

# Un Microprocesseur de Comparaison Dynamique pour la Reconnaissance Vocale<sup>1</sup>

*Georges QUÉNOT<sup>2</sup>, Jean-Luc GAUVAIN, Jean-Jacques GANGOLF, Joseph MARIANI*

LIMSI - CNRS, BP 30, 91406 ORSAY CEDEX Tel: (16 1) 69 85 81 11

## Résumé

Nous présentons un processeur intégré spécialisé pour effectuer les calculs de programmation dynamique pour les systèmes de reconnaissance vocale. Ce processeur délivre une puissance de 10 MIPS efficaces à 20 MHz et son architecture parallèle et pipe-line lui permet d'effectuer en moyenne plus de 30 millions d'opérations par seconde. Un système utilisant un seul processeur de comparaison dynamique peut reconnaître 5000 mots isolés monocuteur, 500 mots isolés multilocuteur ou 600 mots enchaînés monocuteur. Plusieurs de ces processeurs peuvent être utilisés en parallèle dans un même système. L'originalité de ce processeur est d'offrir simultanément: une très grande puissance unitaire, une très grande souplesse d'emploi et une capacité de fonctionnement efficace en multiprocesseur.

Le circuit a été réalisé dans une technologie CMOS 2  $\mu\text{m}$ , il intègre 127309 transistors dans 60 mm<sup>2</sup> et est disponible en boîtier 84 broches de type PGA ou PLCC. Il comprend une mémoire de programme interne de 640 instructions sur 24 bits et fonctionne avec une mémoire de données externe de 256 Kmots de 16 bits. Le développement comprend: un chemin de données et un générateur d'horloges entièrement personnalisés, une mémoire compilée et une réalisation sur base de cellules standard pour la logique de contrôle et les plots d'entrée-sortie.

## I. Introduction

La programmation dynamique est une méthode très utilisée pour la comparaison et l'alignement temporel en reconnaissance vocale<sup>1,2,3,4</sup>. Elle conduit toujours à la résolution récurrente dans un ensemble de matrices d'équations de types suivants:

Distance locale:

$$d(t, i) = \sum_{k=0}^{k=p-1} | a(t, k) - b(i, k) |$$

---

<sup>1</sup>Ce travail a été subventionné par France Télécom (contrat DAI n° 86.35.053) et a été effectué avec la collaboration des sociétés Bull SA, VECSYS SA et VLSI technology.

<sup>2</sup>De la Délégation Générale pour l'Armement

Distance cumulée:

$$g(t, i) = \min \begin{cases} g(t-1, i-2) + c_1 \times d(t, i-1) + c_2 \times d(t, i) \\ g(t-1, i-1) + c_3 \times d(t, i) \\ g(t-2, i-1) + c_4 \times d(t-1, i) + c_5 \times d(t, i) \end{cases}$$

Propagation d'étiquette:

$$b(t, i) = \text{tracemin} \begin{cases} b(t-1, i-2) \\ b(t-1, i-1) \\ b(t-2, i-1) \end{cases}$$

où  $c_1, c_2, c_3, c_4$  et  $c_5$  sont des constantes dont les valeurs peuvent être 0, 1/2, 1, 3/2, 2 ou 3. Dans la troisième équation la fonction *tracemin* retourne l'argument correspondant au minimum des trois termes de la deuxième équation.

L'ensemble de ces équations constitue le traitement d'un point de programmation dynamique. Pour la reconnaissance de mots isolés, la troisième équation n'est pas nécessaire. Les applications sont personnalisées par la forme précise des équations retenues et par la façon dont celles-ci sont initialisées et enchaînées.

Tenant compte du fait que ces calculs sont relativement simples et très répétitifs, nous avons développé un microprocesseur spécialisé pour les effectuer très rapidement. De nombreux autres processeurs spécialisés pour la programmation dynamique existent déjà<sup>5,6,7,8,9</sup>, l'originalité de celui-ci est de fournir simultanément toutes les options suivantes:

- Capacité de traitement supérieure à 600000 points de programmation dynamique par seconde pour les mots isolés et à 300000 points par seconde pour la parole continue,
- Equations locales programmables,
- Possibilité d'ajouter une équation de propagation d'étiquette,
- Architecture flexible et mémoire de programme intégrée permettant l'utilisation de nombreuses techniques d'optimisation comme: contraintes de pente, restrictions de domaine de recherche dynamiques, élagage, reconnaissance hiérarchique et contraintes syntaxiques,
- Possibilité de fonctionnement en multiprocesseur avec une mémoire partagée,
- Interface système très simple par connexion directe au bus du processeur hôte et aux boîtiers constituant la mémoire locale.

## II. Architecture du système de reconnaissance

Le système comporte trois sous-ensembles (Figure 1):

- Un module d'acquisition et d'analyse du signal qui transforme en temps réel le signal acoustique en une forme numérique (matrice de nombres) adaptée pour la comparaison,

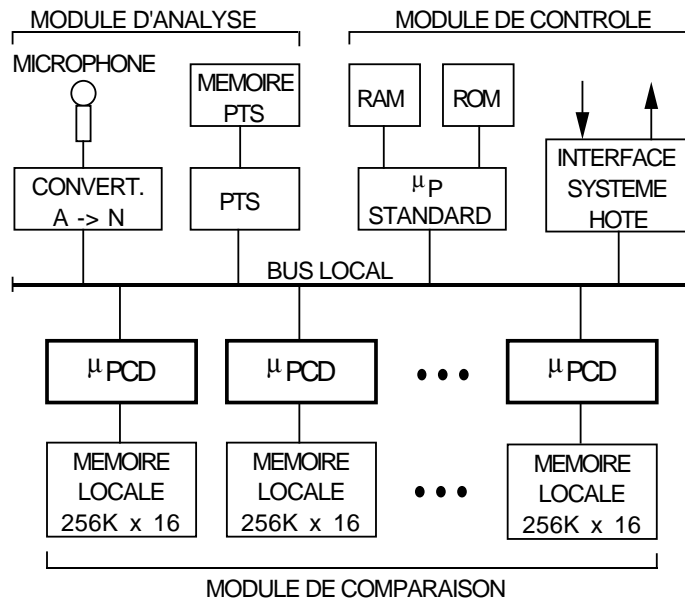


Figure 1: Architecture du système de reconnaissance

- Un module spécialisé pour la comparaison, comportant un ou plusieurs microprocesseurs de comparaison dynamique ( $\mu$ PCD) avec chacun sa mémoire locale, qui effectue en temps réel la comparaison entre la forme acoustique numérique venant du module d'analyse et une bibliothèque de formes de référence stockées dans les mémoires locales,
- Un module de contrôle et d'interface avec l'application.

### III. Description du processeur

Le processeur a deux interfaces, une avec son processeur hôte et une avec sa mémoire locale (Figure 2).

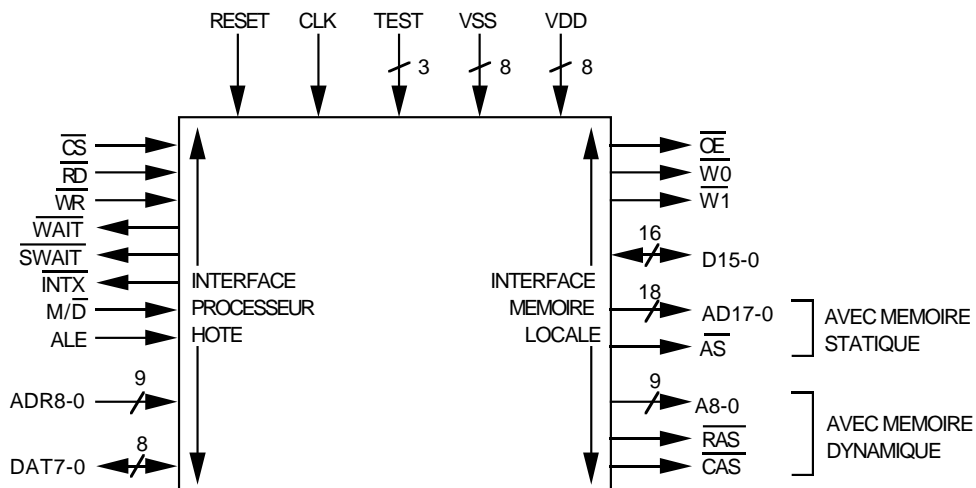


Figure 2: Présentation externe du processeur de comparaison dynamique

### *III.1. Interface avec le processeur hôte*

Le processeur hôte voit le  $\mu$ PCD comme une mémoire de 512 octets. Par des séquences de 1 à 7 accès virtuels en lecture ou en écriture, le processeur hôte peut accéder à tous les registres internes, à la mémoire de programme interne et à la mémoire de données externe (à travers le  $\mu$ PCD), il peut aussi contrôler l'exécution des programmes.

L'interface est physiquement constituée de 9 lignes d'adresses, 8 lignes de données et 8 lignes de contrôle. Les adresses et les données peuvent être multiplexées ou non et l'interface peut fonctionner avec un mode attente ou non. Le circuit peut donc être connecté simplement au bus local de n'importe quel microprocesseur standard.

### *III.2. Interface avec la mémoire locale*

Le  $\mu$ PCD peut se connecter directement avec n'importe quel type de boîtier mémoire standard  $256K \times 1$  statique ou  $256K \times 4$  dynamique. La configuration statique ou dynamique est programmable et les mêmes broches sont utilisées dans les deux cas.

L'interface est physiquement constituée de 18 lignes d'adresses, 16 lignes de données et 4 lignes de contrôle pour une mémoire statique et de 9 lignes d'adresses, 16 lignes de données et 5 lignes de contrôle pour une mémoire dynamique.

### *III.3. Autres Signaux*

Le circuit comprend également 16 lignes d'alimentation (8 VDD et 8 VSS), 1 ligne de RESET, 1 ligne pour l'horloge et 3 lignes réservées pour le test.

### *III.4 Architecture interne*

Le  $\mu$ PCD comprend 7 blocs fonctionnels principaux qui sont (Figure 3):

- Une unité de calcul de distances, spécialisée pour le calcul de la distance L1 entre deux vecteurs, comprenant une mémoire de 8 mots de 16 bits pour le stockage du vecteur test courant, un opérateur spécialisé et un accumulateur sur 12 bits.
- Une unité arithmétique générale, dédiée au calcul des distances cumulées et à la propagation des étiquettes, comprenant 32 registres de travail sur 16 bits et un opérateur capable d'effectuer des additions, des soustractions, des additions saturées, des décalages et des comparaisons.
- Une unité de calcul d'adresses, comprenant des registres bases et indexs, un opérateur spécialisé et un contrôleur de rafraîchissement pour la mémoire dynamique externe.
- Un séquenceur, comprenant un compteur ordinal et une pile de sous programme cablée à 4 niveaux.
- Une mémoire de programme de 640 mots de 24 bits.
- Une interface avec le processeur hôte.
- Un bloc pour le décodage des instructions et la logique de contrôle.

La Figure 4 montre la photo du circuit et la Figure 5 son plan de masse.

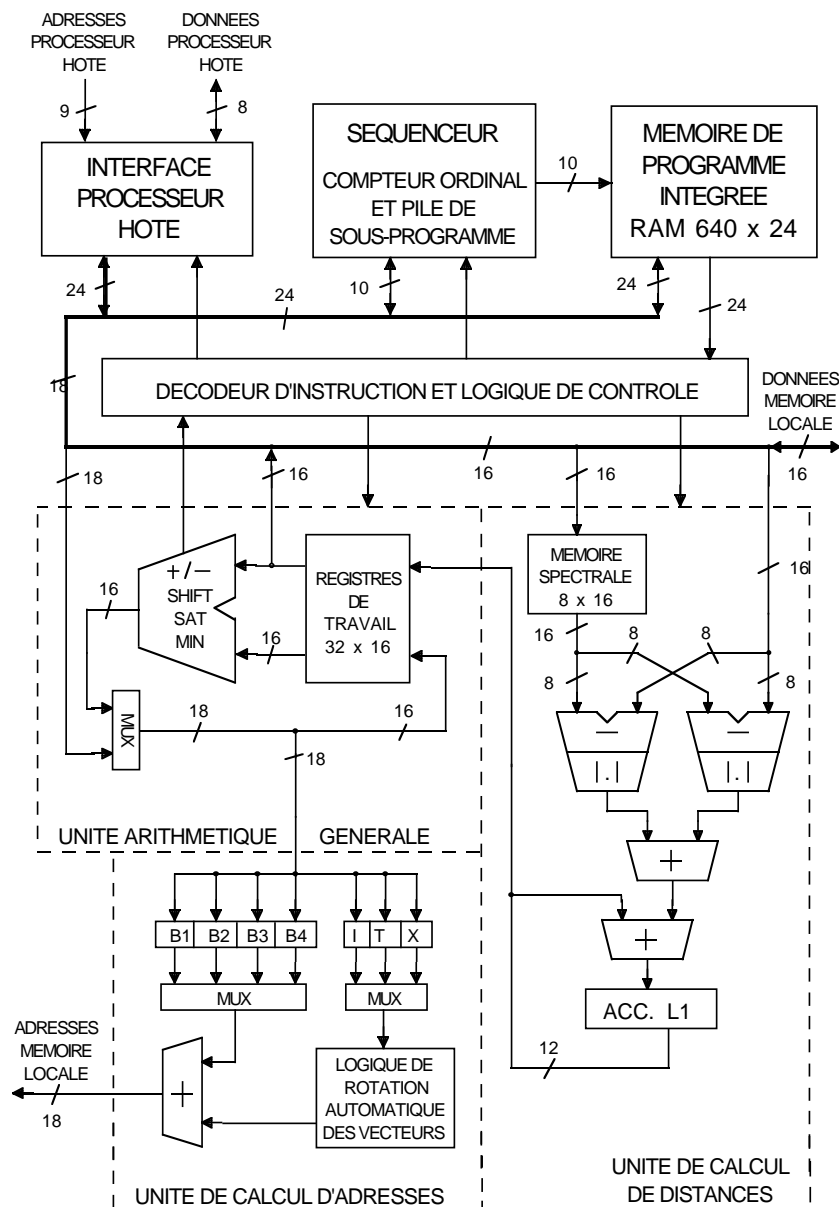


Figure 3: Architecture du processeur de comparaison dynamique

#### IV. Fonctionnement en multiprocesseur

Plusieurs  $\mu$ PCDs peuvent être placés en parallèle dans un système. Le moyen le plus simple est de les connecter tous sur le bus local du processeur hôte comme indiqué sur la Figure 1. Ils apparaîtront chacun à celui-ci comme une mémoire de 512 octets.

Cette organisation convient si les  $\mu$ PCDs n'ont pas de données à échanger entre eux (cas des mots isolés) ou très peu (mots enchaînés sans syntaxe ou détection de mots). Dans ce cas, les données à échanger peuvent l'être sous le contrôle du processeur hôte.

Dans le cas de reconnaissance de mots ou unités phonétiques enchaînés avec syntaxe (le plus intéressant, car nécessaire pour la reconnaissance de parole continue), le volume de données à échanger est plusieurs centaines de fois plus important. Pour résoudre ce problème, nous avons développé une possibilité de fonctionnement avec une mémoire commune comme indiqué sur la Figure 6.

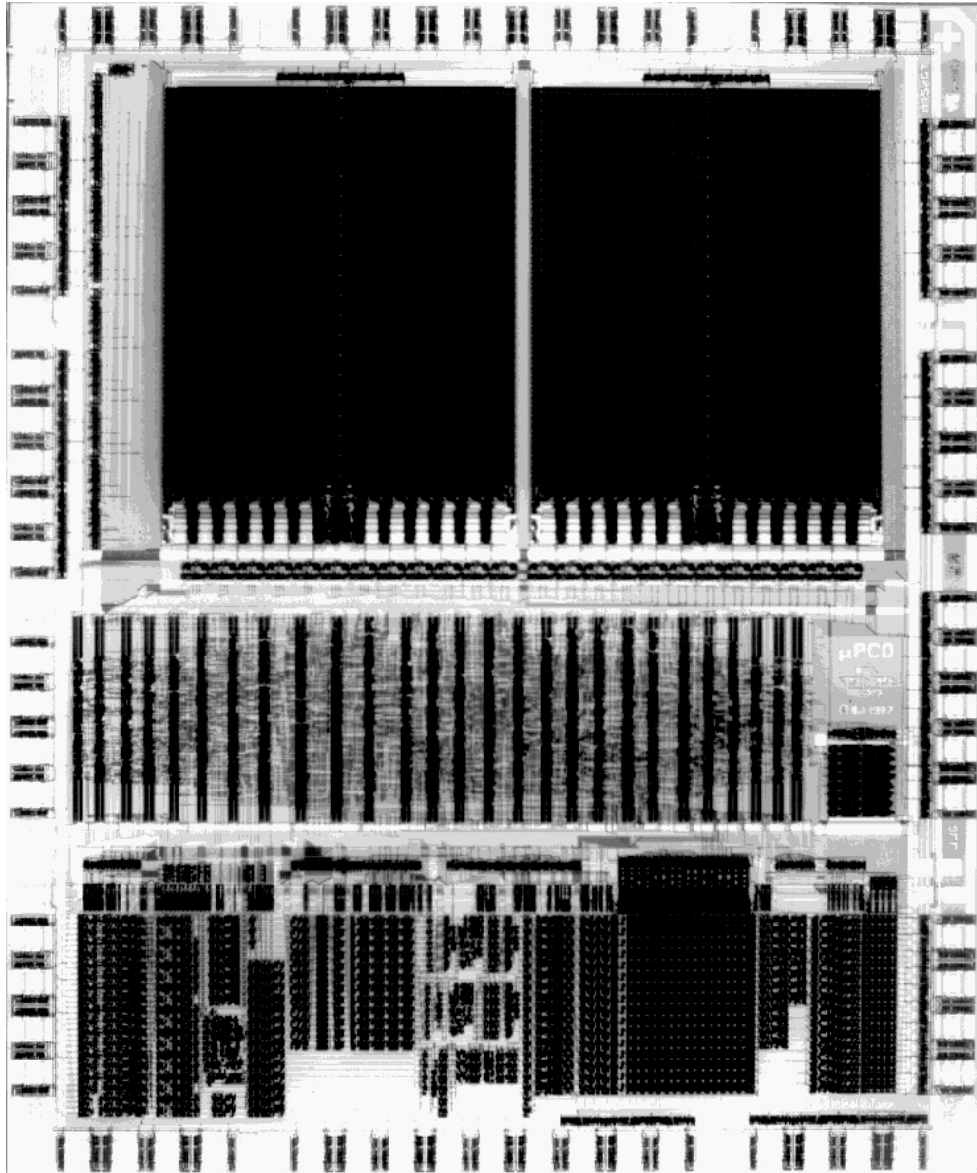
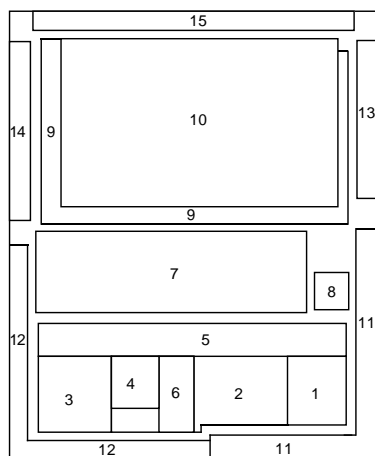


Figure 4: Photo du microprocesseur de comparaison dynamique



1. Unité de calcul de distances
2. Unité arithmétique générale
3. Unité de calcul d'adresses
4. Compteur ordinal et pile de sous-programme
5. Décodeurs d'adresses des bancs de registres et amplis de commandes
6. Unité de traitement des adresses des bancs de registres
7. Décodeur d'instructions et logique de contrôle
8. Générateur d'horloges
9. Interface avec le processeur hôte
10. Mémoire de programme: 2 RAMs  $640 \times 12$
11. Plots d'entrée-sortie des données de la mémoire locale
12. Plots de sortie des adresses de la mémoire locale
13. Plots d'entrée-sortie des données du processeur hôte
14. Plots d'entrée des adresses du processeur hôte
15. Plots d'entrée ou sortie horloge et signaux de contrôle

Figure 5: Plan de masse du processeur de comparaison dynamique

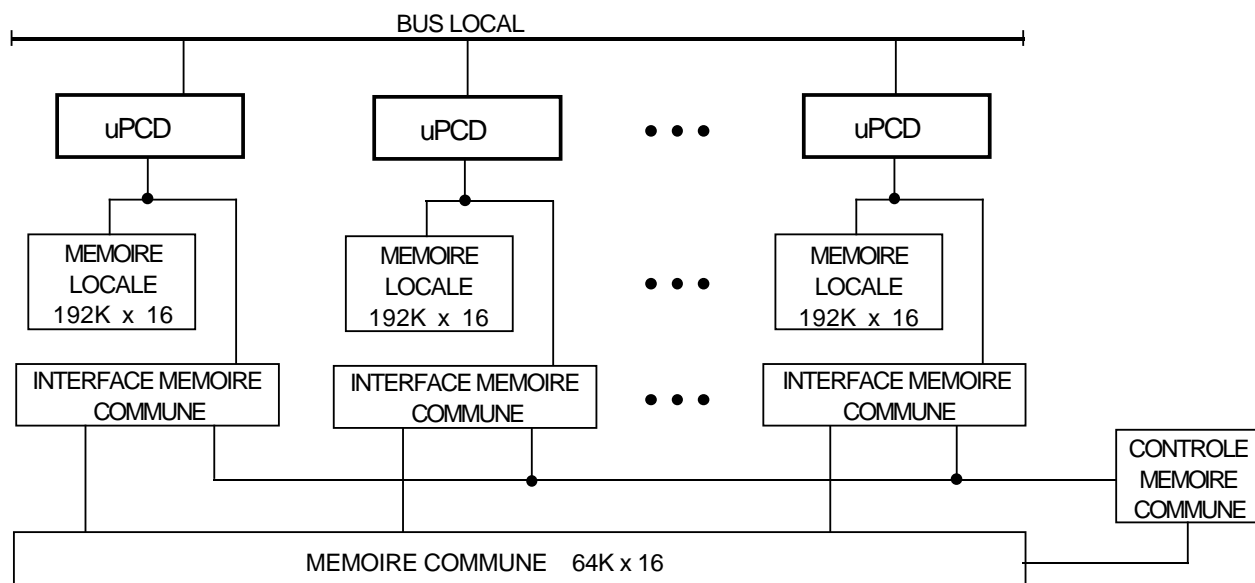


Figure 6: Organisation multiprocesseur avec mémoire commune

La mémoire commune est accédée de manière circulaire par tous les  $\mu$ PCDs. Pour  $N$  processeurs, la bande passante entre un  $\mu$ PCD la mémoire commune est de  $1/(N + 2)$  fois celle de la mémoire locale (20 Moctets/s). La bande passante totale de la mémoire commune est de  $N/(N + 2)$  fois celle d'une mémoire locale et d'environ 50 fois celle des transferts contrôlés par le processeur hôte. Cette bande passante est suffisantes pour des systèmes comportant jusqu'à 16  $\mu$ PCDs.

L'interface avec la mémoire commune nécessite peu de logique externe: 5 circuits SSI et MSI standards par  $\mu$ PCD, plus 8 PALs et circuits SSI standard pour le contrôle de la mémoire commune.

## V. Méthodologie de conception

Toutes les unités de traitement ont été regroupées dans un bloc chemin de données unique avec des sous-sections sur 12, 16 et 18 bits. Des unités spécialisées pour les manipulations des adresses des bancs de registres, sur 3, 4, 5 et 6 bits ont été assemblées en parallèle dans une section sur 18 bits. Ce chemin de données a été entièrement personnalisé et il est constitué d'environ 150 cellules de base qui sont elles-mêmes dérivées de seulement une quarantaine de cellules réellement dessinées et optimisées.

La partie contrôle a été réalisée en un seul bloc par un compilateur logique à partir d'un schéma logique hiérarchique. Il comprend un mélange de cellules standard et personnalisées.

Le cycle machine correspond à deux périodes d'horloge et est divisé en quatre phases. Huit horloges internes différentes, combinant une ou deux des quatre phases de base sont utilisées. Ces horloges sont générées par un bloc spécialisé, entièrement personnalisé. Ce générateur a été conçu sur une organisation en tranche de bits et sur un modèle qui permet de programmer une horloge quelconque par combinaison des phases de base et d'ajuster la sortance de chaque horloge en fonction de la charge qu'elle est supposée commander.

La mémoire de programme a été développée avec un compilateur et est divisée en 2 parties de 640 mots de 12 bit de façon à bien s'intégrer dans la partie supérieure du cœur du circuit.

La couronne de plots est réalisée avec des plots standard.

Toutes les cellules critiques ont été simulées électriquement. Le circuit complet a été simulé sur un simulateur logique avec temps de propagation pour en vérifier la fonctionnalité et identifier les chaînes longues. La structure simulée comprenait simultanément des sous-blocs provenant de schémas logiques, de dessins physiques extraits ou de modèles fonctionnels.

## VI. Testabilité

Lorsque le circuit est installé dans un système, le processeur hôte peut accéder à travers son interface à tous les registres internes et à la mémoire de programme. Un mode d'exécution en pas à pas a également été implémenté. Ceci permet un test du circuit sur site commandé par le processeur hôte et offre en outre un environnement très convivial pour le développement du logiciel.

Pour les tests sur machine, le même principe est utilisé, mais afin de réduire le nombre de vecteurs de test, un mode d'accès directement sur 24 bits a été implémenté en utilisant les 16 lignes de données de l'interface mémoire locale pour compléter les 8 lignes de l'interface processeur hôte. L'accès aux registres et à la mémoire interne en mode test permet de diviser par quatre la taille du programme de test.

Dans les deux cas, l'accessibilité et le mode pas à pas permettent le test du circuit par une méthode équivalente au test non fonctionnel et notamment un taux de couverture très élevé (100 % pour tous les registres, les opérateurs et la mémoire de programme). Grâce à l'utilisation de cellules registres personnalisées à plusieurs entrées et sorties, le matériel supplémentaire ajouté pour le test n'occupe pas plus de 5 % de la surface totale du circuit.

## VII. Caractéristiques et performances

Le tableau suivant résume les caractéristiques du  $\mu$ PCD:

Technologie	CMOS 2 $\mu$ m
Taille de la puce	7169 $\mu$ m $\times$ 8617 $\mu$ m
Nombre de transistors	127309
Boîtier	84 broches PGA ou PLCC
Fréquence d'horloge	20 MHz
Mémoire de programme	640 $\times$ 24
Mémoire de données	256K $\times$ 16
Performance	10 Mips

Le  $\mu$ PCD peut traiter un point de programmation dynamique général en 25 cycles machine, soit 2.5  $\mu$ s, et une version simplifiée en 12 cycles machine, soit 1.2  $\mu$ s, ce qui lui permet d'en calculer de 300000 à 600000 par seconde avec les calculs auxiliaires correspondants.

La taille de la mémoire de programme, de 640 instructions permet l'implémentation d'une grande variété d'algorithmes de reconnaissance vocale et d'équations de programmation dynamique.

Un système de reconnaissance sur deux cartes au format PC, intégrant un seul  $\mu$ PCD, et un ensemble logiciel, appelé AMADEUS, ont été développés par le LIMSI-CNRS et VECSYS SA.

Ce système peut reconnaître en temps réel 1000 mots isolés avec un algorithme basé uniquement sur la programmation dynamique. En utilisant en plus des techniques de reconnaissance hiérarchique, de restriction de domaine et d'élagage, il est possible d'étendre la taille du vocabulaire jusqu'à 5000 mots en maintenant le temps de réponse en dessous de 500 ms. De la même façon, le système peut reconnaître de 100 à 500 mots isolés en multilocuteur avec en moyenne 10 références par mot.

Le système peut également reconnaître 300 mots enchaînés avec uniquement la programmation dynamique et 600 mots avec élagage.

En mots isolés comme en mots enchaînés, il est possible de définir des contraintes syntaxiques, celles-ci sont représentées en mémoire locale et le  $\mu$ PCD ne reconnaît que des séquences vérifiant cette syntaxe. Ceci permet à la fois de réduire le volume de calcul et d'améliorer la qualité de la reconnaissance.

Une carte au format PC comprenant de 4 à 8  $\mu$ PCDs avec une mémoire commune est en cours de développement. Elle devrait permettre la reconnaissance de parole continue avec un algorithme basé sur la reconnaissance de 5000 syllabes ou mots courts enchaînés avec contraintes syntaxiques.

## VIII. Conclusion

Nous avons présenté un microprocesseur de comparaison dynamique avec un architecture parallèle et pipe-line. Ce circuit, réalisé en technologie CMOS 2  $\mu$ m, intègre 127309 transistors dans 60 mm<sup>2</sup>, est délivré dans un boîtier à 84 broches PGA ou PLCC et a une puissance de 10 Mips à 20 MHz.

Comparativement aux autres réalisations dans le domaine, l'originalité de ce processeur est qu'il offre simultanément: une flexibilité totale, une très grande puissance de calcul, une intégration très simple dans un système compact et une possibilité de fonctionnement efficace en multiprocesseur.

Le circuit a été monté dans un système et a permis l'implémentation d'une large gamme d'algorithmes de reconnaissance. Il a été vérifié qu'il permet un gain qualitatif en performance par rapport aux systèmes classiques, aussi bien en taille de vocabulaire qu'en qualité de reconnaissance. L'utilisation de ce circuit dans des systèmes multiprocesseurs permettra d'aborder la reconnaissance de la parole continue. Pour toutes ces applications, outre la puissance de calcul, la souplesse d'emploi et la capacité d'effectuer des calculs complémentaires de ceux de programmation dynamique ont joué un rôle essentiel en permettant, par l'emploi de techniques d'optimisation sophistiquées, de multiplier la puissance brute du circuit par un facteur important.

## Remerciements

Le développement du circuit a été subventionné par France Télécom (contrat DAI n° 86.35.053) et a été effectué avec la collaboration des sociétés Bull SA, VECSYS SA et VLSI Technology. Les auteurs souhaitent remercier tous les membres de ces sociétés qui les ont aidés à réussir ce développement.

## Références

- [1] R. Bellman, "Dynamic Programming", *Princeton University Press*, 1957.
- [2] H. Sakoe & S. Chiba, "Dynamic Programming optimisation for spoken word recognition", *IEEE Trans. on Acoust., Speech, and Signal Proc.*, Vol. ASSP-26, p43, February 1978.
- [3] H. Sakoe, "Two Level DP-Matching - A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition", *IEEE Trans. on Acoust., Speech, and Signal Proc.*, Vol. ASSP-27, p588-595, December 1979.
- [4] J.S. Bridle, M.D. Brown & R.M. Chamberlain, "An Algorithm for Connected Word Recognition", *Proc IEEE ICASSP-82*, p899-902, May 1982.
- [5] Ishizuka et al., "A microprocessor for speech recognition", *Proc. IEEE ICASSP-83*, p.503, April 1983.
- [6] R. Kavaler et al., "A Dynamic Time Warp Integrated Circuit for a 1000-Word Speech Recognition System", *IEEE Journal of Solid-State Circuits*, Vol SC-22, NO 1, February 1987, pp 3-14.
- [7] Y. Kitazume et al., "LSI Implementation of a Pattern Matching Algorithm for Speech Recognition", *IEEE Trans. Acoust., Speech, and Signal Proc.*, Vol. ASSP-33 p.386, February 1985.
- [8] S.C. Glinski et al., "A Processor for Graph Search Algorithm", *ISSCC Dig. Tech. Papers*, 1987, p.162.
- [9] W. Drews et al., "A 1000 Word Speech Recognition System Using A Special Purpose CMOS Processor", *European Conf. on Speech Technology* Vol.2 pp 218-221, 1987.