

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la recherche Scientifique

Université de Batna
Faculté de Sciences de l'Ingénieur
Département d'informatique



THÈSE DE MAGISTER

Présentée par : *Riadh HOCINE*

Spécialité : *Informatique*

Option : *Informatique Industrielle*

THÈME

**VALIDATION PAR RÉSEAUX DE PETRI DE SPÉCIFICATIONS DE
CONTRÔLEURS DÉDIÉS DÉCRITS DANS UN ENVIRONNEMENT
DE SYNTHÈSE DE HAUT NIVEAU**

Soutenue le : 02/07/2002, devant le jury composé de

<i>M. Baatouche</i>	M.C	U. Constantine	Président
<i>M. Benmohamed</i>	M.C	U. Constantine	Rapporteur
<i>M.K. Kholladi</i> (Doctorat)	C.C	U. Constantine	Examineur
<i>B. Bouchana</i> (Doctorat)	C.C	U. Biskra	Examineur

Résumé

La conception de systèmes de commande spécifique est devenue une tâche actuellement pratiquement automatique. Des outils de plus en plus développés ont été conçus pour aider les concepteurs de contrôleurs dédiés.

La conception démarre à partir d'une spécification algorithmique du circuit à concevoir, un compilateur comportemental fournira en sortie une description RTL du circuit à concevoir, d'autres compilateurs vont fournir par la suite jusqu'à la description du masque du circuit.

Dans le cadre de cette thèse, on s'intéresse au niveau comportemental, la spécification fournie par le concepteur est simulée pour valider cette spécification. Actuellement, on essaie d'utiliser des outils formels pour vérifier quelques propriétés au niveau de cette spécification initiale.

Les réseaux de Petri constituent aujourd'hui l'un des modèles les plus utilisés pour la modélisation du comportement dynamique des systèmes discrets. Dans ce travail, on essaie donc d'utiliser ces réseaux pour valider certains aspects de la spécification.

Il s'agit donc de proposer un outil de vérification basé sur ses réseaux de Petri : on part d'une spécification d'un contrôleur dédié décrit dans un langage de spécification de matériel (VHDL), une première compilation est faite pour vérifier certaines erreurs, la sortie sera traduite sous forme d'un réseau de Petri, qui sera par la suite simulé dynamiquement pour vérifier d'autres types d'erreurs.

Cet outil est un modèle formel basé sur les réseaux de Petri interprétés, adapté à la représentation des spécifications comportementales des contrôleurs décrits en VHDL, et un modèle de simulation dirigé par événements pour la simulation du modèle représentant la spécification. Les résultats sont analysés en appliquant quelques propriétés relatives au formalisme des réseaux de Petri pour détecter quelques éventuelles erreurs de spécifications qui ne peuvent être détectées dans la compilation.

Mots-clés

vérification, réseaux de Petri, spécification de contrôleurs, simulation, synthèse de haut niveau, VHDL.



SOMMAIRE

CHAPITRE 1 : INTRODUCTION ET PROBLEMATIQUE	1
1.1 INTRODUCTION	1
1.2 VERIFICATION DES SYSTEMES DIGITAUX	1
1.2.1 <i>La validation et sa problématique</i>	1
1.2.2 <i>Techniques de validation</i>	4
1.3 PLAN DE LA THESE	5
CHAPITRE 2 : CAO DES CONTROLEURS	6
2.1 INTRODUCTION	6
2.2 LES CONTROLEURS	6
2.2.1 <i>Présentation interne d'un contrôleur</i>	6
2.3 MOTIVATIONS	7
2.4 LA CONCEPTION "ZERO DEFAUT" D'ARCHITECTURES COMPLEXES	8
2.5 SYNTHESE DE CONTROLEURS	9
2.5.1 <i>Niveaux d'abstraction et modèles</i>	9
2.5.2 <i>Différents niveaux de synthèse</i>	11
2.6 LA VERIFICATION	15
2.7 CONCLUSION	16
CHAPITRE 3 : LA SYNTHESE DE HAUT NIVEAU	17
3.1 INTRODUCTION	17
3.2 ABSTRACTION AU NIVEAU COMPORTEMENTALE	18
3.3 DOMAINES D'APPLICATION DE LA SYNTHESE COMPORTEMENTALE	18
<i>Circuits orientés flots de données</i>	18
<i>Circuits dominés par contrôle (contrôleurs)</i>	18
3.3.1 <i>Graphe de flot de contrôle (CFG)</i>	18
3.3.2 <i>Graphe de flot de données (DFG)</i>	19
3.3.3 <i>Applications mixtes</i>	20
3.4 PROCESSUS DE SYNTHESE COMPORTEMENTALE	21
3.4.1 <i>Description comportementale</i>	21
3.4.2 <i>Compilation et élaboration de la description comportementale</i>	22
3.4.3 <i>Ordonnancement</i>	22
3.4.4 <i>Allocation</i>	23
3.4.5 <i>Optimisation de haut niveau</i>	23
3.4.6 <i>Génération d'architecture</i>	23
3.5 LES AVANTAGES DE LA SYNTHESE DE HAUT NIVEAU	23
3.6 VERIFICATION ET VALIDATION D'UNE DESCRIPTION COMPORTEMENTALE	24
3.7 OUTILS DE LA SYNTHESE D'ARCHITECTURES COMMERCIALISES	25
3.7.1 <i>Behavioral Compiler</i>	25
3.7.2 <i>Visual Architect</i>	26
3.7.3 <i>AMICAL</i>	26
3.7.4 <i>MACH (Montpellier Architecture Compiler for Hardware)</i>	27
CHAPITRE 4 : LE LANGAGE VHDL	28
4.1 INTRODUCTION	28
4.2 LES LANGAGES DE DESCRIPTION DE MATERIEL	28
4.3 LE LANGAGE VHDL	29
4.4 LES PARTICULARITES DU LANGAGE VHDL	30
4.4.1 <i>Un langage fortement typé</i>	30
4.4.2 <i>Un langage de simulation</i>	31

4.5 SYNTAXE ET SEMANTIQUE DU LANGAGE VHDL	31
4.5.1 L'entité.....	31
4.5.2 L'architecture.....	32
4.5.3 Le processus	32
4.5.4 Les styles de description	35
4.5.5 L'élaboration d'un modèle exécutable	42
4.5.6 La Simulation VHDL	42
4.6 VHDL : UN LANGAGE D'ENTREE A LA SYNTHESE COMPORTEMENTALE.....	46
CHAPITRE 5 : LES RESEAUX DE PETRI	48
5.1 INTRODUCTION	48
5.2 CONCEPTS DE BASE	48
5.2.1 Réseau de Petri.....	48
5.2.2 Réseaux de Petri marqué.....	49
5.2.3 Graphe associé et notations matricielle	49
5.2.4 Franchissement de transition	50
5.2.5 Séquence de franchissement	50
5.2.6 Réseaux de Petri particulier	50
5.3 PROPRIETES DES RESEAUX DE PETRI	51
5.3.1 RdP borné.....	51
5.3.2 RdP Sauf.....	51
5.3.3 RdP vivant	52
5.3.4 RdP conforme.....	52
5.3.5 RdP propre (réinitialisable)	52
5.3.6 Le blocage	52
5.3.7 La persistance.....	52
5.3.8 Les invariants	52
5.4 METHODES D'ANALYSE DES RESEAUX DE PETRI	52
5.4.1 Analyse par énumération de marquage.....	53
5.4.2 Analyse par l'algèbre linéaire.....	53
5.4.3 Analyse par réduction.....	54
5.5 CLASSES DES RESEAUX DE PETRI	54
5.5.1 RdP autonome	54
5.5.2 RdP généralisé.....	54
5.5.3 RdP ordinaire	54
5.5.4 RdP à capacité.....	54
5.5.5 RdP coloré.....	54
5.5.6 RdP à priorité.....	55
5.5.7 RdP continu	55
5.5.8 RdP à prédicat.....	55
5.6 LES RESEAUX DE PETRI NON AUTONOMES.....	55
5.6.1 Réseaux de Petri synchronisés.....	55
5.6.2 Les Réseaux de Petri temporisés	58
5.6.3 Les réseaux de Petri interprétés	60
5.6.4 Algorithme d'interprétation.....	63
5.6.5 Les réseaux de Petri colorés.....	64
5.7 CHOIX DE LA CLASSE(POUR LA VALIDATION).....	65
CHAPITRE 6 : VERIFICATION DE SPECIFICATIONS DES CONTROLEURS : UN ETAT DE L'ART66	
6.1 INTRODUCTION	66
6.2 LA VERIFICATION PAR SIMULATION	66
6.2.1 Etapes d'exécution et structure d'un simulateur	68

6.2.2	<i>Compilateur</i>	68
6.2.3	<i>Génération du programme simulable</i>	68
6.2.4	<i>Exécution de la simulation</i>	69
6.3	LA VERIFICATION FORMELLE	70
6.4	VERIFICATION FORMELLE DU MATERIEL : UN ETAT DE L'ART	71
6.4.1	<i>La vérification formelle</i>	71
6.4.2	<i>Approches de vérification formelle</i>	72
6.4.3	<i>Les langages de spécifications formelles</i>	76
6.5	VERIFICATION DES SYSTEMES DECRITS EN VHDL	82
6.5.1	<i>Vérification des propriétés des systèmes décrits en VHDL</i>	82
6.6	VUE D'ENSEMBLE DE NOTRE CONTRIBUTION	85
CHAPITRE 7 : UN OUTIL DE VALIDATION BASE SUR LES RESEAUX DE PETRI.....		88
7.1	INTRODUCTION	88
7.2	CHOIX DE LA MODELISATION	88
7.2.1	<i>Restriction syntaxique du langage VHDL</i>	88
7.2.2	<i>Pourquoi le formalisme des réseaux de Petri ?</i>	92
7.2.3	<i>Présentation intuitive</i>	92
7.3	DEFINITION FORMELLE DE NOTRE MODELE	94
7.3.1	<i>L'espace des données externes</i>	94
7.3.2	<i>Le formalisme</i>	95
7.3.3	<i>Représentation graphiques</i>	96
7.3.4	<i>Règles de franchissement</i>	97
7.4	COMPILATION DES PROGRAMMES VHDL (TRADUCTION DES SPECIFICATIONS COMPORTEMENTALES DANS LE MODELE FORMEL)	99
7.4.1	<i>Phase 1. Construction de l'espace de données</i>	99
7.4.2	<i>Phase 2. Extraction des squelette de processus (le graphe de contrôle)</i>	100
7.4.3	<i>Phase 3. Construction de l'ordonnanceur</i>	109
7.5	ANALYSE DU MODELE SIMULE	110
7.5.1	<i>Analyse des résultats de la simulation</i>	110
7.5.2	<i>Analyse structurelle du réseau de Petri</i>	111
7.6	EXEMPLE D'ANALYSE	116
7.6.1	<i>Graphe de marquage accessible</i>	116
7.6.2	<i>Analyse des résultats</i>	117
7.7	COMPARAISON ENTRE NOTRE MODELE ET LE MODELE VPN.....	117
CHAPITRE 8 : CONCLUSION ET PERSPECTIVES.....		118
8.1	INTRODUCTION	118
8.2	CONTRIBUTION	118
8.3	PERSPECTIVES	119
BIBLIOGRAPHIE.....		120