



N° d'ordre :MAG/ 24/2008

THÈSE

Présentée au Département de Physique
ÉCOLE NORMALE SUPÉRIEURE DE KOUBA - ALGER

PAR

M^{lle} Hadjira KARTOUT
Ingénieur d'État en Électronique «U.S.T.H.B»

Pour l'obtention du grade

MAGISTÈRE

SPÉCIALITÉ : PHYSIQUE APPLIQUÉE

THÈME

**Implémentation de l'Algorithme Cordic sur un FPGA
(Field Programmable Gate Array)
Application au Modèle de PARK**

Soutenue le : 26 /01/ 2009 , devant le jury :

M ^r A. LATEF	M.C	ENS Kouba	Président
M ^r A. NESBA	M.C	ENS Kouba	Examineur
M ^r M. HADDADI	Pr	ENP Alger	Examineur
M ^r A. MERAGHNI	Pr	ENS Kouba	Rapporteur
M ^r M. MAMOUNE	C.C	Université de Blida	Invité

Sommaire

INTRODUCTION GÉNÉRALE	1
CHAPITRE I : L'ALGORITHME CORDIC	3
1. Introduction	3
2. CORDIC classique	3
2.1. Types de fonctionnement	8
2.1.1. Type rotationnel	8
2.1.2. Type vectoriel	8
3. CORDIC généralisé	9
3.1. Systèmes de coordonnées	9
4. Domaines de convergences	13
4.1. Extension du domaine de convergence	13
5. Analyse de la précision du CORDIC	15
5.1. Erreur d'approximation	15
5.2. Erreur de troncature	16
5.3. Erreur due à la correction du facteur d'échelle	18
5.3.1. Erreur d'approximation	18
5.3.2. Erreur de troncature	18
5.4. Calcul de l'erreur de quantification globale	19
5.5. Normalisation des entrées pour le CORDIC vectoriel	20
5.6. Nombre d'itérations dans les différents modes	21
5.7. Erreur numérique et format de données	21
6. Simulation de l'algorithme	22
6.1. Mode Circulaire	23
6.1.1. Type Rotation	23
6.1.2. Type Vecteur	24
6.2. Mode Linéaire	26
6.2.1. Type Rotation	26
6.2.2. Type Vecteur	27
6.3. Mode Hyperbolique	28
6.3.1. Type Rotation	28
6.3.2. Type Vecteur	29

Sommaire (suite)

7. Conclusion	30
CHAPITRE II : TECHNOLOGIE DES FPGA	31
1. Introduction	31
2. Les circuits ASIC	32
2.1. Les circuits personnalisés	33
2.1.1. Les circuits à la demande	33
2.1.2. Les circuits précaractérisés	33
2.2. Les circuits semi-personnalisés	34
2.2.1. Les réseaux logiques programmables	34
2.2.2. Les prédiffusés	35
2.2.3. Les FPGA	35
2.3. Les avantages et inconvénients des circuits ASIC	35
3. L'architecture des FPGA	37
3.1. Les CLB (configurable logic bloc)	38
3.2. Les IOB (input output bloc)	38
3.3. Les différents types d'interconnexions	39
3.3.1. Les interconnexions à usage général	39
3.3.2. Les interconnexions directes	40
3.3.3. Les longues lignes	41
3.4. Technologie de la programmation	41
3.4.1. La technologie anti-Fuse	42
3.4.2. EPROM/EEPROM	42
3.4.3. Static RAM	43
3.5. Technique de programmation des FPGA	44
3.5.1. Mode maître série	45
3.5.2. Mode esclave	45
4. Le Virtex II	46
4.1. Introduction	46
4.2. Les caractéristiques de la famille Virtex II	46
4.3. Architecture du Virtex II	46
5. Les outils de conception	49
5.1. Le logiciel Xilinx ISE 7.1i	49
5.1.1. Généralités	49
5.1.2. Création d'un projet	50
5.1.2.1. Création d'un projet	50
5.1.2.2. La Saisie du fichier source	52
5.1.3. Les étapes pour la Simulation	53
5.1.3.1. Création d'un signal de test	53
5.1.3.2. Simulation	53

Sommaire (suite)

5.1.4.	Flot d'implantation	54
5.2.	Le logiciel ModelSim	58
6.	La carte de développement	58
6.1.	Introduction	58
6.2.	Système Virtex-II TM V2MB1000	59
6.3.	Description du sigle	59
6.4.	Présentation de la carte	60
6.5.	Le FPGA utilisé	61
6.6.	La génération d'horloge	61
6.7.	Le contact d'utilisateur DIP (SW2)	62
6.8.	Le port de JTAG	62
6.8.1.	Le connecteur de JTAG	62
6.8.2.	La chaîne de JTAG	63
6.8.3.	Les jumpers de JTAG	63
6.9.	Mode basse puissance de Virtex-II :	64
6.10.	Le commutateur de programme (SW2)	64
6.11.	Le port de LVDS	65
6.11.1.	l'interface de LVDS	65
7.	Le chargement du Kit	69
8.	Conclusion	69
CHAPITRE III : IMPLÉMENTATION DE L'ALGORITHME CORDIC		71
1.	Introduction	71
2.	Implémentation de l'algorithme Cordic	72
2.1.	Structure d'implémentation	72
2.2.	Choix de l'implémentation	74
3.	Synthese de l'algorithme CORDIC	74
3.1.	Le langage VHDL	75
3.2.	Synthèse VHDL du CORDIC	77
3.2.1.	Création d'un nouveau projet	79
3.2.2.	Création d'un fichier VHDL	80
3.2.3.	Les Process du code VHDL	81
3.2.3.1.	Process d'incréméntation	81
3.2.3.2.	Process générateur de constantes	82
3.2.3.3.	Process signal de chargement	83
3.2.3.4.	Process signal de validation	84
3.2.3.5.	Process calcul de X	85
3.2.3.6.	Process calcul de Y	88

Sommaire (suite)

3.2.3.7.	Process calcul de Z	88
3.2.4.	Synthèse du code VHDL global	90
3.2.4.1.	Rapport de synthèse	91
3.2.4.2.	Description au niveau RTL	93
4.	Simulation fonctionnelle de la description VHDL	94
4.1.	Création du fichier Test Bench	94
4.2.	Le rapport de simulation	97
5.	Implémentation du Cordic Classique	99
5.1.	Génération du fichier de configuration	100
5.2.	Procédure d'affectation des Entrées/Sorties	101
5.3.	Programmation du FPGA	102
6.	L'implémentation du CORDIC généraliser	103
6.1.	L'entité du Cordic généralisé	103
6.2.	Le Code VHDL du Cordic généralisé	104
6.3.	Synthèse du Cordic généralisé	109
6.4.	Le bloc RTL du Cordic généralisé	110
6.5.	Simulation fonctionnelle	111
6.6.	Implémentation du Cordic Généralisé	114
7.	Conclusion	115
CHAPITRE IV : APPLICATION AU MODÈLE DE PARK		116
1.	Introduction	116
2.	La modélisation de la machine asynchrone	116
2.1.	Modèle de la MAS en grandeurs de phases	117
2.2.	Le modèle de Park	119
2.3.	Modèle linéaire de la machine asynchrone	120
3.	L'implémentation de PARK sur FPGA	122
3.1.	Description	122
3.2.	Synthèse des sous bloc de PARK	125
3.2.1.	Création des symboles	125
3.2.1.1.	Composant ADD/SUB	125
3.2.1.2.	Sous_Composant ajout2bit	126

Sommaire (suite)

3.2.1.3.	Sous_Composant MsC2	127
3.2.1.4.	Sous_Composant subS18bits	129
3.2.1.5.	Sous_Composant addS18bits	130
3.2.1.6.	Sous_Composant c2ms18bits	130
3.2.1.7.	Synthèse du Composant ADD/SUB	131
3.2.1.8.	Analyse fonctionnelle du Composant ADD/SUB	131
3.2.2.	Composant Prerot	132
3.2.2.1.	Analyse fonctionnelle du Composant Prerot	133
3.2.3.	Composant Cordic Classique	134
3.2.3.1.	Analyse fonctionnelle du Composant Cordic	137
3.2.4.	Sous Composant PROD	138
3.2.5.	Le Composant PRODUIT	139
3.2.5.1.	Analyse fonctionnelle du Composant PRODUIT	140
3.2.6.	Le Composant ADD	142
3.2.6.1.	Analyse fonctionnelle du Composant ADD	143
3.2.7.	Le Composant PRODUIT1	145
3.2.7.1.	Analyse fonctionnelle du Composant PRODUIT1	146
3.3.	Synthèse du Modele PARK	147
3.3.1.	Rapport de synthèse de l'implémentation de Park	149
3.3.2.	Le Bloc RTL de PARK	149
3.3.3.	Analyse fonctionnelle du Composant PARK	150
3.3.4.	TEST sur la carte de développement	152
4.	Conclusion	152
	CONCLUSION GÉNÉRALE	153
	ANNEXE	185
•	Le Code developper en VHDL du Cordic classique	185
•	Les circuits interne du niveau RTL du Cordic Classique	185
•	Rapport de synthèse de l'implémentation du CORDIC Généralisé	185
•	Les circuits interne du niveau RTL du Cordic Généralisé	185
•	Rapport de synthèse de l'implémentation du modèle de PARK	185
	Abréviations	185
	Bibliographie	187

INTRODUCTION GÉNÉRALE

L'algorithme CORDIC (Coordinate Rotation Digital Computer) est présenté pour la première fois en septembre 1954 par J.Volder, pour une application temps réel de calculs de navigation. Son idée principale consiste à dire que pour effectuer une rotation d'angle θ , il faut décomposer cet angle en une somme d'angles élémentaires θ_i . Si ces derniers sont correctement choisis, cette méthode ne nécessite alors que de simples additions et décalages pour calculer les fonctions trigonométriques élémentaires.

En 1971, J. S. Walther décrit un algorithme CORDIC généralisé légèrement modifié qui permet de calculer toutes les fonctions mathématiques classiques. En effet en plus des fonctions circulaires déjà définies dans l'algorithme de Volder, s'y rajoutent les fonctions linéaires et hyperboliques.

Comparé aux autres méthodes d'évaluation des fonctions trigonométriques que sont les tables et les approximations polynomiales, l'algorithme CORDIC prend un large avantage du fait qu'il n'utilise pas de multiplieur et exige peu de ressources. De plus, il permet de calculer un grand éventail de fonctions mathématiques.

Malgré son ancienneté, l'algorithme CORDIC a retrouvé une jeunesse avec l'avènement de la technologie des circuits programmables FPGA (Field Programmable Gate Array). En effet, les récents progrès de la microélectronique et des techniques VLSI (Very Large System Integration) ont permis l'émergence de nouvelles solutions de circuits programmables permettant d'intégrer un nombre toujours plus grand de portes logiques et de ressources de calcul sur le même circuit.

D'autre part, la vitesse de fonctionnement de ces circuits ne cesse d'augmenter pour atteindre plusieurs centaines de MHz. Aussi les récents progrès des outils de CAO (Conception Assistée par Ordinateur) ont ajouté de la flexibilité à la conception à l'aide de ces circuits programmables.

De ce fait, les FPGA sont devenus des cibles privilégiées pour la conception d'un grand nombre d'applications.

Les nouvelles techniques de commande des machines électriques asynchrones est un thème de recherche du laboratoire LSIC (Laboratoire des Systèmes Intégrés à base des Capteurs) de l'ENS de Kouba. C'est dans ce cadre que se situe l'objet de notre travail de Magistère.

Le présent projet traite en grande partie de l'implémentation d'un algorithme mathématique nommé CORDIC sur FPGA et d'étendre cette implémentation à un modèle de la commande de la machine asynchrone.

Dans ce travail de recherche, nous étudierons l'algorithme CORDIC et définirons une méthodologie de conception pour l'implémentation numérique sur FPGA, développée autour d'outils Xilinx ISE 7.1i, qui nous permettra d'étudier et valider les étapes d'implémentation conçus progressivement dans le FPGA « XC2V1000-4FG456 ».

Dans le premier chapitre, nous présenterons l'algorithme CORDIC classique et généralisé. Nous définirons les problèmes liés au domaine de calcul, de convergences et de précision et la solution de précalage. Le langage C nous a aidés à traduire le CORDIC pour l'assimiler à travers les exécutions visualisées.

Dans le chapitre suivant, nous introduisons la technologie FPGA et son environnement de conception. Nous situerons la place qu'occupent les FPGA dans les circuits ASIC et nous décrirons d'abord l'architecture des FPGA en général et celle du (XC2V1000-4FG456C) Virtex-II de Xilinx en particulier, puis les outils de conception ainsi que notre carte de développement.

Dans le chapitre trois, nous exposerons les différentes architectures du CORDIC en précisant le choix. Ce chapitre est réservé à la synthèse VHDL de l'algorithme CORDIC en passant par des simulations (ModelSim XE 6.0b et le simulateur de XILINX) pour la validation de la description et en terminant avec les tests fonctionnels et sur la carte de développement « Virtex-II V2MB1000 ».

Le quatrième chapitre traite de l'application de l'algorithme CORDIC, c'est la modélisation de Park, très utilisée dans la commande de la machine asynchrone, nous décrirons les étapes d'implémentation de ce modèle avec des simulations de validations

Enfin, nous tirons nos conclusions sur ce travail et nous évoquons les perspectives de recherches et de développement dans le domaine des architectures reconfigurables.