

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTRE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE COLONEL EL-HADJ LAKHDAR BATNA

FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE



THÈSE DE MAGISTER

Présentée Par: HOUASSI Hichem

Spécialité: INFORMATIQUE

Option: Informatique Industrielle

Thème

Optimisation dans la Synthèse de Haut niveau Pour Circuits de Contrôle Dédiés

Soutenue devant le jury composé de:

A. ZIDANI	M.C	U.Batna	Président
M. BENMOUHAMMED	M.C	U.Constantine	Rapporteur
M.C. BAATOUCHE	M.C	U.Constantine	Examineur
M.K. KHOULADI	C.C	U.Constantine	Examineur

Résumé

L'expansion du marché de l'électronique, de l'informatique, du multimédia et des télécommunications, la complexité des circuits actuels a augmenté de manière exponentielle cette dernière décennie. Dans le but de faire face à un tel accroissement, il est nécessaire d'augmenter la productivité des concepteurs. Pour ce faire, des spécifications de plus haut niveau et des systèmes de synthèse exploitant celle-ci étaient requis. Aujourd'hui des outils de synthèse commençant avec une description comportementale permettant l'automatisation et l'accélération du processus de conception des circuits intégrés.

L'objectif de la synthèse comportementale est la génération d'une architecture, au niveau transfert de registres, composée d'un contrôleur et d'un chemin de données, à partir d'une description de style algorithmique (comportementale). La conception par la synthèse de haut niveau démarre à partir d'une description comportementale du circuit à concevoir et d'une bibliothèque de composants regroupant les unités fonctionnelles (ou unités d'exécution), les unités de stockage et les unités de communication, et par application des algorithmes d'ordonnancement et d'allocation des ressources fournira en sortie une description RTL du circuit à synthétisé, d'autres compilateurs sont fournis par la suite jusqu'à la description du masque du circuit.

Dans le cadre de ce travail on s'intéresse donc aux aspects optimisation à l'étape d'allocation de ressources. Cette optimisation est guidée par les soucis au niveau de la surface, vitesse et consommation. Plusieurs techniques sont connues à ce niveau, il s'agit donc de faire une synthèse et classification de ces techniques et de proposer une nouvelle technique, en utilisant de nouvelles techniques d'intelligence artificielle, et de point de vue pratique, serait de développer un outil qui génère une partie opérative (sous deux formes: graphique et textuelle) à partir de la sortie de l'ordonnancement, après une compilation du source en VHDL.

Des nouvelles techniques d'allocation et d'affectation des ressources utilisent des directives d'optimisation visent à réduire le circuit en terme de nombre de ressource, de temps et de consommation. Cependant ces techniques ne pouvant pas être appliqués aux problèmes de grande taille puisque ce problème est un problème NP-Complet, notre approche intègre une technique de résolution approchée qui est la technique de recuit simulé pour résoudre le problème d'allocation de façon approchée.

Mots-clés:

CAO-VLSI, Synthèse de haut niveau, VHDL, Allocation de ressources, Optimisation Multi-critères, Optimisation combinatoire.

Sommaire

LEXIQUE	V
LISTE DES FIGURE	VI
Introduction Générale	01
1. Introduction	01
2. Objectif du travail	01
3. Plan de travail	02
Chapitre I: CAO des circuits intégrés VLSI	04
1.1. Introduction	05
1.2. Les circuits intégrés VLSI	05
1.3. Les circuits de contrôle	06
1.3.1. La partie contrôle	06
1.3.2. La partie opérative	07
1.4. Conception des circuits de contrôle	07
1.4.1. Domaines de description	08
1.4.2. Niveaux d'abstraction	09
1.4.2.1. Niveau système	09
1.4.2.2. Niveau comportementale	09
1.4.2.3. Niveau Transfert de registres RTL	09
1.4.2.4. Niveau logique	09
1.4.2.5. Niveau physique	09
1.4.3. Les modèles du circuit à différents niveaux d'abstraction	10
1.4.4. Synthèse	11
1.4.4.1. La synthèse au niveau système	11
1.4.4.2. La synthèse architecturale (Synthèse de haut niveau: HLS)	13
1.4.4.3. La synthèse au niveau transfert de registre (RTL)	13
1.4.4.4. La synthèse logique	14
1.4.4.5. La synthèse physique	14
1.4.5. Outils vérification des résultats de la synthèse comportementale	14
1.4.5.1. La simulation	14
1.4.5.2. La vérification formelle	15
1.5. Conclusion	15
Chapitre II: La synthèse de haut niveau	16
2.1. Introduction	17
2.2. Principe général	17
2.3. Nécessité de la synthèse de haut niveau	18
2.4. Domaines d'application de la synthèse de haut niveau	19
2.5. Les flots de synthèse comportementale	19
2.6. Les étapes de la synthèse de haut niveau	21
2.6.1. Description comportementale	22
2.6.1.1. Les langages de description de matériel	22
2.6.1.1.1 Langages déclaratifs	22

2.6.1.1.2. Langages procédéraux	22
2.6.1.1.3. VHDL	23
2.6.2. Compilation et génération de la forme intermédiaire	23
2.6.2.1. Optimisation de haut niveau	23
2.6.2.2. Représentation intermédiaire de la description comportementale	25
2.6.2.2.1. Graphe de flot de données	25
2.6.2.2.2. Graphe de flot de contrôle	26
2.6.2.2.3. Graphe mixte (de flot de contrôle et de données)	27
2.6.3. L'ordonnancement	27
2.6.4. L'allocation des ressources	28
2.6.5. Génération de l'architecture	29
2.6.6. Interdépendance entre les étapes de synthèse	30
2.7. Définition du problème de la synthèse de haut niveau	30
2.8. Conclusion	31

Chapitre III: Allocation des ressources

3.1. Introduction	33
3.2. L'objectif de l'allocation de ressources	34
3.3. Les tâches d'allocation	34
3.3.1. Allocation des unités de stockage	34
3.3.2. Allocation des unités fonctionnelles	35
3.3.3. Allocation des connexions et génération de la partie opérative	35
3.4. Le chemin de données (Partie opérative)	36
3.4.1. Description des composants du chemin de données	37
3.4.1.1. Les unités fonctionnelles	37
3.4.1.2. Les unités de stockage	38
3.4.1.3. Les unités d'interconnexion	38
3.4.1.3.1 Architecture à base de Bus	38
3.4.1.3.2. Architecture à base de multiplexeur	39
3.4.1.3.3. Comparaison entre les différentes architectures	40
3.4.1.4. Les unités de communication	41
3.4.1.5. Les unités de connexion externe	41
3.5. Le partage des ressources	42
3.5.1. Variables et registres	42
3.5.2. Opérations et unités fonctionnelles	42
3.5.3. Transfert de données et unités de communication	43
3.5.3.1. Modèle à base de bus	44
3.5.3.2. Modèle point à point ou à base de multiplexeur	44
3.6. Regroupement des registres en RAMs	45
3.7. Regroupement des constantes en ROMs	46
3.8. Performances du circuit	46
3.8.1. La surface du circuit	46
3.8.2. La Performance de temps	48
3.9. Conclusion	51

Chapitre IV: Optimisation combinatoire

4.1. Qu'est ce qu'une optimisation combinatoire	53
4.1.1. Exemples des problèmes d'optimisation combinatoires	53
4.2. La complexité des algorithmes	54
4.3. Les classes des problèmes	54
4.4. La théorie de la NP-complétude	55
4.5. Le problème d'optimisation des ressources d'un allocateur dans la synthèse architecturale	55

4.6. Classification des algorithmes des problèmes d'optimisation combinatoire	56
4.6.1. Algorithmes exactes	56
4.6.1.1. La méthode de Séparation-évaluation (Branch & Bound)	57
4.6.2. Les heuristiques	57
4.6.2.1. L'heuristique gloutonne	58
4.6.2.2. Les algorithmes de listes	58
4.6.2.3. Les heuristiques aléatoires	58
4.6.2.4. La recherche locale	59
4.6.3. Les méthaheuristiques	60
4.6.3.1. Méthodes de voisinage	60
4.6.3.1.1. Le recuit simulé (Simulated annealing)	61
4.6.3.1.2. GRASP	62
4.6.3.2. Les algorithmes évolutifs	64
4.6.3.2.1. Les algorithmes génétiques	65
4.6.3.2.2. Les colonies de fourmis	65
4.6.3.3. Méthodes hybrides	68
4.7. Conclusion	68

Chapitre V: Approches de résolution du problème d'allocation 69

5.1. Approches de résolution du problème d'allocation des ressources	70
5.1.1. Approche constructive / itérative	70
5.1.1.1. Algorithme de base	70
5.1.1.2. Méthode de résolution (Algorithme de couplage maximal)	71
5.1.1.2.1. Modélisation du problème d'allocation des registres	71
5.1.1.2.1.1. Prise en compte des connexions	73
5.1.1.2.2. Modélisation du problème d'allocation des UFs	74
5.1.1.2.2.1. Le calcul des poids des arcs du graphe biparti	75
5.1.1.2.3. Allocation des interconnexions	76
5.1.1.3. Algorithme classique de Branch & Bound	77
5.1.2. Approches globales	78
5.1.2.1. Programmation linéaire entière (ILP)	77
5.1.2.2. Méthode utilisée dans l'outil Gaut_w	79
5.1.2.3. Méthode de partitionnement d'un graphe en cliques	80
5.1.2.3.1. Allocation des registres	80
5.1.2.3.2. Allocation des unités fonctionnelles	82
5.1.2.3.3. Allocation des interconnexions	82
5.1.2.3.4. Algorithme de C.J. Tseng	84
5.1.2.4. Algorithme d'arrêt gauche	86
5.1.3. Approche de raffinement itératif	88
5.1.3.1. Réallocation des registres et UFs	88
5.1.3.2. Réallocation des interconnexions	89
5.1.3.2.1. L'heuristique de fusionnement de multiplexeurs	90
5.1.3.2.2. L'heuristique de fusionnement de bus	91
5.2. Notre Approche	94
5.2.1. Le but de cette approche	95
5.2.2. La fonction coût	95
5.2.3. Formulation du problème	96
5.3. Conclusion	99

Chapitre VI: Conception & réalisation d'un Allocateur de ressources 100

6.1. Processus de conception	101
6.2. Description de l'entrée de notre logiciel	103
6.2.1. Exemple d'une MEF	103

6.2.2. Les transferts	103
6.2.3. Les opérations	103
6.2.4. Les appels des procédures	104
6.3. Les méthodes utilisées pour l'allocation de ressources	104
6.3.1. Compilation	104
6.3.1.1. Analyseur lexical	104
6.3.1.2. Analyse syntaxique	105
6.3.2. Allocation des unités fonctionnelles	106
6.3.3. Micro-ordonnancement	107
6.3.4. Allocation des registres	107
6.3.5. Allocation des Ports d'E/S	108
6.3.6. Allocation des connexions	108
6.3.7. Allocation d'interconnexions	109
6.4. La sortie de notre logiciel	109
6.4.1. Sortie textuelle	110
6.4.2. Sortie graphique	110
6.5. Exemples	110
6.5.1. Description du factoriel	110
6.5.2. Sortie textuelle	111
6.5.3. Sortie graphique	112
6.6. Description de GCD	113
6.6.1. Sortie textuelle.....	114
6.6.2. Sortie graphique	115
Chapitre VII: Conclusion & Perspectives	116
7.1. Synthèse	117
7.2. Perspectives et travaux futurs	118
Annexe A.....	121
Annexe B	125
Bibliographie	128