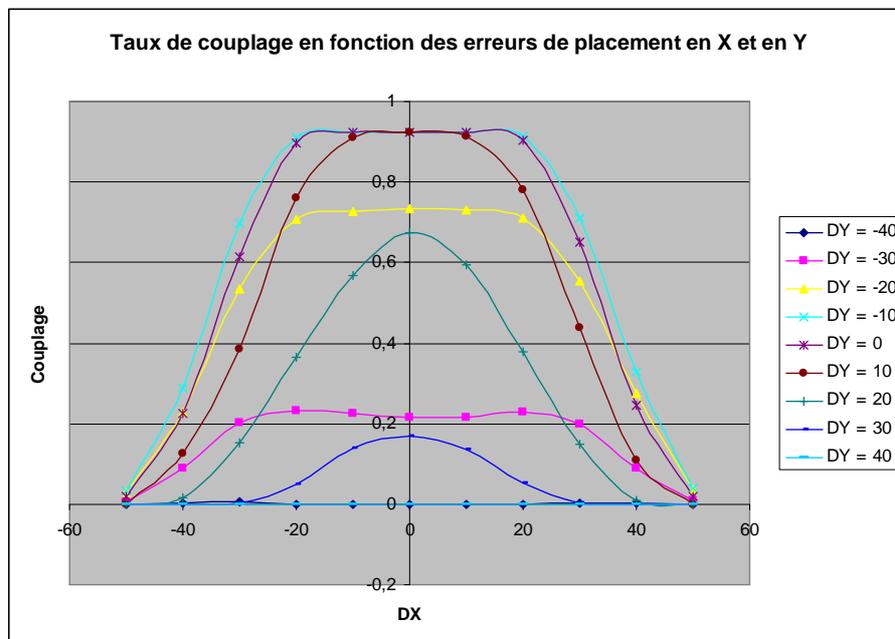


Fig. 35 - Résultats de ce modèle

Si ce modèle ne permet pas de voir un maximum inférieur à 0.922 (soit 2 fois une perte de 4%), il met cependant en évidence la dissymétrie du dispositif suivant Y, ce qui n'était pas le cas du modèle précédent.

Le modèle Mathcad correspondant est disponible en **Annexe II**

- **La modélisation "tout VHDL-AMS"**

Si l'intégralité de la norme VHDL-AMS était implémentée dans les simulateurs, nous pourrions disposer de l'instruction **PROCEDURAL** [LRM1]. Cette instruction permet théoriquement de lancer un bloc d'opérations séquentielles à chaque ASP. De cette manière, nous aurions pu écrire un générateur aléatoire et réaliser les opérations faites en C avec VHDL-AMS.

Par ailleurs, les possibilités du langage SML, propre à Simplorer, permettent de réaliser des calculs à chaque ASP. Nous pourrions donc envisager de créer un modèle interne à Simplorer, qui ne fasse pas intervenir de module C. Dans ce cas, nous perdrons toute possibilité de portage du module puisqu'il serait écrit avec les composants propriétaires de Ansoft.

3.3 - Le couplage Fibre optique-Photodiode

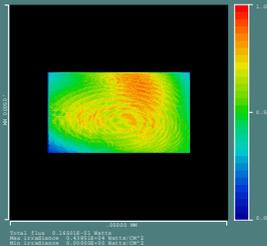
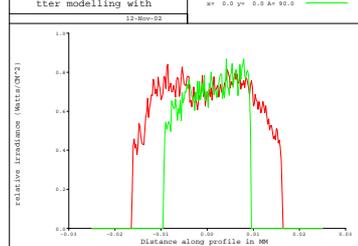
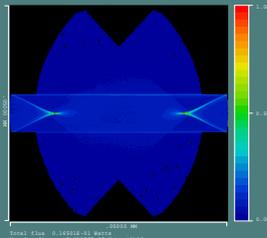
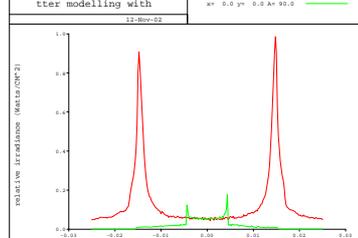
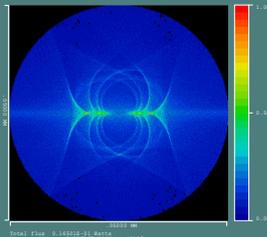
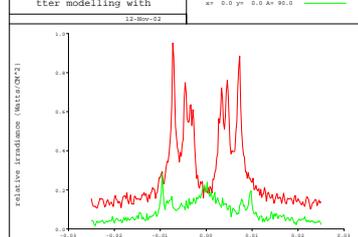
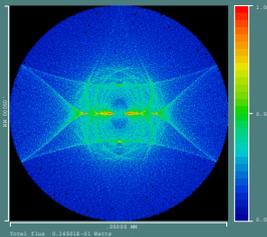
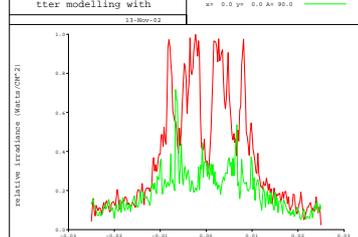
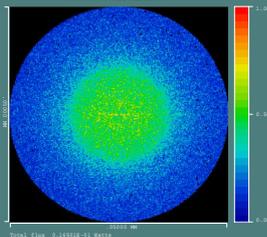
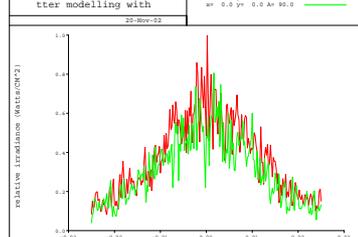
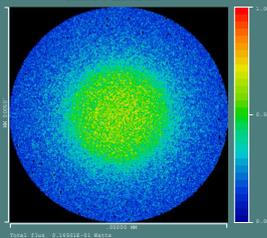
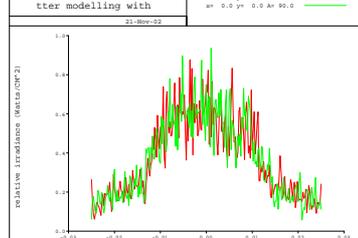
Conçu à l'origine pour utiliser la réflexion du faisceau émergent de la fibre sur un plan incliné taillé dans le substrat en silicium, le couplage entre la fibre optique et la photodiode a été ramené à un système symétrique au couplage entre la VCSEL et la fibre en raison d'un positionnement de la fibre optique trop délicat à assurer dans le premier cas.

Si le système d'extraction est devenu la symétrie physique de l'injection, il n'en va pas de même pour sa modélisation.

Afin de connaître la forme du diagramme de rayonnement au niveau de la photodiode, mais également pour valider les équations de propagation dans la fibre, nous avons essayé de savoir quelle forme prenait le faisceau lumineux en sortie de fibre.

- **Forme du faisceau lumineux dans la fibre optique**

Pour réaliser cette étude, nous avons utilisé les travaux réalisés dans le logiciel CodeV pour le couplage d'entrée. En utilisant les diagrammes de rayonnement de VCSEL expérimentaux présentés *figure 36*, nous observons le diagramme de sortie pour différentes longueurs de fibre (pour des longueurs de fibre de 3 m ou 5 m, il a fallu près d'une journée de calcul à un ordinateur récent pour simuler la propagation).

0,00 m			<p>Juste après réflexion sur le miroir, on obtient une image très peu déformée du spectre d'émission du VCSEL.</p>
0,01 m			<p>Les premières réflexions dispatchent peu à peu l'ensemble des rayons lumineux dans l'ensemble du cœur de fibre</p>
0,05 m			<p>Les modes de la fibre optique s'homogénéisent pour tendre vers une distribution gaussienne</p>
0,10 m			<p>Les différences ne sont plus perceptibles à l'œil, il faut regarder les diagrammes pour voir que la transformation en gaussienne se termine.</p>
1,0 m			<p>Les différences ne sont plus perceptibles à l'œil, il faut regarder les diagrammes pour voir que la transformation en gaussienne se termine.</p>
5,5 m			<p>Les différences ne sont plus perceptibles à l'œil, il faut regarder les diagrammes pour voir que la transformation en gaussienne se termine.</p>

Tab. 05 - Diagramme d'intensité pour différentes longueurs de fibre

L'étude du **tableau 5** semble montrer que le faisceau lumineux en sortie de fibre optique devient très rapidement gaussien. N'ayant pas à notre disposition de diagramme expérimental de forme annulaire (forte excitation), nous n'avons pas pu en tester la propagation. Par ailleurs, pour les mêmes raisons que précédemment, nous n'avons pas traité le cas de la fibre GI.

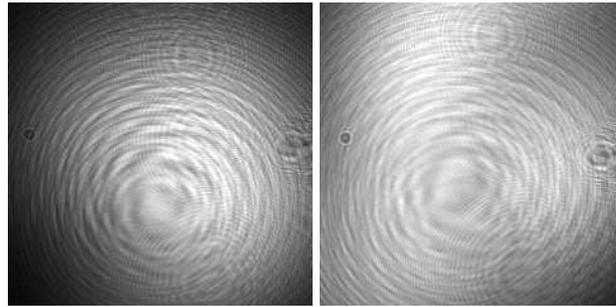


Fig. 36 - Deux vues d'un diagramme de rayonnement de VCSEL expérimental (THALES)

Les résultats du **tableau 5** sont corroborés par des mesures effectuées par THALES avec un VCSEL ULM photonics et une fibre optique multimodes à gradient d'indice.

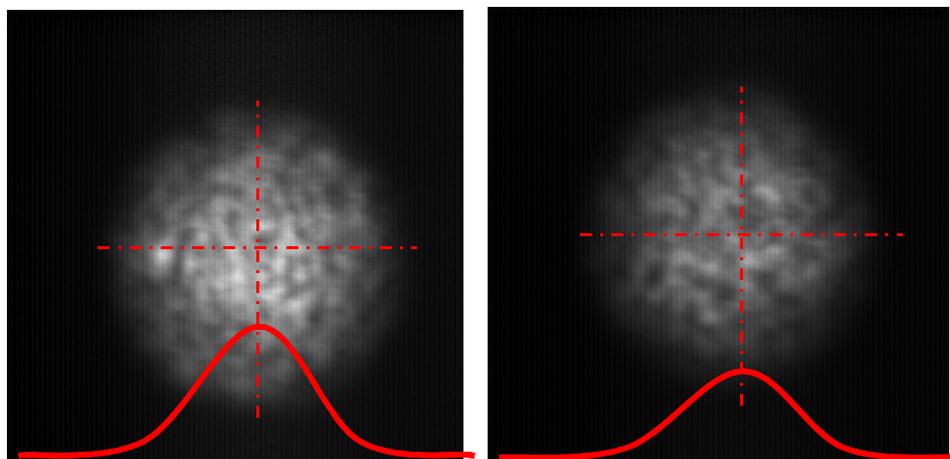


Fig. 37 - Profil du faisceau lumineux après 5 m et 205 m de fibre optique GI [SHA1]

On est en droit, à la vue de ces résultats, de penser que le profil du champ sera gaussien à l'intérieur et en sortie de fibre et cela quelle que soit la longueur de fibre (au delà de 5 m).

Cette étape passée, nous avons commencé à explorer la modélisation du couplage fibre-photodiode, mais la tâche apparaît rapidement trop ardue. En effet, le comportement de l'onde lumineuse dans la fibre optique GI, qui renvoie les rayons centraux avec l'angle le plus grand et les rayons externes dans l'axe de la fibre, nous permet de penser que le diagramme de rayonnement au niveau de la photodiode n'aura rien à voir avec celui que l'on sait gaussien en sortie de fibre (*figures 37 et 38*).

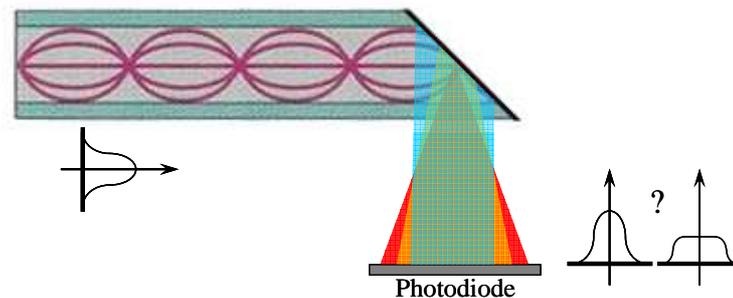


Fig. 38 - Illustration de l'évolution du diagramme de rayonnement en sortie de fibre optique

Nous n'avons jusqu'à présent trouvé aucune littérature qui puisse nous éclairer sur ce point. De ce fait, nous sommes dans l'incapacité de déterminer la distribution et la dimension de la tache lumineuse issue de la fibre optique au niveau de la photodiode. Nous sommes donc incapable de la comparer aux dimensions de la surface active de la photodiode pour prendre en compte l'effet négatif d'un éclairage en dehors de cette surface, modélisé par nos homologues de l'ENST.

Dans l'attente de recherches ultérieures, nous avons choisi de considérer le diagramme au niveau de la photodiode comme uniforme avec un taux de couplage comme paramètre d'ajustement. Ceci prend en compte les pertes dues au couplage lui-même, mais également la quantité de lumière qui ne parvient pas à la photodiode dans le cas où la tache projetée serait plus grande que la surface sensible de la photodiode.

En utilisant l'ouverture numérique de la fibre, on peut néanmoins approximer le diamètre ϕ_p de la surface éclairée par la lumière en sortie de fibre (en conditions d'extraction standards) et estimer ainsi en première approximation les marges de placement possibles.

$$\phi_p = 2 \times z \times \tan(\arcsin(ON))$$

Soit avec une fibre d'ouverture 0.275 et à une distance de 100 μm de la photodiode

$$\phi_p \# 29\mu\text{m}$$

Cette dimension est inférieure d'un ordre de grandeur à la surface réceptrice de la photodiode (250 μm). Ceci laisse une marge d'erreur de placement très conséquente de $\pm 110\mu\text{m}$ et X et en Y, et peu de probabilité que l'intégralité du rayon émergent ne soit pas reçu par la photodiode.

Le code VHDL-AMS du couplage fibre-photodiode est disponible en *Annexe V*

3.4 - Le générateur pseudo-aléatoire de bits

Afin de pouvoir étudier les différentes grandeurs caractéristiques du système au moyen notamment du diagramme de l'œil, nous avons implémenté un modèle de générateur pseudo-aléatoire de bits, générique sur la taille de registre (au choix 7,10 et 15), le débit de données et les valeurs électriques des niveaux logiques.

Le code VHDL-AMS du générateur pseudo-aléatoire est disponible en *Annexe VI*

3.5 - La thermique

Etant donné que l'absence de dissipateur thermique est une des caractéristiques principales du module que nous avons étudié, il semble intéressant, sinon nécessaire de s'intéresser aux phénomènes thermiques qu'il abrite.

Cet aspect est en fait un point essentiel du fonctionnement du module. En effet, les composants électroniques du boîtier (VCSEL, Driver, TIA, Câblages) sont des producteurs de chaleur non négligeables. L'évacuation de cette chaleur est indispensable au bon fonctionnement du système. En effet, les VCSELS et les drivers sont très sensibles à une variation de température.

Pour les diodes lasers, une variation de température entraîne une variation de la puissance optique et de la longueur d'onde émise (*figure 39*).

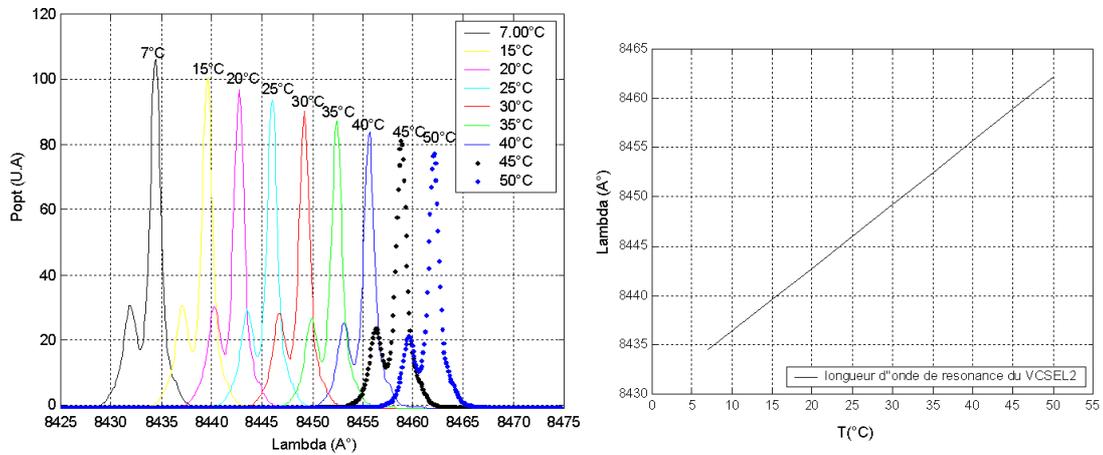


Fig. 39 - Evolution du spectre d'une VCSEL en fonction de la température (Supaero) [SHA1]

Pour les Drivers, ce sont les transistors qui sont très sensibles à la température et font dériver les niveaux de sortie (*figure 40*). Ces variations influent en retour sur la VCSEL et la puissance optique produite

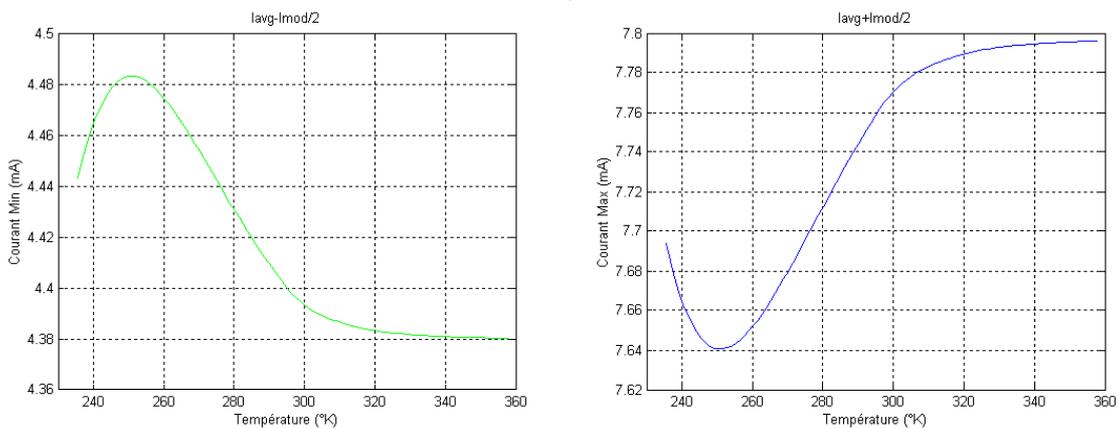


Fig. 40 - Influence de la température sur les niveaux de sortie du driver (ENST) [SHA1]

Il apparaissait donc indispensable de prendre en compte ces effets dans nos modélisations VHDL-AMS.

A cette fin, des modèles thermiques équivalents ont été mis au point par les spécialistes du projet (*figures 41 et 42*).

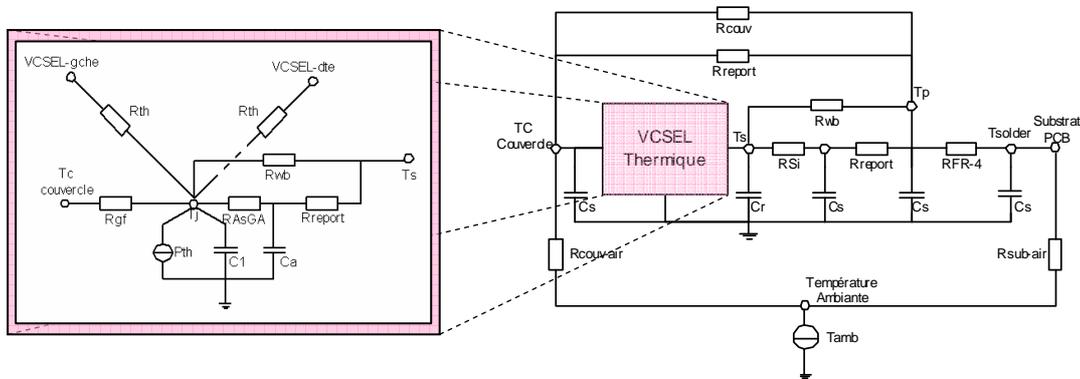


Fig. 41 - Modèle thermique du VCSEL [SHA1]

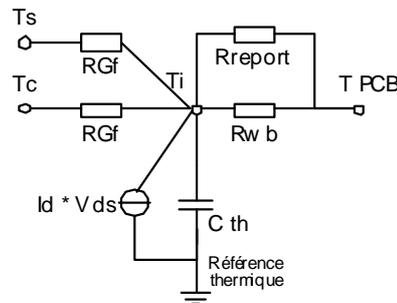


Fig. 42 - Modèle thermique du Driver [SHA1]

Etant donné que le langage VHDL-AMS inclut de manière native le support des différents domaines de la physique, il est assez aisé d'introduire des aspects thermiques dans les modèles. D'un point de vue graphique, Simplorer permet la représentation graphique des différents domaines physiques au moyen d'un code de couleurs. De plus, il existe une bibliothèque de composants de base associés à chaque discipline. Il devient alors assez intuitif de construire des circuits ou de se repérer sur des schémas de systèmes multidisciplinaires.

3.6 - La thermomécanique

A l'origine prévue pour évaluer les marges additionnelles d'erreur de position VCSEL-fibre et fibre-Photodiode dues aux dilatations thermiques des matériaux du boîtier, une étude simple des positions relatives des objets en fonction de la température a permis de procéder en cours de projet à une amélioration de l'architecture globale.

En effet les cylindres de positionnement en acier stub ont été remplacées par des espaceurs en silice (fibre optique) rendant la structure quasiment indépendante de la température, d'une part à cause du coefficient de dilatation thermique 1000 fois moins important de la silice, et d'autre part du fait de l'homogénéisation des coefficients de dilatation thermique de toutes les parties. En effet, toute la structure en silice va se dilater en même temps, ce qui conserve mieux les placements relatifs.

Nous avons donc décidé de ne pas prendre en compte l'évolution thermomécanique du système dans notre modèle, mais les marges d'erreurs mécaniques dues aux procédés de fabrication sont, elles, prises en compte.

3.7 - Le module complet

Un objectif du projet était d'intégrer les différents éléments VHDL-AMS modélisés, par nous et nos collègues de l'ENST, dans une structure plus globale permettant de simuler l'ensemble du dispositif aussi bien d'un point de vue électrique, optique, mécanique que thermique.

Nous avons dissocié cette démarche en deux parties en passant par une étape de modélisation d'un canal du module pris indépendamment des autres.

3.7.1 - La simulation avec Cosis d'un canal de transmission

Le logiciel Cosis ne prend pas en compte les interactions thermiques et mécaniques présentes au sein du module. Il permet cependant de simuler les performances d'une chaîne de transmission comportant des composants optoélectroniques.

L'équipe d'IPSIS a donc simulé un canal de transmission du POM. La chaîne de transmission simulée comporte un driver, un VCSEL, des pertes de couplage pour la partie

émission et réception et une fibre multimode pour la propagation de l'information. En réception il y a une photodiode suivie d'un filtre passe-bande tenant compte de la bande passante de la photodiode et de l'électronique de réception (amplificateur transimpédance et limiteur).



Fig. 43 - Schéma d'un canal de transmission du module dans Comsis

Chaque composant est ici schématisé par une boîte et est défini par un ensemble de paramètres qu'il faut impérativement renseigner avant de lancer la simulation. Une fois cette étape franchie, il est alors possible de lancer une simulation.

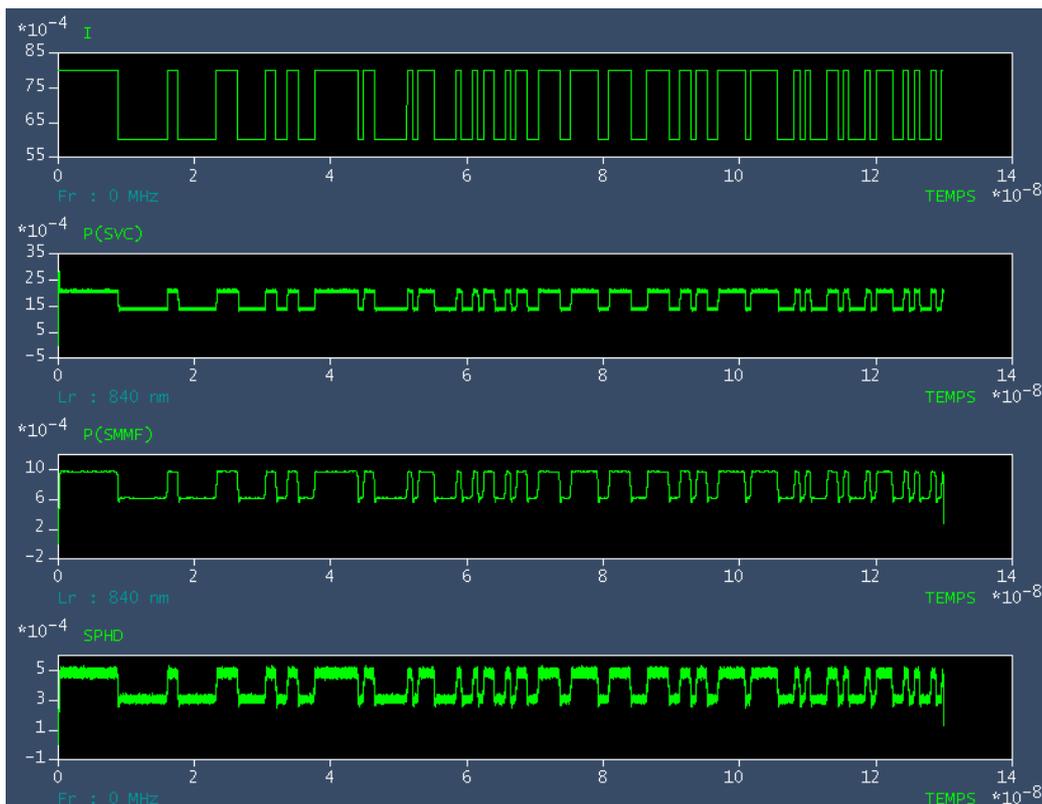


Fig. 44 - Résultats de simulation Comsis pour un canal (Courant en sortie du driver, Puissance optique en sortie du VCSEL, Puissance optique en sortie de la fibre, Courant en sortie du bloc de la photodiode) [SHA1]

Comsis effectue cette dernière dans le mode temporel. Il calcule le signal à la sortie de chaque composant, le signal de sortie d'un composant constituant le signal d'entrée du composant suivant. De par cette conception, ce modèle est dit "*Signal Flow*".

Au terme d'une simulation, nous avons accès au signal à la sortie de chaque composant. Nous avons alors pu observer simultanément la réponse temporelle en sortie du driver, la puissance en sortie du VCSEL et de la fibre optique multimode, ainsi que la réponse temporelle en sortie de la photodiode (*figure 44*).

Nous avons également tracé le diagramme de l'œil de la liaison (*figure 45*).

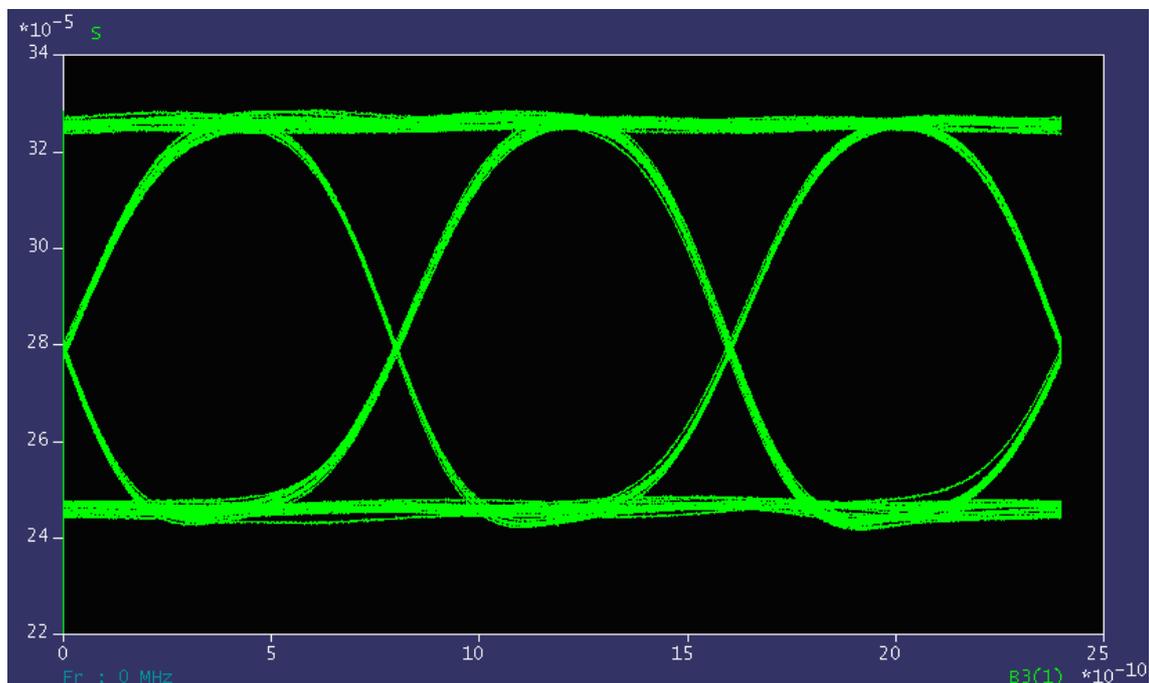


Fig. 45 - Diagramme de l'œil en sortie de la chaîne [SHA1]

3.7.2 - La simulation VHDL-AMS d'un canal de transmission

Nos collègues de l'ENST ayant développé leurs modèles en utilisant le logiciel ADVance MS (Mentor Graphics), nous avons du porter leurs composants vers Simplorer pour pouvoir modéliser un canal de transmission. En effet, comme mentionné précédemment, le code VHDL-AMS devrait être portable d'un outil à l'autre, mais à cause de bibliothèques différentes et d'implémentations différemment incomplètes de la norme,

ce n'est pas encore tout à fait le cas. Tout particulièrement si l'on ne prend pas garde à la conception des modèles.

- **Le portage des différents éléments développés par l'ENST**

L'ensemble des composants qui suivent a été développé et validé avec ADVance MS par nos collègues de l'ENST.

- Le VCSEL

Pour ce composant central du dispositif, il a fallu faire quelques adaptations pour que le modèle fonctionne sous Simplorer. Tout d'abord, les sources de bruit du modèle initial ont été portées à l'extérieur du modèle. Cette démarche est le fruit d'une nécessité due au fait que la fonction mathématique appelée dans ADVance MS n'apparaît pas dans Simplorer, mais également celui de l'opportunité d'une amélioration parce que Simplorer offre la possibilité, grâce à son langage propriétaire SML, d'utiliser des générateurs aléatoires synchronisés sur les ASP fournis par le noyau de simulation VHDL-AMS. En VHDL-AMS pur (comme dans le modèle initial), cette possibilité n'est pas offerte, notamment à cause de l'absence du PROCEDURAL qui permettrait de contourner la difficulté, et il faut générer des points aléatoires sous une condition d'horloge, ce qui ralentit alors considérablement la simulation en imposant des points de calcul ou en n'ayant pas de changement de bruit entre deux points de calcul rapprochés.

Par ailleurs le portage direct du modèle avec ses 3 bruits distincts (sur la diffusion des porteurs, des photons et de la phase) ne semble pas fonctionner correctement sous Simplorer. Nous avons alors exploité la piste d'un bruit global qui présente l'avantage d'être issu de la feuille de caractéristiques du composant : le bruit d'intensité relative (*Relative Intensity Noise* – RIN). Au lieu d'intervenir dans les calculs intrinsèques du VCSEL, ce bruit est simplement superposé à la puissance optique en sortie du composant. De ce fait, 3 fois moins de sources de bruit sont nécessaires et le modèle semble mieux se comporter en simulation (*figure 46*).

Nous avons cependant rencontré quelques problèmes non résolus quant à la relation entre la puissance optique émise et le courant appliqué au VCSEL, et l'influence de la température comme en témoignent les résultats de la *figure 47*.

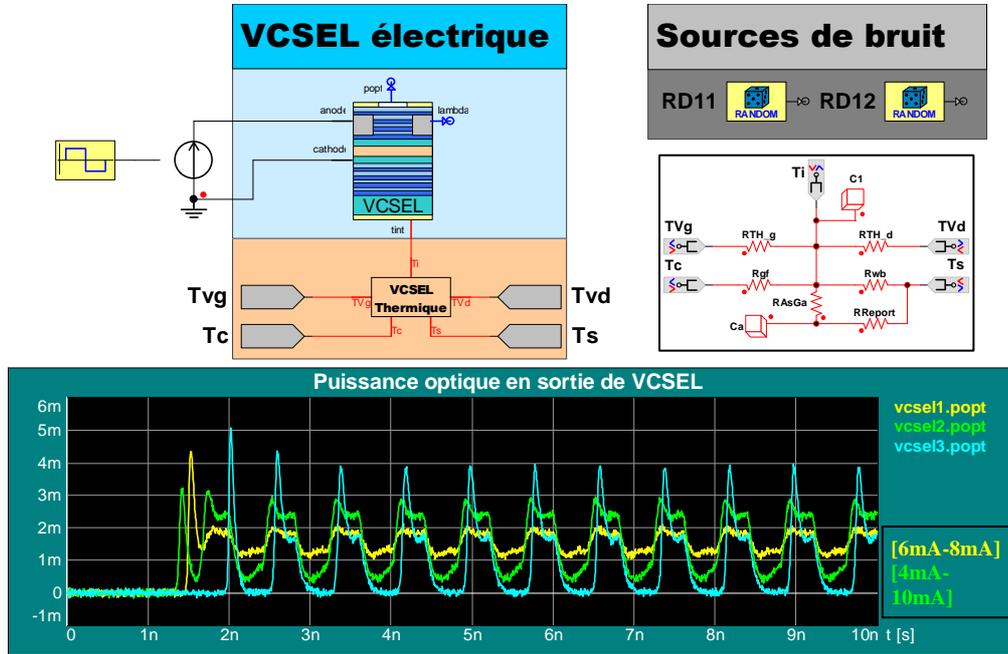


Fig. 46 - Schéma Simplorer du VCSEL et simulation non thermique avec bruit

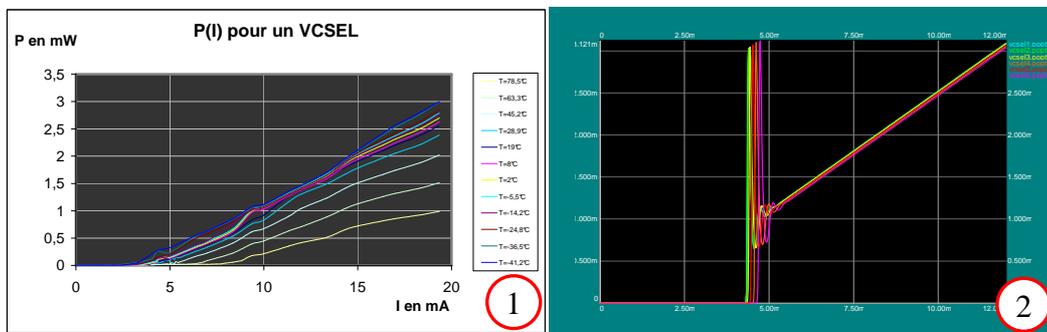


Fig. 47 - $P_{opt}(I)$ mesurée par THALES (1) et simulée avec Simplorer (2) pour différentes températures

On observe que l'ordre de grandeur semble bon, puisque autour de 10mA on retrouve la valeur maximale mesurée d'environ 1mW. Il y a par contre un saut de puissance optique inexplicable autour de 5mA qui semble dépendre des paramètres de simulation et une réponse à la variation en température un peu faible comparée aux mesures. Ces problèmes

ne semblent pas apparaître dans les simulations de nos collègues de l'ENST. Nous envisageons la possibilité que ces différences de comportement soient dues au portage du modèle sous Simplorer ou que le noyau de simulation de ce dernier pose des problèmes que nous n'avons pas encore résolus.

Le code VHDL-AMS associé au modèle de diode VCSEL se trouve en *Annexe VII*

- Le driver du VCSEL

Ce convertisseur tension-courant est un point sensible de la modélisation du système. En effet, il commande la diode laser en intensité et il est très sensible aux variations de température. Il est donc important d'avoir un modèle électrothermique fiable des transistors MOS qui composent le dispositif.

Notre tentative de portage direct du modèle de l'ENST s'est avérée être un échec. En effet, le code complexe du driver, faisant appel à de nombreuses fonctions imbriquées et des instanciations de composant qui ne se gèrent pas de la même manière sous Simplorer sont difficiles à gérer, n'a pas pu être simulé avec Simplorer.

Dans un premier temps, nous avons envisagé un modèle comportemental extrêmement simple qui avait pour fonction de restituer un signal de sortie en courant à deux valeurs fixées en fonction de l'entrée.

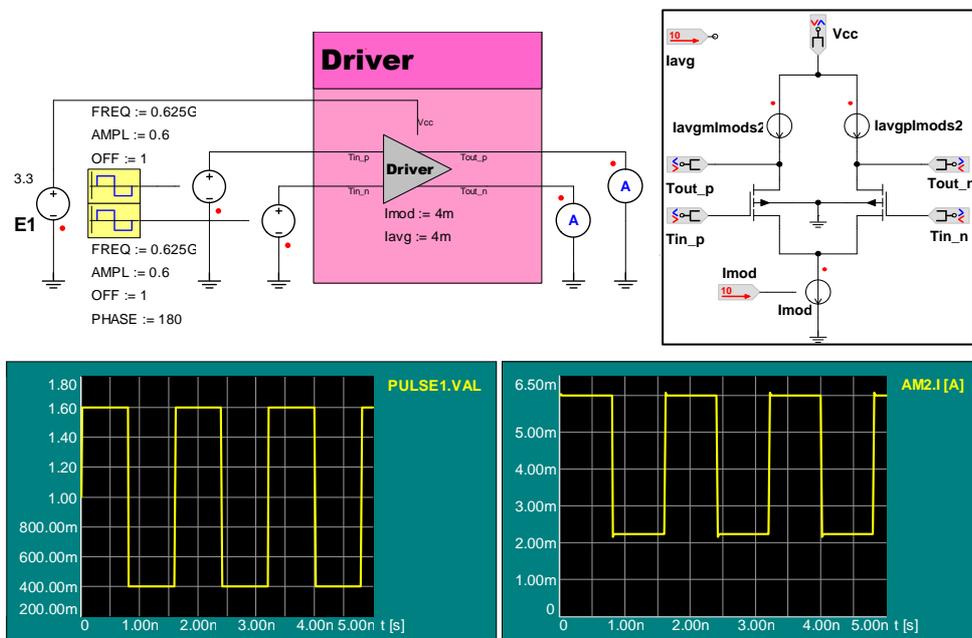


Fig. 48 - Schéma Simplorer et résultats de simulation du Driver

Par la suite, l'utilisation des modèles de transistors MOS compatibles SPICE disponibles dans Simplorer, nous a permis de réaliser un driver fonctionnel (*figure 48*).

Le modèle SPICE de niveau 1 correspond, pour sa partie électrique à celui implémenté par nos collègues de l'ENST. Pour ce qui est de la thermique, les modèles SPICE la gèrent, mais pas de manière dynamique. C'est à dire que la température de fonctionnement doit être fixée en début de simulation et est considérée comme constante après. Il faudra donc encore améliorer notre modèle pour pouvoir prendre en compte de manière dynamique les influences de la température sur le driver.

- La photodiode

Pour ce composant, comme pour le VCSEL, nous avons renvoyé à l'extérieur du modèle purement VHDL-AMS la gestion des sources aléatoires. Nous pourrions utiliser un bruit RIN comme dans le cas du VCSEL, mais étant donné que le portage du modèle incluant 3 sources de bruits sous forme de générateurs de courant ne nous a pas posé de problème de modélisation sous Simplorer, nous avons conservé sa forme initiale (*figure 49*).

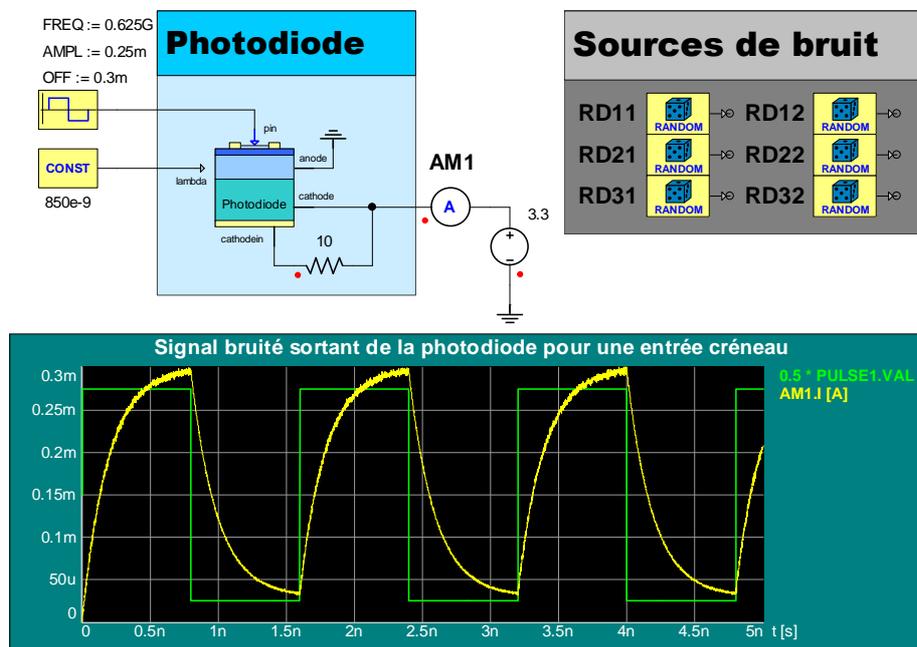


Fig. 49 - Schéma de la photodiode, sources de bruit associées et résultats de simulation

Notre étude s'étant principalement portée sur la partie émettrice du module, le point de vue thermique de la photodiode n'a pas été étudié.

Le code VHDL-AMS de la photodiode se trouve en *Annexe VIII*

- L'étage de sortie

Comme pour le driver, nous avons utilisé des modèles de transistors MOS issus de la bibliothèque compatibles SPICE de Simplorer. Ici aussi, nous avons utilisé des modélisations de niveau 1, avec le même inconvénient en termes de comportement thermique que pour le Driver.

La *figure 50* simule le comportement du TIA en lui injectant un signal bruité du type de celui attendu en sortie de photodiode pour vérifier son bon fonctionnement.

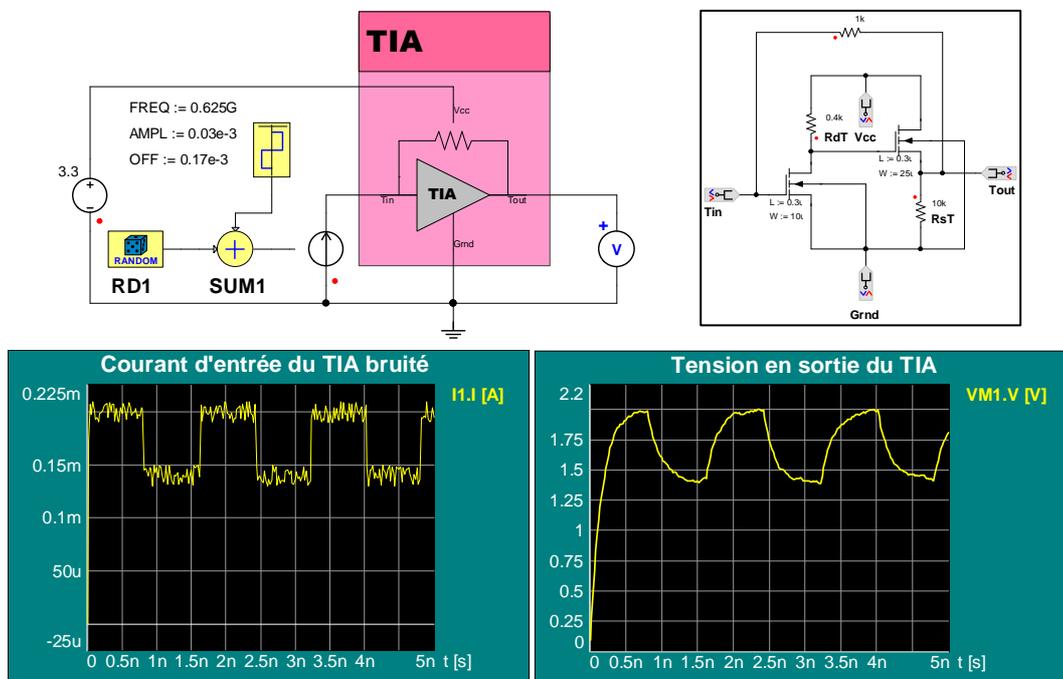
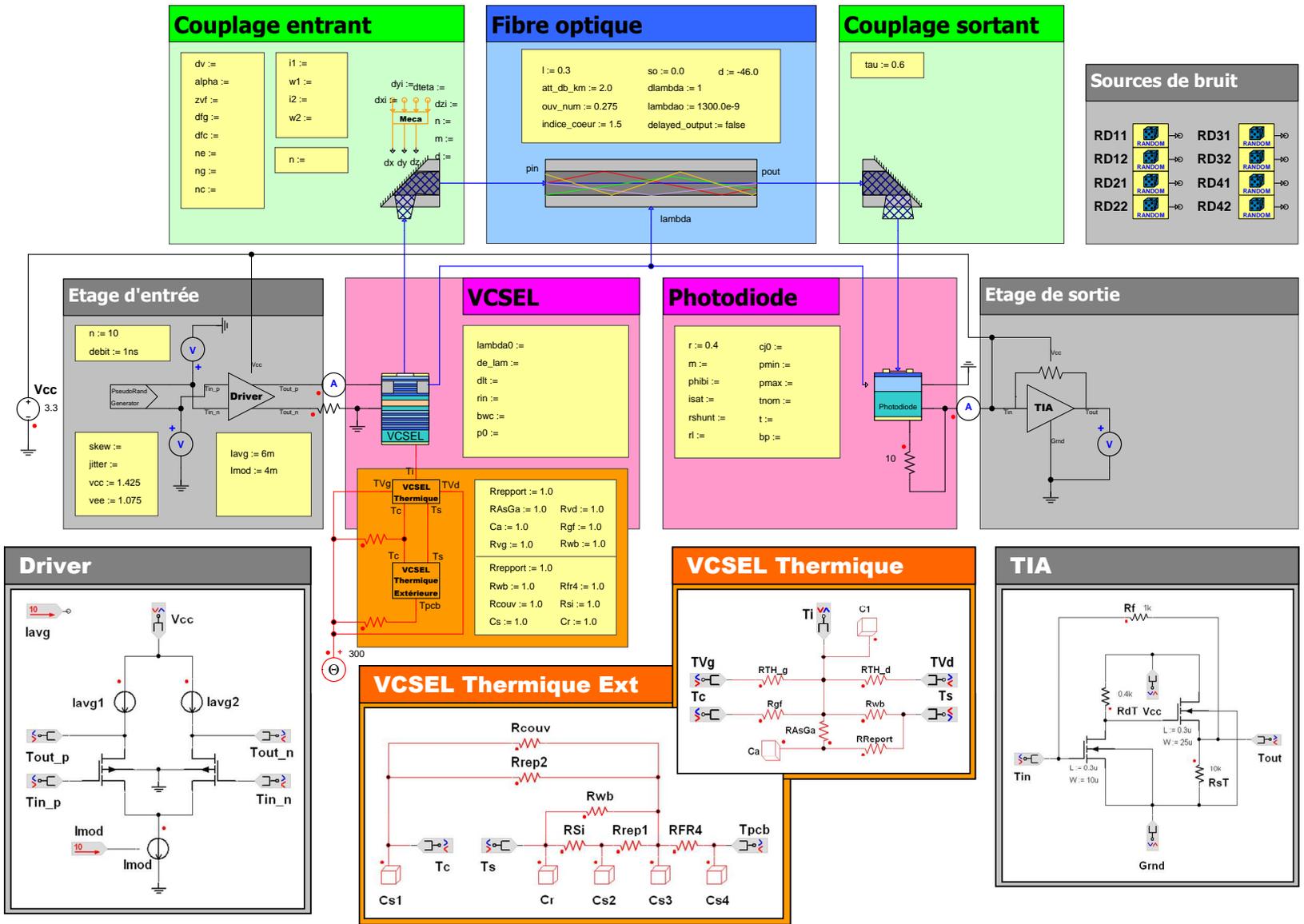


Fig. 50 - Circuit de test et résultats de simulation associés au TIA

- La modélisation d'un canal de transmission complet

En utilisant les possibilités offertes par l'interface de Simplorer, il est possible de matérialiser un canal complet par l'assemblage de ses différents constituants sur une même feuille de schéma (les variables laissées en blanc prennent les valeurs définies par défaut dans le modèle) :

Fig. 51 - Feuille de schéma Simplorer assemblant les composants d'un canal de transmission



Dans la *figure 51*, les encarts présentant les composants thermiques, ainsi que le driver et le TIA ne font pas partie de la feuille de schéma au niveau 0, mais sont des "sous-feuilles" de schéma (*subsheets*) qui permettent de détailler graphiquement le fonctionnement d'éléments sans surcharger la description du système global. Ils y ont été incorporés pour mettre en avant les possibilités offertes par Simplorer et distinguer les parties écrites en code des parties composées de composants préexistants.

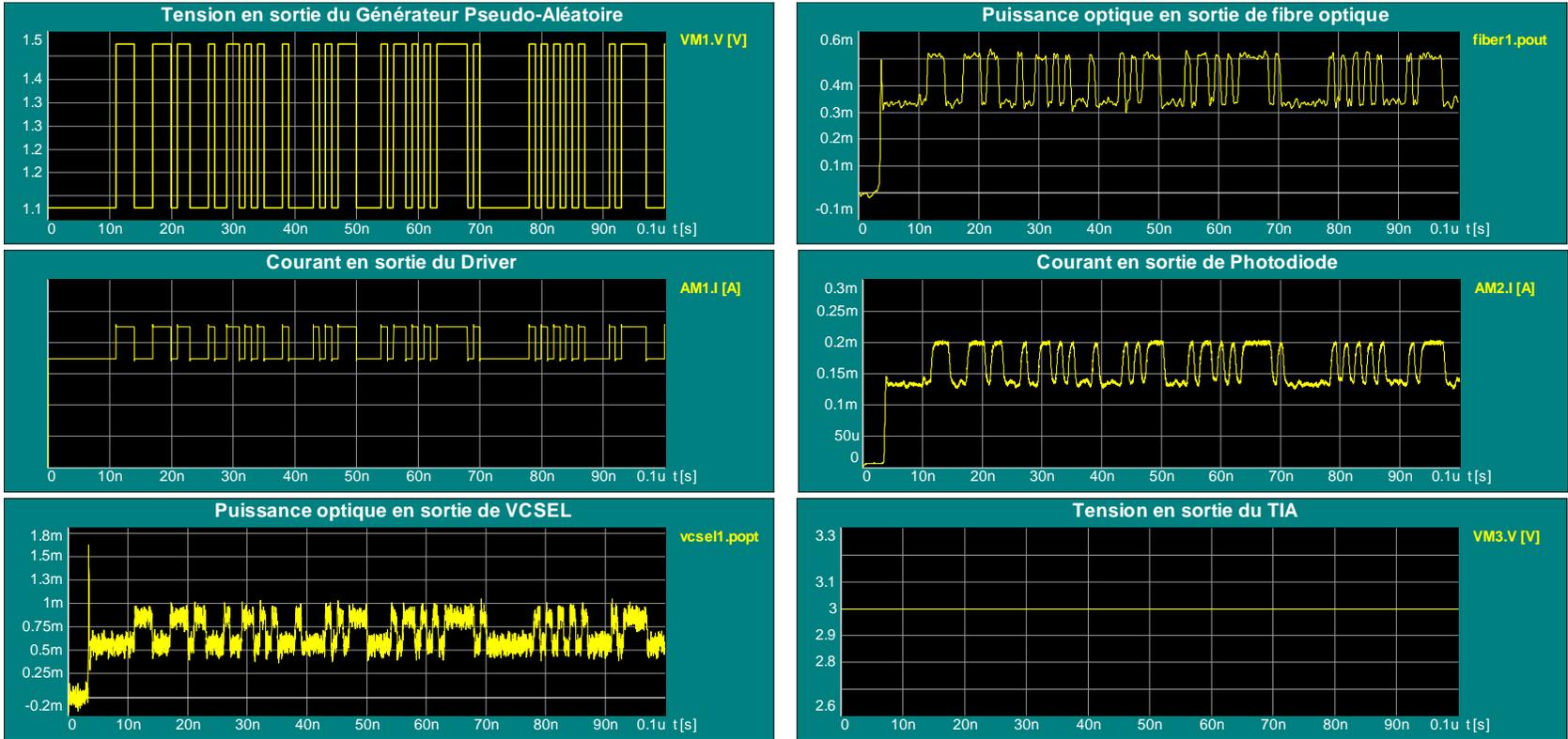
Cette *figure 51* illustre bien le haut degré d'abstraction auquel on peut prétendre avec les outils modernes. L'utilisateur qui aurait un tel schéma sous les yeux n'aurait nullement besoin de savoir de quoi les différents composants sont faits, du moment qu'ils se connectent et que la simulation tourne sans problèmes. Dans cette vue, les seules choses dont l'utilisateur doit se préoccuper sont les paramètres des modèles (mis ici en évidence dans les cadres jaunes) qui vont traduire les choix technologiques et modifier en conséquence les résultats.

Pour des raisons de présentation, nous avons dissocié les résultats de la *figure 52*, de la représentation du système (*figure 51*), mais ces deux ensembles de données peuvent sans problème apparaître sur la feuille de schéma, permettant ainsi d'afficher et d'archiver les résultats en vis-à-vis des paramètres utilisés pour la simulation.

Ces courbes sont issues d'une simulation où l'amplificateur transimpédance a été court-circuité, car les simulations l'incluant dans le circuit provoquent des erreurs de divisions par zéro dues probablement aux structures de bruit et aux mécanismes du simulateur Simplorer 6.0. Ce problème, en cours d'étude, n'a pas encore trouvé de solution.

Afin de créer des moyens d'étude du dispositif (*Annexe IX*), nous avons tenté d'obtenir un diagramme de l'œil du système. Pour ce faire, nous avons analysé le signal en sortie de photodiode. Simplorer ne disposant pas, contrairement à son concurrent SystemVision, d'outil automatique de création du diagramme de l'œil, nous avons utilisé un signal en dent de scie de période équivalente à trois bits. Avec l'outil d'analyse post-traitement de Simplorer, il est possible de représenter les signaux mémorisés les uns par rapport aux autres (représentation X/Y). On applique cette possibilité au signal que l'on veut étudier, en utilisant le signal en dent de scie en abscisse (*figure 53*).

Fig. 52 - Résultats de simulation du canal de transmission



Afin d'éviter l'affichage de traits horizontaux barrant le diagramme de l'oeil, nous passons en affichage des points de simulation.

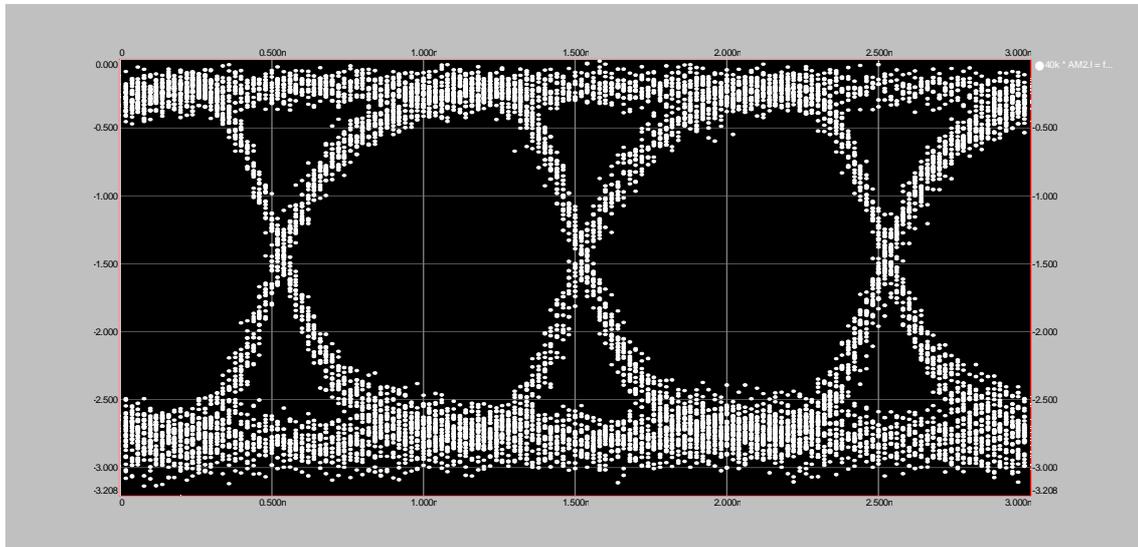


Fig. 53 - Exemple de diagramme de l'œil du canal de transmission

Une fois la méthode établie, cette opération est applicable à tout signal du dispositif, pour autant que l'utilisateur en ait requis la sauvegarde. Le logiciel de post-traitement permet aussi d'avoir recours à Matlab ou Mathcad pour travailler sur les données stockées.

3.7.3 - La simulation VHDL-AMS du POM complet

Pour pouvoir travailler sur le module complet, il a tout d'abord fallu transformer le schéma d'un canal en élément d'un schéma qui en instanciera plusieurs. Cette démarche a été relativement aisée, car Simplorer permet de créer un objet, à partir d'une sélection graphique, qui constituera un élément dans une vue d'abstraction supérieure. Il faut alors spécifier les grandeurs qui vont constituer les entrées/sorties de l'abstraction du canal.

Cette opération a été réalisée dans la **figure 54** où nous avons supprimé les éléments de mesure et "externalisé" les générateurs de signaux, ainsi que l'alimentation du circuit.

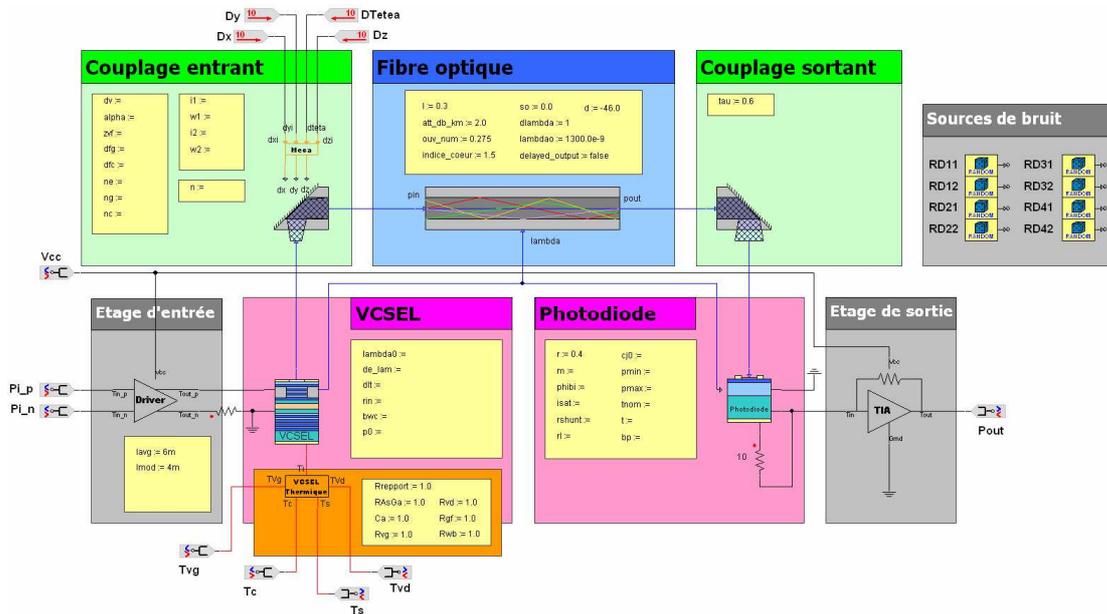


Fig. 54 - Transformation d'un schéma en élément d'un système plus important

En utilisant ce mécanisme, il a été possible de réaliser un schéma du montage global. A titre d'exemple, mais également pour la lisibilité du schéma, nous avons travaillé avec un module à trois canaux (**figure 55**).

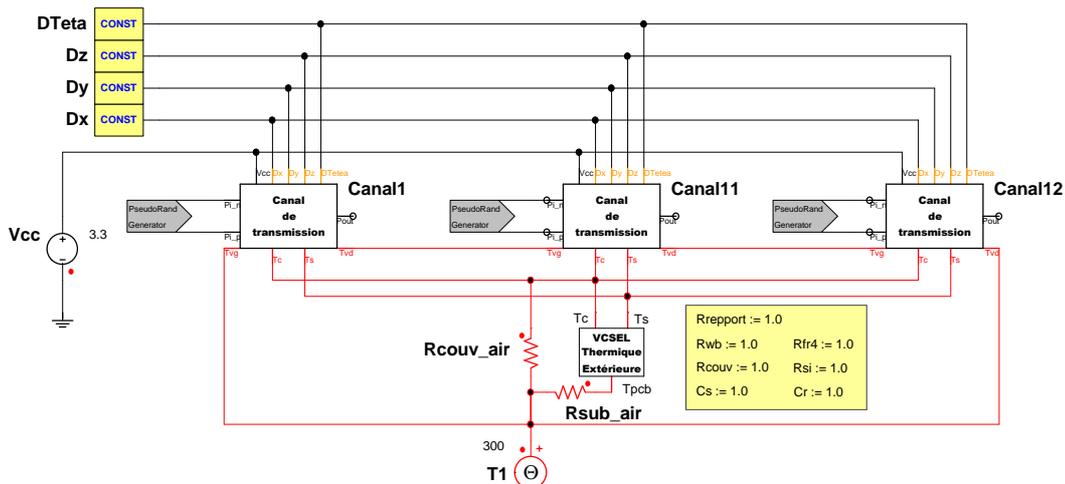


Fig. 55 - Feuille de schéma du module complet à trois canaux avec circuit thermique des VCSELs

A partir de ce circuit, nous avons voulu simuler le comportement des trois canaux simultanément et tenter ainsi d'observer les effets thermiques des VCSELs les uns sur les autres. Cette simulation nous a cependant semblé être d'un intérêt limité, étant donné que le réseau thermique n'influe pas sur le comportement des transistors MOS du driver dans cette modélisation, et qu'il est donc impossible d'en évaluer l'influence sur la puissance optique émise par un VCSEL. En effet, comme nous l'avons déjà évoqué, la modélisation SPICE utilisée pour simuler efficacement le comportement des transistors du circuit, utilise la température comme un paramètre initial qui n'est plus actualisé durant la simulation. Même si l'ensemble des fonctions du dispositif n'est actuellement pas complètement analysable du fait de l'insuffisance de certains modèles, la structure de simulation est bien en place et prête à accueillir des changements dans les composants sans que cela ne change l'apparence du circuit, et donc la vision de l'utilisateur.

3.8 - Conclusion

A l'issue du travail effectué sur le projet SHAMAN, nous avons donc pu réaliser la capitalisation des connaissances issues des membres du groupe au travers d'une bibliothèque de modèles VHDL-AMS qui propose des modèles majoritairement opérationnels, excepté dans certains aspects thermiques. Le commanditaire du projet pourra donc disposer, après quelques modifications, d'un ensemble de composants paramétrables et réutilisables lui permettant de simuler son module optoélectronique tant au niveau comportemental pour les industriels que système pour ses recherches et développements.

Méthodologie de modélisation des systèmes complexes

Alors que les systèmes conçus par l'industrie se complexifient à un rythme effréné, leurs méthodes de conception n'ont globalement pas subi d'avancées majeures par rapport à la structure classique du cycle de conception en V (*figure 56*).

La propagation des langages de modélisation, comme VHDL-AMS, depuis les domaines universitaires vers ceux de la R&D industrielle, s'accompagne d'une révision en profondeur des méthodologies de conception utilisées. L'outil informatique permet en effet d'entrevoir des gains conséquents en temps et en argent qui sont des enjeux capitaux pour les entreprises. L'expression de ce potentiel passe par une communication efficace des nombreux acteurs de la conception au travers d'un protocole défini et d'un outil commun.

En comparant le cycle de développement classique au travail que nous avons réalisé, mais aussi à ce que nous aurions rétrospectivement pu faire au cours de ce projet, nous allons tenter de dégager les grandes lignes méthodologiques du travail de modélisation de systèmes complexes, à l'image de celui sur lequel nous avons travaillé.

4.1 - Le cycle de conception traditionnel en V

Le travail effectué couramment par les équipes de développement au sein des entreprises peut être schématisé suivant l'organigramme de la figure ci-dessous.

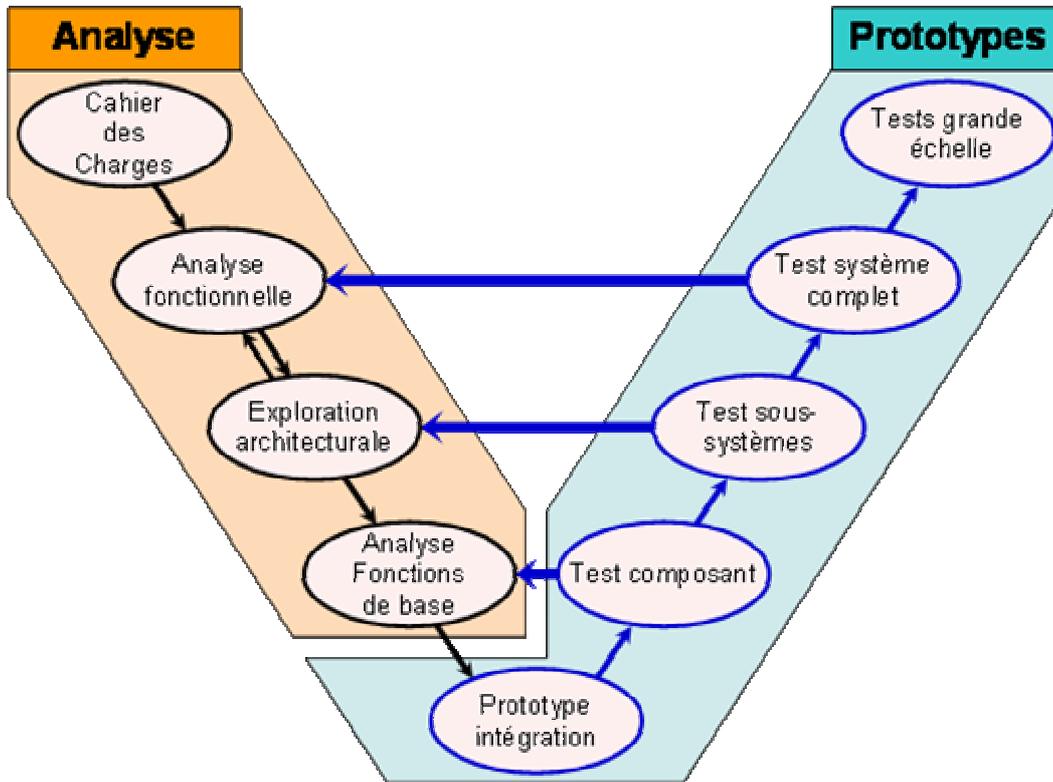


Fig. 56 - Cycle de conception en V

Ce graphique comporte deux zones distinctes :

- Tout d'abord un travail d'analyse en descendant

Ce travail permet de déterminer, à partir du cahier des charges, le découpage d'un système en blocs fonctionnels. Comme leurs noms l'indiquent, ces entités correspondent aux différentes tâches que le système doit accomplir, mais n'ont pas forcément de relations directes avec les composants qui les réaliseront. Ce découpage donne ensuite lieu à une exploration architecturale dont "l'objectif consiste [...] à définir l'architecture, le nombre et le type de ressources ou de composants (d'exécution, de communication et de mémorisation) pour exécuter l'ensemble des fonctionnalités du système tout en respectant

les contraintes dictées par le contexte d'utilisation (performances, taille mémoire...) ainsi que la date d'arrivée sur le marché [...]. Une tendance est d'utiliser de plus en plus de processeurs programmables, de composants sur étagère et d'IPs; la part de conception de logiciel devient ainsi largement prépondérante par rapport à la conception de matériel (souvent nulle)" [MIRI]. Cette dernière tendance s'expliquant probablement par le fait que la conception de composants dédiés serait tellement fastidieuse avec ces méthodologies qu'il est plus intéressant d'utiliser des briques matérielles simples et d'y ajouter une couche logicielle plus élaborée. Une fois ce travail accompli, les composants des couches les plus basses du circuit sont définis.

- Ensuite un travail de test et de correction d'erreurs

Une fois que les éléments constitutifs du système ont été déterminés par l'analyse précédente, un prototype en est réalisé. On détermine alors par le test systématique de chaque composant s'ils remplissent bien leurs offices. Dans le cas contraire, un retour à l'analyse des fonctions de base est nécessaire, afin de définir une nouvelle palette de composants et de créer un nouveau prototype. Cette étape de test passée avec succès, vient la phase de vérification du bon fonctionnement des sous-systèmes du dispositif, ou ensemble de composants, qui peut déboucher comme précédemment sur des erreurs et un retour sur les analyses structurelles et fonctionnelles qui permettront de créer de nouveaux prototypes jusqu'à atteindre la validation des tests. Finalement, viennent les tests du système complet, qui, s'ils réussissent, débouchent sur l'ultime phase de tests à grande échelle précédant la mise sur le marché.

Notons que même s'il s'agit d'un cycle de conception simple, exploité à titre indicatif, et pouvant comporter des variantes, des parties récursives ou plus complexes dans sa mise en œuvre sur le terrain, il n'en reste pas moins l'illustration du temps et de l'argent passés à la correction des erreurs commises dans le cadre de cette méthodologie.

Par l'introduction des nouvelles technologies de modélisation du matériel et du logiciel, le Prototypage Virtuel permet aujourd'hui de développer des méthodologies qui devraient amener à terme à court-circuiter la quasi-totalité du processus de correction d'erreur sur prototype réel, favorisant ainsi de larges économies d'argent, mais également de temps.

4.2 - Le Prototypage Virtuel

4.2.1 - Les objectifs du Prototypage Virtuel

Dans le cadre très large d'un projet visant à mettre un produit sur le marché, la partie interne du projet concernant le prototypage virtuel, qui a pour but de réduire les temps et les coûts d'analyse et de production, peut avoir un ou plusieurs objectifs à atteindre qui constituent sa raison d'être.

Ces derniers peuvent être très différents. Il faut tout d'abord prendre en compte l'existence préalable ou non d'un produit physique (qu'il soit à l'état de prototype ou commercialisé). Dans le cas d'un produit virtuel, la seule base de travail dont disposent les modélisateurs repose sur les spécifications fonctionnelles de ce dernier. Par contre, dans le cas d'un dispositif préexistant, les spécifications originelles ont été complétées par les feuilles de caractéristiques qui rendent compte des résultats réels du dispositif et il est toujours envisageable de mener une campagne de mesure sur le système pour observer un comportement particulier (réaction à la température, au rayonnement, ...) et en constituer une modélisation empirique.

Par ailleurs, les objectifs diffèrent également par leur nature. Il peut, par exemple, s'agir d'une étude sur la conception d'un ou plusieurs composants du dispositif, d'une étude sur les possibilités de réduire les contraintes des procédés de fabrication, d'une étude sur le choix des composants à utiliser en fonction des performances ou du prix, ou encore de l'étude des modes de défaillance d'un système. Toutes ces directions de recherche n'auront probablement pas les mêmes implications en terme de niveaux d'abstraction pour la modélisation des composants.

Avant d'entamer la phase de conception des modèles, il convient d'établir un bilan des ressources dont le groupe d'étude dispose. Il faut en effet faire l'examen des bibliothèques de modèles existantes (celles du groupe, celles des fabricants ou d'autres IP tierces [MIL]) à la recherche d'éléments, physiques ou comportementaux, pouvant subvenir directement à nos besoins ou du moins servir de base de développement. Il faut également recenser les compétences techniques nécessaires à la modélisation et vérifier leur adéquation avec les ressources humaines du groupe d'étude. Enfin, la définition des normes et standards

auxquels le système doit adhérer, donne des détails importants sur les caractéristiques limites du système.

Une fois ces éléments analysés, on peut alors établir les spécifications du système que l'on veut modéliser et mobiliser les compétences nécessaires au bon déroulement de la modélisation. Cette étape est encore aujourd'hui le fruit d'un travail manuel, car il n'existe pas pour le moment d'outils formels permettant de réaliser cette étape d'expression des spécifications de manière formelle [HER4].

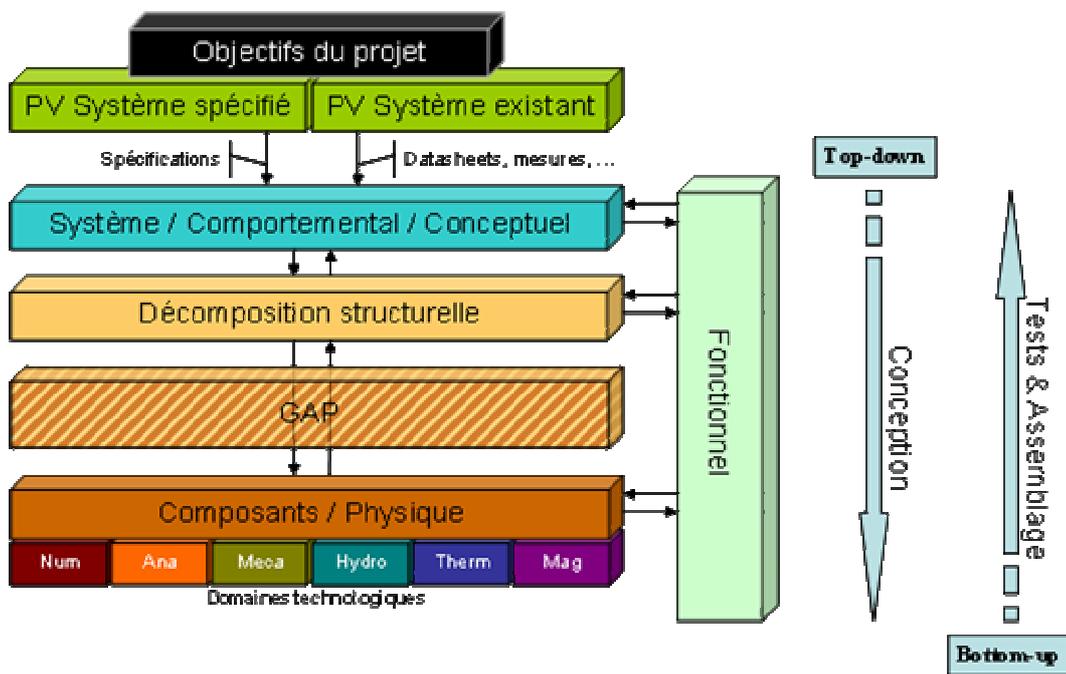


Fig. 57 - Niveaux d'abstraction pouvant intervenir dans le processus de modélisation d'un système

Alors que dans le domaine de la modélisation numérique les niveaux d'abstraction d'un système sont assez bien définis, il n'en va pas de même pour les modélisations analogiques ou à signaux mixtes et encore moins pour les modélisations multi-domaines (**figure 57**).

En effet, le monde analogique offre une multitude de niveaux d'abstraction possibles selon les besoins du concepteur (vitesse ou précision) et les données dont il dispose (paramètres physiques, feuilles de caractéristiques, mesures). On peut cependant essayer de regrouper les modèles selon leur appartenance à trois catégories qui s'articulent autour du développement d'un modèle : les modèles comportementaux, les modèles physiques et les modèles fonctionnels.

4.2.2 - L'analyse et la conception "Top-down" des modèles

Comparable à l'analyse menée dans le cycle en V traditionnel, le procédé de conception que l'on utilise est qualifié de "Top-down", car il part d'une vue extérieure au dispositif et à chaque étape de l'analyse il s'enfonce un peu plus dans une vue détaillée du système jusqu'à ce que l'on ait atteint le niveau de détail suffisant pour remplir les objectifs du projet.

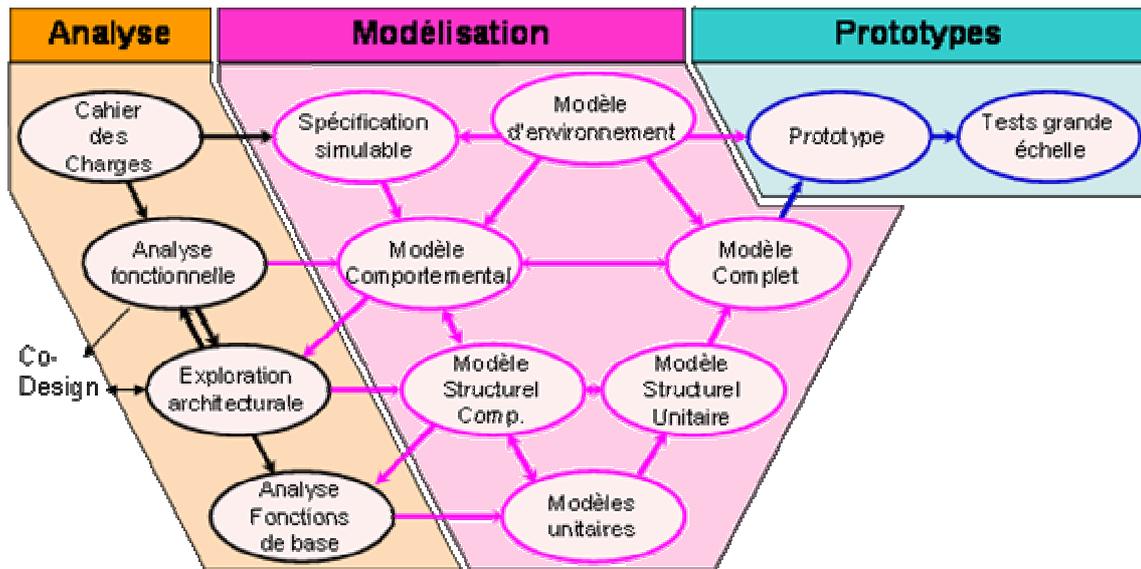


Fig. 58 - Cycle idéal de conception par Prototypage Virtuel

Une fois les spécifications établies, la modélisation commence avec un niveau dit comportemental qui décrit le système comme une "boîte noire". On descend ensuite progressivement dans les niveaux d'abstraction selon nos besoins (*figure 58*).

Si l'on souhaite modéliser un système à un haut niveau afin, par exemple, de pouvoir en présenter l'efficacité à un client, la simulation devra être la plus rapide possible et il ne sera alors pas nécessaire de le décrire plus en détail qu'au niveau comportemental. Si en revanche il s'agit d'étudier l'influence d'un changement de technologie de gravure dans les transistors employés, il faudra alors descendre bien plus bas dans le niveau de modélisation de certaines parties du système.

- **La description comportementale globale du système**

Ce passage obligatoire de la conception d'un dispositif consiste à traduire les spécifications du système établies plus haut en un "cahier des charges simulable" qui reproduise le

fonctionnement du système en établissant des relations entre les entrées et les sorties comme s'il s'agissait d'une "boîte noire". Ce travail est actuellement réalisé par des spécialistes des HDLs, car il n'existe pas encore d'outils permettant de passer automatiquement de la description des spécifications au modèle comportemental global [HER4].

Comme son nom l'indique, ce modèle ne fait que reproduire le comportement du système en utilisant les données disponibles le concernant. Dans le cas des spécifications fonctionnelles, la modélisation comportementale se situe au niveau de ce qui serait souhaité par le groupe d'étude. Il se peut que devant des impératifs financiers ou technologiques, des solutions, donnant un résultat en désaccord avec les spécifications, s'imposent. Dans le cas où l'équipe de modélisation utilise des données contenues dans les feuilles de caractéristiques ou issues d'une série de mesures, la modélisation proposée peut refléter la réalité avec une précision extrême, mais sans aucun lien avec le fonctionnement interne du dispositif. Ce type de modèle descriptif, conçu pour simuler le fonctionnement du système dans une plage d'utilisation "normale", ne peut en aucun cas gérer les cas de figures pathologiques où le système est mis en déroute à cause de son fonctionnement interne ou d'une utilisation en dehors de son domaine de validité.

Les apports d'un modèle comportemental sont multiples. Il fournit tout d'abord une base à laquelle on pourra se référer pour valider toute simulation ultérieure des éléments de plus bas niveau constitutifs du système. Ensuite, un modèle comportemental est jusqu'à 1000 fois plus rapide [MEN2] à simuler qu'un modèle bas niveau et rend bien compte de son fonctionnement standard.

Les modèles comportementaux globaux sont donc des modèles descriptifs d'un système et constituent une étape indispensable qui servira de référence vers une modélisation de plus bas niveau.

- *L'étape structurelle*

Il est rare que les objectifs d'un projet soient satisfaits par la simulation comportementale d'un dispositif. Il faut alors ôter le couvercle de la "boîte noire" du système pour en observer le fonctionnement et isoler les groupes physiques ou fonctionnels qui le constituent.

Il ne s'agit pas d'un niveau de modélisation au sens de la simulation, mais plutôt au sens de l'expression du modèle. En effet, l'analyse structurelle établit les liens entre les blocs

constituant le système, indépendamment de leur implémentation comportementale ou physiques.

Sur le plan de la simulation, nous utilisons la représentation fonctionnelle qui substitue aux composants abstraits entre lesquels des liens ont été établis, dans le niveau structurel, leurs modèles comportementaux ou physiques, selon les besoins.

Sur de gros systèmes, ou sur des systèmes multidisciplinaires, l'identification des éléments constitutifs d'un modèle structurel requiert l'intervention de spécialistes des domaines physiques ayant trait au dispositif. Il s'agit en effet, pour eux, de choisir les sous-ensembles à isoler, en fonction des interactions possibles entre eux et des niveaux de modélisation que l'on souhaite associer à chacun selon les objectifs recherchés.

Effectivement, chaque élément du modèle fonctionnel peut à son tour faire l'objet d'une analyse structurelle, et ainsi de suite, jusqu'à descendre au niveau de modélisation physique des différents composants.

Dans l'état actuel de la normalisation VHDL-AMS, les connexions spécifiées dans une ENTITY (interface de communication entre le modèle et son environnement), ne peuvent être modifiées en fonction de l'architecture simulée. Or, en fonction du niveau de modélisation, nous pouvons être amenés à percevoir les interfaces différemment.

A titre d'exemple, si l'on considère un circuit RC au niveau comportemental, nous pourrions être tentés de le modéliser par un filtre très polyvalent utilisant une interface à base de quantités. Si par contre nous descendons au niveau physique, nous percevons l'association d'une résistance et d'un condensateur comme un circuit électrique et nous utiliserions probablement des interfaces à base de terminaux pour représenter le système. Une telle démarche n'est pas possible directement, il faut actuellement utiliser toujours des terminaux et insérer des étapes supplémentaires dans notre modèle comportemental pour traiter des quantités simples en entrée et en sortie.

En attendant que ce problème trouve une solution, avec l'abstraction des connexions, dans de futurs développements de VHDL-AMS, les concepteurs doivent être vigilants sur leurs choix d'entrées/sorties dès le départ.

- ***La description physique des composants***

Lorsque l'on souhaite une modélisation prédictive très fine d'un composant, il est nécessaire de descendre très profondément dans ses mécanismes internes pour faire

ressortir les équations qui gouvernent son comportement et permettent d'appréhender les effets parasites les plus fins.

Ces modèles qui sont en général beaucoup plus lents que leurs homologues comportementaux ont pour eux la "prédictivité" et la précision de leurs résultats. De tels modèles sont en effet capables de traduire les comportements atypiques qui échapperaient totalement à une modélisation de plus haut niveau.

La mise au point de ces modèles nécessite l'intervention de tous les domaines de compétences. Les spécialistes de la physique doivent développer les modèles mathématiques rendant compte au mieux de la subtilité du fonctionnement des composants, alors que les spécialistes des langages tenteront de trouver la solution informatique la mieux adaptée au traitement du modèle physique.

Une fois ces étapes explorées, nous disposons d'une bibliothèque de modèles VHDL-AMS qui permettent de décrire tout ou partie d'un système. Il reste ensuite, éventuellement, à assembler ces différentes briques pour tester et valider notre modélisation.

4.2.3 - La vérification "Bottom-up" des modèles

La vérification des modèles n'intervient pas forcément à la toute fin du travail. Elle peut, et même devrait, intervenir après chaque nouveau développement.

Cette étape décisive du processus de modélisation est dans une logique dite "Bottom-up" car chaque niveau de modélisation est comparé au fonctionnement des objets de niveaux d'abstraction supérieurs créés en amont.

Le premier stade de ce processus se situe au niveau du modèle comportemental global qui doit bien évidemment se conformer aux spécifications ou autres données qui ont servi à établir ses lois de fonctionnement. Dans le cas d'un produit existant, on peut confronter la simulation à la mesure pour valider la modélisation.

En descendant à l'intérieur du découpage structurel du système, nous devons bien sûr vérifier la cohérence entre la simulation de l'association des sous modèles comportementaux et celle du modèle comportemental global. Plus généralement, chaque décomposition comportementale d'un élément doit correspondre à la modélisation comportementale du composant.

En arrivant au bas de l'échelle d'abstraction, les modèles que nous devons vérifier deviennent physiques. Si ces modèles prédictifs correspondent à un composant aux grandeurs accessibles et mesurables, le meilleur indice de leur bon fonctionnement est celui de la confrontation avec les mesures expérimentales. Dans le cas contraire, il est important d'inclure le modèle bas niveau au sein d'une structure comportementale vérifiée qui permette de le valider dans la plage de fonctionnement "normale" du dispositif par comparaison avec les simulations purement comportementales du système.

On peut ainsi progressivement vérifier le bon fonctionnement de chaque constituant d'un système, puis décliner les assemblages à loisir jusqu'à simuler, par exemple, un schéma ne comportant que des modèles de bas niveaux. Ce dernier sera probablement beaucoup plus lent que le modèle comportemental global, mais il aura l'avantage de fonctionner dans n'importe quelle condition.

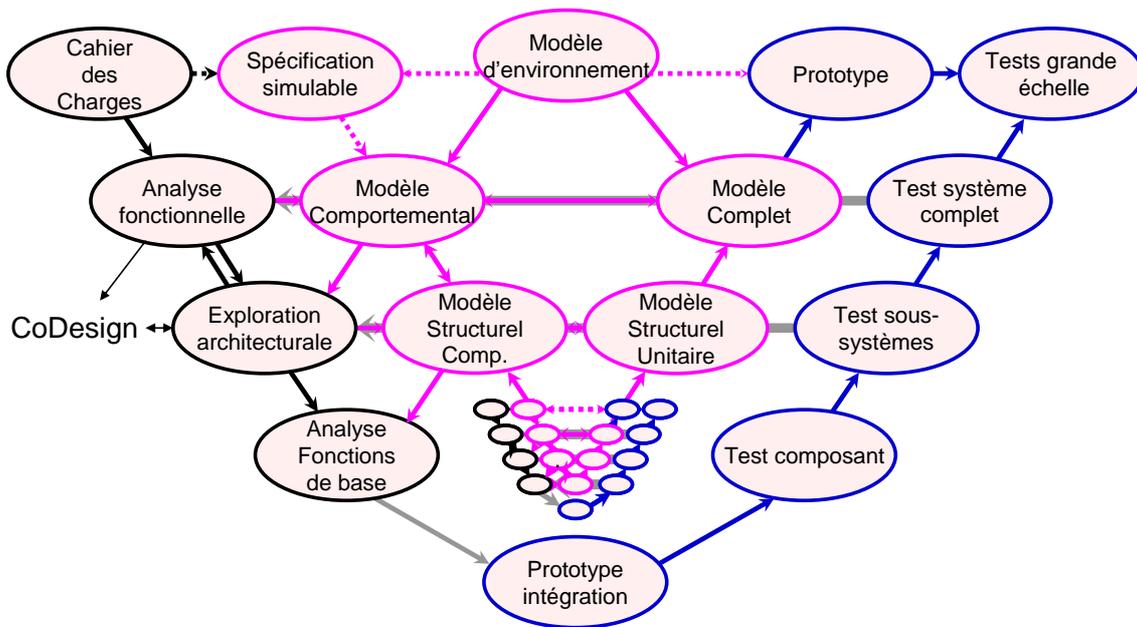


Fig. 59 - Cycle récursif réel de conception par Prototypage Virtuel

Comme l'illustre la **figure 59**, le cycle de conception par le Prototypage Virtuel n'en est pas encore à l'extrémité idéale présentée dans l'organigramme de la **figure 58**. Les méthodes de modélisation permettent certes d'accélérer le processus de conception en économisant des prototypes réels intermédiaires trop nombreux, mais ces derniers restent souvent indispensables à cause du manque, voir de l'absence, de maîtrise des outils et des méthodes

de modélisation HDLs. Notons que la faute incombe également à l'implémentation incomplète de la norme qui limite l'expression du potentiel complet de ces langages.

Une fois l'ensemble de ces étapes passées, le système est validé et les résultats de simulations prêts à passer dans les mains d'une autre équipe qui guidera le produit vers sa réalité physique. Il nous semble important de remarquer ici que, devant la palette des possibilités de modélisation, peu de systèmes restent non simulables, même s'ils ne peuvent pas tous être implémentés à un niveau physique. Cependant, les projets devenant de plus en plus complexes, il est maintenant rare qu'un seul corps de métier ou groupe de spécialistes puisse maîtriser suffisamment bien les différents éléments d'un système pour en permettre une modélisation satisfaisante. C'est pourquoi des collaborations interviennent à l'intérieur de ces cycles de conception.

4.3 - L'intégration des collaborations dans le PV

La collaboration entre différents acteurs de la modélisation ou bien la sous-traitance d'une partie à des spécialistes nécessite l'emploi d'un formalisme commun de description qu'apporte la méthodologie, les langages et les logiciels à des niveaux différents. En effet, la méthodologie et les normes de gestion de projets apportent un découpage standardisé des tâches et une garantie de résultats. Les langages, quant à eux, autorisent une communication très aisée entre les différents intervenants qui peuvent, par exemple, définir en commun des interfaces à leurs composants, puis développer, en interne, les modèles comportementaux ou physiques qui leur incombent. L'utilisation de logiciels de simulation multi-langages permet de connecter entre eux des modèles issus de milieux différents et de pallier parfois à l'impossibilité d'une communication basée sur un langage de modélisation unique.

Nous illustrons ci-dessous notre propos avec deux exemples de collaborations.

La *figure 60* propose un schéma qui peut s'apparenter au projet SHAMAN, puisqu'il s'agissait pour nous de développer un modèle complet d'un module existant déjà. On peut donc considérer que THALES nous a fourni le cahier des charges et l'analyse fonctionnelle du module, ainsi que toutes les données nous permettant de développer rapidement le

modèle comportemental et un environnement de test. Dans le cadre d'une maîtrise du VHDL-AMS par THALES, il aurait été intéressant que ce soit eux qui nous fournissent les modèles comportementaux et d'environnement. Ceci aurait probablement permis une communication plus aisée, plus rapide, et peut-être aussi un contexte de test plus neutre des "sous-modèles" que nous avons développés.

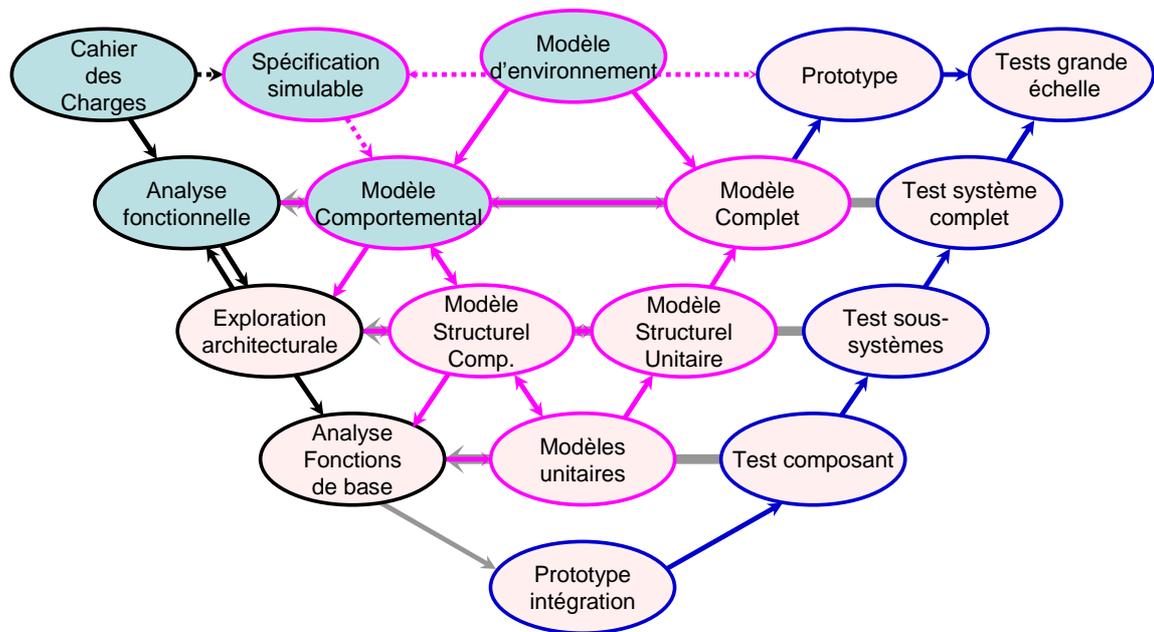


Fig. 60 - Données fournies par le commanditaire (en bleu), dans le cadre du développement d'un modèle complet

On peut également considérer l'exemple d'un fabricant de voitures qui a besoin d'un nouveau composant n'entrant pas dans son domaine d'action, mais dans celui des équipementiers auxquels il sous-traite une partie du travail. Une organisation comme celle présentée en **figure 61** suppose que la société ait développé en interne l'étude et la modélisation structurelle de son système, hormis le composant manquant pour lequel seul un modèle comportemental est disponible. Ce dernier peut permettre dans un premier temps de lancer un appel d'offre auprès des sous-traitants, afin qu'ils étudient les performances à fournir et estiment la faisabilité et les coûts. Il peut ensuite servir à tester les composants développés par comparaison avec les résultats comportementaux.

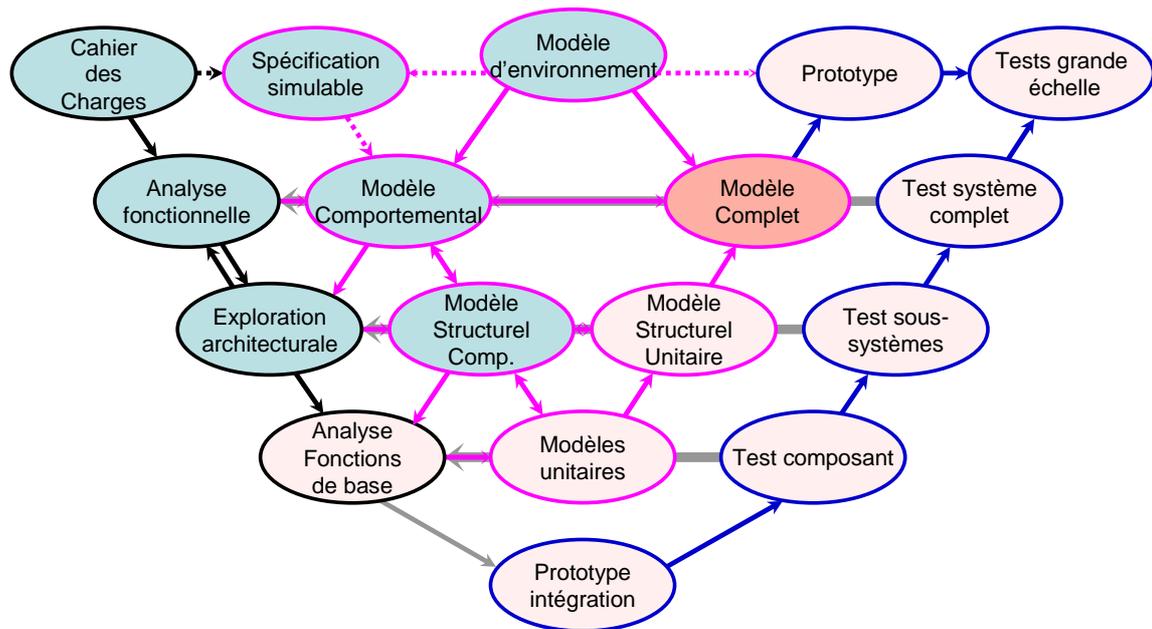


Fig. 61 - Données fournies par le commanditaire (en bleu), dans le cadre du développement d'un composant manquant pour une modélisation complète (en rouge)

Ces deux exemples d'organisation permettent d'imaginer les conditions de communication à développer au sein des collaborations pour permettre une amélioration des conditions d'échange de données et une réduction des délais de développement.

4.4 - Conclusion

Alors que les HDLs, et notamment VHDL-AMS, arrivent à maturité et s'implantent de plus en plus profondément dans le paysage industriel, la productivité et les contraintes du marché imposent la création de nouvelles méthodologies de développement et de conception qui sont aujourd'hui à l'essai, mais promettent d'améliorer considérablement l'organisation, la communication et les délais au sein des projets les plus complexes

Conclusion et Perspectives

Conclusion

Devant la réduction constante des temps de gestation des projets industriels, la miniaturisation et la complexification galopante qui l'accompagne, les outils de CAO multi-domaines, multi-sigaux et multi-abstractions sont devenus une nécessité pour améliorer la productivité et réduire les coûts de développement à l'aide du prototypage virtuel.

Le langage VHDL-AMS, encore jeune, et les outils les plus récents qui lui sont associés permettent aujourd'hui d'envisager la construction, par les spécialistes du langage, de bibliothèques de modèles généralistes dans lesquelles les concepteurs de systèmes pourront puiser, sans pour autant avoir besoin d'une connaissance approfondie des HDLs sous-jacents.

Le travail que nous avons effectué sur le projet SHAMAN est une illustration des possibilités dont le monde industriel dispose pour modéliser des systèmes complexes et multidisciplinaires. Le fait qu'il s'agisse d'un projet pilote dans le domaine nous a fait rencontrer beaucoup des problèmes liés à cette activité, et nos tentatives pour les résoudre nous ont fait sortir quelque peu du cadre industriel et de ses impératifs temporels. Cette démarche aura été salutaire puisque le RMNT qui a encadré et subventionné le projet nous a cité comme projet ayant fait tomber des barrières technologiques dans le domaine de la conception [RMN1]

Les problèmes auxquels nous avons été confrontés nous ont donné l'opportunité de réfléchir aux méthodes à appliquer dans le cadre du travail de modélisation des systèmes complexes. Nous espérons que les pistes que nous avons explorées et décrites ici donneront

aux futurs groupes de modélisation une base méthodologique qui leur fera gagner du temps en économisant les tâtonnements inhérents à toute nouvelle discipline.

A la lumière de notre étude, nous pensons que la modélisation des systèmes complexes, qui est une nécessité pour les entreprises, devient aujourd'hui une réalité grâce aux langages de modélisation comme VHDL-AMS et ceux qui lui sont associés (C/C++, Matlab/Simulink, Mathcad et outils de CAO spécifiques), à condition de savoir mettre en relation des spécialistes d'horizons divers et de développer des projets de prototypage virtuel en respectant les étapes méthodologiques de conception et de validation des modèles.

Même si VHDL-AMS est très adapté au travail de décomposition structurelle et à la conception de modèles, il ne couvre pas aujourd'hui les besoins associés à l'expression et à la traduction des spécifications d'un système et n'offre pas de mécanismes automatisés de vérification des modèles.

Perspectives

Les marchés et les technologies sont toujours appelés à évoluer. Les outils de conception que sont les langages de modélisation des systèmes doivent le faire en parallèle. Des langages normalisés suite au travail et aux réflexions communes des chercheurs et des industriels, comme VHDL-AMS, ont les moyens d'être toujours en phase, voir en avance sur les besoins industriels de développement.

Une grande partie des besoins actuels se situent dans l'expression et la traduction automatisée des spécifications d'un système. Différentes ébauches de solutions sont proposées actuellement au travers de langages autres que VHDL-AMS (SysML, HiLeS), mais il existe aussi des possibilités d'apporter ces fonctions à VHDL-AMS au moyen d'extensions du langage [HER4].

L'accroissement des puissances de traitement et de simulation, par l'évolution des ordinateurs et des langages autorisera la conception de systèmes toujours plus complexes, et permettra peut-être à long terme de maîtriser la synthèse analogique comme on peut le faire aujourd'hui dans le monde du numérique. Ceci pourrait constituer les premier pas vers des systèmes intelligents qui pourraient se concevoir, se réparer, s'améliorer ...

Bibliographie

- [ANS1] "ANSOFT Corporation"
<http://www.ansoft.com/products/em/simplorer/>
- [ANS2] "Simulation System Simplorer 6.0 User Manual"
Documentation Simplorer, fichier Manual60.pdf
- [ANS3] "SIMPLORER C-interface"
Fichiers d'aide de Simplorer, fichier C_interf.hlp
- [ANS4] Fichiers d'aide électronique du logiciel Simplorer 6.0 (Ansoft)
- [ASH1] "The VHDL Cookbook"
Peter J. ASHENDEN, 1990
- [ASH2] "The student's guide to VHDL"
Peter Ashenden, Hartcourt Publisher, ISBN: 1558605207
- [BER1] "Modeling in analog design"
Jean-Michel BERGE et al, Kluwer academic publisher, 1995, ISBN:
0792395697
- [BRU1] "SHAMAN, une approche multidisciplinaire pour la conception de modules
d'interconnexion optique"
Christine LE BRUN et al, REE N°9, Octobre 2002, p. 68
- [CHR1] "Analog and Mixed-Signal Modeling Using the VHDL-AMS Language"
Ernst CHRISTEN et al, DAC 1999, New Orleans, June 21-25
- [DAV1] "Modélisation du comportement et du couplage de lasers à cavité verticale et
émission par la surface pour la simulation d'interconnexions optiques"
Lauréline DAVID, Stage THALES, 2003
- [DES1] "SoC modelling for virtual prototyping with VHDL-AMS"
Patricia DESGREYS et al, FDL 2004, 13-17 septembre, Lille
- [DOL1] Documentaiton de SMASH 4.9.12 de DOLPHIN INTEGRATION
- [EDA1] "EDA Industry Working Groups"
<http://www.vhdl.org>
- [HAM1] Ancienne page d'accueil du simulateur VHDL-AMS de SIMEC : "hAMSter"
<http://www.hamster-ams.com/index2.htm>

- [HAY1] "LA MÉTHODE DE MONTE CARLO METROPOLIS"
Marc HAYOUN, Jussieu, 29-31 mai 2002, Paris
- [HER1] "Cours d'Electronique Numérique et de méthodologie de CAO Electronique"
Yannick HERVE
<http://www-ensps.u-strasbg.fr/coursen/>
- [HER2] "VHDL-AMS : Applications et enjeux industriels : Cours et exercices corrigés"
Yannick HERVE, Dunod, 2002, ISBN: 2100058886
- [HER3] "Langages de haut niveau et flot de conception des systèmes complexes : vers le prototypage virtuel ? "
Yannick HERVE, Thèse de HDR, 1er juillet 2003, ULP, numéro d'ordre 481
- [HER4] "Requirements and Verification through an extension of VHDL-AMS"
Yannick HERVE et A. FAKHFAKH, FDL 2004, 14-17 septembre, Lille
- [HER5] Documents internes
Yannick HERVE.
- [INT1] Article relatif à la loi de Moore sur le site d'Intel
<http://www.intel.com/research/silicon/mooreslaw.htm>
- [IPS1] Site Internet d'IPSI distributeur de COMSIS
<http://www.ipsis.com>
- [KAR1] "Modèle VHDL-AMS d'un module optoélectronique : photorécepteur"
Mohamed KARRAY et al, JSF 2003, 20-22 décembre, Tozeur, Tunisie
- [LRM1] "IEEE Standard VHDL Analog and Mixed-signal Extensions" (1076.1-1999 Language Reference Manual)
320 pages [0738116408], commandable sur <http://www.vhdl.org/analog/>
- [LRM2] "Verilog-AMS Language Reference Manual"
<http://www.eda.org/verilog-ams/htmlpages/public-docs/lrm/>
- [LRM3] "ADA 95 Reference Manual"
<http://www.adahome.com/rm95/>
- [LRM4] "SystemC 2.0.1 Language Reference Manual"
<http://www.systemc.org/web/sitedocs/lrm.html>
- [LRM5] "SpecC Language Reference Manual"
http://www.specc.gr.jp/eng/tech/SpecC_LRM_20.pdf
- [LRM6] "SystemVerilog 3.1a Language Reference Manual"
http://www.eda.org/sv/SystemVerilog_3.1a.pdf
- [MAN1] "Modeling with an Analog Hardware Description Language"
H. Alan MANTOOTH and Mike FIEGENBAUM, Kluwer academic publishing, 1994, ISBN: 0792395166
- [MAR1] "Conception de systèmes hétérogènes contenant des microsystèmes optiques"
Sergio O. MARTINEZ et al, ISRN: TIMA--RR-01/11-1—FR, 2001
- [MAT1] Site Internet du distributeur de MATLAB
<http://www.mathworks.com>

- [MEN1] Documentation de ADV MS de Mentor Graphics / Anacad (Version 1.5)
- [MEN2] "ADVance MS - datasheet"
http://www.mentor.com/ams/datasheets/advancems_ds.pdf
- [MEN3] "SystemVision for Successive Verification – datasheet"
http://www.mentor.com/systemvision/datasheets/successive_ds.pdf
- [MIL1] "Behavioural Library Development: Models Documentation and Qualification"
 Noëlle MILET-LEWIS et al, FDL 2002, 24-27 septembre, Marseille
- [MIR1] "Exploration architecturale", Benoît MIRAMOND
<http://www.lami.univ-evry.fr/~miramond/CV/CVlg/node20.html>
- [PEC1] "VHDL-AMS and Verilog-AMS as Alternative Hardware Description Languages for Efficient Modeling of Multidiscipline Systems"
 F. Pêcheux, C. Lallement, A. Vachoux, IEEE TCAD, Vol. 24 No. 2 Pages 204-225, février 2005
- [PET1] "The System Designer's Guide to Vhdl-Ams: Analog, Mixed-Signal, and Mixed-Technology Modeling"
 Gregory PETERSON, Pater ASHENDEN and Darrell TEEGARDEN, Morgan Kaufmann Publishers, 2002, ISBN: 1558607498
- [PEZ1] "Interconnexions optiques"
 Mathias PEZ, Techniques de l'ingénieur – Traité d'électronique, E3 652
- [PEZ2] "Comparaison des liaisons optiques et électriques"
 Mathias PEZ, Techniques de l'ingénieur – Traité d'électronique, E3 680
- [RMN1] "RMNT - Livre blanc"
http://www.rmnt.org/a_charger/Livre_blanc_RMNT.zip
- [SHA1] "Rapport final du projet SHAMAN", Compte rendu ministériel de fin de projet
 Morgane MAREC et al, 12 décembre 2003
- [SHI1] "VHDL-AMS Design Objectives and Rationale"
 C.J Richard SHI, A. VACHOUX, in "Modeling in analog design", 1995
- [SNA1] "Final User Oriented SOC Modeling"
 Sébastien SNAIDERO et Yanick HERVE, DCIS 2004, 24-26 novembre, Bordeaux
- [SYN1] "SYNOPSIS Inc, MAST Datasheet"
http://www.synopsys.com/products/mixedsignal/saber/mast_ds.html
- [TOF1] "Multilevel Behavioral Simulation of VCSEL based Optoelectronic Modules"
 Zeno TOFFANO et al, IEEE J. Sel. Top. Quant., Vol. 9 No. 3, mai/juin 2003
- [TOF2] "Optoélectronique – Composants photoniques et fibres optiques"
 Zeno TOFFANO, Ellipses, 2001, ISBN : 2-7298-0703-9
- [TUI1] "Spice: A guide to Circuit Simulation Analysis using Pspice"
 Paul W. TUINEGA, Prentice Hall, 1995, ISBN: 0138346070

- [VAC1] "Modélisation de Systèmes Analogiques et Mixtes - Introduction à VHDL-AMS"
Alain VACHOUX, notes de cours 2003 EPFL
http://lsmwww.epfl.ch/design_languages/Model_Sys_Mix/Documents/modelmix03.pdf
- [WIL1] "VHDL RF & MW: Radio Frequency & Microwave Extensions to VHDL for Mixed Signal Design & Verification"
John WILLIS, DAC 2001
http://www.eda.org/vhdl-ams/ftp_files/wg_meetings/DASC_jun01/vhdlrfmw_status.pdf

Glossaire

AMS	<i>Analog and Mixed-Signal</i>
ASIC	<i>Application-Specific Integrated Circuit</i>
BPM	<i>Beam Propagation Method</i>
CAO	<i>Conception Assistée par Ordinateur</i>
CNRS	<i>Centre National de la Recherche Scientifique</i>
DLL	<i>Dynamic Link Library</i>
EDA	<i>Electronic Design Automation</i>
ECAD	<i>Electronic Computer-Aided Design</i>
ENST	<i>Ecole Nationale Supérieure des Télécommunications</i>
FPGA	<i>Field-Programmable Gate Array</i>
HDL	<i>Hardware Description Language</i>
IC	<i>Integrated Circuit</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IP	<i>Intellectual Property</i>
LAN	<i>Local Area Network</i>
MEMS	<i>Micro-Electro-Mechanical Systems</i>
MOEMS	<i>Micro-Opto-Electro-Mechanical Systems</i>
OIF	<i>Optical Internetworking Forum</i>
PHASE	<i>Laboratoire de PHysique Applications des SEmi-conducteurs</i>
PIN	<i>Positive-Intrinsic-Negative</i>
POM	<i>Parallel Optical Module</i>
PV	<i>Prototypage Virtuel</i>
SHAMAN	<i>Simulation Hybride pour Applications Multi-domaines Analogiques ou Numériques</i>
SiP	<i>System in Package</i>
SoC	<i>System on a Chip</i>
VCSEL	<i>Vertical-Cavity Surface-Emitting Laser</i>
VHDL	<i>VHSIC HDL</i>
VHSIC	<i>Very-High-Speed Integrated Circuit</i>
VLSI	<i>Very Large Scale Integration</i>
VSR	<i>Very Short Reach</i>

Annexe I : Modèle comportemental du POM

Annexe II : Modèles de couplage VCSEL-Fibre optique

Annexe III : Modèle de Fibre optique multimode

Annexe IV : Modèle de connexion inter Fibre optique

Annexe V : Modèle de couplage Fibre optique-Photodiode

Annexe VI : Modèles de générateur pseudo-aléatoire

Annexe VII : Implémentation Simplorer du modèle de VCSEL

Annexe VIII : Implémentation Simplorer de la Photodiode

Annexe IX : Diagramme de l'œil

Annexe I : Modèle comportemental du POM

Le code ci-dessous permet d'instancier 8 canaux de transmission en parallèle avec leurs entrées/sorties pour constituer le module complet. Ce modèle, conçu sous ADVance MS, ne peut pas fonctionner sous Simplorer, car il fait appel à l'instruction GENERATE qui n'y est pas disponible. L'interface graphique de Simplorer permet par contre d'imiter cette instruction en déposant 8 instances du composant représentant un canal de transmission sur la feuille de schéma.

```
entity TRT_POM is
    port (signal Com(1 to 8):in bit_vector;
          quantity Po(1 to 8):out real_vector);
end entity TRT_POM

architecture VSR_1_BEH of TRT_POM is
begin
    for i in 1 to 8 generate
        way:entity vcsel_beh(VSR_1_BEH) port map(com(i),Po(i));
    end generate;
end architecture;
```

Le modèle d'un canal de transmission a été tout d'abord développée sous ADVance MS puis porté vers Simplorer. On simule ici le comportement des informations circulant depuis le driver du VCSEL jusqu'à la sortie électrique de la photodiode. Lors du portage vers Simplorer, nous avons dû externaliser les sources de bruits car les fonctions mathématiques utilisées n'y sont pas implémentées. Cette démarche fait malheureusement perdre la portabilité du modèle par l'emploi d'un composant propre à Simplorer. Même si pour d'autres modèles une action comme celle-ci permet de s'affranchir du signal d'horloge interne du dispositif qui peut ralentir et fausser la simulation, le traitement des relations de décalage temporel lié à l'introduction de jitter aléatoire nous obligent à la garder dans le modèle.

```
library ieee;
use ieee.electrical_systems.all;

entity POM_beh is
    generic (skew : time := 5000 ps;
            jitter : time := 50 ps;
            P_l : real := 0.1; -- en mw
            Text : real := 4.0;
            Tr : real := 260.0e-12; -- 260 ps de 20 à 80%
            Bp : real := 1.5e9; -- bande passante
            P_noise : real := 0.025;
            port (signal data : in bit;
                  quantity UN1,UN2 : in real; -- Sources externes de bruit
                  quantity Po : out real);
end entity POM_beh;
```

```

architecture VSR_1_beh of POM_beh is
    signal Q_skew, Q_jitt, clock : bit;
    signal Q_ideal, Q_noise : real := 0.0;
    quantity Q_noised: real;

begin
    clock <= not clock after 100 ps;

    process
        begin
            wait on clock;
            Q_noise <= (UN1-0.5) * P_noise;
        end process;

    Q_skew <= transport data after skew-jitter/2;

    process
        begin
            wait on Q_skew;
            Q_jitt <= transport Q_skew after UN2 * jitter;
        end process;

    break on Q_jitt, Q_noise;

    process
        begin
            wait on Q_jitt;
            if Q_jitt='1' then
                Q_ideal <= P_1;
            else
                Q_ideal <= P_1/Text;
            end if;
        end process;

    Q_noised == Q_ideal'ramp(Tr*10.0/6.0)+Q_noise;
    Po + Po'dot/(2.0*math_pi*bp) == Q_noised;

end architecture VSR_1_beh;

```

Annexe II : Modèles de couplage VCSEL-Fibre optique

Scripts CodeV

- *Script de création de la partie couplage*

```
! *****
! * SHAMAN EMITTER WITH INT FILE FILTER *
! *****
! This file describes the VCSEL and a piece of the fiber with the reflective surface.
! It is considered to be the minimal system on which will be added straight parts of
! fiber.
! *****
! ARG0 "Set the parameters to fit your configuration"
!
! ARG1 NAME "Environment Index"
! ARG1 TYPE LITERAL
! ARG1 DEFAULT AIR
! ARG1 HELP "Index of the material surrounding the fiber and the VCSEL"
!
! ARG2 NAME "Optical Fiber Cladding index"
! ARG2 TYPE LITERAL
! ARG2 DEFAULT FK6
! ARG2 HELP "Index of the Cladding surrounding the Core of the fiber"
!
! ARG3 NAME "Optical Fiber Core index"
! ARG3 TYPE LITERAL
! ARG3 DEFAULT FK50
! ARG3 HELP "Index of the Core of the fiber"
!
! ARG4 NAME "X shift (mm)"
! ARG4 TYPE NUM
! ARG4 DEFAULT 0
! ARG4 HELP "troubles in X placement (mm)"
!
! ARG5 NAME "Y shift (mm)"
! ARG5 TYPE NUM
! ARG5 DEFAULT 0
! ARG5 HELP "troubles in Y placement (mm)"
!
! ARG6 NAME "Z shift (mm)"
! ARG6 TYPE NUM
! ARG6 DEFAULT 0
! ARG6 HELP "troubles in Z placement (mm)"
!
! ARG7 NAME "Tilt (deg)"
! ARG7 TYPE NUM
! ARG7 DEFAULT 0
! ARG7 HELP "Tilt (deg)"
!
! ARG8 NAME "VCSEL emission angle (°)"
! ARG8 TYPE NUM
! ARG8 DEFAULT 9
! ARG8 HELP "Angle of the light produced by the VCSEL"
!
! ARG9 NAME "F_Length (mm)"
! ARG9 TYPE NUM
! ARG9 DEFAULT 0.0
! ARG9 HELP "Optical Fiber length from the center of the mirror (mm)"
! *****
*
lcl num ^angl ^leng ^X_shift ^Y_shift ^Z_shift ^Tilt
```

```

^angl == #8
^angl == absf(^angl)

^pi == 3.1415926535898
^deg_rad == ^pi/180

! Fiber dimensions
^h_dia == 0.05 ! Fiber Heart Diameter
^c_dia == 0.125 ! Fiber Diameter
^leng == #9 ! Length

^X_shift == #4
^Y_shift == #5
^Z_shift == #6
^Tilt == #7

LEN
TITLE 'SHAMAN Project : Emitter modelling with INT filter'
EPD 0.02
DIM M
WL 850.
XOB 0.
YOB 0.
MXH 1000

! Cone d'émission de la diode
SO 0.0 0.0065/TANF(^angl*^deg_rad/2) ! The diode is 13µm of diameter (0.0065
mm radius)

S1 0.0 0.0 ! Aperture of the VCSEL
REX 0.013
REY 0.013
REX EDG 0.013
REY EDG 0.013

n 0.0 AIR REFR #1 ! Environnement context
REX 0.013
REY 0.013
REX EDG 0.013
REY EDG 0.013

!*****
!* FIBER CLADDING DEFINITION *
!*****

n 0.0 #2 REFL ! cladding mirror surface
XDE ^X_shift
YDE ^Y_shift
ZDE ^Z_shift+0.038+^c_dia/2
ELY ^c_dia/2
ELX ^c_dia/2*1.4142136
ELY EDG ^c_dia/2
ELX EDG ^c_dia/2*1.4142136
ELY HOL ^h_dia/2
ELX HOL ^h_dia/2*1.4142136
BDE -45

n -^c_dia/2 #2 TIR #1 ! first toroid (+z direction)
YTO ! Change to Y-type toroid
CUX 0
XDE ^X_shift-^leng/2
YDE ^Y_shift
ZDE ^Z_shift+0.038+^c_dia
REY EDG 0.355*^c_dia
REX EDG ^c_dia/2+^leng/2
REY 0.355*^c_dia
REX ^c_dia/2+^leng/2

n -^c_dia/2 #2 TIR #1 ! second toroid (-z direction)
YTO ! Change to Y-type toroid
CUX 0
XDE ^X_shift-^leng/2

```

```

YDE    ^Y_shift
ZDE    ^Z_shift+0.038
ADE    180
REY    EDG    0.355*^c_dia
REX    EDG    ^c_dia/2+^leng/2
REY    0.355*^c_dia
REX    ^c_dia/2+^leng/2

n      -^c_dia/2    #2    TIR    #1    ! third toroid (+y direction)
YTO                                         ! Change to Y-type toroid
CUX    0
XDE    ^X_shift-^leng/2
YDE    ^Y_shift+^c_dia/2
ZDE    ^Z_shift+0.038+^c_dia/2
ADE    90
REY    EDG    0.355*^c_dia
REX    EDG    ^c_dia/2+^leng/2
REY    0.355*^c_dia
REX    ^c_dia/2+^leng/2

n      -^c_dia/2    #2    TIR    #1    ! fourth toroid (-y direction)
YTO                                         ! Change to Y-type toroid
CUX    0
XDE    ^X_shift-^leng/2
YDE    ^Y_shift-^c_dia/2
ZDE    ^Z_shift+0.038+^c_dia/2
ADE    -90
REY    EDG    0.355*^c_dia
REX    EDG    ^c_dia/2+^leng/2
REY    0.355*^c_dia
REX    ^c_dia/2+^leng/2

n      0.0    #2    REFR    AIR    ! Output cladding surface
XDE    ^X_shift-^leng-^c_dia/2
YDE    ^Y_shift
ZDE    ^Z_shift+0.038+^c_dia/2
BDE    90
CIR    ^c_dia/2
CIR    HOL    ^h_dia/2
CIR    EDG    ^c_dia/2

!*****
!*    FIBER CORE DEFINITION    *
!*****

n      0.0    #3    REFL    ! core mirror surface
XDE    ^X_shift
YDE    ^Y_shift
ZDE    ^Z_shift+0.038+^c_dia/2
ELY    ^h_dia/2
ELX    ^h_dia/2*1.4142136
ELY    EDG    ^h_dia/2
ELX    EDG    ^h_dia/2*1.4142136
BDE    -45

n      -^h_dia/2    #3    TIR    #2    ! first toroid (+z direction)
YTO                                         ! Change to Y-type toroid
CUX    0
XDE    ^X_shift-^leng/2
YDE    ^Y_shift
ZDE    ^Z_shift+0.038+(^c_dia+^h_dia)/2
REY    EDG    0.355*^h_dia
REX    EDG    ^c_dia/2+^leng/2
REY    0.355*^h_dia
REX    ^c_dia/2+^leng/2

n      -^h_dia/2    #3    TIR    #2    ! first toroid (-z direction)
YTO                                         ! Change to Y-type toroid
CUX    0
XDE    ^X_shift-^leng/2
YDE    ^Y_shift
ZDE    ^Z_shift+0.038+(^c_dia-^h_dia)/2

```

```

ADE 180
REY EDG 0.355*^h_dia
REX EDG ^c_dia/2+^leng/2
REY 0.355*^h_dia
REX ^c_dia/2+^leng/2

n  -^h_dia/2      #3      TIR      #2      ! first toroid (+y direction)
YTO                                     ! Change to Y-type toroid
CUX 0
XDE ^X_shift-^leng/2
YDE ^Y_shift+^h_dia/2
ZDE ^Z_shift+0.038+^c_dia/2
ADE 90
REY EDG 0.355*^h_dia
REX EDG ^c_dia/2+^leng/2
REY 0.355*^h_dia
REX ^c_dia/2+^leng/2

n  -^h_dia/2      #3      TIR      #2      ! first toroid (+y direction)
YTO                                     ! Change to Y-type toroid
CUX 0
XDE ^X_shift-^leng/2
YDE ^Y_shift-^h_dia/2
ZDE ^Z_shift+0.038+^c_dia/2
ADE -90
REY EDG 0.355*^h_dia
REX EDG ^c_dia/2+^leng/2
REY 0.355*^h_dia
REX ^c_dia/2+^leng/2

n  0.0      #3      REFR      AIR      ! Output core surface
XDE ^X_shift-^leng-^c_dia/2
YDE ^Y_shift
ZDE ^Z_shift+0.038+^c_dia/2
BDE 90
CIR ^h_dia/2
CIR EDG ^h_dia/2

S  0.0      0.0
CIR ^h_dia/2

S  0.0      0.0
CIR ^h_dia/2

SI 0.0      0.0
CIR ^h_dia/2

NSS S2..14
NSP S14  GL2  FOR  0

GO

```

• **Script de création d'un tronçon de fibre optique**

```

!*****
! Sequence:          FIBER_ADD
!
! Purpose:           Sequence to insert a piece of glass fiber in front of a
!                   specified surface, using four NSS toroids. The fiber is
!                   assumed to be used in TIR
!
! Syntax:            in FIBER_ADD surface [cladding_glass] cladding_diameter [core_glass]
!                   core_diameter fiber_length
!
! Notes:            Insert 13 surfaces
!*****
! ARG0 "Inserts a fiber using NSS at the specified surface (cannot be "
! ARG0 "the image surface). The fiber is round and straight, and cannot"
! ARG0 "be made of a gradient index glass."
!
! ARG1 NAME         "Surface number"
! ARG1 TYPE         lit

```

```

! ARG1 DEFAULT 0
! ARG1 HELP "Surface where insertion occurs."
!
! ARG2 NAME "Cladding Glass"
! ARG2 TYPE LITERAL
! ARG2 DEFAULT FK6
! ARG2 HELP "Material (glass) of the fiber's cladding"
!
! ARG3 NAME "Cladding Diameter"
! ARG3 TYPE NUM
! ARG3 DEFAULT 0
! ARG3 HELP "Fiber cladding diameter"
!
! ARG4 NAME "Core Glass"
! ARG4 TYPE LITERAL
! ARG4 DEFAULT FK50
! ARG4 HELP "Material (glass) of the fiber's core"
!
! ARG5 NAME "Core Diameter"
! ARG5 TYPE NUM
! ARG5 DEFAULT 0
! ARG5 HELP "Fiber core diameter"
!
! ARG6 NAME "Fiber Length"
! ARG6 TYPE NUM
! ARG6 DEFAULT 0
! ARG6 HELP "Length of the piece of Fiber to add"
!*****

rfd 0 FK6 0 FK50 0 0
lcl num ^surf ^h_dia ^c_dia ^leng

^surf == str_to_num(rfstr(1))
^surf == roundf(absf(^surf))

^c_dia == #3
^c_dia == absf(^c_dia)

^h_dia == #5
^h_dia == absf(^h_dia)

^leng == #6
^leng == absf(^leng)

if (^surf<=0) or (^h_dia=0) or (^c_dia=0) or (^leng=0)
  wri
  wri "Macro to insert a TIR fiber using toroids in NSS"
  wri
  wri "Syntax: in ADD_FIBER surface [cladding_glass] cladding_diameter [core_glass]
core_diameter fiber_length"
  wri
  rtn
else if ^surf=1
  ins s1
  ^surf == 2
else if ^surf=(num s)
  ins si
else if ^surf>(num s)
  wri
  wri "Error in surface specification."
  rtn
end if

! check for FNO or NA

if (typ ape)="FNO" or (typ ape)="NA"
  wri
  wri q"WARNING - Pupil specification changed from 'ccc' to EPD." (typ ape)
  wri
  epd (epd)
end if

```

```

! insert the fiber surfaces and make them an NSS range

pim n
ins s^surf..^surf+12 0 0
nss s^surf..^surf+11
nsp s^surf+11 gl2 for 0

!*****
!*   FIBER CLADDING DEFINITION   *
!*****

n^surf 0      AIR      refr   #2          ! front face of fiber
      CIR      ^c_dia/2
      CIR      EDG      ^c_dia/2
      CIR      HOL      ^c_dia/2

n      -^c_dia/2  #2      TIR      AIR      ! first toroid (+x direction)
      YTO
      CUX      0
      XDE      ^c_dia/2
      ZDE      ^leng/2
      BDE      -90
      REY      EDG      0.355*^c_dia
      REX      EDG      ^leng/2
      REY      0.355*^c_dia
      REX      ^leng/2

n      -^c_dia/2  #2      TIR      AIR      ! second toroid (-x direction)
      YTO
      CUX      0
      XDE      -^c_dia/2
      ZDE      ^leng/2
      BDE      90
      REY      EDG      0.355*^c_dia
      REX      EDG      ^leng/2
      REY      0.355*^c_dia
      REX      ^leng/2

n      -^c_dia/2  #2      TIR      AIR      ! third toroid (+y direction)
      YTO
      CUX      0
      YDE      ^c_dia/2
      ZDE      ^leng/2
      ADE      90
      CDE      90
      REY      EDG      0.355*^c_dia
      REX      EDG      ^leng/2
      REY      0.355*^c_dia
      REX      ^leng/2

n      -^c_dia/2  #2      TIR      AIR      ! fourth toroid (-y direction)
      YTO
      CUX      0
      YDE      -^c_dia/2
      ZDE      ^leng/2
      ADE      -90
      CDE      90
      REY      EDG      0.355*^c_dia
      REX      EDG      ^leng/2
      REY      0.355*^c_dia
      REX      ^leng/2

n      0          #2      refr   AIR      ! back face of fiber
      ZDE      ^leng
      CIR      ^c_dia/2
      CIR      EDG      ^c_dia/2
      CIR      HOL      ^h_dia/2

!*****
!*   FIBER CORE DEFINITION   *
!*****

```

```

n      0      AIR      refr      #4      ! front face of fiber
      CIR      ^h_dia/2
      CIR      EDG      ^h_dia/2

n      -^h_dia/2      #4      TIR      #2      ! first toroid (+x direction)
      YTO
      CUX      0
      XDE      ^h_dia/2
      ZDE      ^leng/2
      BDE      -90
      REY      EDG      0.355*^h_dia
      REX      EDG      ^leng/2
      REY      0.355*^h_dia
      REX      ^leng/2

n      -^h_dia/2      #4      TIR      #2      ! second toroid (-x direction)
      YTO
      CUX      0
      XDE      -^h_dia/2
      ZDE      ^leng/2
      BDE      90
      REY      EDG      0.355*^h_dia
      REX      EDG      ^leng/2
      REY      0.355*^h_dia
      REX      ^leng/2

n      -^h_dia/2      #4      TIR      #2      ! third toroid (+y direction)
      YTO
      CUX      0
      YDE      ^h_dia/2
      ZDE      ^leng/2
      ADE      90
      CDE      90
      REY      EDG      0.355*^h_dia
      REX      EDG      ^leng/2
      REY      0.355*^h_dia
      REX      ^leng/2

n      -^h_dia/2      #4      TIR      #2      ! fourth toroid (-y direction)
      YTO
      CUX      0
      YDE      -^h_dia/2
      ZDE      ^leng/2
      ADE      -90
      CDE      90
      REY      EDG      0.355*^h_dia
      REX      EDG      ^leng/2
      REY      0.355*^h_dia
      REX      ^leng/2

n      0      #4      refr      AIR      ! back face of fiber
      ZDE      ^leng
      CIR      ^h_dia/2
      CIR      EDG      ^h_dia/2

S      0.0      0.0
      CIR      ^h_dia/2

```

• *Script d'étude de l'influence de la distance sur la distribution d'intensité*

```

!      *****
!      *      SHAMAN EMITTER : Distance influence study      *
!      *****
! Sequence:      SHAMAN_DISP
!
! Purpose:      Sequence to study the consequences of the length of the fiber on the
!               intensity profile of the light into the fiber
!
! Notes:
! *****
! ARG0 "Dispersion study"

```

```

!
! ARG1 NAME      "Environment glass"
! ARG1 TYPE      LITERAL
! ARG1 DEFAULT   AIR
! ARG1 HELP      "Material surrounding the fiber"
!
! ARG2 NAME      "Cladding Glass"
! ARG2 TYPE      LITERAL
! ARG2 DEFAULT   FK6
! ARG2 HELP      "Material (glass) of the fiber's cladding"
!
! ARG3 NAME      "Core Glass"
! ARG3 TYPE      LITERAL
! ARG3 DEFAULT   FK50
! ARG3 HELP      "Material (glass) of the fiber's core"
!
! ARG4 NAME      "Fiber shortest length (mm)"
! ARG4 TYPE      NUM
! ARG4 DEFAULT   0
! ARG4 HELP      "Basic length of the Fiber"
!
! ARG5 NAME      "Fiber longest length (mm)"
! ARG5 TYPE      NUM
! ARG5 DEFAULT   0
! ARG5 HELP      "Final length of the Fiber"
!
! ARG6 NAME      "Steps"
! ARG6 TYPE      NUM
! ARG6 DEFAULT   1
! ARG6 HELP      "Number of steps to complete the transformation"
!
! ARG7 NAME      "VCSEL emission angle (°)"
! ARG7 TYPE      NUM
! ARG7 DEFAULT   9
! ARG7 HELP      "Angle of the light cone produced by the VCSEL"
!
! ARG8 NAME      "LUM NRA"
! ARG8 TYPE      NUM
! ARG8 DEFAULT   1E3
! ARG8 HELP      "Number of rays to draw in the LUM process"
!
! ARG9 NAME      "INT File (without .int)"
! ARG9 TYPE      LITERAL
! ARG9 DEFAULT   VTRT10P
! ARG9 HELP      "File to use for the Illumination without the .int termination"
!*****

rfd AIR FK6 FK50 0 0 1 9 1E3 VTRT10P
lcl num ^start_l ^final_l ^steps ^angl ^step_val ^dist
lcl str ^file_base ^file_name

^start_l == #4
^final_l == #5
^steps == #6
^angl == #7

!*****
!*      Notification des répertoires      *
!*****
PTH      SEQ      C:\CVUSER\SHAMAN\SOURCES\
^res_dir == "C:\CVUSER\SHAMAN\RESULTATS\"

^file_base == concat(^res_dir,"Vue_")
^dist == ^start_l

! creation of the fiber base :
IN      Fiber_base      #1 #2 #3 0 0 0 0 ^angl ^start_l
IN      Plot_disp      #8 #9 ^angl/2 0.05 241 ^dist ^file_base

! creation of further pieces if required :
if (^final_l-^start_l)>0 and ^steps>0
      ^step_val == (^final_l-^start_l)/^steps

```

```

for ^i ^start_l (^final_l-^step_val) ^step_val
    ^dist == ^dist + ^step_val
    IN    Fiber_base    #1 #2 #3 0 0 0 0 ^angl ^dist
!       IN    Fiber_add    16 #2 0.125 #3 0.05 ^step_val
    IN    Plot_disp     #8 #9 ^angl/2 0.05 241 ^dist ^file_base
end for
end if

```

• *Script d'étude de l'influence des erreurs de positionnement*

```

! *****
! *          SHAMAN EMITTER : Shifts influence study V2.0          *
! *****
***
! Sequence:          SHAMAN_SHIFT
!
! Purpose:           Sequence to study the consequences of the mechanical variations in
the
!                   device
!
! Notes:
! *****
***
! ARG0 "Dispersion study"
!
! ARG1 NAME          "X min"
! ARG1 TYPE          NUM
! ARG1 DEFAULT       0
! ARG1 HELP          "Minimum of the X range"
!
! ARG2 NAME          "X max"
! ARG2 TYPE          NUM
! ARG2 DEFAULT       1
! ARG2 HELP          "Maximum of the X range (>Xmin)"
!
! ARG3 NAME          "X points"
! ARG3 TYPE          NUM
! ARG3 DEFAULT       2
! ARG3 HELP          "Number of points from min to max (>1)"
!
! ARG4 NAME          "Y min"
! ARG4 TYPE          NUM
! ARG4 DEFAULT       0
! ARG4 HELP          "Minimum of the Y range"
!
! ARG5 NAME          "Y max"
! ARG5 TYPE          NUM
! ARG5 DEFAULT       1
! ARG5 HELP          "Maximum of the Y range (>Ymin)"
!
! ARG6 NAME          "Y points"
! ARG6 TYPE          NUM
! ARG6 DEFAULT       2
! ARG6 HELP          "Number of points from min to max (>1)"
!
! ARG7 NAME          "Z min"
! ARG7 TYPE          NUM
! ARG7 DEFAULT       0
! ARG7 HELP          "Minimum of the Z range"
!
! ARG8 NAME          "Z max"
! ARG8 TYPE          NUM
! ARG8 DEFAULT       1
! ARG8 HELP          "Maximum of the Z range (>Zmin)"
!
! ARG9 NAME          "Z points"
! ARG9 TYPE          NUM
! ARG9 DEFAULT       2
! ARG9 HELP          "Number of points from min to max (>1)"
! *****
rfd 0 1 2 0 1 2 0 1 2

```

```

lcl  num    ^X_pts ^Y_pts ^Z_pts
lcl  num    ^X_step ^Y_step ^Z_step ^Sim_num
lcl  num    ^X_pos ^Y_pos ^Z_pos                ! Current point

lcl  num    ^angl          ^nb_rays
lcl  str    ^titre ^res_dir ^base_name        ! res_dir is the results
directory
gbl  num    ^FILE1 ^FILE2 ^FILE3

!*****
!*  Fiber parameters definition (not enough params in the dialog box)  *
!*****
^angl == 14          ! VCSEL emission angle
!^nb_rays == 1E5     ! Number of rays to trace un the LUM mode (0.01%)
^nb_rays == 1E2     ! Number of rays to trace un the LUM mode (1%)

!*****
!*  Parameters initialization                                          *
!*****
^X_pos == #1
^Y_pos == #4
^Z_pos == #7

^X_pts == #3
^Y_pts == #6
^Z_pts == #9

!*****
!*  Notification des répertoires  *
!*****
PTH  SEQ    C:\CVUSER\SHAMAN\SOURCES\
^res_dir == "C:\CVUSER\SHAMAN\RESULTATS\"

!if #3>1 and #6>1 and #9>1 and #2>#1 and #5>#4 and #7>#8
^X_step == (#2-#1)/(^X_pts-1)
^Y_step == (#5-#4)/(^Y_pts-1)
^Z_step == (#8-#7)/(^Z_pts-1)

OPE  NEW    U^FILE1 concat(^res_dir,"X_mat")
FOR  ^x     1      ^X_pts
WRI  U^FILE1 Q"d.ddd" ^X_pos
      ^X_pos == ^X_pos + ^X_step
END FOR
CLO  U^FILE1
OPE  NEW    U^FILE2 concat(^res_dir,"Y_mat")
FOR  ^y     1      ^Y_pts
WRI  U^FILE2 Q"d.ddd" ^Y_pos
      ^Y_pos == ^Y_pos + ^Y_step
END FOR
CLO  U^FILE2
OPE  NEW    U^FILE3 concat(^res_dir,"Z_mat")
FOR  ^z     1      ^Z_pts
WRI  U^FILE3 Q"d.ddd" ^Z_pos
      ^Z_pos == ^Z_pos + ^Z_step
END FOR
CLO  U^FILE3

^Z_pos == #7
FOR  ^z     1      ^Z_pts
      ^X_pos == #1
      FOR  ^x     1      ^X_pts
            ^Y_pos == #4
            FOR  ^y     1      ^Y_pts
                  ^sim_num == (^z-1)*^X_pts*^Y_pts+(^x-1)*^Y_pts+^y
                  IN    Fiber_base  AIR  FK6  FK50  ^X_pos  ^Y_pos  ^Z_pos  0
^angl  0
                  ^titre == concat("Simulation ",num_to_str(^sim_num))
                  tit    ^titre
                  ^base_name == concat("Shift_sim_",num_to_str(^sim_num))
                  IN    Plot_shift  ^nb_rays  ellipse  ^angl/2  0.05  241
concat(^res_dir,^base_name)

```

```
        ^Y_pos == ^Y_pos + ^Y_step
      END FOR
      ^X_pos == ^X_pos + ^X_step
    END FOR
    ^Z_pos == ^Z_pos + ^Z_step
  END FOR
!else
!  wri
!    wri "Parameters ERROR"
!  wri
!end if
```

Modélisation Mathcad

Couplage d'entrée (VCSEL-fibre) :

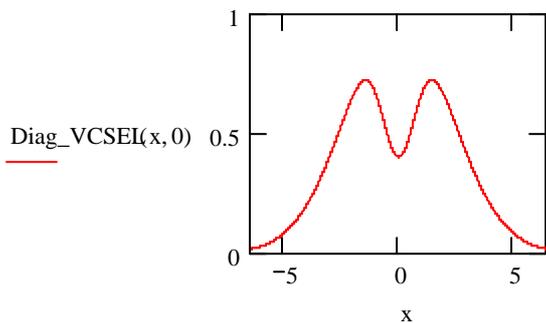
Fonctions utiles :

tand(x) := tan(x deg)

Pour d'autres params (Rb, Rm, H, Im et Po), voire annexe J rapport[DAV]

Diamètre de la surface émissive du VCSEL :	$\phi_v := 13.0$	$R_v := \frac{\phi_v}{2}$
Angle de propagation VCSEL :	$\alpha_v := 20$	
Décalage de la fibre en X :	$\Delta X := 0.0$	
Décalage de la fibre en Y :	$\Delta Y := 0.0$	
Décalage de la fibre en Z :	$\Delta Z := 0.0$	
Distance théorique VCSEL-bord de fibre :	$D_{vbf} := 38.0$	
Diamètre coeur de fibre :	$\phi_{fc} := 50$	
Diamètre gaine de fibre :	$\phi_{fg} := 125$	
Distance entre la VCSEL et le centre de la fibre :	$D_{vf} := D_{vbf} + \frac{\phi_{fg}}{2} + \Delta Z$	
Distribution d'intensité en sortie VCSEL :	$I_1 := 1.0$	
	$w_1 := 4.5$	
	$I_2 := 0.6$	
	$w_2 := 1.5$	

Fonction résultante :
$$\text{Diag_VCSEL}(R, \theta) := I_1 \cdot \exp\left(\frac{-2 \cdot R^2}{w_1^2}\right) - I_2 \cdot \exp\left(\frac{-2 \cdot R^2}{w_2^2}\right)$$



Diamètre VCSEL équivalent au niveau de la fibre :

$$\phi_{v_eq} := (D_{vf} \cdot \text{tand}(\alpha_v) + \phi_v)$$

$$R_{v_eq} := \frac{\phi_{v_eq}}{2}$$

$$\rho\phi := \frac{\phi_v}{\phi_{v_eq}} \quad \phi_{v_eq} = 49.579$$

$$\rho\phi = 0.262$$

Contour de la surface d'intégration équivalente au centre de la fibre :

$$R_{\text{int}}(\theta) := \frac{\phi c}{2}$$

Etablissement de la proportion de puissance transmise :

Nombre de points de simulation (Monté-Carlo) : $N := 1000$

Définition du vecteur de simulation : $i := 1.. N$

Définition des points de la distrib :

$$X_i := \text{runif}(1, -Rv_eq, Rv_eq)_{0,0} \quad Y_i := \text{runif}(1, -Rv_eq, Rv_eq)_{0,0}$$

Définition du ratio de puissance transmise :

Coordonnées polaires des points de la distrib : $\text{Dist_v}_i := \text{xy2pol}(X_i, Y_i)$

Puissance émise par la VCSEL en unité arbitraire :

$$Pv := \sum_i \text{if} \left[\left(\text{Dist_v}_i \right)_0 \leq Rv_eq, \text{Diag_VCSEL} \left[\left(\text{Dist_v}_i \right)_0 \cdot \rho\phi, \left(\text{Dist_v}_i \right)_1 \right], 0.0 \right]$$

$$Pv = 176.114$$

Distance et angle du point au centre de la fibre :

$$\text{Dist_f}_i := \text{xy2pol}(X_i - \Delta X, Y_i - \Delta Y)$$

Puissance reçue par la fibre en unité arbitraire :

$$Pf := \sum_i \text{if} \left[\left(\text{Dist_v}_i \right)_0 \leq Rv_eq \wedge \left(\text{Dist_f}_i \right)_0 \leq R_{\text{int}} \left[\left(\text{Dist_f}_i \right)_1 \right], \text{Diag_VCSEL} \left[\left(\text{Dist_v}_i \right)_0 \cdot \rho\phi, \left(\text{Dist_v}_i \right)_1 \right], 0.0 \right]$$

$$Pf = 176.114$$

Ratio de puissance :

$$\text{resultat} := \frac{Pf}{Pv} \quad \text{resultat} = 1$$

Modélisation C/C++

Ce listing est celui du fichier C++ créé par l'assistant, modifié ensuite selon nos besoins:

```
// Template File for C-Models under Simplorer 6.0
// Interface version 6.x

#include <math.h>
#include <stdlib.h>
#include <process.h>
#include "SimplorerAfx.h"
#include "Sim2000User.h"
#include "Complex.h"

// Set the model name
#define STRG_MODELNAME      "coupl_in"

// Adjust the function names in case of a multiple-model-DLL
#define PREP_FCN      Prepare_coupl_in
#define INIT_FCN      Initialize_coupl_in
#define SIMU_FCN      Simulate_coupl_in
#define VALI_FCN      Validate_coupl_in
#define CLOSE_FCN     Close_coupl_in

// Its suggested to define Parameter, Output and internal State names here
#define STRG_NCNAME_INPUT      "Pin"
#define STRG_NCNAME_OUTPUT    "Pout"
#define STRG_NCNAME_NBPTS      "N" // Nombre de points à calculer
#define STRG_NCNAME_OUVNUM     "ON" // Ouverture Numérique de la diode
#define STRG_NCNAME_DIAMD      "Dd" // Diamètre de la surface d'emission VCSEL
#define STRG_NCNAME_DIAMF      "Df" // Diamètre coeur de fibre
#define STRG_NCNAME_DELTAX     "Dx" // Erreur de placement en X
#define STRG_NCNAME_DELTAY     "Dy" // Erreur de placement en Y
#define STRG_NCNAME_DELTAZ     "Dz" // Erreur de placement en Z

double aleat(double);
double distrib(double, double);

////////////////////////////////////
///
/// Function prototypes
////////////////////////////////////
///
BOOL check_param ( CModUser * ); // TRUE if parameters OK, FALSE if not

////////////////////////////////////
///
/// build the model - interface
////////////////////////////////////
///
FCTDECL PREP_FCN( CModUser *pMod )
{
    // Inputs (non-conservative nodes)
    pMod->AddNode_nc( STRG_NCNAME_INPUT, 0.0, DIRIN );
    pMod->AddNode_nc( STRG_NCNAME_NBPTS, 1000.0, DIRIN );
    pMod->AddNode_nc( STRG_NCNAME_OUVNUM, 0.2, DIRIN );
    pMod->AddNode_nc( STRG_NCNAME_DIAMD, 13.0, DIRIN );
    pMod->AddNode_nc( STRG_NCNAME_DIAMF, 62.5, DIRIN );
    pMod->AddNode_nc( STRG_NCNAME_DELTAX, 0.0, DIRIN );
    pMod->AddNode_nc( STRG_NCNAME_DELTAY, 0.0, DIRIN );
    pMod->AddNode_nc( STRG_NCNAME_DELTAZ, 0.0, DIRIN );
    // Input Info - English
    pMod->SetInfoNode_nc( STRG_NCNAME_INPUT, "Optical power input" );
    pMod->SetInfoNode_nc( STRG_NCNAME_NBPTS, "Nombre de points à calculer" );
    pMod->SetInfoNode_nc( STRG_NCNAME_OUVNUM, "Ouverture Numérique de la diode" );
    pMod->SetInfoNode_nc( STRG_NCNAME_DIAMD, "Diamètre de la surface d'émission de la
VCSEL" );
    pMod->SetInfoNode_nc( STRG_NCNAME_DIAMF, "Diamètre du coeur de fibre" );
}
```

```
pMod->SetInfoNode_nc( STRG_NCNAME_DELTAX, "Erreur de placement en X" );
pMod->SetInfoNode_nc( STRG_NCNAME_DELTAY, "Erreur de placement en Y" );
pMod->SetInfoNode_nc( STRG_NCNAME_DELTAZ, "Erreur de placement en Z" );
// Outputs
pMod->AddNode_nc( STRG_NCNAME_OUTPUT, 0.0 );
// Output Info - English
pMod->SetInfoNode_nc( STRG_NCNAME_OUTPUT, "Optical power output" );
return lL;
}

////////////////////////////////////
///
/// Initialization of model internal variables and states
////////////////////////////////////
///
FCTDECL INIT_FCN( CModUser *pMod )
{
    srand(_getpid());

    //double I0 = pMod->GetValNode_nc( STRG_NCNAME_INPUT );

    //pMod->SetValNode_State( STRG_NCNAME_OUTPUT, I0 );

    return TRUE;
}

////////////////////////////////////
///
/// Simulate function
////////////////////////////////////
///
FCTDECL SIMU_FCN( CModUser *pMod )
{
    //////////////////////////////////////////////////
    ////////////////////////////////////////////////// Getting Values from modelinterface
    //////////////////////////////////////////////////
    double input = pMod->GetValNode_nc( STRG_NCNAME_INPUT );
    double N = pMod->GetValNode_nc( STRG_NCNAME_NBPTS );
    double ON = pMod->GetValNode_nc( STRG_NCNAME_OUVNUM );
    double Dd = pMod->GetValNode_nc( STRG_NCNAME_DIAMD );
    double Df = pMod->GetValNode_nc( STRG_NCNAME_DIAMF );
    double Dx = pMod->GetValNode_nc( STRG_NCNAME_DELTAX );
    double Dy = pMod->GetValNode_nc( STRG_NCNAME_DELTAY );
    double Dz = pMod->GetValNode_nc( STRG_NCNAME_DELTAZ );

    double L = 38.0 + (125.0 - Df)/2.0;
    double Rf = Df/2.0;
    double output;
    int i; // Variable compteur
    double Rreal; // Rayon approx du faisceau VCSEL sur l'axe fibre
    double Num, Den; // Ratio de pts entrant/total
    double Xi, Yi, Dist_d, Dist_f, Teta;

    Rreal = Dd/2.0 + tan(asin(ON)) * (L+Dz+Rf);
    Num=Den=0.0;

    for(i=0;i<N;i++) // Calcul pour chaque point
    {
        Xi = aleat(Rreal); // Le point considéré
        Yi = aleat(Rreal);

        Dist_d = sqrt(pow(Xi,2.0)+pow(Yi,2.0));
        Dist_f = sqrt(pow(Xi-Dx,2.0)+pow(Yi-Dy,2.0));
        Teta = atan2(Yi,Xi);

        if(Dist_d<Rreal)
        {
            Den += distrib(Dist_d,Teta);
            if(Dist_f<Rf)
            {

```

```

        Num += distrib(Dist_d,Teta);
    }
}
output = Num/Den * input;
//output = Dx;

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% setting values to interface %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pMod->SetValNode_nc( STRG_NCNAME_OUTPUT,    output );

return TRUE;
}

```

```

////////////////////////////////////
///
/// Fonctions annexes
////////////////////////////////////
///
double aleat(double X)
{
    /* Retourne un nombre aléatoire entre -X et +X
    suivant une distribution uniforme */
    return ((rand()/(double)RAND_MAX)*2.0-1.0)*X;
}
double distrib(double R, double T)
{
    // C'est ici que l'on peut insérer la fonction de pondération
    //return exp(-2.0*pow(R,2.0));
    return 1.0;
}

////////////////////////////////////
///
/// Validate function
////////////////////////////////////
///
FCTDECL VALI_FCN( CModUser *pMod )
{
    // Do some validation work after every timestep
    return TRUE;
}

////////////////////////////////////
///
///
///
///
FCTDECL CLOSE_FCN( CModUser *pMod )
{
    // Clean up and free dynamically allocated memory
    return TRUE;
}

////////////////////////////////////
///
/// register all the models of the dll
////////////////////////////////////
///
FCTDECL RegisterUserModel( long lIndex, CUModDecl *pUModDecl )
{
    switch( lIndex )
    {
        case 0: pUModDecl->SetDefName( STRG_MODELNAME );
                pUModDecl->SetModelType( UMODEL );
                pUModDecl->SetSimPref( ECM );
                break;
    }
}

```

```

        default:
            break;
    }
    return TRUE;
}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///
// register the functions of the model
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////
///
FCTDECL RegisterUMODELFct( CModUser *pMod )
{
    if( !strcmp( pMod->GetDefName(), STRG_MODELNAME ) )
    {
        switch( pMod->GetAnalysisType() )
        {
            case TR_:
                pMod->SetUMODPrepFct( PREP_FCN ); // Getting
ModelInterface
                pMod->SetUMODInitFct( INIT_FCN ); // Initialize
Function
                pMod->SetUMODSimFct( SIMU_FCN ); // Simulate
Function
                pMod->SetUMODValidFct( VALI_FCN ); // Validate
Function
                pMod->SetUMODCloseFct( CLOSE_FCN ); // Close Function
                break;
            case DC_:
                break;
            case AC_:
                break;
            default:
                break;
        }
    }
    return TRUE;
}
}

```

Second modèle – Modélisation Mathcad

Couplage d'entrée II 3D (VCSEL-fibre) :

$$d2r := \frac{\pi}{180} \quad r2d := \frac{180}{\pi}$$

Diamètre de la surface émissive du VCSEL :

$$\phi_v := 13.0$$

$$R_v := \frac{\phi_v}{2}$$

Demi angle d'ouverture du VCSEL :

$$\alpha_v := 7$$

$$Z_v := \frac{R_v}{\tan(\alpha_v \cdot d2r)}$$

Distance VCSEL-bord de fibre :

$$Z_{vf} := 38.0$$

Erreurs de placement :

$$\Delta X := 0.0$$

$$\Delta Y := 0.0$$

$$\Delta Z := 0.0$$

Diamètre coeur de fibre :

$$\phi_{fc} := 50$$

Diamètre gaine de fibre :

$$\phi_{fg} := 125$$

$$R_{fc} := \frac{\phi_{fc}}{2} \quad R_{fg} := \frac{\phi_{fg}}{2}$$

Distance entre la source virtuelle et le centre de la fibre : $Z_{tot} := Z_v + Z_{vf} + R_{fg} + \Delta Z$

Distribution d'intensité en sortie VCSEL :

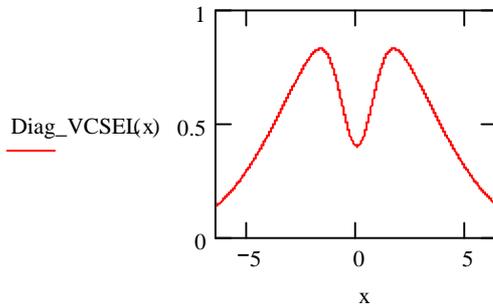
$$I_1 := 1.0$$

$$w_1 := 6.5$$

$$I_2 := 0.6$$

$$w_2 := 1.5$$

Fonction résultante :
$$\text{Diag_VCSEL}(R) := I_1 \cdot \exp\left(\frac{-2 \cdot R^2}{w_1^2}\right) - I_2 \cdot \exp\left(\frac{-2 \cdot R^2}{w_2^2}\right)$$



Etablissement de la proportion de puissance transmise :

Nombre de points de simulation (Monté-Carlo) : $N := 1000$

Définition du vecteur de simulation : $i := 0..N$

Définition des points de la distrib :

$$X_i := \text{runif}(1, -R_v, R_v)_{0,0} \quad Y_i := \text{runif}(1, -R_v, R_v)_{0,0} \quad R\theta_i := \text{xy2pol}(X_i, Y_i)$$

$$\text{puissance}(R) := \begin{cases} \text{Diag_VCSEL}(R) & \text{if } R < R_v \\ 0 & \text{otherwise} \end{cases} \quad \alpha\theta_i := \text{atan}\left[\frac{(R\theta_i)_0}{Z_v}\right]$$

Puissance optique IN limitée par surface VCSEL $P_i := \text{puissance}\left[(R\theta_i)_0\right]$

Vecteurs issus du VCSEL $I0_i := \begin{pmatrix} X_i \\ Y_i \\ Z_v \end{pmatrix}$

Calcul du point d'impact fibre

$$a_i := 1 + \frac{(X_i)^2}{Z_v^2} \quad b_i := -2 \cdot \left(Z_{tot} + \Delta X \cdot \frac{X_i}{Z_v} \right) \quad c_i := \Delta X^2 + Z_{tot}^2 - R_{fg}^2$$

$$\Delta D_i := (b_i)^2 - 4 \cdot a_i \cdot c_i \quad \text{sol}(D, a, b) := \begin{cases} 0 & \text{if } D \leq 0 \\ \min\left(\frac{-b + \sqrt{D}}{2a}, \frac{-b - \sqrt{D}}{2a}\right) & \text{if } D > 0 \end{cases}$$

$$Iz_i := \text{sol}(\Delta D_i, a_i, b_i) \quad Ix_i := \frac{Iz_i \cdot X_i}{Z_v} \quad Iy_i := \frac{Iz_i \cdot Y_i}{Z_v}$$

$$I_i := \begin{pmatrix} Ix_i \\ Iy_i \\ Iz_i \end{pmatrix} \quad F_i := \begin{pmatrix} \Delta X \\ Iy_i \\ Z_{tot} \end{pmatrix} \quad R_i := I_i - F_i \quad Z2_i := \frac{-R_i}{|R_i|} \quad X2_i := \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \times Z2_i$$

$$\alpha_{1i} := \text{acos}\left(\frac{I0_i \cdot Z2_i}{|I0_i|}\right) \quad \text{interface}(\alpha_i, n1, n2) := \text{asin}\left(\frac{n1}{n2} \cdot \sin(\alpha_i)\right)$$

$$\alpha_{2i} := \text{interface}(\alpha_{1i}, 1.0, 1.45) \quad \text{sincond}(\alpha, x) := \begin{cases} \sin(\alpha) & \text{if } x \geq 0 \\ -\sin(\alpha) & \text{otherwise} \end{cases}$$

Vecteur issu de la gaine dans le nouveau repère

Vecteur incident dans le nouveau repère

$$Z := \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$I2_i := \begin{pmatrix} \frac{I0_i \cdot X2_i}{|I0_i|} \\ \frac{(I0_i)_1}{|I0_i|} \\ \frac{I0_i \cdot Z2_i}{|I0_i|} \end{pmatrix} \quad M_i := \begin{bmatrix} (I2_i \times Z)_0 & (I2_i \times Z)_1 \\ (I2_i)_0 & (I2_i)_1 \end{bmatrix} \quad V_i := \begin{bmatrix} 0 \\ \cos(\alpha_{1i} - \alpha_{2i}) - (I2_i)_2 \cdot \cos(\alpha_{2i}) \end{bmatrix}$$

$$S_i := \text{Isolve}(M_i, V_i) \quad I3_i := \begin{bmatrix} (S_i)_0 \\ (S_i)_1 \\ \cos(\alpha_{2i}) \end{bmatrix}$$

Calcul du point d'impact coeur fibre

$$a2_i := 1 + \frac{\left[(I3_i)_0 \right]^2}{\left[(I3_i)_2 \right]^2} \quad b2_i := -2 \cdot Rfg \quad c2_i := Rfg^2 - Rfc^2$$

$$\Delta D2_i := (b2_i)^2 - 4 \cdot a2_i \cdot c2_i$$

$$Iz2_i := \text{sol}(\Delta D2_i, a2_i, b2_i) \quad Ix2_i := Iz2_i \cdot \frac{(I3_i)_0}{(I3_i)_2} \quad Iy2_i := Iz2_i \cdot \frac{(I3_i)_1}{(I3_i)_2}$$

$$Ic_i := \begin{pmatrix} Ix2_i \\ Iy2_i \\ Iz2_i \end{pmatrix} \quad Fc_i := \begin{pmatrix} 0 \\ Iy2_i \\ Rfg \end{pmatrix} \quad R2_i := Ic_i - Fc_i \quad Z3_i := \frac{-R2_i}{|R2_i|}$$

$$\alpha3_i := \text{acos} \left(\frac{I3_i \cdot Z3_i}{|I3_i|} \right)$$

$$\alpha4_i := \text{interface}(\alpha3_i, 1.45, 1.46)$$

Segment gaine-coeur dans le repère originel : $Ic0_i := Ix2_i \cdot X2_i + Iy2_i \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + Iz2_i \cdot Z2_i$

Coordonnées du point d'impact coeur dans le repère originel :

$$Itot_i := I_i + Ic0_i$$

Comparaison avec le plan du miroir :

$$\text{fmirror}(y, yf, zf) := y - yf + zf$$

$$\text{Mirror}_i := \text{if} \left[\text{fmirror} \left[\left(Itot_i \right)_1, \Delta Y, Z_{tot} \right] \geq \left(Itot_i \right)_2 \wedge \text{fmirror} \left[\left(Itot_i \right)_1, \Delta Y, Z_{tot} \right] \leq (Z_{tot} + Rfc), 1, 0 \right]$$

$$\text{Den} := \sum_i P_i \quad \text{Den} = 323.964$$

$$\text{Num} := \sum_i \text{if} \left(Iz1_i > 0 \wedge Iz2_i > 0, P_i \cdot 0.96 \cdot 0.96 \cdot \text{Mirror}_i, 0 \right) \quad \text{Num} = 298.566$$

$$\rho := \frac{\text{Num}}{\text{Den}} \quad \rho = 0.922$$

Annexe III : Modèle de Fibre optique multimode

Ce modèle comportemental de fibre optique contient deux architectures différentes pour pouvoir traiter le cas des fibres à saut et à gradient d'indice et permet d'utiliser indifféremment une valeur de dispersion modale fixe ou bien sa pente.

```
ENTITY fiber IS
    generic (L:real                := 0.3;           -- [km] Longueur de fibre
            att_db_km:real         := 2.0;           -- [dB/Km] atténuation
            Ouv_num:real           := 0.275;        -- Eponyme
            indice_coeur:real      := 1.5;         -- indice du coeur de la fibre
            D:real                 := 0.0;         -- [ps/(nm.km)] Dispersion chromatique
            So:real                := 0.0;         -- [ps/(nm².km)] pente dispersion
    chroma.
        Lambdao:real              := 1300.0e-9;    -- [m] λ de dispersion chromatique
    nulle
        delayed_output:boolean);    -- [true/false] sortie décalée ou non ?
    port  (quantity Pin           : in real;        -- Puissance optique d'entrée
          quantity Pout          : out real;       -- Puissance optique de sortie
          quantity Lambda        : in real;        -- [m] Longueur d'onde source
          quantity dLambda       : in real);       -- [nm] Etalement spectral de la source
END ENTITY fiber;
```

Les 2 architectures diffèrent dans cette version simplement par le calcul de la dispersion modale.

- *Fibre à gradient d'indice (GI)*

```
library ieee;
use ieee.math_real.all;

ARCHITECTURE grad2 OF fiber IS
    constant c      : real := 299_792.458;         -- vitesse lumière dans le vide [km/s]
    constant att    : real := 10.0**(att_db_km*L/1.0e4); -- Attenuation [dB/km]
    -- Approximation du filtre gaussien par un Bessel du 8ème ordre
    constant num    : real := 2_027_025.0;
    constant den   : real_vector(0 to 8) :=
        (2_027_025.0, 2_027_025.0, 945_944.0, 270_270.0, 51_975.0, 6_930.0, 630.0, 36.0, 1.0);
    -- Quantités locales
    quantity Dtm,Dtc,DT,Bp: real;                 -- Dispersion et Bande passante
    quantity delay         : real;                 -- Retard de Propagation
    quantity PnoDelayed,pdelayed : real;          -- Sorites retardée ou non
    quantity pdot1,pdot2,pdot3,pdot4,pdot5,pdot6,pdot7,pdot8 : real;
                                                    -- Quantités intermédiaires de
    dérivation
BEGIN
    Dtm == (L * ouv_num**4) / (8.0 * indice_coeur**3 * c); -- Dispersion modale [s]
    if (So = 0.0) use
        Dtc == -L * dLambda * D * 1.0e-12;             -- Dispersion chromatique [s]
    else
        Dtc == (L * dLambda * So*1.0e3 * ((Lambda**4-Lambdao**4)/Lambda**3)) / 4.0;
                                                    -- Dispersion chromatique [s]
    end use;
    DT == sqrt(Dtm**2 + Dtc**2);                       -- Dispersion totale
    Bp == sqrt(12.0) * sqrt(2.0*log(2.0)) / (2.0*math_pi*DT); -- Bande passante totale
    delay == L*indice_coeur/c - 1.0/Bp;               -- retard corrigé
```

```

-- Filtrage et atténuation
pdot1 == Pnodelayed'dot/Bp;
pdot2 == Pdot1'dot/Bp;
pdot3 == Pdot2'dot/Bp;
pdot4 == Pdot3'dot/Bp;
pdot5 == Pdot4'dot/Bp;
pdot6 == Pdot5'dot/Bp;
pdot7 == Pdot6'dot/Bp;
pdot8 == Pdot7'dot/Bp;

num*Pin / att == den(0) * Pnodelayed
                + den(1) * Pdot1
                + den(2) * Pdot2
                + den(3) * Pdot3
                + den(4) * Pdot4
                + den(5) * Pdot5
                + den(6) * Pdot6
                + den(7) * Pdot7
                + den(8) * Pdot8;

Pdelayed == Pnodelayed'delayed(delay);

if delayed_output use
    Pout == Pdelayed;
else
    Pout == Pnodelayed;
end use;

END ARCHITECTURE grad2;

```

- **Fibre à saut d'indice (SI)**

```

library ieee;
use ieee.math_real.all;

ARCHITECTURE grad2 OF fiber IS
    constant c      : real := 299_792.458;      -- vitesse lumière dans le vide [km/s]
    constant att    : real := 10.0**(att_db_km*L/1.0e4); -- Atténuation [dB/km]
-- Approximation du filtre gaussien par un Bessel du 8ème ordre
    constant num    : real := 2_027_025.0;
    constant den    : real_vector(0 to 8) :=
        (2_027_025.0, 2_027_025.0, 945_944.0, 270_270.0, 51_975.0, 6_930.0, 630.0, 36.0, 1.0);
-- Quantités locales
    quantity Dtm,Dtc,DT,Bp: real;              -- Dispersion et Bande passante
    quantity delay          : real;            -- Retard de Propagation
    quantity PnoDelayed,pdelayed : real;      -- Sorites retardée ou non
    quantity pdot1,pdot2,pdot3,pdot4,pdot5,pdot6,pdot7,pdot8 : real;
                                                -- Quantités intermédiaires de
dérivation
BEGIN
    Dtm == (L * ouv_num**2) / (2.0 * indice_coeur * c); -- Dispersion modale [s]
    if (So = 0.0) use
        Dtc == -L * dLambda * D * 1.0e-12;          -- Dispersion chromatique [s]
    else
        Dtc == (L * dLambda * So*1.0e3 * ((Lambda**4-Lambdao**4)/Lambda**3)) / 4.0;
                                                -- Dispersion chromatique [s]
    end use;
    DT == sqrt(Dtm**2 + Dtc**2);                    -- Dispersion totale
    Bp == sqrt(12.0) * sqrt(2.0*log(2.0)) / (2.0*math_pi*DT); -- Bande passante totale
    delay == L*indice_coeur/c - 1.0/Bp;             -- retard corrigé

-- Filtrage et atténuation
    pdot1 == Pnodelayed'dot/Bp;
    pdot2 == Pdot1'dot/Bp;
    pdot3 == Pdot2'dot/Bp;
    pdot4 == Pdot3'dot/Bp;
    pdot5 == Pdot4'dot/Bp;
    pdot6 == Pdot5'dot/Bp;
    pdot7 == Pdot6'dot/Bp;
    pdot8 == Pdot7'dot/Bp;

```

```
num*Pin / att == den(0) * Pnodelayed
               + den(1) * Pdot1
               + den(2) * Pdot2
               + den(3) * Pdot3
               + den(4) * Pdot4
               + den(5) * Pdot5
               + den(6) * Pdot6
               + den(7) * Pdot7
               + den(8) * Pdot8;

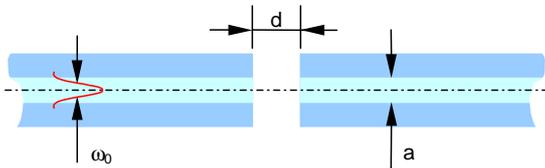
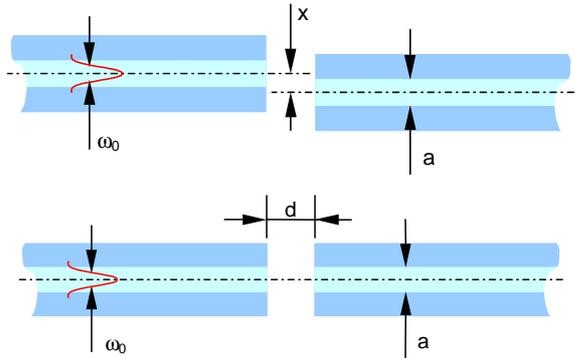
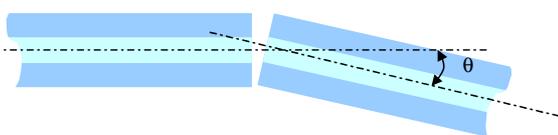
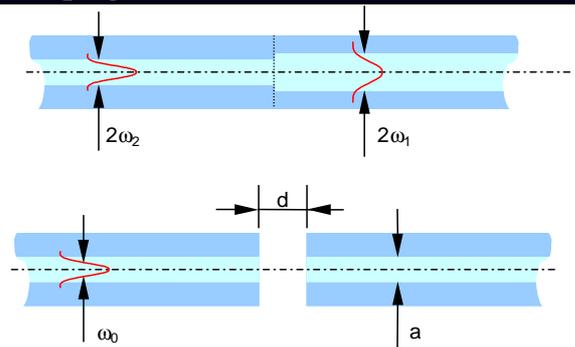
Pdelayed == Pnodelayed'delayed(delay);

if delayed_output use
    Pout == Pdelayed;
else
    Pout == Pnodelayed;
end use;

END ARCHITECTURE grad2;
```

Annexe IV : Modèle de connexion inter Fibre optique

Les rubans de fibres optiques du module d'émission et du module de réception sont reliés dans le module complet grâce à un connecteur. L'étude de la connexion entre deux rubans de fibres doit tenir compte des désalignements transversal et angulaire ainsi que de l'écart longitudinal, qui induisent des pertes de couplages.

Désalignement longitudinal		Désalignement latéral	
			
Monomode	Multimode	Monomode	Multimode
$L_{long} = -10 \log \left[\frac{1}{\left(\frac{d\lambda}{2\pi n_0 \omega_0^2} \right)^2 + 1} \right]$	$L_{long} = -10 \log \left[1 - \frac{d \cdot ON}{2an_0} \right]$	$L_{lat} = -10 \log \left[e^{-\left(\frac{x}{\omega_0} \right)^2} \right]$	$L_{lat} = -10 \log \left[1 - \frac{8x}{3\pi a} \right]$
Désalignement angulaire		Couplage de mode et diamètre différents	
			
Monomode	Multimode	Monomode	Multimode
$L_{lat} = -10 \log \left[e^{-\left(\frac{n_0 \pi a_0 \sin \theta}{\lambda} \right)^2} \right]$	$L_{ang} = -10 \log \left[1 - \frac{8n_0 \sin \theta}{3\pi ON} \right]$	$L_{mode} = -10 \log \left[\frac{4}{\left(\left(\frac{\omega_2}{\omega_1} \right) + \left(\frac{\omega_1}{\omega_2} \right) \right)^2} \right]$	$L_{core} = -10 \log \left[\left(\frac{\omega_2}{\omega_1} \right)^2 \right]$

Mode fondamental d'une fibre monomode	Nombre de mode (N)
$\omega_0 = a \left(0.65 + \frac{1.619}{V^{1.5}} + \frac{2.879}{V^6} \right)$	$V = \frac{2\pi a}{\lambda} ON$ Saut d'indice : $N=V^2/2$ Gradient d'indice : $N=V^2/4$

Tab. 06 - Equations de couplage permettant de calculer les pertes

Le modèle de connecteur est décrit par le code VHDL-AMS suivant:

```

entity fiber_connector is
  generic(ON1      : real := 0.18;      -- Ouv. Num. fibre entrant dans le connecteur
          a1       : real := 62.5;     -- Diam. coeur fibre entrant dans le connect.
  )
  (m)
          ON2      : real := 0.18;     -- Ouv. Num. fibre sortant du connecteur
          a2       : real := 62.5;     -- Diam. coeur fibre sortant du connecteur (m)
          n0       : real := 1.48;     -- Indice du coeur de fibre entrante
          d_long   : real := 0.0;      -- Désalignement longitudinal (m)
          d_lat    : real := 0.0;      -- Désalignement latéral (m)
          d_theta  : real := 0.0;      -- Désalignement angulaire
          f_loss   : real := 0.0;      -- Pertes de Fresnel (dB)
  --
          lambda   : real := 0.850e-6; -- longueur d'onde centrale

  port    (quantity Pin  : in real;    -- Puissance optique d'entrée
          quantity Pout : out real;    -- Puissance optique de sortie
          quantity Lambda : in real);  -- Longueur d'onde centrale de la source
end entity;

library ieee;
use ieee.math_real.all;

architecture behavioural of fiber_connector is
  constant long_loss : real := -10.0 * log(1.0 - (d_long*ON1) / (2.0*a1*n0));
  constant lat_loss  : real := -10.0 * log(1.0 - (8.0*d_lat) / (3.0*math_pi*a1));
  constant ang_loss  : real := -10.0 * log(1.0 - (8.0*n0*sin(d_theta)) /
(3.0*math_pi*ON1));
  constant V1       : real := 2.0 * math_pi * a1 * ON1 / lambda;
  constant V2       : real := 2.0 * math_pi * a2 * ON2 / lambda;
  constant omegal   : real := a1 * (0.65 + 1.619 / V1**1.5 + 2.879 / V1**6);
  constant omega2   : real := a2 * (0.65 + 1.619 / V2**1.5 + 2.879 / V2**6);
  constant diff_loss : real := -20.0 * log(omegal/omega2);
  constant tot_loss  : real := long_loss + lat_loss + ang_loss + diff_loss + f_loss;

begin
  Pout == Pin * 10.0**(-tot_loss/10.0);
end architecture;

```

Annexe V : Modèle de couplage Fibre optique- Photodiode

```
ENTITY Coupling_OUT IS
    Generic (tau : real := 0.6);           -- Taux de couplage
    port    (quantity Popt_in : in real;
             quantity Popt_out : out real);
END ENTITY Coupling_OUT;
```

```
ARCHITECTURE beh OF coupling_out IS
BEGIN
    Popt_out == tau * Popt_in;
END ARCHITECTURE beh;
```

Annexe VI : Modèles de générateur pseudo-aléatoire

```
library ieee;
use ieee.electrical_systems.all;

ENTITY generator IS
    generic (n      : integer := 10;      -- Degré du polynôme de génération : 7, 10 ou
15
            debit  : time := 1 ns;      -- Débit maximal (durée minimale d'un état)
            skew   : time := 25 ps;     -- Temps de propagation électronique
(invariable)
            jitter : time := 30 ps;     -- Jitter en distrib. gaussienne à + ou - 3
sigma
            Vcc    : real := 3.3;
            Vee    : real := 0.0);
    port (terminal T_p,T_n : electrical);-- Terminaux de sortie différentielle
END ENTITY generator;

library ieee;
use ieee.math_real.all;

ARCHITECTURE ideal_behav OF generator IS
    --- Signaux
    signal SR      : bit_vector(1 to n) := (others =>'1');
    signal clock, S : bit;
    signal Sn      : real := Vee;
    signal Sp      : real := Vcc;

    --- Quantités de ports
    quantity V_p across I_p through T_p to electrical_ref;
    quantity V_n across I_n through T_n to electrical_ref;
BEGIN
    --- Création de l'horloge
    clock <= not clock after (debit);

    --- Processus de generateur aléatoire de bit
    process
    begin
        wait on clock;                    -- Attente front d'horloge

        for i in n-1 downto 1 loop
            SR(i+1) <= SR(i);
        end loop;

        if SR(n)='1' then
            Sp <= Vcc;
            Sn <= Vee;
        else
            Sp <= Vee;
            Sn <= Vcc;
        end if;

        case n is
            when 1 => SR(1) <= clock;
            when 7 => SR(1) <= SR(3) xor SR(7);
            when 10 => SR(1) <= SR(3) xor SR(10);
            when 15 => SR(1) <= SR(1) xor SR(15);
            when others => assert false report "Attention : seuls 7, 10, 15 sont
definis" severity failure;
        end case;
    end process;

    V_p == Sp;
    V_n == Sn;
END ARCHITECTURE ideal_behav;
```

Annexe VII : Implémentation Simplorer du modèle de VCSEL

Les sources de bruit ont été extraites du modèle et les grandeurs correspondantes entrent par les quantités "UN11 et UN12" car on utilise qu'une seule source de bruit blanc gaussien appliquée au bruit RIN:

```
library ieee;
use ieee.electrical_systems.all;
use ieee.thermal_systems.all;

ENTITY VCSEL IS

    generic (eta      : real := 1.0      ; -- Pente du Laser
             etai     : real := 1.0     ; -- Injection efficiency
             beta     : real := 1.0e-6 ; -- Fraction d'emission spontanee dans le mode
             ng      : real := 3.5     ; -- Indice de refraction de groupe
             ni       : real := 1.0    ; -- Indice du milieu
             eps     : real := 5.0e-7 ; -- Coefficient de compression de gain
             AlphaH  : real := 2.0     ; -- Facteur de Henry
             m       : real := 0.33    ; -- Parametre empirique
             Phibi   : real := 0.2     ; -- "built-in voltage" aux bornes de la jonction
             Cj0     : real := 0.5e-12 ; -- Capacité de jonction "zero-bias"
             Alpham  : real := 2.0e3   ; -- Coefficient de pertes miroir
             Go      : real := 1.6e4   ; -- Taux de gain differentiel
             No      : real := 1.940e7 ; -- Nombre de porteurs à la transparence
             TauN    : real := 5.0e-9  ; -- Temps de vie des porteurs
             TauP    : real := 2.28e-12; -- Temps de vie des photons
             a0      : real := 138.929e-3; -- Coef. du courant en temperature
             a1      : real := -100.978e-5; -- Coef. du courant en temperature
             a2      : real := 15.5145e-7; -- Coef. du courant en temperature
             a3      : real := 29.949e-10; -- Coef. du courant en temperature
             a4      : real := -6.4491e-12; -- Coef. du courant en temperature
             ag0     : real := -0.4    ;
             ag1     : real := 0.00147 ;
             ag2     : real := 7.65e-7 ;
             bg0     : real := 1.3608  ;
             bg1     : real := -0.00974;
             bg2     : real := 1.8e-5  ;
             T0      : real := 298.0   ; -- Température ambiante en °K
             Lambda0: real := 850.0e-9; -- Wavelength
             De_Lam : real := 0.1e-9   ; -- Spectral width (FWHM)
             DLT     : real := 0.06e-9 ; -- Wavelength temperature coefficient
             Rvo     : real := 50.0    ; -- Resistance thermique entre VCSEL et 1er
voisin
             RGf    : real := 52.0e3   ; -- Resistance thermique entre VCSEL et 1er
voisin
             RASGa  : real := 45.6     ; -- Resistance thermique entre VCSEL et 2eme
voisin
             Rrep   : real := 3.5      ; -- Resistance thermique entre VCSEL et le
boitier
             Rwb    : real := 14.8     ; -- Resistance thermique entre VCSEL et le
substrat
             C1     : real := 1.0      ; -- Capacité thermique
             Ca     : real := 1.0      ; -- Capacité thermique
             P0     : real := 2.4e-3   ; -- Puissance Optique nominale
             Dnn    : real := 0.0      ; -- Coeffitition de diffusion des porteurs
             Dss    : real := 0.0      ; -- Coeffitition de diffusion des photons
             Dpp    : real := 0.0      ; -- Coeffitition de diffusion de phase
             RIN    : real := -130.0   ; -- Relative Intensity noise en sortie (dB/Hz)
end ENTITY VCSEL IS
```

```

        Po    : real := 2.3e-3 ;    -- Puissance optique nominale
        BWC   : real := 8.0e9) ;    -- Bande passante

    port    (terminal    anode,cathode : electrical;
            terminal    Tint : thermal;
            quantity    Popt,PdBm,Lambda : out real;
            quantity    UN11,UN12 : real);    -- Modèle avec bruit

END ENTITY VCSEL;

```

Architecture utilisant le bruit RIN (les effets thermiques sont commentés):

```

use ieee.math_real.all;

ARCHITECTURE avec_bruit_RIN OF vcsl IS

--- déclaration des quantités des ports
    quantity Vd across I,ic through Anode to Cathode
    quantity Ti across Tint to Thermal_ref;

--- Déclaration des constantes
    constant C : real := 3.0e8;    -- Speed of light in vacuum, m/s
    constant physical_Q : real := 1.602_177_33e-19; -- Charge of electron, coulomb
    constant physical_K : real := 1.380_658e-23; -- Boltzman's constant, joule/kelvin;
    constant Vg : real := C/ng ;    -- Vitesse de groupe
    constant physical_H : real := 6.626_075_5e-34; -- Plank's constant, joule*second

--- déclaration des quantités libres
    quantity Ioff : real ;    -- Offset Current
    quantity N : real ;    --:=1.0e7;    -- Electron Number
    quantity S : real ;    --:=0.5e5;    -- Photon Number
    quantity nu : real ;    -- Frequence optique
    quantity G : real := 3.0e50; -- Gain G
    quantity Pop : real ;    -- Puissance optique
    quantity Cj : real ;    -- Capacit  de jonction
    quantity nu0,QLambda,eta_eff : real ;
    quantity GWN : real ;
    quantity Prin : real ;

BEGIN

--- Initialisation du syst me

--- Calcul du bruit blanc gaussien utilis  pour le RIN
GWN == SQRT( - 2.0*Log(UN11)) * cos(math_2_pi*UN12);

--- Calcul de la capacit  de jonction
IF Vd>=0.0
    USE
        Cj == Cj0 * (1.0 + m * Vd / Phibi);
    ELSE
        Cj == Cj0 / (1.0 - Vd / Phibi)**m;
END USE;
--- Courant de la capacit  de jonction
ic == Cj * Vd'dot;

--- Tension aux bornes de la diode
Vd == (physical_K * Ti/ physical_Q) * Log (ABS(1.0 + (I/ 1.0e-14))) ;

--- Calcul du courant Ioff et du gain G en fonction de la temp rature
Ioff == 0.0;    --a0 + a1 * Ti + a2 * Ti**2.0 + a3 * Ti**3.0;-- + a4 * Ti**4.0 ;
G == Go;    --* (ag0 + ag1*Ti + ag2*Ti**2) /(bg0 + bg1*Ti + bg2*Ti**2);

--- Equation de continuit  des porteurs
N'dot == etai * (I-Ioff)/physical_Q - N/TauN - G * (N - No) * S / (1.0 + eps * S) ;
--- Equation de continuit  des photons
S'dot == - S/TauP + Beta * N / TauN + G * (N - No) * S / (1.0 + eps * S) ;
--- Equation de continuite de la phase

```

```

nu == nu0 + AlphaH / 2.0 * ( G * (N - No)/(1.0 + eps * S) - 1.0/TauP );

nu0 == C / QLambda ;
QLambda == De_Lam + Lambda0 + (Ti-T0)* DLT ;

--- Calcul du rendement efficace
eta_eff == eta *(1.0 + (Ti - 298.0) * 0.65e-2);

--- Puissance optique
Pop == (physical_H * nu * Alpham * Vg / 2.0) * S + Prin;
--- Bruit optique
Prin == GWN * sqrt(10.0**(RIN/10.0) * BWC * Po**2);

--- Sorties
Popt == Pop;
PdBm == 10.0 * Log (ABS(Pop) / 1.0e-3);
Lambda == QLambda;

END ARCHITECTURE avec_bruit_RIN;

```

Annexe VIII : Implémentation Simplorer de la Photodiode

Les sources de bruit ont été extraites du modèle et les grandeurs correspondantes entrent par les quantités "UN11 ... UN32" :

```

library ieee;
use ieee.electrical_systems.all;

ENTITY photodiode IS
    generic (tr      : real := 100.0e-12;      -- Temps de montée
            tf      : real := 100.0e-12;      -- Temps de descente
            R        : real := 0.6;           -- Sensibilité de la photodiode en
mA/mW
            m        : real := 0.33;         --
            Phibi    : real := 0.2;         --
            Isat     : real := 10.0e-9;      -- Courant de saturation
            Tnom     : real := 295.0;        -- Temperature nominale
            T        : real := 298.0;        -- Température ambiante en °K
            Rshunt   : real := 1.0e9;        -- Resistance
            Rl       : real := 1.0e3;        -- Resistance de charge
            Cj0      : real := 0.5e-12;      -- Capacité de jonction
            Bp       : real := 5.0e12;
            Pmin     : real := 0.0159e-3;
            Pmax     : real := 0.5e-3;
            Eg       : real := 1.519;
            alphaT   : real := 5.405e-4;
            betaT    : real := 204.0;
            tau      : time := 1ns);        -- Noise

    port (terminal Anode,Cathode,CathodeIn : electrical;
          quantity UN11,UN12,UN21,UN22,UN31,UN32,Lambda,Pin : in real);
END ENTITY photodiode;

```

```

use ieee.math_real.all;

ARCHITECTURE behav OF photodiode IS
--- Déclaration des constantes
    constant C          : real := 3.0e8;    -- Speed of light in vacuum, m/s
    constant physical_Q : real := 1.602_177_33e-19; -- Charge of electron, C
    constant physical_K : real := 1.380_658e-23; -- Boltzman's constant,
J/K
    constant physical_H : real := 6.626_075_5e-34; -- Plank's constant, J.s

--- déclaration des quantités des ports
    quantity Vd across Anode to CathodeIn;
    quantity I,In_d,In_q,In_t through Cathode to Anode;
--- déclaration des quantités libres
    quantity Iphoto : real; -- Courant photonique
    quantity TauOpt : real; -- Constante de temps optique
    quantity TauRC  : real; -- Constante de temps électrique
    quantity Cj     : real; -- Capacité
    quantity Popteff : real; -- Puissance optique efficace
    quantity Idark  : real := 0.1; -- Courant d'obscurité
    quantity Rdt    : real; -- Responsivité fonction de la T
    quantity EgT,EgTnom : real;
    quantity IsatT  : real;

```

```

quantity      GWN1,GWN2,GWN3 : real;
quantity      fact              : real;

BEGIN

--- Génération des bruits
GWN1 == SQRT( - 2.0*Log(UN11)) * cos(math_2_pi*UN12);
GWN2 == SQRT( - 2.0*Log(UN21)) * sin(math_2_pi*UN22);
GWN3 == SQRT( - 2.0*Log(UN31)) * cos(math_2_pi*UN32);

--- Calcul des bruit
In_d  == 0.1 * SQRT (ABS(2.0 * physical_Q * Bp * Idark)) * GWN1;
In_q  == 0.1 * SQRT (ABS(2.0 * physical_Q * Bp * Rdt * Pin)) * GWN2;
In_t  == 0.1 * SQRT (ABS(4.0 * physical_K * Bp * T /Rshunt)) * GWN3;

--- Calcul de la capacité de jonction
IF Vd>=0.0
  USE
    Cj == Cj0 * (1.0 + m * Vd / Phibi);
  ELSE
    Cj == Cj0 / (1.0 - Vd / Phibi)**m;
  END USE;

--- Calcul de la puissance optique efficace
IF Pin<Pmin USE
  Popteff == Pmin;
ELSIF Pin>Pmax USE
  Popteff == Pmax;
ELSE
  Popteff == Pin;
END USE;

--- variation de la responsivite de la PIN en fonction de la Temperature (a travers la
variation de Lambda)
Rdt  == R;
--- Courant d'obscurité
Idark == IsatT * (exp((physical_Q * Vd) / (physical_K * T)) - 1.0);
--- Courant total
I == -Iphoto + IsatT*(exp((physical_Q*Vd)/(physical_K*T))-1.0) - Cj*Vd'dot -
Vd/Rshunt;
--- Courant photonique
Rdt * Popteff == Iphoto + TauOpt / 2.2 * Iphoto'dot ;
--- Constante de temps optique
TauOpt == SQRT(ABS(tr**2 - TauRC**2));
--- Constante de temps électrique
TauRC == 2.2 * Cj * Rl;
--- Effet de la temperature

--- Courant de saturation
fact == ((physical_Q * EgTnom) / (physical_K*Tnom)) - ((physical_Q * EgT) /
(physical_K*T)) + (3.0 * Log(T/Tnom));
IsatT == Isat * exp(fact);

--- Niveau Energie
EgT == 1.16 - (7.02e-4 * (T**2/ (T+1108.0))) ;
EgTnom == 1.16 - (7.02e-4 * (Tnom**2/ (Tnom+1108.0))) ;

END ARCHITECTURE behav;

```

Annexe IX : Diagramme de l'oeil

La validation de la simulation du module complet est effectuée par des procédures de test. L'entrée est constituée d'un signal bruité spécifié par les normes. A la sortie nous obtenons le « diagramme de l'œil » dimensionné suivant un gabarit standard. Ce diagramme de l'œil pourra donner le pire cas pour un module de datacom parallèle. Les paramètres tels que la température, le taux de modulation, la puissance, les contraintes mécaniques et géométriques seront utilisés comme des paramètres génériques. Les résultats de simulation sont directement influencés par les valeurs de ces paramètres. A partir du diagramme de l'œil on extrait le rapport signal à bruit S/B (ou le facteur Q) et la gigue. Le diagramme de l'œil constitue une donnée essentielle quant au bon fonctionnement du système.

C'est la mesure que nous pourrons effectuer par la suite et qui sera déterminante quant aux conclusions sur le fonctionnement général du module.

Les normes de communication IEEE802.3ae (10Gbps), STM8/OC24 (1.25Gbps), 10G Fiber channel (10Gbps) et VSR préconisent des tests basés sur la mesure du diagramme de l'œil.

Selon ces normes, les caractéristiques de la forme de l'impulsion de sortie seront spécifiées par un masque dans le diagramme de l'œil. Le masque consiste en une fenêtre hexagonale et deux bandes interdites, il définit une zone où le BER est inférieur à une valeur standard (10^{-12}). Les temps normalisés pour le « 0 » et le « 1 » sur l'échelle de l'intervalle unitaire seront déterminés par le croisement à une valeur moyenne du diagramme de l'œil.

Les spécifications en matière de diagramme de l'œil sont les suivantes :

- Masque de type hexagonal
- Rectangle intérieur : 0,2 UI
- Hexagone extérieur : 0,4 UI
- Ouverture verticale : 50 %

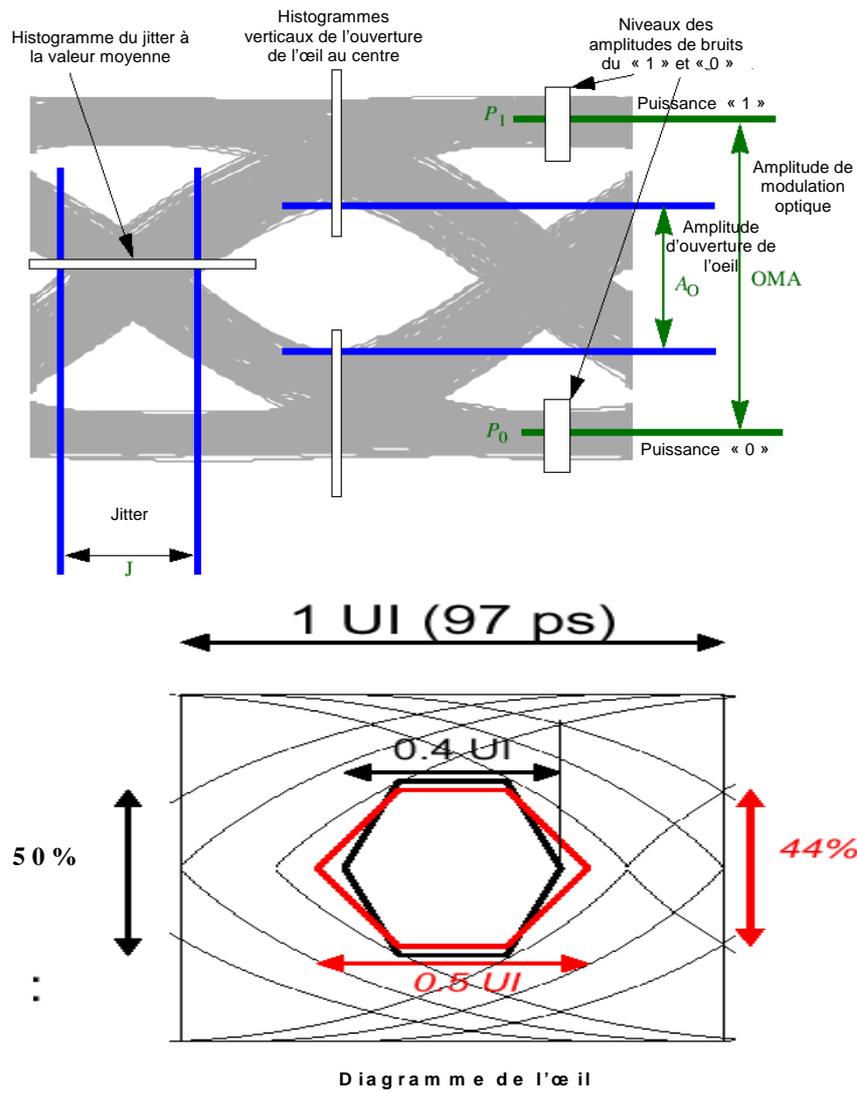


Fig. 62 - Dimensions caractéristiques et masques pour les diagrammes de l'œil

Modélisation multidisciplinaire VHDL-AMS de systèmes complexes : vers le Prototypage Virtuel

Les systèmes industriels sont de plus en plus miniaturisés et intégrés. Ils font cohabiter sur le même substrat différentes disciplines techniques (électronique analogique et numérique, optique, mécanique, thermique, différents capteurs ...). Le cycle de conception traditionnel en V ne répond plus aux exigences industrielles et beaucoup de problèmes restent à identifier et à résoudre pour avancer dans ce domaine. Les phénomènes perturbateurs prennent de plus en plus d'importance à mesure que la taille des systèmes diminue. Leur conception doit en tenir compte dès le début de la démarche.

L'évolution industrielle dans ce domaine passe par la fourniture de procédures réalistes et productives pour améliorer le cycle de conception. Le Prototypage Virtuel constitue une voie intéressante pour cela en offrant la possibilité de réduire les temps et les coûts de développement par la formalisation, la capitalisation et la réutilisation de modèles informatiques directement simulables, ainsi que par l'application d'une méthodologie rigoureuse quant à leur conception. Parmi les langages de description matériels utilisables, VHDL-AMS semble un des plus adaptés. En effet, il permet de modéliser les systèmes multidisciplinaires de manière native et offre des possibilités de modélisation à haut niveau d'abstraction. De plus, ce langage se déploie de plus en plus au sein des industries, ce qui génère un nouveau public d'utilisateurs, n'ayant pas de connaissances des HDLs, pour lesquels la nouvelle génération de simulateurs graphiques s'avèrent très utiles.

Cette thèse présente tout d'abord le travail effectué dans le cadre du RMNT sur la modélisation d'un dispositif optoélectronique à haut débit, courte portée et à fortes interactions thermiques et mécaniques de la société THALES. Cette étude a ensuite permis de dégager des éléments de méthodologie qui pourraient rendre la conception de modèles et la simulation de systèmes complexes plus en phase avec un marché toujours plus exigeant.

Mots clés : VHDL-AMS, Prototypage Virtuel, modélisation, simulateurs, optoélectronique

VHDL-AMS multidisciplinary modelling of complex systems: toward Virtual Prototyping

The industrial devices are more and more miniaturized and integrated. They allow various technical fields (digital and analog electronics as well as optics, mechanics, thermals or sensors...) to share the same chip. The traditional V design cycle does not fulfil the industrial requirements any more and a lot of bolts remain to identify and solve to breakthrough this field. The disturbing phenomena take more and more importance as the size of the systems decreases. Their design must take this into account from the very beginning of the step.

The industrial evolution in this field implies the development of realistic and productive procedures to improve the design cycle. Virtual Prototyping offers an interesting solution making it possible to reduce times and costs of development by the formalization, capitalization and re-use of the models which can be directly simulated, like by the application of a rigorous methodology as for their design. Among the Hardware Description Languages available, VHDL-AMS seems to fit the most. Indeed, it makes it possible to model the multidisciplinary systems in a native way and offers possibilities of modelling at high levels of abstraction. Moreover, this language is spread more and more within industries, which generates a new public of users, not having knowledge of HDLs, for which the new generation of graphic simulators prove to be very useful.

This thesis first of all presents the work carried out within the framework of the RMNT on the modelling of an optoelectronic device at high flow, short range and with strong thermal and mechanical interactions of the THALES Company. This study then made it possible to bring out elements of methodology which could make the design of models and the simulation of complex devices more in phase with a market increasingly more demanding.

Keywords: VHDL-AMS, Virtual Prototyping, modelling, simulators, optoelectronics