

THESE

présenté par

Bachir BERKANE

pour obtenir le grade de DOCTEUR

de L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

*(arrêté ministériel du 30 mars 1992)*

(Spécialité : Signal - Image - Parole)

---

---

**Vérification des systèmes matériels numériques  
séquentiels synchrones**

Application du langage Lustre et de l'outil de vérification  
Lesar

---

---

Date de soutenance : 2 octobre 1992

Composition du jury :	Président	G. Mazaré
	Rapporteurs	F. Anceau P. Prinetto
	Examineurs	G. Thuau V. Olive

Thèse préparée au sein du Laboratoire de Génie Informatique

# Sommaire

<b>Introduction Générale</b>	<b>1</b>
Problématique .....	1
Objectif de l'étude .....	2
Plan de lecture .....	3
<b>I Les systèmes matériels numériques séquentiels synchrones : caractéristiques et validation</b>	<b>5</b>
<b>1 Les systèmes matériels séquentiels synchrones : SMSS</b> .....	<b>7</b>
1.1 Caractéristiques d'un SMSS .....	8
1.1.1 Modèle mathématique .....	8
1.1.2 Représentation d'un SMSS .....	9
1.2 Interconnexion des SMSS .....	11
1.2.1 Interconnexion de systèmes évoluant sur la même horloge .....	11
1.2.2 Exemple .....	13
1.2.3 Cas des systèmes évoluant sur des horloges différentes .....	14
1.3. Conclusion .....	15
<b>2 Validation des systèmes matériels séquentiels</b>	<b>17</b>
2.1 Description des systèmes matériels .....	17
2.2 Les différentes approches pour la vérification formelle des systèmes matériels séquentiels .....	18
2.2.1 La preuve déductive .....	19

2.2.2	L'évaluation d'une formule sur un modèle de machine d'états finis " associé au système matériel : "model checking" .....	20
2.3	Un cadre unifié pour la description et la vérification des SMSS .....	22
<b>II</b>	<b>Une approche de vérification unifiée</b> .....	<b>25</b>
<b>3</b>	<b>Spécification des propriétés temporelles</b> .....	<b>27</b>
3.1	Terminologie et notations .....	27
3.2	Les différentes classes de propriétés d'un système matériel .....	28
3.3	Spécification des propriétés de sûreté .....	29
3.3.1	Événements définis .....	30
3.3.2	Intervalle d'observation .....	31
3.3.3	Réaction d'une propriété .....	32
3.3.4	Propriétés de sûreté .....	33
3.3.5	Opérateurs de base .....	34
3.4	Conclusion .....	35
<b>4</b>	<b>Vérification des propriétés de sûreté</b> .....	<b>37</b>
4.1	Les machines associées aux propriétés de séquences finies .....	37
4.1.1	Les machines associées aux événements définis .....	38
4.1.2	Les machines associées aux propriétés d'intervalles et de réactions .....	39
4.2	Les machines associées aux propriétés de sûreté .....	42
4.3	La méthode de vérification .....	44
4.4	Conclusion .....	45
<b>5</b>	<b>Vérification de conformité</b> .....	<b>47</b>
5.1	Les opérateurs de comparaison .....	48
5.1.1	L'opérateur de synchronisation $\Sigma$ .....	48
5.1.2	L'opérateur de retard $\Delta$ .....	50
5.1.3	L'opérateur parallèle $\Pi$ .....	50
5.2	Les machines comparables .....	51
5.2.1	Cas d'une réalisation et d'une spécification de même structure .....	51
5.2.2	Cas d'une réalisation pipeline .....	51
5.2.3	Cas d'une réalisation parallèle et d'une spécification série .....	51
5.2.4	Cas d'une réalisation série et d'une spécification parallèle .....	51

5.3 Critères d'observation .....	54
5.4 La méthode de vérification .....	55
5.5 Conclusion .....	56
<b>6 Vérification sous un environnement</b> .....	<b>57</b>
6.1 Le langage accepté et les propriétés d'environnement.....	58
6.2 Modélisation de l'environnement d'un système matériel .....	58
6.2.1 Le modèle reconnaisseur .....	58
6.2.2 Le modèle générateur .....	60
6.3 La vérification sous un environnement .....	62
6.3.1 Vérification avec un modèle reconnaisseur de l'environnement.....	62
6.3.2 Vérification avec un modèle générateur de l'environnement.....	63
6.4 Conclusion .....	64
<b>III Application du Langage Lustre de l'outil de vérification Lesar</b> .....	<b>65</b>
<b>7 Le langage Lustre</b> .....	<b>67</b>
7.1 Horloges et flots .....	68
7.2 Variables, équations et expressions .....	69
7.2.1 Opérateurs sur les données.....	69
7.2.2 Opérateurs sur les suites .....	69
7.3 Les assertions .....	71
7.4 Structuration des programmes.....	71
7.5 Tableaux et structures.....	73
7.5.1 Opérateurs de construction .....	73
7.5.2 Opérateur de sélection .....	74
7.5.3 Opérateur de concaténation.....	74
7.5.4 Assertions sur les structures et tableaux .....	74
7.5.5 Extension homomorphe des opérateurs .....	75
7.5.6 Paramètres statiques et récursivité .....	75
7.6 Conclusion .....	76
<b>8 Sémantique de Lustre en termes de machines d'états finis</b> .....	<b>79</b>
8.1 Les opérateurs sur les données .....	79
8.2 Les opérateurs temporels.....	80

8.2.1	L'opérateur "pre" .....	80
8.2.2	L'opérateur "→" .....	81
8.2.3	L'opérateur "when" .....	82
8.2.4	L'opérateur "current" .....	83
8.3	Un programme Lustre .....	84
8.4	Les assertions .....	86
8.5	Conclusion .....	88
<b>9</b>	<b>Un cadre unifié pour la description et la vérification fonctionnelle</b>	
	<b>des SMSS</b> .....	<b>89</b>
9.1	Description des SMSS .....	90
9.1.1	Description de la spécification d'un SMSS en Lustre .....	90
9.1.2	Exemple de spécification d'un SMSS en Lustre .....	91
9.1.3	Description de la réalisation d'un SMSS en Lustre .....	93
9.1.4	Exemple de description d'une réalisation d'un SMSS en Lustre .....	93
9.2	Vérification des propriétés de sûreté .....	96
9.2.1	Spécification des propriétés de séquences finies en Lustre .....	96
9.2.2	Spécification des propriétés de sûreté en Lustre .....	97
9.2.3	Exemple de spécification d'une propriété de sûreté .....	98
9.2.4	Programme de vérification .....	99
9.2.5	Exemple .....	100
9.3	Vérification de conformité .....	102
9.3.1	Les opérateurs de comparaison .....	102
9.3.2	Obtention des machines comparables en Lustre .....	104
9.3.3	Programme de vérification .....	105
9.4	Vérification sous un environnement .....	106
9.4.1	Description du modèle reconnaisseur de l'environnement .....	106
9.4.2	Description d'un environnement générateur .....	108
9.4.3	Vérification avec un modèle reconnaisseur de l'environnement .....	109
9.4.4	Vérification avec un modèle générateur de l'environnement .....	110
9.4.5	Exemples de vérification sous un environnement .....	111
9.5	Conclusion .....	115
<b>10</b>	<b>L'outil de vérification Lesar</b> .....	<b>117</b>
10.1	Obtention de la machine d'états finis .....	117
10.1.1	Normalisation du programme .....	118
10.1.2	Identification du modèle .....	119

10.2 Méthodes d'exploration de la machine de vérification .....	119
10.2.1 La méthode énumérative .....	120
10.2.2 La méthode symbolique "en arrière" .....	122
10.2.3 Résultats expérimentaux .....	123
10.2.4 Discussion.....	124
10.3 Conclusion .....	125
<b>11 Diagnostic</b> .....	<b>127</b>
11.1 Exécution du programme de vérification .....	128
11.2 Technique des assertions .....	129
11.3 Un analyseur logique virtuel .....	131
11.4 Conclusion .....	131
<b>Conclusion et perspectives</b> .....	<b>133</b>
<b>Bibliographie</b> .....	<b>137</b>
<b>Annexe : Bibliothèque d'opérateurs temporels en LUSTRE</b> .....	<b>145</b>

**Résumé de thèse :** La validation fonctionnelle d'un système matériel consiste à vérifier le système vis-à-vis de son fonctionnement attendu. Il existe deux façons de spécifier ce fonctionnement attendu. D'une part, la spécification peut être donnée sous forme d'une description fonctionnelle complète. D'autre part, l'expression de cette spécification peut être donnée sous forme d'un ensemble de propriétés temporelles critiques. Ces deux façons de spécifier les systèmes matériels ont donné lieu à deux problèmes de vérification. Notre domaine d'étude concerne les *systèmes matériels numériques séquentiels synchrones*. Ce document développe une approche de vérification unifiée, fondée sur le modèle de machines d'états finis, pour résoudre les deux problèmes de vérification sur ces systèmes. Dans cette approche, tout problème de vérification se ramène à définir une machine d'états finis sur laquelle la vérification sera réalisée. L'application du langage Lustre et de l'outil de vérification Lesar associé a été étudiée dans le but de valider cette approche. Dans cette application, la résolution des deux problèmes de vérification se ramène à définir un programme Lustre ayant une seule sortie. La vérification consiste à vérifier que cette sortie est la constante booléenne 1. Cette vérification est réalisée automatiquement par l'outil de vérification Lesar.

**Mots-clés :** systèmes matériels numériques séquentiels synchrones, machines d'états finis, vérification fonctionnelle, propriétés temporelles, environnement d'un système matériel, langage de description du matériel, outil de vérification, diagnostic.

**Abstract:** The functional verification of hardware design consists in checking that a given system operates as specified. There are mainly two kinds of high level specifications used in hardware design: (1) a complete specification given by a high abstraction level and (2) a partial high level specification made up of a set of properties that the system must satisfy. These two different kinds of hardware specifications give rise to two different verification problems. This thesis presents a unified framework to handle the two verification problems on synchronous sequential digital hardware systems. Each verification problem comes down to defining a verification machine with a single output Flag and to verifying that Flag is always true. The application of the language Lustre and the verification tool Lesar is studied. Within this application, the designers can face their verification problems by defining a Lustre program with a single output flow. The verification comes down to checking that this output flow is the constant true. This verification is automatically performed by the verification tool Lesar.

**Keywords:** synchronous sequential digital hardware systems, finite state machines, functional verification, temporal properties, hardware system environment, hardware description language, verification tool, diagnosis.