

Les outils de synthèse de haut niveau

Partie intégrante des méthodologies de conception au niveau système, les outils de synthèse de haut niveau visent à transformer un algorithme comportemental en une description au niveau RTL d'une partie d'un circuit. Et, depuis quelques années, l'offre en outils commerciaux s'est étoffée.

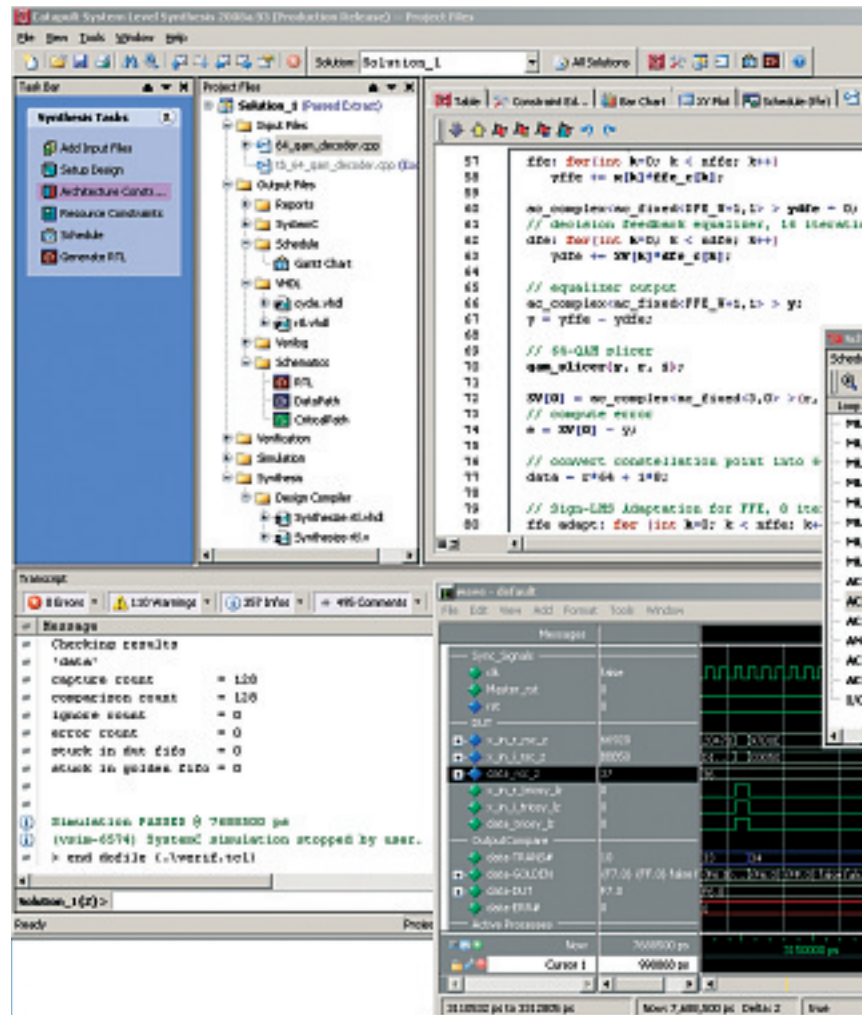
La volonté de transformer la description d'un circuit à un haut niveau d'abstraction directement au niveau du silicium n'est pas une idée neuve. Dans les années quatre-vingt, de nombreuses équipes de recherche dans le monde se sont attelées à ce problème. On parlait alors de Silicon Compilation, avec l'idée qu'il était possible de construire, comme dans le monde du logiciel, un compilateur capable de transformer un code logiciel en un circuit (tout du moins sa représentation au niveau portes). Mais, dans la pratique, ce fut un échec. Puis, au milieu des années quatre-vingt-dix, est apparu le produit commercial Behavioral Compiler de Synopsys qui utilisait comme langage d'entrée Verilog ou VHDL associé à des contraintes de timing, de surface... gérées par l'utilisateur. Mais, là aussi, ce fut un échec. L'outil ne s'est jamais réellement imposé sur le marché et dix ans plus tard, en 2004, Synopsys a décidé de jeter l'éponge en arrêtant de le commercialiser.

Un regain d'intérêt pour ce type de technologie est venu au début des années deux mille de la société anglaise Celoxica, qui a popularisé le concept de compilation d'algorithmes complexes sur FPGA, en utilisant comme point d'entrée le langage Handel-

C, un dérivé du C doté d'extensions non standard en particulier pour le contrôle du parallélisme au niveau matériel. Issue des travaux du laboratoire de l'université d'Oxford, cette approche a eu un succès certain sur le marché, notamment dans le domaine du prototypage.

Mais le véritable intérêt vis-à-vis de cette technologie, dans le monde de la CAO, est né, d'une part, du succès rencontré par la société américaine Forte Design Systems avec son outil Cynthysizer, très implanté sur le marché japonais et, d'autre part, de l'arrivée en 2005 de l'outil de synthèse de haut niveau CatapultC de Mentor Graphics. Avec ces sociétés, auxquelles on peut adjoindre l'américain Synfora dont la technologie est issue des laboratoires de Hewlett-Packard, on est passé à une dimension industrielle et opérationnelle de la mise en œuvre de la synthèse comportementale au sein d'équipes de conception d'Asic, de SoC ou de FPGA (en Asie et en Europe essentiellement).

Pourquoi cette évolution ? D'abord parce que ce type d'outil permet, théoriquement, de réduire le temps de conception d'un bloc fonctionnel avec une grande fiabilité du code généré par rapport à un travail de codage en « manuel » du RTL, dont la qualité dépend largement de



L'outil CatapultC de Mentor Graphics procure une voie de transformation du C/C++ vers une implantation matérielle au niveau RTL, avec en parallèle la génération automatique de fichiers de test ad hoc (les testbenchs).

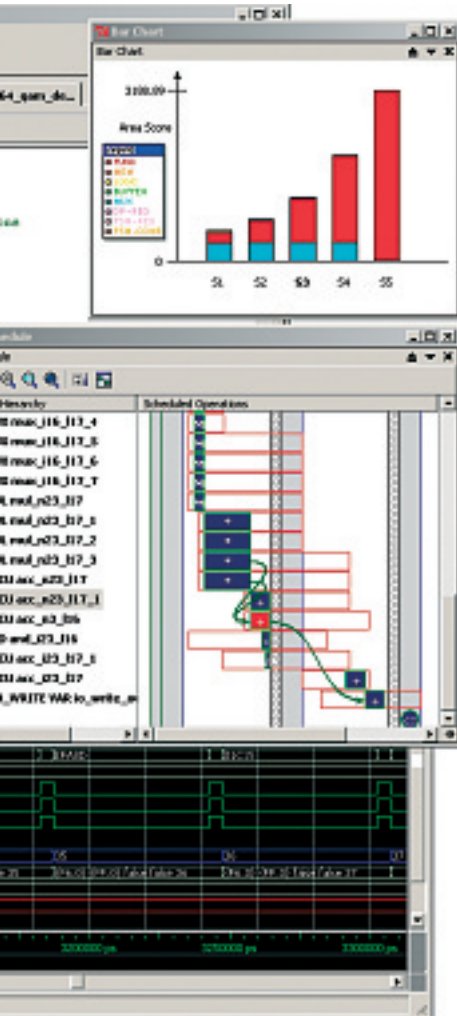
l'expérience et du talent de l'ingénieur concerné. Ensuite, la montée en puissance de ces technologies s'inscrit dans le mouvement plus large du déploiement des méthodes de développement au niveau système, qui visent à remonter le niveau d'abstraction de la conception d'un circuit. Et au fond, il faut considérer les outils de synthèse de haut niveau comme une des

composantes de l'ESL (Electronic system level).

Cependant, cet optimisme quant au développement de ces outils doit être tempéré à l'aune des chiffres du marché. Car, malgré le fort intérêt suscité par cette approche, le domaine des outils de l'ESL ne dépasse pas les 200 M\$, selon les chiffres de l'Edac présentés lors de la dernière Dac par Wally Rhines, le

CEO de Mentor Graphics. Un chiffre à comparer aux 4 à 5 milliards de dollars correspondant à l'ensemble du domaine de la CAO électronique.

On le voit, ce marché des outils de synthèse de haut niveau est encore loin d'être florissant. D'autant plus que selon les chiffres de l'analyste américain Gary Smith, Mentor Graphics avec son outil CatapultC détiendrait aux alentours de 67% de parts de marché, devant Forte Design



Systems et Synfora. Certes, cette très forte dominance de Mentor montre que la société a su imposer sa technologie auprès des développeurs. Mais, en même temps, cet état de fait n'est pas forcément un bon indice de la santé globale du secteur des outils de synthèse de haut niveau, qui devraient faire l'objet d'une concurrence plus forte entre acteurs. Mais celle-ci viendra peut-être de Cadence, qui à son tour a décidé de rentrer sur ce marché avec son outil C-to-Silicon, annoncé en 2008.

Deux grands groupes d'acteurs

Il est possible de séparer les fournisseurs de solutions de synthèse de haut niveau en deux grandes catégories. Tout d'abord, ceux qui délivrent des solutions destinées en priorité à accélérer au niveau matériel des algorithmes de calcul sur une plate-forme FPGA. Les fabricants de circuits, comme Altera avec C2H, proposent ce type de logiciel, au même titre que des sociétés indépendantes des fondateurs comme Impulse Accelerated Technologies (dont la solution est utilisée par Xilinx), Altium ou Mentor Graphics avec son environnement DK Design Suite (technologie issue de Celoxica, acquise via le rachat de certains actifs d'Agility qui, sous le nom, de Catalytic, avait racheté Celoxica). Cette accélération matérielle, de tout ou partie d'un code écrit en C, est généralement associée à la présence d'un processeur embarqué sur le FPGA. Une partie du code est ainsi traitée par le cœur de processeur alors que l'autre partie est codée en «dur», avec une parallélisation intense des opérations. Et les résultats de cette accélération sont généralement transmis au processeur via un bus. Ces solutions peuvent aussi toutefois créer un bloc matériel capable de gérer cette accélération de manière autonome.

L'autre catégorie concerne les sociétés qui commercialisent des outils beaucoup plus ambitieux, aptes à s'insérer dans des flots de conception classiques d'Asic, de SoC ou de systèmes sur FPGA. Ces solutions ont pour objectif de transformer des fonctions de traitement de données complexes (filtres) et/ou de transmission de données (modem), décrites dans un langage de haut niveau (C, C++, SystemC), en une architecture matérielle qui s'insère dans un circuit plus large. Il peut s'agir de «simples» accélérateurs matériels jusqu'à des blocs IP complexes, réalisant par exemple des fonctions de codage/décodage de flux vidéo, en passant par des blocs de contrôle mémoire ou des contrôleurs d'entrées-sorties. Ici, on trouve notamment les solutions proposées par Mentor Graphics et son

outil CatapultC, Forte Design et sa solution Cynthesizer, Synfora et sa technologie Pico Extreme, et, dernier arrivé sur le marché, Cadence et son logiciel C-to-Silicon.

Preuve s'il en est que ces deux types d'outils sont très différents, les fabricants de FPGA, Altera et Xilinx, ont signé des accords de partenariat et de coopération avec Mentor et, plus récemment avec Cadence pour adapter cette approche à leur architecture dans le cas de conceptions complexes. Alors que, dans le même temps, ils proposent des solutions de synthèse algorithmique visant à l'accélération de calculs.

C ou SystemC?

Il n'existe pas de différences fondamentales entre les outils de la première catégorie. Mis à part DK Design Suite qui utilise en entrée un langage propriétaire, Handel-C, tous les autres travaillent à partir d'algorithmes écrits en C. Et en ce qui concerne la problématique de ces technologies, elle est identique: transformer un algorithme qui décrit une suite d'opérations sous forme séquentielle en une architecture matérielle qui parallélise ces opérations (d'où forcément un gain très important en termes de vitesse de calcul).

Les outils du second groupe présentent, eux, des différences technologiques sont plus importantes. Le débat fait notamment rage entre ceux qui prônent le langage C, sans aucune référence temporelle, comme point d'entrée de la synthèse et ceux qui militent pour l'utilisation de SystemC. Dans le premier camp, on trouve Mentor Graphics et, dans une moindre mesure, Synfora (qui intègre des constructions spécifiques au C de départ), et dans le second Forte Design Systems et Cadence. Les premiers estiment que le langage C est le mieux à même de gérer un haut niveau d'abstraction, sans aucun présupposé sur l'architecture matérielle à venir. Une caractéristique qui permet notamment des possibilités étendues d'exploration d'architectures et une meilleure réutilisabilité du code. De plus, toujours selon les promoteurs du C, SystemC est mieux adapté à la description de l'architecture d'un

circuit qu'à l'écriture d'un algorithme. C'est d'ailleurs ce langage, le SystemC, qui est massivement utilisé pour la programmation des plates-formes virtuelles d'un SoC au niveau transaction (TLM, Transaction level modeling).

Mais pour les défenseurs de SystemC, ces arguments sont précieux. Ils font remarquer, d'une part, que SystemC n'est pas un «langage exotique venu d'ailleurs», mais une extension (un superset) du C++ qui autorise des descriptions d'algorithmes à un haut niveau d'abstraction sans référence temporelle, comme avec le C. En revanche, contrairement au C, SystemC offre la possibilité de rajouter des informations sur le parallélisme, la notion de calculs concourants, de hiérarchie... Autant de caractéristiques qui permettent, selon les défenseurs de SystemC, d'obtenir une bien meilleure flexibilité dans la synthèse.

Au fond, la question qui est posée ici est: à quel niveau doit-on rajouter des informations sur la structure du matériel? Dans le cas du C, ces données sont rajoutées sous la forme de contraintes externes, alors qu'avec SystemC, certaines de ces données sont formalisées dans le code lui-même, facilitant le travail d'optimisation de la synthèse, selon ses promoteurs. Autre point d'achoppement, les tenants du C estiment qu'il est plus aisé de prouver la correspondance fonctionnelle entre un code écrit en C et son équivalent synthétisé en VHDL ou Verilog, qu'entre un fichier en SystemC et sa version compilée au niveau RTL. D'un autre côté, il est plus facile d'intégrer du code en SystemC dans des plates-formes de validation au niveau transaction que du code C pur (pour lequel il faut faire l'effort de générer des wrappers).

Synthèse de flot de données ou de blocs de contrôle?

Autre sujet de discussion intense entre fournisseurs d'outils: la possibilité ou non de gérer les logiques de contrôle dans un outil de synthèse de haut niveau au même titre que les flots de données (datapath). Une chose est certaine, d'un point de vue

théorique, plus il existe de branchements conditionnels dans un code, plus il est difficile de le synthétiser, contrairement par exemple à un algorithme de traitement de signal. Car, dans ce dernier cas, la synthèse consiste à réaliser une parallélisation massive des opérations. Mais dans la pratique, bien souvent, les fonctions à synthétiser sont plutôt un mélange de flots de données et de contrôle, dans des proportions variables. Cadence estime que son outil C-to-Silicon est capable de traiter ce type de fonctions, avec un niveau élevé de fiabilité, grâce à la maîtrise des timings, son logiciel intégrant un moteur de synthèse logique classique en parallèle à la synthèse de haut niveau. Ce qui permet, selon cet éditeur, de maîtriser les délais entre les blocs de contrôle et les blocs de calculs, parfois « éloignés » les uns des autres sur un circuit. Ce qui n'est pas possible, toujours selon Cadence, avec des outils plus orientés flots de données qui abordent ce problème des timings via des bibliothèques de blocs précaractérisés en termes de délais.

De toute évidence cependant, les sociétés qui se sont lancées sur les outils de synthèse de haut niveau centrés sur les blocs de contrôle ne sont pas des plus florissantes. Il en est ainsi d'Esterel-EDA qui est actuellement en cours de cession par sa maison-mère Esterel Technologies, ou de Bluespec qui ne perçoit pas vraiment sur le marché avec sa solution à base de System Verilog associée à un ensemble de règles. On peut aussi se demander si cette voie de synthèse de haut niveau pour la logique de contrôle est pertinente. « Est-ce qu'il n'est pas finalement plus aisé et plus efficace de programmer une logique de contrôle sous forme de machine d'états sur un microcontrôleur embarqué dans un circuit ? », s'interroge Frédéric Pétrot, professeur au laboratoire Tima et animateur de la solution académique de synthèse comportementale UGH (User guided high level synthesis).

Pour l'avenir, la tendance est à l'élargissement des capacités des outils de synthèse en termes de taille de bloc à traiter, qui atteint d'ores et déjà pour les outils plus

évolués entre 7 et 10 millions de portes. Chez Synfora, par exemple, le système de compilation de haut niveau proposé par la société, Pico Extreme, est désormais capable de prendre en compte un assemblage hiérarchisé de plusieurs blocs pour former un moteur de calcul complexe. Avec cet outil, le développeur utilise des primitives compilées à partir d'un code en C (une FFT par exemple), les assemble et construit une composition de blocs

imbriqués, hiérarchisés qui forme un sous-système complet de calcul au sein d'un SoC. Certes, il est clair qu'il n'existera pas dans l'avenir une solution de synthèse globale d'un circuit. Car les notions de réutilisation de blocs IP et d'intégration de ces blocs, reliés entre eux par un bus ou un NOC (Network on chip), sont désormais incontournables et incompatibles avec cette notion de synthèse globale de circuit. On estime, par exemple, qu'aujourd'hui au moins la moitié des blocs IP présents sur un SoC ne rentrent pas dans le champ des outils de synthèse de haut niveau, mais dans celui du marché de la propriété intellectuelle.

En revanche, pour certains blocs IP, là où il existe encore de la conception de matériel, véritable facteur de différenciation entre concurrents, les outils de synthèse de haut niveau ont un rôle très important à jouer à l'avenir.

FRANÇOIS GAUTHIER

Critères de choix

Nature de la cible et type de synthèse structurent le choix des outils

Destinés avant tout aux développeurs de matériels, les outils de synthèse de haut niveau offrent, pour un objectif final identique, des manières très différentes de travailler. Les aspects vérification, gestion des interfaces et facilité d'utilisation doivent être examinés avec soin.

La synthèse algorithmique, comportementale, ou synthèse de haut niveau, vise avant tout à créer une implantation performante au niveau matériel d'une fonction décrite sous forme logicielle. Selon la cible visée, FPGA ou SoC, et suivant le type de synthèse souhaitée, accélération matérielle d'une fonction de calcul, ou création d'un bloc IP plus ou moins complexe, le choix de l'outil va être fondamentalement différent.

Si l'on souhaite accélérer des algorithmes de calcul écrits en C sur du FPGA, on choisira des logiciels simples d'emploi, capa-

bles de s'adapter aussi bien aux ingénieurs logiciel qu'aux ingénieurs matériel. La solution d'Impulse Accelerated Technologies est, par exemple, à même d'accélérer des logiciels de calcul sur des cartes coprocesseur dotées d'interfaces standard. L'ensemble des opérations d'optimisation vis-à-vis du matériel est alors réalisé de manière automatique sans recours à une quelconque expertise.

Cependant, mis à part ce cas de figure, il faut reconnaître que la très grande majorité des utilisateurs de ces technologies demeurent des ingénieurs matériel qui

vont pouvoir, à travers ces outils de synthèse, améliorer la productivité de leur activité ainsi que la qualité du code généré.

Analyser les restrictions sur le code d'entrée

Une des premières choses à regarder sur ces solutions est la nature du code d'entrée utilisé. Pour ceux qui utilisent le langage C, il est impératif de demander quelles restrictions les éditeurs imposent au départ. Car si tous affichent qu'ils travaillent à partir d'un code C standard, leur logiciel de synthèse ne prend pas forcément en compte toutes les structures du C (pointeurs, virgule flottante, largeur des données...). Pour SystemC, la démarche doit être équivalente avec un degré de prudence supplémentaire, car il existe différentes variantes de ce langage en circulation.

Ensuite, un des aspects fondamentaux à évaluer est la vérification. En effet, rien ne sert de

Suite p.58

L'offre en synthèse comportementale

Editeur (fournisseur)	Nom du produit	Langages supportés en entrée	Fichiers de sortie	Outils de vérification associés	Cibles supportées	Fonctions spécifiques	Observations
Altera	C2H Compiler	Ansi C	Structure hardware sur FPGA	Non	FPGA d'Altera avec un CPU Nios II	Connexion de l'accélérateur hardware au processeur Nios II, automatisée	Outil d'accélération matériel, piloté via l'environnement de développement C du processeur Nios II.
Altium	Altium Designer C to hardware Compilation	C	RTL (VHDL ou Verilog)	Non	FPGA (Altera et Xilinx)	Génération de circuits « combinatoires » ou « multicycles »	Fonction intégrée dans la suite de conception sur FPGA Altium designer.
AutoESL	Autopilot	C, C++, SystemC	RTL	Non	Asic, FPGA	Plate-forme au niveau système centrée sur les communications	Commercialisation sous licence du logiciel de synthèse xPilot (UCLA VLSICAD Laboratory).
Bluespec	Bluespec Compiler	System Verilog avec règles et méthodes d'interfaces	RTL (Verilog)	Oui (Simulateur au niveau système)	Asic	Méthodes de description des interfaces et de règles de comportement	Compilation correcte par construction pour logiques de contrôle et flot de données.
Cadence	C-to-Silicon Compiler	C, C++, SystemC	RTL (Verilog)	Oui (vérification sur un modèle matériel généré automatiquement)	Asic, FPGA	Gestionnaire de contraintes de design, synthèse incrémentale	Base de données unique qui stocke les modifications du design; séparation entre description fonctionnelle et contraintes.
Cebatech	C2R Compiler	Ansi C	RTL (Verilog)	Non	Asic, FPGA	Macro architecture insérée dans le code source C via des directives spécifiques	Fonction de définition des interfaces de communication génériques entre modules et de blocs IP.
Esterel-EDA (1)	Esterel Studio	Langage Esterel	SystemC, RTL VHDL ou Verilog)	Oui (Simulation à base de vérification formelle)	Asic, SoC	Spécification exécutable à base de méthodes formelles	Programmation graphique de machines d'états et langage textuel pour les algorithmes.
Forte Design Systems	Cynthesizer	SystemC	RTL (VHDL ou Verilog)	Non	Asic, SoC, FPGA	Génération automatique d'interfaces (mémoire, buffer...)	Optimisation des chemins de données pour obtenir des timings précis pour les additionneurs, multiplieurs...
Impulse Accelerated Technologies (Csylia)	Impulse CoDeveloper	C standard	Structure Hardware sur FPGA	Non	FPGA (Altera et Xilinx)	Vérification et débogage avec des outils de développement du langage C	Fonction de partitionnement et de « profiling » entre le CPU embarqué et l'accélérateur matériel sur le FPGA.
Mentor	CatapultC Synthesis	C/C++ sans référence temporelle	SystemC, RTL (VHDL et Verilog)	Oui (flot de vérification avec création automatique de testbench)	Asic, FPGA	Optimisation des interfaces via une synthèse spécifique	Génération automatique de transacteurs et wrappers en SystemC; réutilisation de testbench en C.
	DK Design Suite (2)	Handel C (C modifié)	Structure Hardware sur FPGA	Non	FPGA (Altera et Xilinx)	Outil de gestion de partitionnement du logiciel entre CPU et hardware	Intégration avec des plates-formes de développement matériel et des bibliothèques de design pour le prototypage.
Nec	CyberWorkBench	C standard	RTL (VHDL et Verilog)	Oui (Simulation, equivalent checking)	Asic, FPGA	Données sur la physique du circuit fournies à l'outil de synthèse	Simulation au niveau cycle et cosimulation matériel/logiciel intégré au flot de synthèse; génération de testbench.
Synfora	Pico Extreme	C standard sans référence temporelle + blocs IP	RTL (VHDL ou Verilog)	Oui (création de testbench)	SoC, FPGA	Procédures avancées de gestion du clock gating, interfaces OCP IP	Méthode de composition récursive (bloc en C pris comme une primitive de calcul); compilation de hiérarchies de blocs.
Synopsys	Synplify DSP	Blocs DSP en Simulink et Matlab	RTL	Non	Asic, FPGA	Blocs fonctionnels (filtres, FFT...) prédéfinis écrits en Simulink	Utilisation du langage M pour décrire des machines d'états et des logiques de contrôle; extraction de blocs mémoire.
The MathWorks	Simulink HDL Coder	Langages graphiques Simulink et StateFlow	RTL (VHDL et Verilog)	Oui (génération de testbench)	Asic, FPGA	Modèle Simulink « multirate » synthétisé en un seul domaine d'horloge	Liens avec ModelSim pour la simulation de blocs IP existants (mémoires...) avec les modèles Simulink.

(1) Esterel EDA Technologies, filiale d'Esterel Technologies, est actuellement en cours de cession par sa maison-mère. (2) DK Design Suite est un produit issu de la société Agility, dont certains actifs (DK Design Suite en fait partie) ont été récemment rachetés par Mentor Graphics. Ce produit a été à l'origine développé par la société Celoxica (voir Electronique n°199, p.20).

► transformer automatiquement du code C en un code au niveau RTL, en ayant gagné du temps par rapport à un codage «manuel», pour en perdre par la suite en passant beaucoup de temps à écrire les bancs de tests nécessaires à la vérification du RTL d'un point de vue fonctionnel. Sur ce plan, les sociétés fournissent des outils de vérification plus ou moins sophistiqués et plus ou moins automatisés. Cela peut aller de simples «checkers» jusqu'à des testbenchs complets, à différents niveaux d'abstraction.

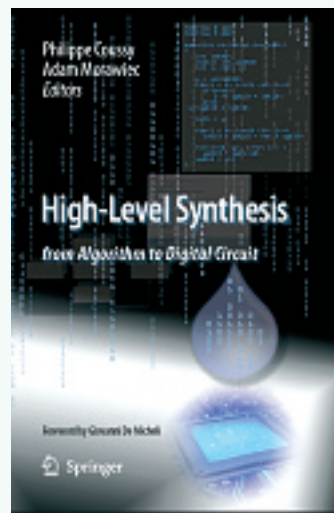
Attention aux interfaces

Vis-à-vis de ces outils de synthèse, il faut aussi s'interroger sur leur capacité à explorer de manière plus ou moins étendue plusieurs architectures matérielles possibles. Même si, lors de la création d'un bloc IP, les concepteurs ont souvent une idée directrice, la finesse d'exploration de microarchitectures est un atout pour optimiser ses choix en termes de performance/surface/

Pour tout savoir sur la synthèse de haut niveau

→ Ce livre, consacré exclusivement à la synthèse de haut niveau, regroupe des contributions issues d'approches académiques, d'éditeurs d'outils et d'expériences d'industriels. Préfacé par Giovanni De Micheli, on y trouve des réflexions sur les besoins des utilisateurs (Pascal Urard de STMicroelectronics), une rétrospective de l'évolution de ce type de technologie ainsi que des descriptions fouillées des différentes approches mises en œuvre par les éditeurs de logiciels (Mentor, Nec, Synfora...).

High-Level Synthesis from algorithm to digital circuit par Philippe Coussy, Adam Morawiec (Editeurs), 2008, XVI, 300 pages, ISBN: 978-1-4020-8587-1
www.springer.com/engineering/circuits-+&+systems/book/978-1-4020-8587-1



consommation. De plus, dans le cas de la réutilisation d'un fichier écrit en C et/ou en SystemC pour le porter sur une nouvelle technologie, cette possibilité d'exploration d'architecture, plus ou

moins évoluée selon les fournisseurs, devient essentielle.

Un autre point à considérer avec attention est celui des interfaces. En effet, créer rapidement un bloc matériel de calcul perfor-

mant, mais isolé, n'a pas beaucoup de sens lorsqu'il faut l'insérer dans un ensemble plus vaste. Les solutions disponibles sur le marché apportent donc une aide à la génération des interfaces (connexion à des bus, aux blocs mémoire...), mais avec des approches différentes. La manière dont ces interfaces sont générées ou fournies en tant que blocs prédéfinis varie d'un éditeur à l'autre. Or, c'est un point crucial à examiner, car la performance d'un code RTL créé automatiquement peut dépendre très étroitement de la qualité des interfaces associées.

Enfin, il ne faut pas négliger l'aspect facilité d'utilisation de ces produits, bien que cette caractéristique soit difficilement quantifiable, car il ne s'agit en aucun cas d'outils de type presse-bouton, malgré quelques «excès marketing» rencontrés sur les brochures des éditeurs; leur emploi nécessite une solide connaissance en développement matériel. FRANÇOIS GAUTHIER

**ELECTRONIQUE
INTERNATIONAL**

Abonnez-vous à **Electronique international** !

A renvoyer à Electronique International – Service abonnements – B 984 – 60 732 Ste Geneviève Cedex



OUI, je choisis de m'abonner à Electronique international pour :



1 an (22 numéros): 150 € au lieu de 176 €*



6 mois (11 numéros): 75 € au lieu de 88 €*

Mes coordonnées :

Mme Mlle Mr Nom : _____ Prénom : _____

Société : _____ Fonction : _____

Adresse : _____ Code postal : | | | | | Ville : _____

Email : _____ Tel : | | | | |

Je souhaite être informé(e) des offres commerciales des partenaires du groupe Tests.

Je règle par : Chèque bancaire ou postal à l'ordre de **Electronique international**
 Carte bancaire (CB, EC, MC, VISA)

N° : | | | | | | | | | | | | | | | | | | | | | |

N° de contrôle : | | | | | Expire fin : | | | | |

3 derniers chiffres au dos de votre carte bancaire

Je souhaite recevoir une facture acquittée.

Si l'adresse de facturation est différente de celle de la livraison ci-contre, merci de nous le préciser.

Date et signature obligatoires

PPEIP8



Il existe un droit de rétractation. Réception du premier numéro sous 4 semaines à réception du règlement. En application de l'article 27 de la loi du 6.1.78, les informations demandées ici sont nécessaires à l'exécution de votre commande et aux services qui y sont associés. Les catégories de destinataires sont uniquement celles nécessaires à l'exécution de cet abonnement, aux services associés et partenaires du Groupe sauf opposition de votre part.

Vous pourrez accéder aux informations vous concernant et procéder éventuellement aux rectifications nécessaires auprès du service Abonnements.

*Prix de vente au numéro. Offre valable jusqu'au 30/06/09 pour les nouveaux abonnés en France métropolitaine uniquement.