Zeitschrift: Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

Herausgeber: Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

Band: 75 (1984)

Heft: 1

Artikel: Méthodes de conception de circuits intégrés complexes

Autor: Piguet, C.

DOI: https://doi.org/10.5169/seals-904328

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Mehr erfahren

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. En savoir plus

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. Find out more

Download PDF: 10.12.2025

ETH-Bibliothek Zürich, E-Periodica, https://www.e-periodica.ch

Méthodes de conception de circuits intégrés complexes

C. Piguet

Placer environ 100 000 transistors sur une seule puce de silicium est aujourd'hui possible dans des temps de développement raisonnables. Néanmoins, pour y parvenir avec quelques chances de succès, il s'agit d'avoir les idées très claires sur les objectifs à atteindre et sur la méthode de conception. Cet article se propose de caractériser les méthodes de conception. Puis, ayant explicité les objectifs souhaités, il décrit une méthode de conception basée sur trois caractéristiques fondamentales:

- *concevoir des architectures régulières, c'est-à-dire chercher à dessiner le plus petit nombre de transistors;*
- *concevoir un plan directeur de la puce pour tenir compte des contraintes topologiques avant de concevoir en détail les différents blocs logiques du circuit;*
- *utiliser une technique de layout symbolique dite orientée métal de manière à augmenter la productivité du concepteur.*

Es ist heute innert nützlicher Frist möglich, etwa 100 000 Transistoren auf einem einzigen Silizium-Chip zu plazieren. Dazu muss man jedoch sehr klare Vorstellungen über die zu erreichenden Ziele sowie die Entwurfsmethode haben. Im vorliegenden Aufsatz werden die Entwurfsmethoden zuerst dargestellt. Anschliessend wird aufgrund der verfolgten Ziele eine Methode vorgestellt, die auf den folgenden drei Grundsätzen basiert:

- *regelmässige Architekturen entwerfen, d.h., eine möglichst kleine Zahl Transistoren zeichnen;*
- *unter Berücksichtigung der Topologie des Chips einen Leitplan entwerfen, bevor die einzelnen logischen Blöcke der Schaltung im Detail konzipiert werden;*
- *eine «metallgerichtete» symbolische Layouttechnik verwenden, um die Produktivität des Entwerfers zu erhöhen.*

Ce travail est partiellement financé par la Commission pour l'Encouragement des Recherches Scientifiques en Suisse, Crédit 1085.

Adresse de l'auteur

Dr C. *Piguet*, Centre Electronique Horloger S.A., Maladière 71, 2000 Neuchâtel 7.

1. Introduction

Les progrès technologiques permettent aujourd'hui de placer 100 000 transistors sur une seule puce de silicium. En quelques années, les concepteurs de circuits intégrés ont dû faire face à cet accroissement spectaculaire de la complexité. Il est dès lors évident que les problèmes méthodologiques liés à cette conception sont fondamentaux.

En effet, il ne suffit pas de disposer d'outils d'aide à la conception sur ordinateur (CAO), si performants soient-ils, pour résoudre le problème. Il s'agit de dégager une méthode de conception en fonction d'objectifs clairs, de mettre au point manuellement cette méthode pour juger si ses résultats sont satisfaisants, puis d'automatiser certaines étapes de la méthode adoptée.

Il est alors certain que les programmes CAO, générés à partir de la méthode adoptée, correspondront à un besoin des concepteurs de circuits intégrés complexes.

2. Circuits intégrés VLSI existants

Le sigle VLSI (Very Large Scale Integration en anglais) est pris dans le sens de complexe; cette complexité est mesurée par le nombre de transistors sur la puce.

L'analyse présentée au tableau I indique que la première loi de *Moore* [1], soit le doublement de la complexité tous les deux ans, est encore vérifiée. Par contre, certains circuits s'écartent de la deuxième loi de Moore, soit le doublement du temps de conception par circuit toutes les 2,66 années.

Le temps de dessin des plans de masques (ou layout) est en général compris entre 25 et 50% du temps de développement total. Ce temps de dessin représente donc un effort considérable. L'analyse doit être faite en introduisant les notions suivantes:

- La régularité $R = \dfrac{\text{nombre total de transistors sans les mémoires}}{\text{nombre de transistors dessinés}}$
- La régularité apparente R_{app} définie en incluant les mémoires
- La productivité de dessin P_{des}, mesurée en nombre de MOS qu'un opérateur dessine au niveau graphique
- La productivité réelle $P = P_{des} \cdot R$, tenant compte de la régularité
- La productivité apparente $P_{app} = P_{des} \cdot R_{app}$, en incluant les mémoires.

Le tableau II indique les régularités obtenues pour les différents circuits analysés [5]. La figure 1 représente les différentes productivités mesurées pour ces circuits. On constate que la productivité P_{des} est très faible: en des-

Tableau I

Circuit		Nombre de transistors	Surface mm²	Temps de développement en homme-année	
Zilog Z8000	[2]	17 500	40	6,5	
Motorola 68000	[3]	68 000	44	14	[5]
Intel 432	[4]	220 000	3 puces	100	
432 01		110 000	65	21,5	[5]
432 02		49 000	74	22,5	
432 03		60 000	75	15	
RISC I	[5]	44 000	78	3,5	
Hewlett Packard CPU 32 bits	[6]	450 000	40	< 10 ?	

Tableau II

Circuit	Nombre de transistors dessinés	Nombre de transistors sans mémoire	Nombre de transistors avec mémoire	Régularité R	Régularité apparente R_{app}	Temps de dessin du layout en homme-année
Zilog Z8000	3 500	17 500	17 500	5	5	5,8
Motorola 68000	3 000	37 000	68 000	12,3	22,7	5,8
Intel 43201	5 600	60 000	110 000	10,7	19,6	8
Intel 43202	9 500	37 000	49 000	3,9	5,2	9,5
Intel 43203	5 700	49 000	60 000	8,6	10,5	4,5
RISC I	2 000	44 000	44 000	22	22	1
Hewlett Packard CPU 32 bits	6 500	100 000	450 000	15,4	69,2	2,3

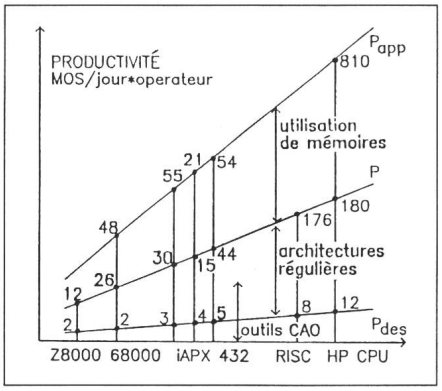

Fig. 1 Evolution de la productivité

sinant le layout géométriquement, on obtient en général de 4 à 6 MOS/jour. Des techniques de layout utilisant des outils CAO plus sophistiqués ont pour but d'augmenter cette productivité (RISC, CPU de HP). Mais il est évident que le recours à des architectures régulières et à l'utilisation de mémoires très régulières est indispensable pour atteindre les productivités nécessaires à la conception de circuits VLSI.

3. Caractérisation de méthodes de conception

Tout circuit intégré admet trois modes de description, à savoir [7] la description de comportement, la description de la structure et la description physique. Pour chacun de ces modes de description, il est nécessaire d'appliquer les principes:

- de hiérarchie, de manière à ce que la conception de chacun des blocs résultant de la décomposition hiérarchique soit aisée,
- d'abstraction, de manière à cacher les détails superflus au niveau d'abstraction où l'on travaille.

Le processus de conception est dit descendant; il consiste à passer d'un mode de description à l'autre par des traductions. La figure 2 représente ce processus descendant: un mode de description est décomposable en plusieurs niveaux d'abstraction. La traduction entre deux niveaux peut s'effectuer par synthèse automatique ou par synthèse manuelle; dans ce dernier cas, il est nécessaire de disposer d'un outil de vérification automatique pour s'assurer qu'il n'y a pas d'erreur.

4. Objectifs

Une méthode de conception doit satisfaire à quatre objectifs:

- sûreté de layout, c'est-à-dire obtenir un circuit sans erreur au premier tour d'intégration,
- réduction du temps et du coût de développement,
- obtenir une grande densité d'intégration, la surface des puces obtenues devant être comparable à celle obtenue par un dessin manuel,
- indépendance des règles de layout, un même circuit devant pouvoir être traduit en layout géométrique selon les exigences de plusieurs fabricants.

5. Méthode de conception proposée

Le nombre et la nature des niveaux d'abstraction des trois modes de description permettent une bonne caractérisation d'une méthode de conception. En outre, le recours à des architectures régulières et à une utilisation intensive de mémoires impose le choix d'architectures du type micro-ordinateur; la spécificité du cahier des charges réside alors dans un programme d'application logé dans une mémoire.

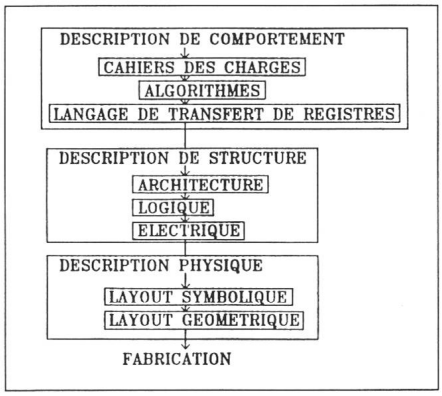

Fig. 2 Modes de description

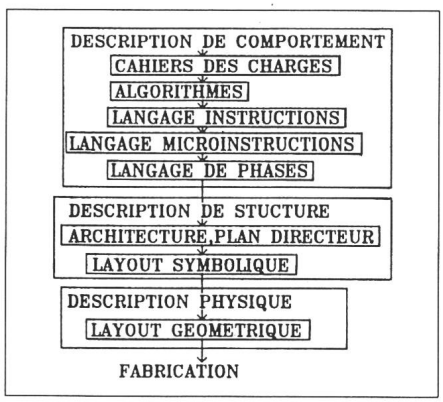

Fig. 3 Méthode proposée

La figure 3 représente le processus de conception de la méthode proposée. Le choix des algorithmes pour satisfaire au cahier des charges est laissé à l'expérience du concepteur. La définition du répertoire d'instructions permet alors de traduire les algorithmes en un programme d'application, soit dans le langage dit instructions. Chaque instruction doit être interprétée par un algorithme d'interprétation écrit dans un langage dit de micro-instructions, permettant de définir les étapes d'exécution des instructions. Plusieurs couches d'interprétation sont possibles pour parvenir au langage dit de phases représentant l'exécution des micro-instructions au rythme d'horloges non recouvrantes [9]. Les algorithmes d'interprétation écrits dans le langage de phases adoptent le formalisme d'un langage de transfert de registres [8].

La correspondance avec l'architecture est alors immédiate. De manière à tenir compte des contraintes topologiques en cours de conception de l'architecture, un plan directeur de la puce du circuit est dessiné pour indiquer les implantations et les formes estimées des blocs du circuit [10; 11]. Cela peut déboucher sur des modifications des algorithmes d'interprétation facilitant l'implantation. Le plan directeur de la puce permet de fixer les formes, les dimensions et les coordonnées relatives des points d'interconnexion des éléments, de manière à pouvoir les concevoir en détail en tenant compte des contraintes topologiques. Il est dès lors

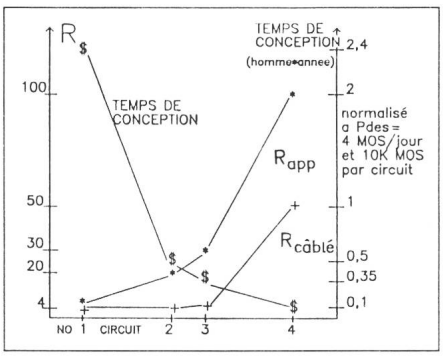

Fig. 4 Régularité et architecture
(se rapportant aux quatre circuits de la fig. 5)

évident que le nombre de niveaux de description de la structure doit être fortement réduit pour qu'il soit possible de prendre en considération les aspects topologiques au niveau de l'architecture. Ainsi, les éléments logiques doivent être conçus directement en layout symbolique à partir des descriptions de comportement. La traduction du layout symbolique en layout géométrique s'effectue alors automatiquement.

6. Régularité et architecture

Le choix de l'architecture est le plus important en ce qui concerne la régularité, puisqu'une logique câblée présente une régularité $R = 1$ à 10, des architectures de micro-ordinateurs $R = 10$ à 50, et enfin des architectures spéciales type réseau de processeurs $R \geqq 50$. Il est évident que le recours à des architectures type réseau de processeurs n'est pas applicable à n'importe quel problème: la régularité dépend donc aussi du type de cahier des charges. La figure 4 représente les courbes de régularité et du temps de conception pour les quatre circuits de la figure 5.

L'avantage d'architectures type processeur est l'utilisation intensive de mémoires très régulières. La spécificité du cahier des charges est entièrement laissée au programme résidant en ROM: l'architecture de l'unité de traitement est alors relativement universelle et peut se concevoir en tranches. Certains processeurs (circuit 2) ont une unité de contrôle très complexe pour l'exécution des instructions. Il est très avantageux de recourir à des architectures dites de machines de décision binaire [12] où la logique de contrôle est limitée (3% de la surface du circuit 3). Enfin, pour certains types de problèmes (traitement de signal), il faut rechercher le parallélisme inhérent à tout algorithme pour implanter des réseaux de processeurs travaillant en parallèle (circuit 4). En outre, toute fonction logique admet un mode de description dit algorithme de décision binaire [12]; celui-ci peut être matérialisé par de la logique câblée ou programmée dans une machine de décision binaire. Cette équivalence matériel-logiciel permet de choisir des architectures de machines très régulières pour des fonctions logiques qui étaient traditionnellement réalisées par des portes logiques.

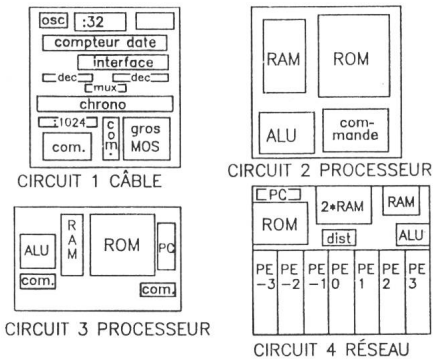

Fig. 5 Architectures de circuits intégrés

7. Plan directeur

Le plan directeur de la puce, conçu en même temps que l'architecture du circuit, a pour but de prendre en compte les aspects topologiques le plus tôt possible, de manière à pouvoir modifier immédiatement l'architecture si nécessaire. Il est utilisable pour n'importe quel type d'architecture, et sera d'autant plus fiable que l'expérience de l'équipe de conception est grande. C'est un outil à la fois concret, simple à mettre en œuvre et indispensable pour obtenir une surface minimum de la puce.

Pour une architecture de micro-ordinateur, les mémoires sont en général les blocs les plus gros. Connaissant la surface d'une cellule-mémoire pour la technologie choisie, il suffit de déterminer leurs organisations en nombre de lignes et colonnes pour obtenir les dimensions des blocs. Plusieurs organisations seront essayées pour retenir la plus favorable. Des blocs comme la banque de compteurs de programme (PC) ou l'unité arithmétique et logique (ALU) sont estimés en tenant compte de leurs décompositions en tranches. Une tranche ALU comporte le système logique combinatoire réalisant les opérations désirées, un ou deux accumulateurs et des portes à trois états pour la connexion au bus de données. Il est possible d'estimer le nombre de transistors que comporte chacune de ces cellules. Connaissant la densité en nombre de transistors par mm² pour la technologie choisie, pour la technique de layout qui sera utilisée et pour le genre de circuits (dans ce cas, une tranche ALU), il est possible d'estimer la surface de la tranche ALU. On procède de même pour l'estimation des blocs de logique de commande, de décodage et du séquenceur du micro-ordinateur.

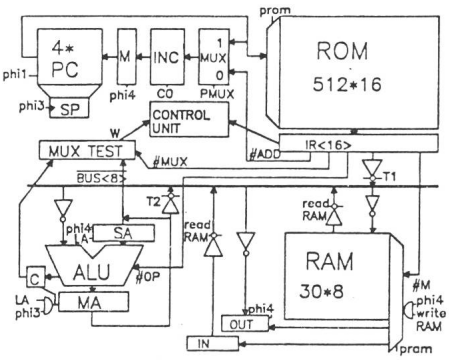

Fig. 6 Architecture du micro-ordinateur (circuit 5)

La figure 6 représente l'architecture d'un micro-ordinateur comportant les quatre blocs principaux que sont la mémoire morte (ROM), la mémoire vive (RAM), une banque de quatre compteurs de programmes de neuf bits (PC) et une unité arithmétique et logique de huit bits (ALU). La figure 7 représente le plan directeur établi à partir de cette architecture pour une technologie CMOS 6 μm. Les cotes sont données en μm. Ce plan directeur peut alors être utilisé pour la conception en layout symbolique de chaque bloc: le concepteur sait alors parfaitement où placer les points d'entrée/sortie des cellules pour minimiser la longueur des interconnexions. En outre, cela lui permet de dessiner ses cellules

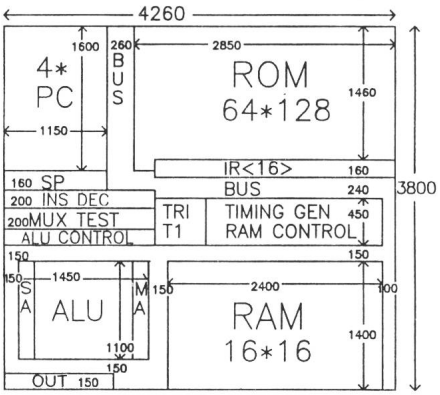

Fig. 7 Plan directeur en technologie CMOS 6 μm (circuit 5)

de manière à ce qu'elles s'imbriquent parfaitement dans le dessin d'ensemble de la puce.

En outre, pour faciliter l'assemblage final, le concepteur doit prévoir que les cellules seront assemblées par simple collage. Les interconnexions entre cellules voisines doivent donc correspondre. De plus, pour éviter de longs détours aux variables d'interconnexion entre cellules non voisines, ces variables traverseront de part en part les cellules placées sur le plus court chemin [10].

8. Layout orienté métal

Une technologie MOS à grille de silicium polycristallin (poly) présente essentiellement trois couches de conducteurs: métal, diffusion et poly. Le croisement des couches diffusion et poly réalise le transistor MOS. De manière à faciliter l'obtention automatique de layouts géométriques, il est indispensable de restreindre très fortement les degrés de liberté dont jouit le concepteur de layout. La technique de layout proposée, dite layout orienté métal, est caractérisée par le croisement systématique des couches métal et diffusion [18]. Les conducteurs métalliques sont placés uniquement horizontalement à distance minimum, tandis que les diffusions sont uniquement verticales. Le transistor MOS est créé en plaçant une grille poly connectée à une ligne métal et croisant une diffusion. Des variables de commande implémentées en poly peuvent traverser la cellule verticalement.

La méthode de synthèse de cellules est illustrée par la figure 8. Les tables de vérités des fonctions logiques à

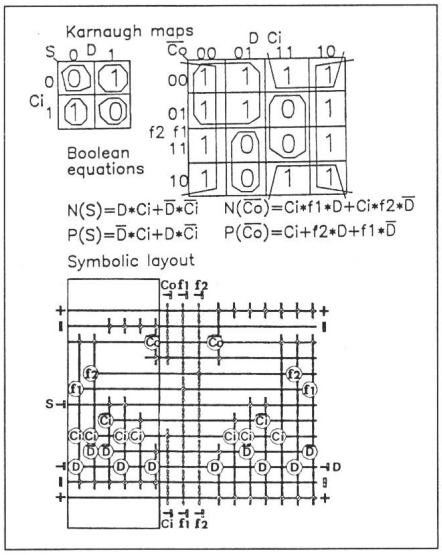

Fig. 8 Méthode de synthèse

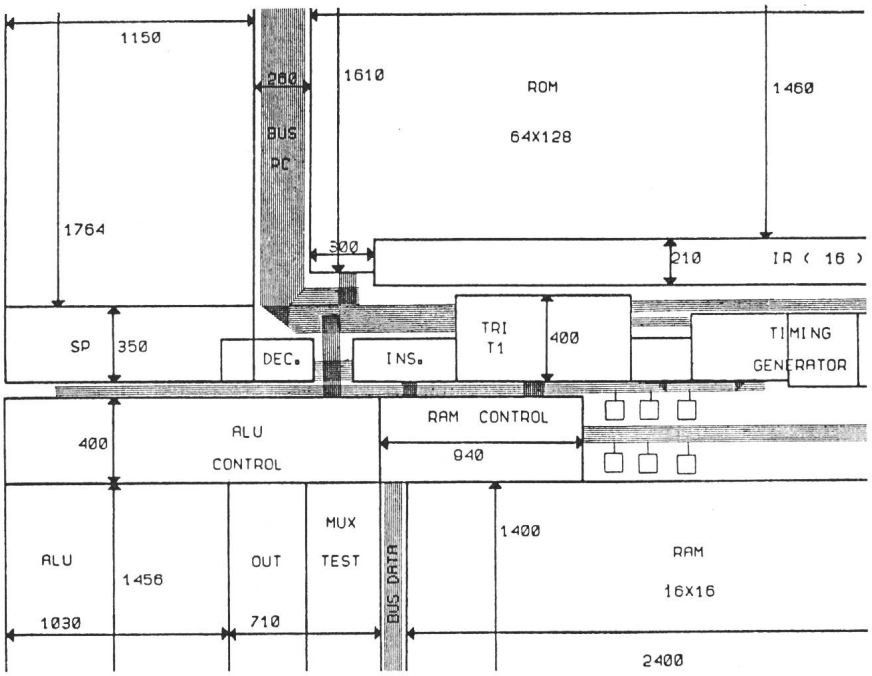

Fig. 9 Layout symbolique

concevoir, qui sont un mode de description de comportement, permettent d'obtenir par la méthode de Karnaugh les équations logiques simplifiées pour les deux réseaux de transistors n-MOS et p-MOS en technologie CMOS. Le ET logique, représentant la mise en série de transistors et le OU logique la mise en parallèle, ces équations sont directement traduites en un layout symbolique orienté métal. Un programme de traduction permet alors d'obtenir le layout géométrique correspondant automatiquement, ceci pour différentes technologies.

Cette méthode a été testée sur différents exemples qui démontrent son intérêt. En effet, la productivité au niveau graphique passe de 4–6 MOS/jour à près de 20 MOS/jour, soit une augmentation d'un facteur 3 à 5. Les densités obtenues sont comparables à celles obtenues pour des layouts manuels: 450 MOS/mm² pour une technologie 6 μm et 1200 MOS/mm² pour une technologie 4 μm.

La figure 9 représente une partie du dessin d'ensemble en layout symbolique du micro-ordinateur du chapitre 7. On constate que cette partie traditionnellement peu régulière, comportant le pointeur de pile, les décodeurs d'instructions, le séquenceur, la logique de contrôle pour la RAM et l'ALU, s'implante parfaitement par simple collage de cellules. La technique de layout proposée facilite grandement l'assemblage par collage de cellules; tous les conducteurs métalliques étant à distance minimum, les cellules s'adaptent parfaitement à des structures de bus métal.

9. Comparaison entre techniques de layout [13][14]

Nous comparerons quatre styles de layout:

A. Le layout manuel, consistant à dessiner géométriquement la quasi totalité d'un circuit intégré. Aucune réelle méthodologie n'est apparente, et comme toutes les étapes sont manuelles, il subsiste toujours des erreurs. Son seul succès: la densité.

B. Le layout structuré, décrit par *Mead/Conway* [15], qui considère chaque bloc fonctionnel comme une matrice de $m \times n$ cellules si possible toutes identiques, ou du moins comprenant un faible nombre de cellules. Ces cellules sont de faible complexité et dessinées à la main. On cherche donc ici à obtenir la plus grande régularité pour augmenter la productivité. Les erreurs sont moins fréquentes et plus facilement corrigées du fait de la structuration. On obtient en général une bonne densité.

C. Le layout symbolique, sur grille ou compacté [16] qui permet à l'utilisateur de ne plus travailler au niveau graphique. Il n'y a donc plus d'erreurs à ce niveau-là. La densité est en général moins bonne que pour les deux premières techniques de layout, bien que certaines orientations parviennent aux mêmes densités. Il est évident que ce style de layout est aussi structuré: sim-

plement les cellules sont dessinées avec une autre technique que manuelle.

D. Cellules standards avec placement-routage automatique. Cette technique est basée sur l'existence de cellules prédéfinies dans une bibliothèque, en général dessinées à la main et testées par des circuits intégrés test. Les cellules ont toutes la même hauteur pour pouvoir s'interconnecter en bandes. Le programme CAO assure un placement optimal des cellules dans les bandes et des bandes dans le circuit, avant de procéder au routage automatique en utilisant les canaux d'interconnexions entre les bandes. La méthode est cohérente, elle produit des circuits sans erreurs. Néanmoins la densité est mauvaise, ce qui n'est pas étonnant: cette méthode est à l'opposé de celle qui introduit très tôt le plan directeur de la puce. Par contre la productivité est élevée. Il est intéressant de noter qu'une telle méthode ne profite pas d'une quelconque régularité: choisir des cellules identiques ou non ne change rien puisqu'elles existent déjà, et que le travail est le même si l'on interconnecte des cellules identiques ou non. En outre, une telle méthode est de moins en moins efficace pour les circuits VLSI: plus le nombre de cellules est grand, plus les canaux d'interconnexions sont importants, et plus la densité est mauvaise.

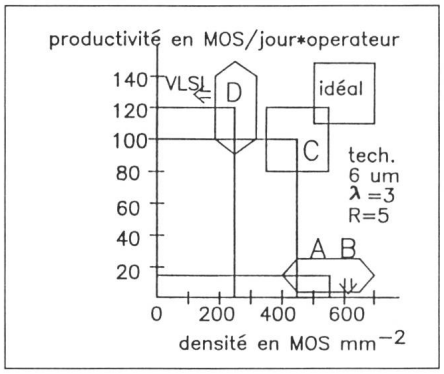

Fig. 10 Productivité et densité

A layout manuel
B layout structuré
C orienté métal
D cellules standards

La figure 10 [14] représente la comparaison entre les quatre techniques de layout au point de vue densité et productivité. La densité est calculée pour une technologie 6 μm (λ = 3 μ). La productivité réelle est $P = P_{des} \cdot R$, en choisissant une régularité R = 5 pour des blocs en logique câblée.

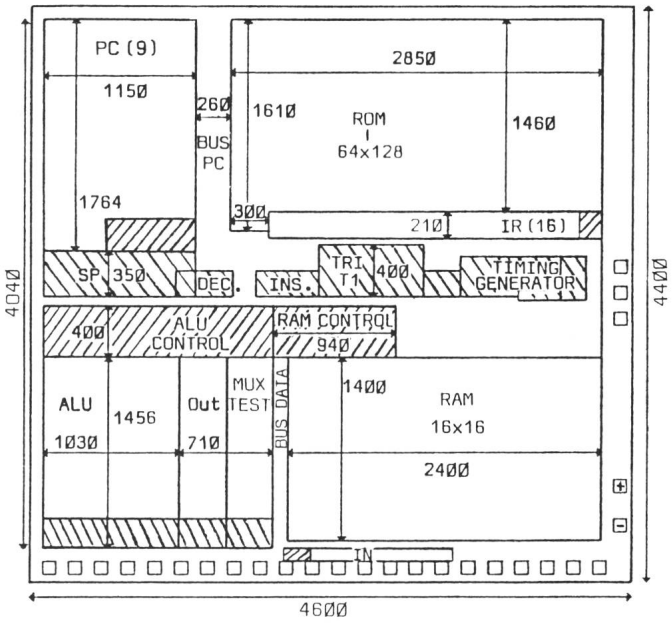

Fig. 11 Dessin d'ensemble de la puce

10. Conclusion

L'utilisation d'un plan directeur associé à l'architecture du circuit permet un gain de temps considérable à l'assemblage final des blocs du circuit. En effet, comme le représente la figure 11, les cellules sont pour la plupart assemblées par simple collage. En outre, elles contiennent en général des interconnexions entre d'autres cellules. On constate par ailleurs que les différents blocs du plan directeur de la figure 7 se retrouvent sur la figure 11 avec des formes et des emplacements quelque peu modifiés, indiquant bien que le plan directeur est un outil prédictif. Néanmoins, l'estimation de surface du noyau du processeur (4260 μm × 3800 μm) est tout à fait acceptable (4260 μm × 4040μm). La figure 11 donne encore une appréciation graphique de la régularité: les parties hâchurées ont été dessinées alors que les autres ont été répétées ou générées par des outils automatiques (ROM, RAM). La

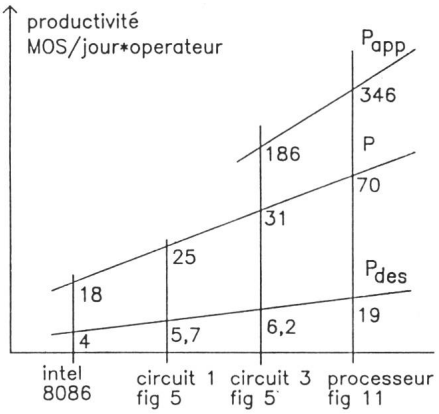

Fig. 12 Productivité de différents circuits

régularité du processeur est R = 3,7 (sans les mémoires) et R_{app} = 18,2 (avec les mémoires).

Il y a évidemment conflit entre régularité et densité: deux cellules fonctionnellement identiques peuvent devoir être dessinées deux fois parce que leurs contraintes topologiques sont différentes.

La figure 12 compare les valeurs des

Tableau III

Circuit	Densité et technologie	Densité équivalente en technologie 4 μ	Nombre total de MOS
Circuit type INTEL 8086 [3]	880 MOS/mm²; MOS 4 μ	880 MOS/mm²	29 000
Circuit 1 figure 5	140 MOS/mm²; CMOS CEH 6 μ	350 MOS/mm²	2 000
Circuit 3 figure 5	500 MOS/mm²; CMOS CEH 6 μ	1250 MOS/mm²	20 000
CPU 16 bits cellules standards [17]	270 MOS/mm²; nMOS 3 μ double niveau de métal	150 MOS/mm²	20 000
Processeur figure 11	660 MOS/mm²; CMOS CEH 6 μ	1650 MOS/mm²	13 500

productivités pour différents circuits. On constate une augmentation notable de la productivité pour la méthode proposée (processeur de la fig. 11) due à l'utilisation du plan directeur et des techniques de layout symbolique du chapitre 8.

Le tableau III indique les valeurs des densités pour différents circuits. Elle confirme que l'approche des cellules standards en bibliothèque est peu favorable sur le plan densité. Enfin, il apparaît que la densité obtenue par la méthode proposée est aussi dense sinon plus que celle obtenue par des conceptions manuelles. C'est là un résultat tout à fait intéressant.

Mais il est évident que ces hautes densités sont obtenues pour des circuits comportant des mémoires de densité très élevée. Pour la logique câblée du processeur de la figure 11, on a obtenu en technologie CMOS CEH 6 μ 370 MOS/mm² pour l'ensemble des cellules et 510 MOS/mm² pour le compteur de programme. Ce sont là des densités tout à fait comparables à celles obtenues pour des layouts manuels non orientés.

Il ressort donc que la méthode proposée présente un certain nombre d'avantages, et satisfait aux objectifs du chapitre 4. Elle convient bien à une organisation verticale [7] de l'équipe de conception d'un circuit, à savoir où chaque personne est responsable d'une partie d'un circuit depuis l'architecture jusqu'au layout.

Bibliographie

[1a] *G. E. Moore:* Are we really ready for VLSI. Proceedings of the Caltech Conference on Very Large Scale Integration, 22...24 January 1979. Pasadena, California Institute of Technology; p. 3...14.

[1b] *G. E. Moore:* VLSI: some fundamental challenges. IEEE Spectrum 16(1979)4, p. 30...37.

[2] *M. Shima:* Demystifying microprocessor design. IEEE Spectrum 16(1979)7, p. 22...30.

[3] *D. F. Barbe:* Very Large Scale Integration. VLSI fundamentals and applications. Second edition. Berlin/Heidelberg/New York, Springer-Verlag, 1982.

[4] *W. W. Lattin* a. o.: A 32b VLSI micromainframe computer system. IEEE International Solid-State Circuits Conference 1981 (ISSCC '81). Digest of technical papers p. 110...111.

[5] *D. T. Fitzpatrik* a. o.: A riscy approach to VLSI. Lambda, Magazine of VLSI Design 2(1981)4, p. 14...20.

[6] *J. W. Beyers* a. o.: A 32-bit VLSI CPU chip. IEEE Journal of Solid-State Circuits 16(1981)5, p. 537...542.

[7] *W. W. Lattin* a. o.: A methodology for VLSI chip design. Lambda, Magazine of VLSI Design 2(1981)2, p. 34...44.

[8] *C. Piguet* e. a.: Le séquenceur d'un microprocesseur. Bull. ASE/UCS 70(1979)3, p. 126...132.

[9] *M. Obreska:* Efficiency and performance comparison of different design methodologies for control parts of microprocessors. Microprocessing and Microprogramming 10(1982)–, p. 163...178.

[10] *F. Anceau* and *R. A. Reis:* Complex integrated circuit design strategy. IEEE Journal of Solid-State Circuits 17(1982)3, p. 459...464.

[11] *R. A. Reis:* TESS: A topological evaluator tool. IEEE International Conference on Circuits and Computers, New York, Sept. 28...Oct. 1, 1982 (ICCC '82); p. 539...542.

[12] *D. Mange:* Arbres de décision pour systèmes logiques câblés ou programmés. Bull. ASE/UCS 69(1978)22, p. 1238...1243.

[13] Special issue on VLSI design: problems and tools. Proc. IEEE 71(1983)1.

[14] *H. De Man:* Computer-aided design techniques for VLSI. In: Design Methodologies for VLSI circuits. NATO Advances Study Institute Series, Series E, No. 47. Alphen aan der Rijn/Rockville Maryland, Sijthoff and Noordhoff, 1982.

[15] *C. Mead* and *L. Conway:* Introduction to VLSI systems. Reading/Massachusetts, Addison-Wesley, 1980.

[16] *J. C. Martin* e. a.: Synthèse automatiques de circuits CMOS. Bull. ASE/UCS 74(1983)5, p. 221...223.

[17] *T. Tokuda:* a. o.: 16b CPU design by a hierarchical polycell approach. IEEE International Conference on Circuits and Computers, New York, Sept. 28...Oct. 1, 1982 (ICCC '82); p. 102...105.

[18] *C. Piguet* and *M. Bertarionne:* Automatic generation of metal oriented layout for CMOS logic. Ninth European Solid-State Circuits Conference, Lausanne 1983 (ESSCIRC '83). Digest of technical papers p. 167...170.