

# Construction d'une bibliothèque de cellules standard par un groupe d'étudiants

**Jean Louis Noullet, Philippe Bourdeu d'Aguerre,  
Atelier Interuniversitaire de Micro-Electronique  
AIME Toulouse**

## 1. Contexte du projet

Les méthodes efficaces de conception d'ASICs qui sont maintenant enseignées reposent sur la méthodologie descendante ("top down") et la synthèse logique, ce qui suppose l'existence d'une bibliothèque de cellules standard ("design kit").

L'utilisation de bibliothèques de cellules standard pré-définies et validées permet une conception rapide et sûre des ASICs.

Pour les concepteurs d'ASICs, la bibliothèque est souvent vue comme un produit "fermé", qu'on achète tout fait. Mais il ne faut pas perdre de vue que la construction de telles bibliothèques occupe de nombreux concepteurs, ce dont nous avons eu confirmation en observant les sujets de stages proposés aux étudiants ces dernières années par les entreprises.

Cela nous a donc paru un sujet intéressant pour les "Bureaux d'études" en conception de circuits intégrés. La réalisation d'un tel projet apporterait en outre un bénéfice supplémentaire : une bibliothèque d'origine universitaire serait exempte des clauses de confidentialité qui interdisent aux utilisateurs d'avoir connaissance du layout interne des cellules des bibliothèques commerciales. L'intérêt pédagogique de systèmes "ouverts" a déjà été démontré dans le cadre de projets plus ambitieux que le notre [1].

L'idée n'est pas nouvelle, mais il y a encore quelques années les méthodes de conception logique manuelle imposaient l'utilisation de bibliothèques assez conséquentes (par exemple 70 cellules). Cependant les progrès spectaculaires des logiciels de synthèse permettent à présent de travailler efficacement avec des jeux de cellules plus restreints, ce qui rend plus abordable l'entreprise de construction d'une bibliothèque "maison".

Enfin ce projet devait s'insérer dans un environnement imposé, comportant les choix suivants :

- en 1997-98, un groupe de 8 étudiants de DESS micro-électronique
- technologie : CMOS "AMS 0.6 microns"
- outil pour schématisation, layout, placement-routage : Cadence 4.4
- outil pour la simulation électrique : HSPICE
- outil pour la synthèse : Synopsys
- langage pour la synthèse et la simulation : Verilog
- format de description temporelle : TLF

L'idée d'une bibliothèque "portable" n'a pas été retenue, mais il a été posé en principe que la méthode utilisée devait permettre une migration facile vers d'autres technologies.

## **2. Etapes du travail**

### **2.1 Choix des fonctions à réaliser**

Les cellules d'un design kit rentrent dans trois catégories :

1. cellules standard du coeur
2. cellules de la périphérie ("pads")
3. blocs matriciels (mémoires, chemins de données)

Concernant les cellules du coeur, nous nous sommes basés sur une étude systématique [2] donnant une liste de 18 cellules ("Reduced ASIC Cell Library") nécessaires à une utilisation efficace de la synthèse logique, à laquelle on doit ajouter le buffer "trois-états".

Concernant les cellules de la périphérie, l'accent a été mis sur l'économie de surface, l'expérience devant être poussée jusqu'à la réalisation de prototypes. Dans cet esprit, on a mis dans les cellules de coins ("corners") les pads d'alimentation de la périphérie pour libérer de l'espace dans les rangées de pads standard.

La réalisation de blocs matriciels a été reportée à une seconde phase, au cours de laquelle un générateur de RAM et ROM devra être réalisé.

### **2.2 Conception au niveau transistor et validation par simulation HSPICE**

La conception de cellules de base comme Nand et Nor ne laisse pas beaucoup de place à l'imagination, et ne justifie pas de validation du principe par simulation.

Par contre la bascule statique (sensible au front) justifiait une étude sérieuse. Le choix s'est porté sur la solution minimale, la bascule à contention, dont le fonctionnement est assez sensible au dimensionnement des transistors. Cette bascule étant réputée engendrer des pics de consommation aux commutations, il a fallu trouver un dimensionnement qui ramène ces pics à un niveau comparable à celui des autres types de bascules.

L'extraction de paramètres temporels par simulation HSPICE dans différents cas de charge capacitive ("fanout") pour chaque cellule n'a été qu'ébauchée. Il est apparu clair que cette tâche ne peut être menée à bien qu'en automatisant la gestion des simulations et la collecte des résultats.

### **2.3 Dessin des masques aux règles du fondeur, vérifications**

Les cellules standard du coeur devant être aboutées latéralement, il a fallu imposer aux concepteurs un jeu de consignes garantissant que l'aboutement se fasse toujours sans violation des règles.

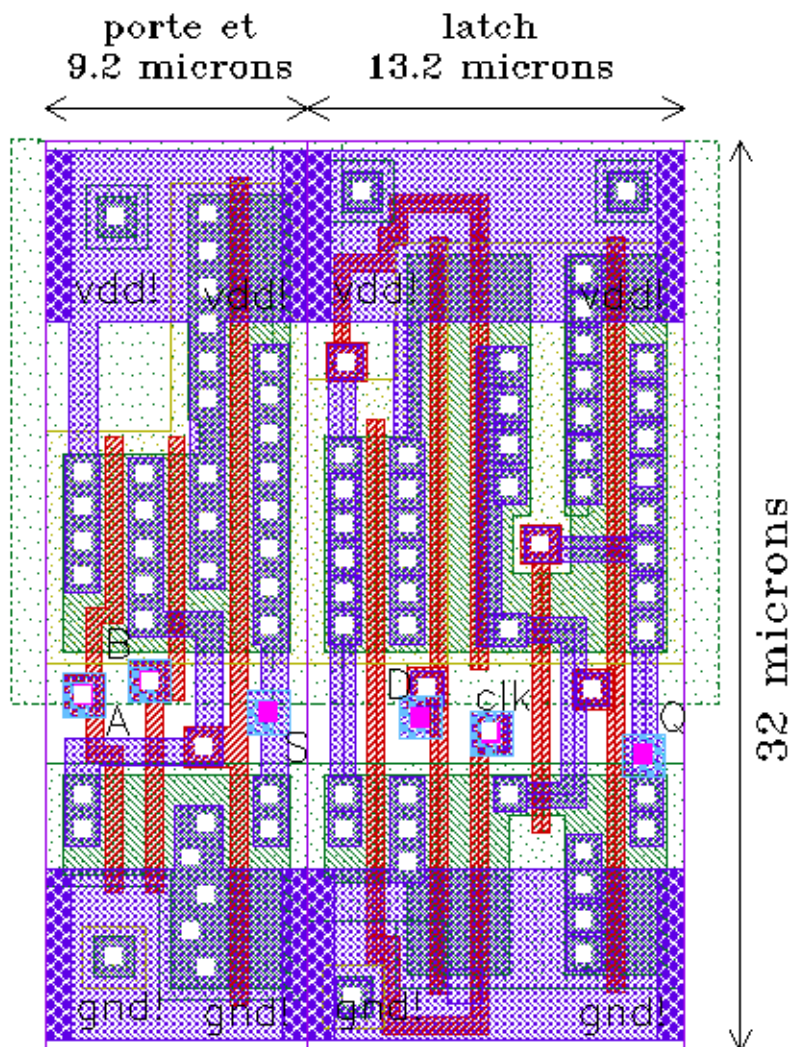
Il a fallu ensuite configurer le système Cadence pour la génération automatique des vues "abstract" qui sont des layouts simplifiés mais contenant toutes les informations nécessaires pour le placement-routage automatique.

La vérification ne s'est pas limitée au DRC de chaque cellule, il a fallu effectuer le placement-routage de plusieurs circuits complexes et effectuer chaque fois sur l'ensemble un DRC et un LVS (comparaison layout-schéma) au niveau transistor.

La figure 1 montre le layout de deux cellules, “porte Et” et “latch”.

Le layout des pads peut être extrêmement fastidieux, car il faut placer des centaines de contacts et de vias. Pour alléger cette corvée, on a essayé de profiter au maximum de la possibilité de hiérarchiser le layout. Ainsi les diodes de protection, les “snaps” (écrêteurs à avalanche) et les transistors de sortie ont été réalisés par juxtaposition de tranches identiques.

**FIGURE 1. Layout de deux cellules du coeur**



## 2.4 Création de modèles pour la synthèse et la simulation logique

Pour que le programme de synthèse Synopsys puisse assembler les cellules, il faut lui fournir leur description fonctionnelle dans un langage spécifique (format “lib”) qu’on compile pour obtenir une bibliothèque de synthèse (format “db”).

Cette description comprend une partie purement logique, qui est facile à gérer dans le cas d’un petit nombre de cellules, et une partie temporelle qui est plus délicate car on doit y incorporer des résultats extraits des simulation SPICE.

Les modèles pour la simulation Verilog contiennent pratiquement les mêmes informations, malheureusement dans un langage différent. Jusqu'à présent, il a fallu effectuer la traduction manuellement.

Cependant la cohérence logique de ces données a pu être vérifiée en synthétisant quelques circuits et en vérifiant que la simulation après synthèse (niveau portes) donnait les mêmes résultats que la simulation avant synthèse (au timing près).

La méthode la plus efficace pour obtenir des simulations réalistes du point de vue temporel avec éventuellement prise en compte des capacités effectives du routage (rétroannotation) est basée sur l'utilisation d'un calculateur de délai ("CDC") utilisant une description des caractéristiques temporelles des cellules en format "TLF" (Timing Library Format).

Un groupe de deux étudiants (parmi les 8) s'est vu confier la tâche d'élucider les mystères de l'utilisation du format TLF. Les résultats sont arrivés malheureusement trop tard pour être appliqués à la version actuelle de la bibliothèque.

### 3. Validation sur silicium

Pour concrétiser les résultats modestes de l'étape 1997-98 du projet, un circuit de validation a été envoyé à la fabrication par l'intermédiaire du C.M.P..

L'objectif était de faire un circuit capable de se tester à des fréquences d'horloge proches du maximum, sans pour autant faire appel à des instruments de test à hautes performances.

A cet effet, il a été inclus :

- un oscillateur en anneau pour produire le signal d'horloge (de quelques centaines de MHz),
- un compteur synchrone 3 bits "modulo 5" qui constitue le circuit "cible" du test
- une cascade de diviseurs de fréquence asynchrones, pour ramener les signaux à observer dans une gamme inférieure à 2 MHz.

L'oscillateur en anneau contient un multiplexeur qui permet de choisir 3 longueurs d'anneau, soit 11, 15 et 21 étages, devant donner 3 fréquences inversement proportionnelles à ces nombres.

Un autre multiplexeur permet d'envoyer vers la cascade de diviseurs soit l'horloge divisée par 5 (compteur synchrone) soit l'horloge divisée par 4 (2 étages de diviseur asynchrone).

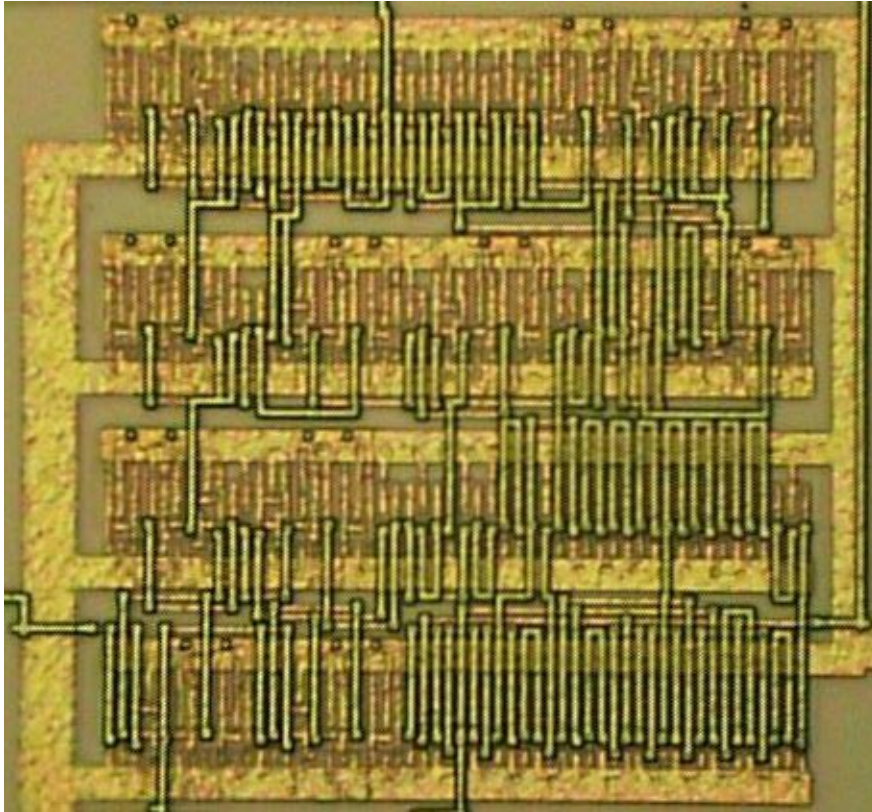
Ainsi par un simple jeu de mesures de fréquences (basses), on peut :

- vérifier que le compteur synchrone fonctionne,
- savoir à quelle fréquence d'horloge,
- ceci pour trois fréquences différentes.

Ce circuit utilise seulement 5 cellules différentes (nor2, nand2, inv, dff, and2) et ses dimensions sont d'environ 160 x 160 microns. Il a été incorporé à un circuit d'un autre projet pour être envoyé à la fabrication dans des conditions économiquement acceptables.

Sur la photo de la [figure 2](#), on distingue les connexions de métal 2 au premier plan et en arrière plan les connexions de métal 1, dont les larges pistes de masse et d'alimentation. (Les grilles de MOS de 0.6 microns sont pratiquement indiscernables au microscope optique.)

**FIGURE 2. Photographie du circuit de test**



Les tests ont porté sur 5 exemplaires du circuit. Les table 1 et 2 donnent les valeurs de fréquence d'horloge mini et maxi déduites des mesures, et la valeur obtenue par simulation HSPICE (modèles "typiques-nominaux").

Dans la table 1 on s'est intéressé aux trois longueurs possibles de l'anneau (à 5 Volts), dans la table 2 à l'influence de la tension d'alimentation (pour l'anneau de 11 étages).

Dans **tous les cas**, le compteur synchrone a fonctionné parfaitement.

**Table 1. Test à 5 Volts**

	<b>Mesure Mini</b>	<b>Mesure Maxi</b>	<b>Simulation HSPICE</b>
<b>Anneau à 11 étages</b>	379 MHz	388 MHz	410 MHz
<b>Anneau à 15 étages</b>	275 MHz	282 MHz	288 MHz
<b>Anneau à 21 étages</b>	212 MHz	217 MHz	216 MHz

**Table 2. Anneau 11 étages**

	<b>Mesure Mini</b>	<b>Mesure Maxi</b>	<b>Simulation HSPICE</b>
<b>Vdd = 6 Volts</b>	412 MHz	422 MHz	450 MHz
<b>Vdd = 3.3 Volts</b>	281 MHz	288 MHz	295 MHz
<b>Vdd = 2.5 Volts</b>	204 MHz	210 MHz	208 MHz

Notons que ces valeurs ont été obtenues en incluant dans la netlist HSPICE les aires et périmètres de drains et sources de tous le MOS et une capacité de routage forfaitaire (12 fF) pour chaque piste de routage.

Sans ces éléments les fréquences obtenues étaient supérieures de 30%, ce qui illustre la grande sensibilité des MOS submicroniques aux éléments parasites.

## 4. Conclusion et perspectives

Bien que d'ampleur très modeste, le projet a abouti à des résultats très encourageants.

La bibliothèque qui ne contient actuellement que 5 cellules de coeur et 6 cellules de périphérie est déjà entièrement opérationnelle au niveau synthèse, simulation logique et layout. Par contre les modèles de timing ne sont pas au point.

Une des conclusions nettes est que pour faire efficacement ce type de travail il est indispensable de faire un peu de programmation. Par exemple on devra créer des programmes pour gérer les simulations de caractérisation de timing et engendrer les fichiers aux formats Synopsys, Verilog et TLF de façon cohérente et automatique.

En 98-99, le projet sera continué avec un groupe d'étudiants plus nombreux, avec les propositions d'objectifs suivantes :

- exploitation du troisième niveau de métal introduit par AMS
- gestion efficace des données de timing
- bibliothèque complète de 19 cellules
- générateur de RAM et ROM

## 5. Références

[1] "A Portable Layout Approach well Adapted to Education",  
Greiner Alain, Pétrot Frédéric, Wajsbürt Franck,  
Proceedings of the 1st European Workshop Microelectronics Education, Grenoble, France, February 1996, pp. 97-102

[2] "Do we need so many cells for digital ASIC synthesis ?",  
J.I. Noullet, A. Ferreira-Noullet,  
Proc. of the 5th International Workshop on Mixed Design of Integrated Circuits and Systems (MIXDES'98), Lodz (Pologne), June 1998