

Avant-propos

Au cours des quinze dernières années, les méthodes de conception des fonctions numériques ont subi une évolution importante. Dans les années soixante-dix, la majorité des applications de la logique câblée étaient construites autour de circuits intégrés standard, souvent pris dans la famille TTL. Au début des années quatre-vingt apparurent, parallèlement, les premiers circuits programmables par l'utilisateur du côté des circuits simples et les circuits intégrés spécifiques (ASICs) pour les fonctions complexes fabriquées en grande série. La complexité de ces derniers a nécessité la création d'outils logiciels de haut niveau qui sont à la description structurelle (schémas au niveau des portes élémentaires) ce que les langages évolués sont au langage machine dans le domaine de la programmation. A l'heure actuelle, l'écart de complexité entre circuits programmables et ASICs s'est restreint : on trouve une gamme continue de circuits qui vont des héritiers des premiers PALs (programmable array logic), équivalents de quelques centaines de portes, à des FPGAs (Field programmable gate array) ou des LCAs (Logic cell array) de quelques dizaines de milliers de portes équivalentes. Les outils d'aide à la conception se sont unifiés ; un même langage, VHDL par exemple, peut être employé quels que soient les circuits utilisés, des PALs aux ASICs.

Circuits numériques et synthèse logique, un outil : VHDL est l'un des résultats de la réflexion faite à l'IUT de CACHAN sur l'évolution du contenu de l'enseignement de l'électronique numérique, pour proposer aux étudiants une formation en accord avec les méthodes de conceptions actuelles.

La gageure était d'éviter le piège du « cours VHDL », impensable à ce niveau de formation, compte tenu du volume horaire imparti (une centaine d'heures, travaux d'applications compris). Nous avons décidé de nous restreindre à un sous-ensemble cohérent de VHDL, qui soit strictement synthétisable. Ce choix étant fait, nous avons mené en synergie l'apprentissage des bases de la logique et celui des concepts des langages de description de haut niveau. Chaque élément nouveau, vu sous son aspect circuit, est immédiatement transcrit en VHDL, ce qui précise sa fonction, et permet à l'étudiant de se familiariser progressivement avec la syntaxe du langage.

L'objectif étant la maîtrise du processus de synthèse, jusqu'à la réalisation, nous avons utilisé un compilateur VHDL qui permet très simplement de générer les fichiers JEDEC utilisés pour la programmation des circuits. Expérimentée pour la première fois au cours de l'année scolaire 1994-95, cette approche un peu différente de l'électronique numérique semble avoir rencontré un accueil favorable de la part des étudiants. La création de machines d'états adaptées à un problème donné, ne semble plus constituer un obstacle pour eux.

Que soient remerciés ici les collègues de l'IUT, pour les nombreuses discussions que nous avons eues sur le sujet, et, surtout, les étudiants qui se sont lancés, parfois avec fougue, dans l'exploration de cette terre inconnue.

Le livre est subdivisé en cinq chapitres et une annexe :

- I. Une introduction générale au monde du numérique. On y définit les notions de base telles que la représentation des nombres, les conventions logiques etc.
- II. Un panorama des caractéristiques électriques, statiques et dynamiques, des circuits numériques et leur emploi pour déterminer les limites de fonctionnement d'une application. Ce chapitre n'est pas forcément traité en cours de façon chronologique linéaire ; certaines parties ne prennent sens qu'au vu des applications de la logique séquentielle (calcul d'une fréquence maximum de fonctionnement, par exemple).
- III. La définition des opérateurs élémentaires, combinatoires et séquentiels. Le principe adopté est de ne pas attendre d'avoir vu toute la logique combinatoire pour aborder la logique séquentielle. Les bascules (synchrones principalement) doivent devenir des objets aussi familiers qu'une porte ET. C'est à l'occasion de la découverte des opérateurs élémentaires que les premiers rudiments du langage VHDL apparaissent. Les difficultés de ce langage (dont la principale concerne le passage combinatoire → séquentiel asynchrone → séquentiel synchrone) sont abordées sur des objets très simples, elles sont donc faciles à expliciter. Une fois ces difficultés surmontées, la puissance du langage récompense très largement l'utilisateur de son effort intellectuel.
- IV. Les principales catégories de circuits, fonctions standard et circuits programmables. L'étude de cette partie, volontairement restreinte, ne peut se faire qu'en illustrant le cours de nombreuses analyses de notices techniques.
- V. Les méthodes de synthèse qui sont l'aboutissement de cet enseignement. On y fait connaissance avec les machines d'états, les architectures de Moore et de Mealy, leur transcription en VHDL. Les méthodes de simplification sont vues très rapidement, essentiellement pour comprendre ce que fait un optimiseur, de façon à apprendre à le piloter.
- VI. Une annexe qui résume et explicite les principales constructions qu'autorise le langage VHDL. Sont exclues, volontairement, toutes les constructions non synthétisables, qui servent exclusivement à la modélisation et à la simulation.

De nombreux exemples illustrent les principes abordés, et des exercices permettent au lecteur d'asseoir sa compréhension.

La connaissance d'un langage de programmation procédural, comme C ou Pascal, n'est pas indispensable, mais elle facilite la compréhension de certains passages. Quelques notions sur les composants, transistors et capacités, et sur les lois élémentaires des circuits électriques sont souhaitables. On notera cependant que les aspects structurels internes des circuits ne sont pas abordés, sauf quand ils sont incontournables (sorties non standard).