L'ABC DE LA MICRO-INFORMATIQUE

LES LCA

Le sigle qui sert de sous-titre à notre article du jour est certainement inconnu de la majorité d'entre vous. C'est bien normal car, comme nous vous l'avons annoncé le mois dernier, les circuits dont nous allons parler aujourd'hui sont des produits très récents et dont la commercialisation réelle en France ne date que du début de cette année. Nous avons néanmoins estimé utile de vous les présenter en détail car d'une part ils sont promis à un brillant avenir dans l'industrie, et d'autre part il est possible de les utiliser dans des réalisations d'amateurs, sous certaines conditions.

TOUJOURS PLUS

Nous avons vu dans nos précédents numéros que les PAL ou assimilés permettaient une réduction sensible du nombre de boîtiers logiques à utiliser dans un circuit donné. Cette réduction est sans cesse recherchée par les fabricants de matériels utilisant des circuits logiques car elle présente de nombreux avantages :

- Réduction de la taille globale du montage.
- Diminution du nombre de soudures nécessaires et, donc, augmentation de la fiabilité.
- Simplification du tracé du circuit imprimé.
- Diminution du nombre de références différentes de circuits à tenir en stock pour réaliser un montage donné.
- Evolution du produit ou modification (légère) de ses possibilités plus facile qu'avec des boîtiers logiques classiques.

Les PAL permettent, bien sûr, de satisfaire certaines de ces contraintes mais, comme nous l'avons vu sur quelques exemples, leur structure interne prédéfinie ne permet pas de tout remplacer par des PAL. En outre, leur « programmabilité » par l'utilisateur conduit à intégrer dans le boîtier des éléments qui ne servent qu'à cela et qui n'ont ensuite aucune utilité dans la fonction logique même du PAL.

Pour remédier à cela, et pour des ensembles logiques très complexes, les fabricants de matériels font donc appel à ce que l'on appelle des *Gate Arrays*, que l'on peut traduire en français par des réseaux de portes.

De tels circuits sont des boîtiers logiques qui contiennent un très grand nombre de portes élémentaires (jusqu'à plusieurs dizaines de milliers !) non interconnectées. L'utilisateur d'un tel boîtier, après avoir étudié la fonction logique qu'il veut réaliser, définit au moyen de programmes spéciaux les interconnexions à établir entre les portes pour parvenir à ses fins. Celles-ci sont alors faites, lors de la fabrication même du Gate Array, par le fabricant de circuits intégrés (un peu comme pour les ROM programmables par masque). Le Gate Array devient donc un circuit particulier dédié à l'application voulue par le client.

Il est évident que, du fait du procédé utilisé, et comme pour les ROM programmables par masque, de tels circuits ne peuvent être « programmés » que si une production en très grande série est envisagée. C'est le cas pour de très nombreux micro-ordinateurs qui utilisent intensivement de tels boîtiers. Le meilleur exemple en est peut-être le célèbre mais déjà ancien ZX-81, dont toute la logique était contenue dans un seul boîtier à 40 pattes, baptisé ULA par Sinclair (Uncommited Logic Array) et qui n'était autre qu'un (petit) Gate Array.

Ces problèmes de très grande série conduisent nombre de petits constructeurs à ignorer les Gate Arrays, puisque leur utilisation ne peut pas être rentable pour des productions de quelques centaines à quelques milliers de pièces. En outre, le développement de tels

réseaux demande des connaissances assez poussées en matière de conception de circuits intégrés, même si les programmes prévus pour cela facilitent un peu les choses. Ces programmes eux-mêmes, d'ailleurs, coûtent fort cher et ne sont disponibles, pour l'instant, que sur de gros calculateurs (HP 9000, Vax ou appareils similaires), ce qui les met hors de portée de petites entreprises.

UNE SOLUTION INTERMEDIAIRE

Les LCA, ce qui signifie Logic Cell Array (ou réseau de cellules logiques), proposés pour la première fois par Xilinx et fabriqués maintenant en seconde source par MMI, représentent une solution intermédiaire entre les PAL et les Gate Arrays comme le montre le diagramme de la figure 1.
Ce graphique représente la complexité d'un boîtier en fonction de son temps de conception. La solution la plus rapide à concevoir, mais aussi la plus simple, fait appel à de la logique classique. Les PAL permettent une conception rapide (dix jours environ) mais ne peuvent contenir qu'une centaine de portes au maximum, tandis que les Gate Arrays peuvent contenir jusqu'à 100 000 portes mais demandent près de mille jours de conception (il faut compter dans ce délai l'étude, la fabrication et le contrôle des prototypes, et la production en série des pièces). La solution « LCA » offre l'avantage d'un temps de conception du même ordre de grandeur que celui des PAL mais, par contre, permet d'intégrer de 1 000 à 8 000 portes environ.
Il existe d'autres données importantes, que le graphique ne met pas en évidence, et qui concernent l'utilisation des ressources contenues dans un circuit programmable. Dans un PAL, du fait de la structure relativement rigide du réseau et des entrées/sorties, diverses études ont montré que l'on n'utilisait en général que 15 à 20 % des fonctions disponibles. Dans un Gate Array, par contre, une utilisation de 80 à 90 % des fonctions est souvent atteinte en raison de la plus grande souplesse de programmation et d'interconnexion. Dans un LCA, un taux d'utilisation de 60 à 70 % peut être atteint sans difficulté.
Enfin, si les PAL sont presque essentiellement des circuits bipolaires (sauf les EPLD, dont nous ne parlerons pas ici) et, de ce fait, consomment beaucoup d'énergie, les Gate Arrays, comme les LCA, sont des circuits C-MOS dont la consommation est donc très faible. Malgré cela, des vitesses de transfert de l'ordre de 15 ns par porte peuvent être atteintes sans difficulté. Tout ceci pour vous dire qu'un LCA

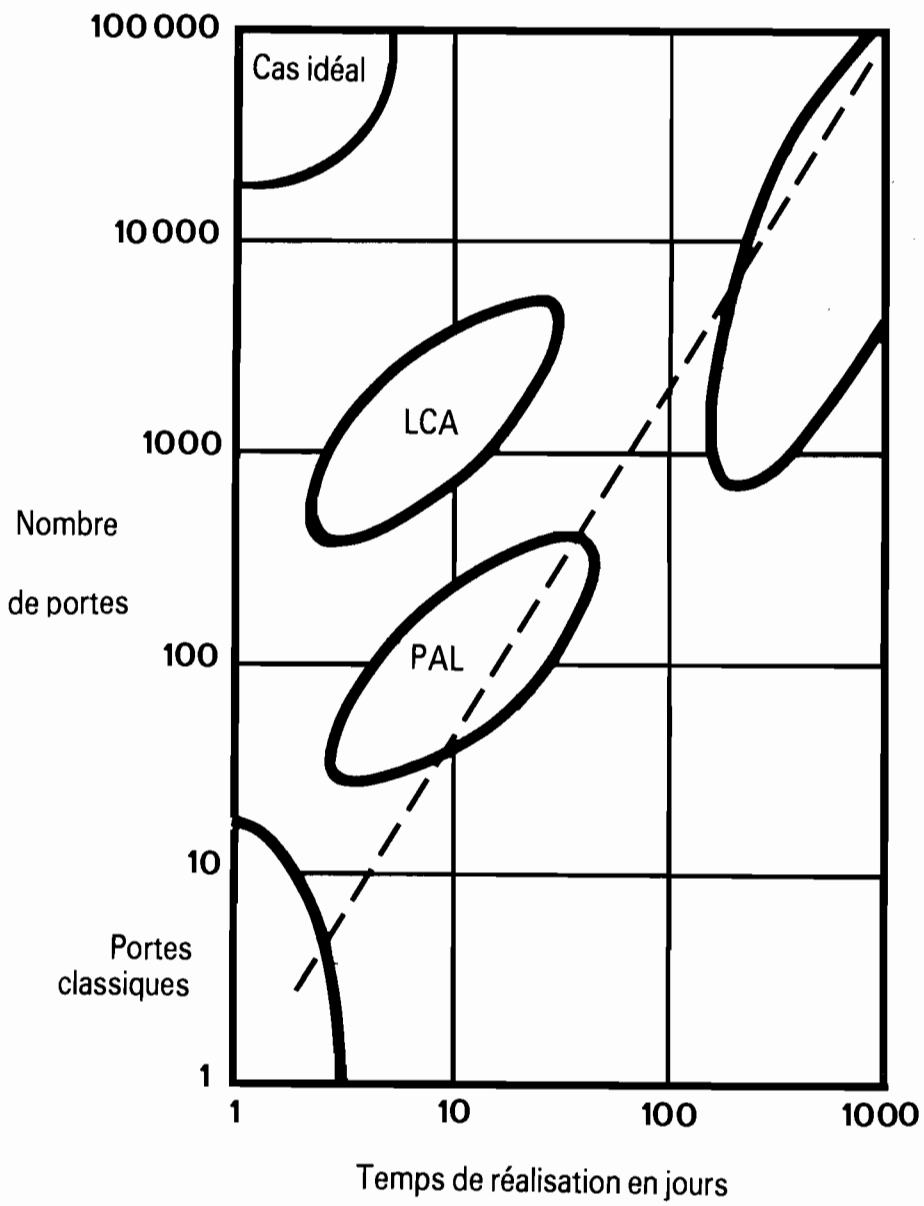


Fig. 1. – Comparaison des différentes familles logiques programmables.

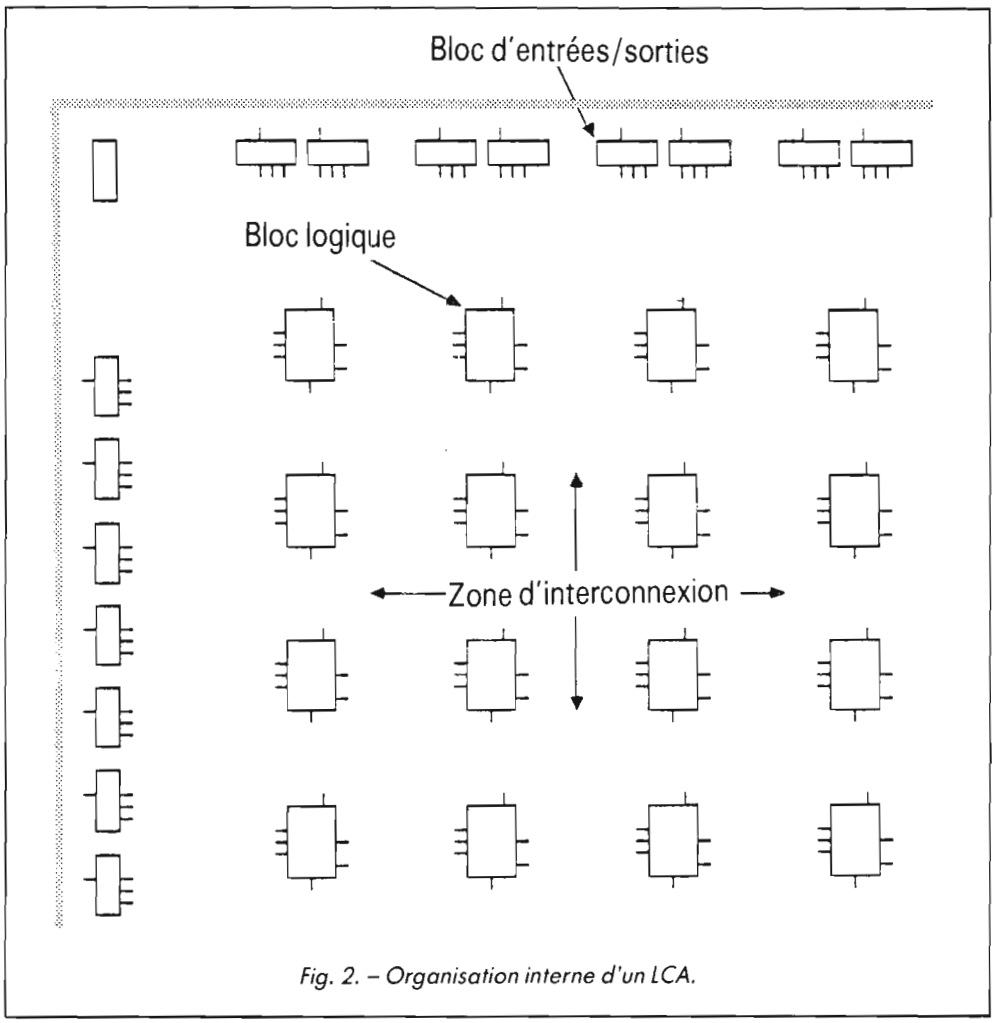
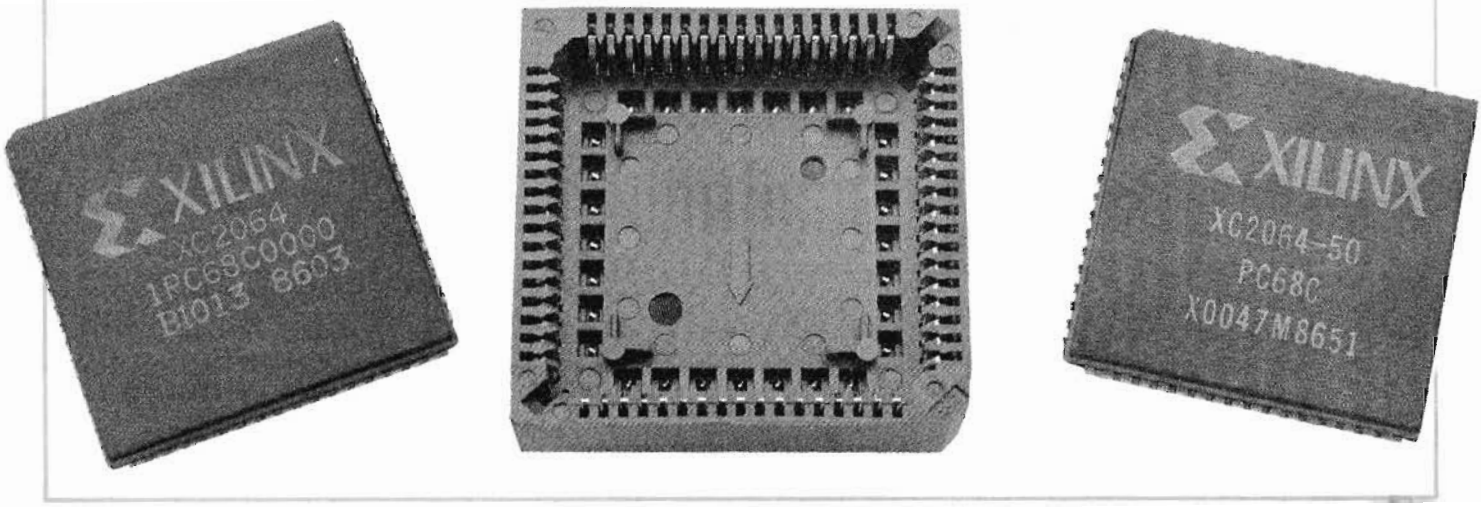


Fig. 2. – Organisation interne d'un LCA.

n'est pas un « super » PAL mais bien au contraire un produit différent, comme nous allons le voir maintenant.

STRUCTURE D'UN LCA

Un LCA est un circuit C-MOS qui contient un certain nombre de « blocs logiques » (logic block) et un certain nombre de blocs d'entrées/sorties (I/O block) non interconnectés, comme vous pouvez le constater à l'examen de la figure 2. L'utilisateur du LCA définit les connexions à établir entre ces blocs en fonction du schéma global à réaliser et, donc, compte tenu de la fonction logique à obtenir. Ces connexions sont alors programmées dans le LCA par écriture dans une mémoire interne. En effet, comme nous allons le voir dans un instant, un grand nombre de lignes traversent le LCA de part en part et passent dans des matrices de connexion. Ces matrices sont des transistors MOS pilotés par le contenu d'une mémoire RAM qui se trouve « sous » la partie logique de la puce du LCA. Pour configurer un LCA, il suffit donc de charger cette RAM avec les informations adéqua-



Deux LCA en boîtiers PLCC avec le support adéquat.

tes pour que les connexions désirées soient réalisées. Comme cette configuration est en RAM, elle est modifiable lorsque vous le souhaitez et sans nécessiter d'équipement spécial ou de passage chez le fabricant du circuit intégré. Ces grands principes étant posés, voyons d'un peu plus près comment tout cela est agencé.

Les blocs d'entrées/sorties ont tous l'aspect schématisé figure 3. Chaque patte est bidirectionnelle, et dispose donc d'un buffer d'entrée et d'un buffer de sortie. En sortie, le buffer peut être toujours ON auquel cas la sortie est toujours active, en trois états auquel cas la sortie est commandée par l'état de la ligne TS, ou toujours OFF auquel cas la sortie est inutilisée et la patte ne sert qu'en entrée. Le choix se fait par le multiplexeur qui commande le buffer et qui est, évidemment, programmable comme tout le reste. L'entrée peut être directe ou « latchée » par la bascule D commandée par l'horloge I/O Clock. Cette horloge est commune à chaque ligne de blocs d'entrées/sorties. Le choix entre les deux modes se fait, là aussi, par le multiplexeur programmable. Comble de raffinement, le niveau logique d'entrée est programmable pour être compatible TTL (1,4 V) ou compatible C-MOS alimenté sous 5 V (2,2 V).

Un circuit comme le XC 2064 contient 58 blocs de ce type, alors que le XC 2018 en contient 74. Cela laisse pas mal de liberté de conception...

Les blocs logiques internes, quant à eux, revêtent l'aspect indiqué figure 4 (attention à ne pas confondre les multiplexeurs programmables avec des portes logiques !). Ils disposent de quatre entrées logiques à usage général (A, B, C, D), d'une entrée d'horloge et de deux sorties. Le bloc marqué *comb. logic* permet de réaliser n'importe quelle fonction logique combinatoire de

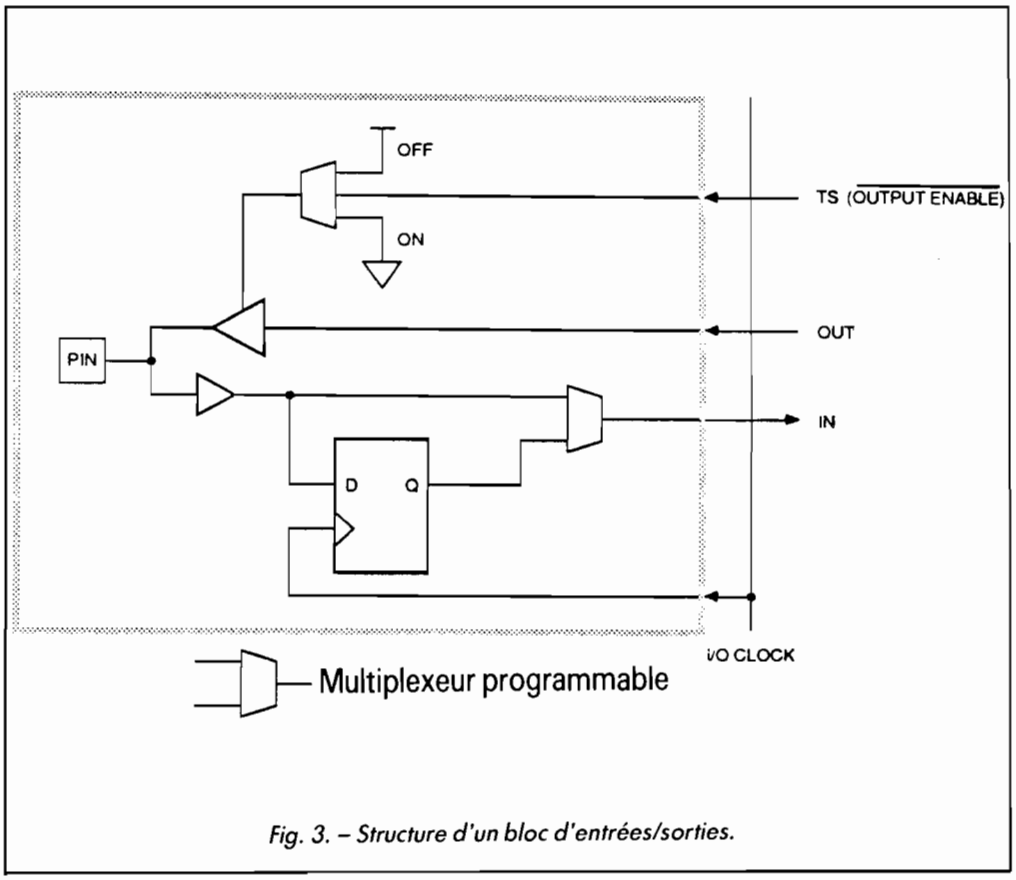


Fig. 3. – Structure d'un bloc d'entrées/sorties.

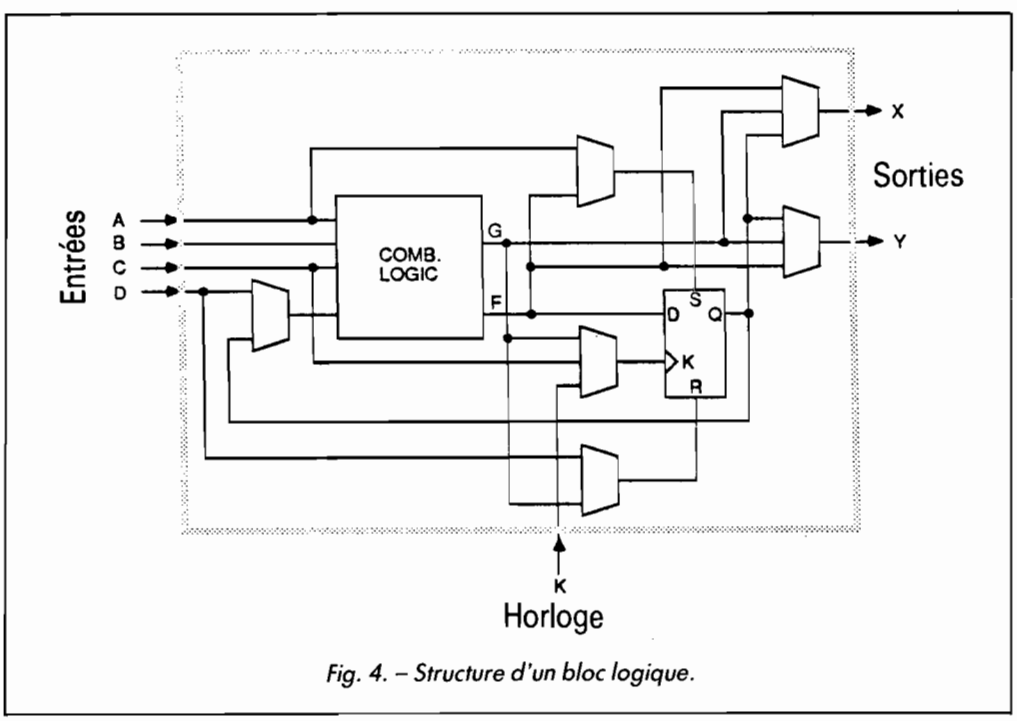


Fig. 4. – Structure d'un bloc logique.

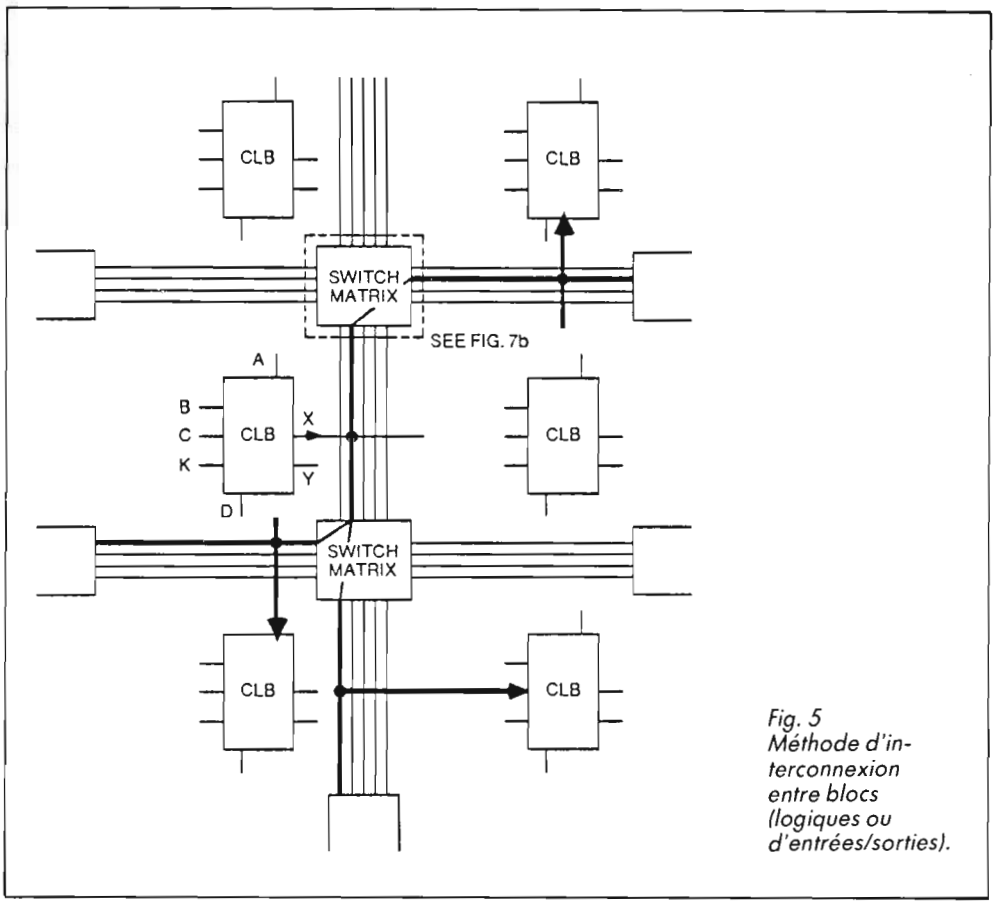
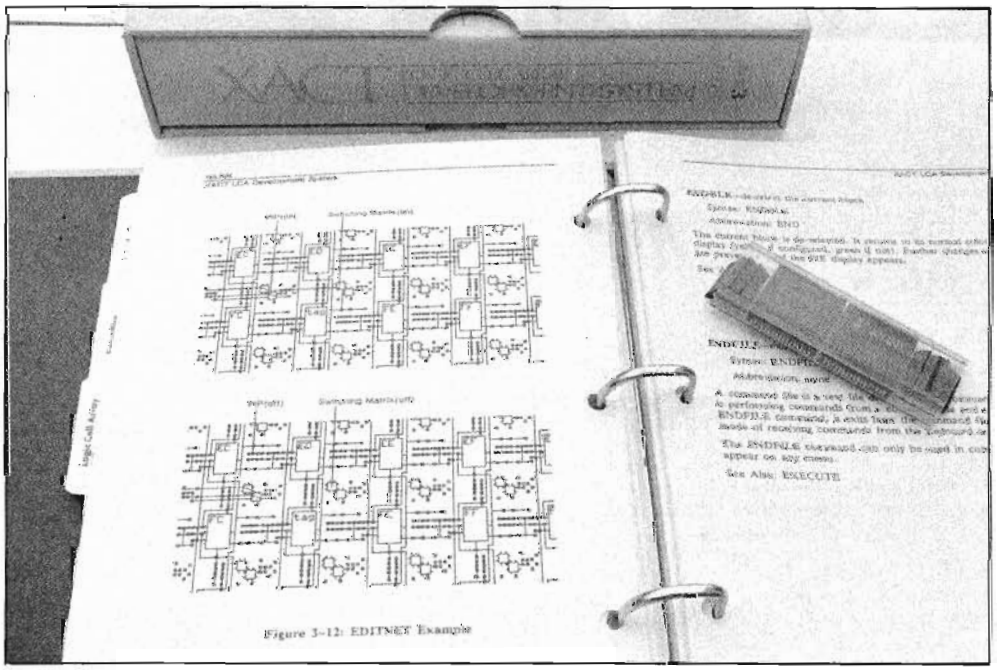


Fig. 5 Méthode d'interconnexion entre blocs (logiques ou d'entrées/sorties).



quatre variables. Ces dernières peuvent être les entrées logiques A à D, mais également la sortie Q de la bascule D contenue dans le bloc.
Le résultat de cette fonction logique peut être disponible directement en sortie X ou Y, mais peut aussi être appliqué à la bascule D afin de le rendre synchrone de l'horloge K. Cette bascule peut elle-même être programmée pour être sensible à un flanc ou à un niveau appliqué sur son entrée D. Enfin, il est également possible de programmer le front actif de l'horloge (montant ou descendant). Ces possibilités de programmation font évidemment économiser un nombre considérable de circuits, même si certains ne sont que de simples inverseurs.
Un tel bloc permet donc de réaliser déjà bon nombre de fonctions logiques mais, lorsque nous vous aurons dit qu'un XC 2064 en contient 64 et qu'un XC 2018 en contient 100, vous commencerez à avoir une meilleure idée de ce que peut faire un « simple » LCA.
Pour relier tout cela, un certain nombre de lignes sont à votre disposition, comme schématisé figure 5. Les différents blocs logiques peuvent se raccorder directement sur ces lignes qui se croisent au niveau de matrices de commutation (switching matrix). Ces matrices ne sont rien d'autre que les transistors à effet de champ dont nous avons parlé ci-avant ; transistors qui peuvent établir les connexions de votre choix entre les lignes.
Certains signaux ayant à transiter dans tout le LCA (signal d'horloge ou de validation commun par exemple), des lignes ne passant pas par ces matrices sont également disponibles, mais en moins grand nombre bien sûr, comme schématisé figure 6.

Le logiciel XACT permet une programmation facile à partir d'un compatible IBM PC.

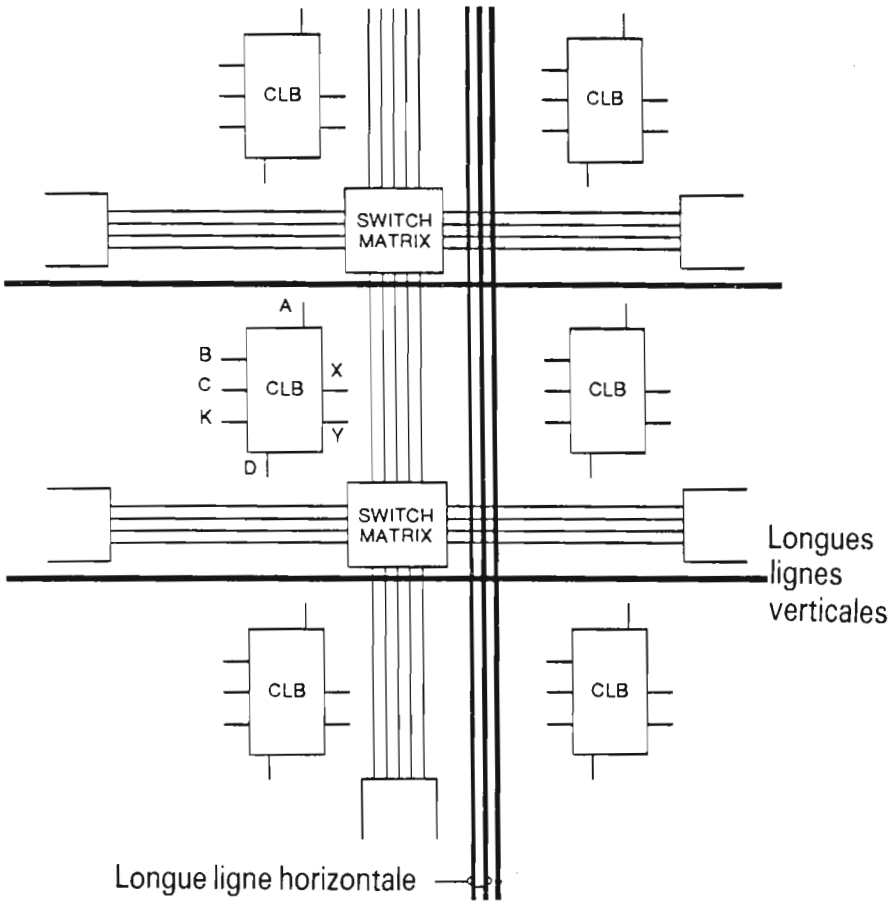


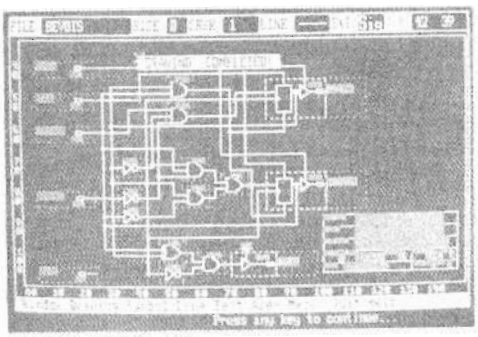
Fig. 6. – Des « longues lignes » sont disponibles pour les signaux qui doivent traverser tout le boîtier.

matiquement les informations de programmation du LCA sélectionné. Si, du fait des choix que vous avez réalisés, les interconnexions ne peuvent être établies, ils permettent de déplacer des portions entières de schémas dans le LCA lui-même. La version la plus complète de ces programmes est capable de simuler le comportement du LCA tel que vous l'avez défini, et vous permet ainsi de vérifier sa conception.

Lorsque tout est au point, ce programme vous permet de programmer, avec votre compatible PC ou avec n'importe quel programmateur de mémoire (avec lequel il sait dialoguer), une PROM ou une UV-PROM qui se chargera de configurer automatiquement votre LCA lors de la mise sous tension de ce dernier. En effet, comme le montre la figure 7, et du fait que les commandes des matrices de commutation du LCA sont contenues en RAM, il faut recharger ces dernières à chaque mise sous tension. Cela se fait en quelques millisecondes par recopie de la PROM de configuration au moyen du montage fort simple de la figure 7.

A QUEL PRIX ?

Bien que d'introduction récente sur le marché, un LCA n'est pas cher eu égard au nombre de boîtiers qu'il permet de remplacer. Pour fixer les idées, un XC 2064, c'est-à-dire un LCA avec 64 blocs logiques et 58 blocs d'entrées/ sorties comportant 1 200 portes en interne, est vendu, à l'unité, aux environs de 250 F. Si vous regardez la figure 8, qui représente le schéma d'un buffer d'imprimante pour microprocesseur type 6800 ou 6809, et si vous tenez compte du fait que son homologue à circuits conventionnels nécessitait 36 boîtiers TTL classiques, vous mesurez la simplification apportée par l'usage de tels composants.



Exemple de saisie de schéma sur compatible IBM PC en vue de la programmation d'un LCA.

LA PROGRAMMATION

Vous vous doutez bien que, vu la profusion de possibilités offertes, des moyens de programmation performants doivent être utilisés pour exploiter efficacement un LCA. Ces moyens sont en fait un programme, ou plus exactement un ensemble de programmes, disponibles sur machines IBM PC ou compatibles. Ils autorisent la saisie de schémas logiques ou d'équations, ou encore de diagrammes de Karnaugh, et établissent auto-

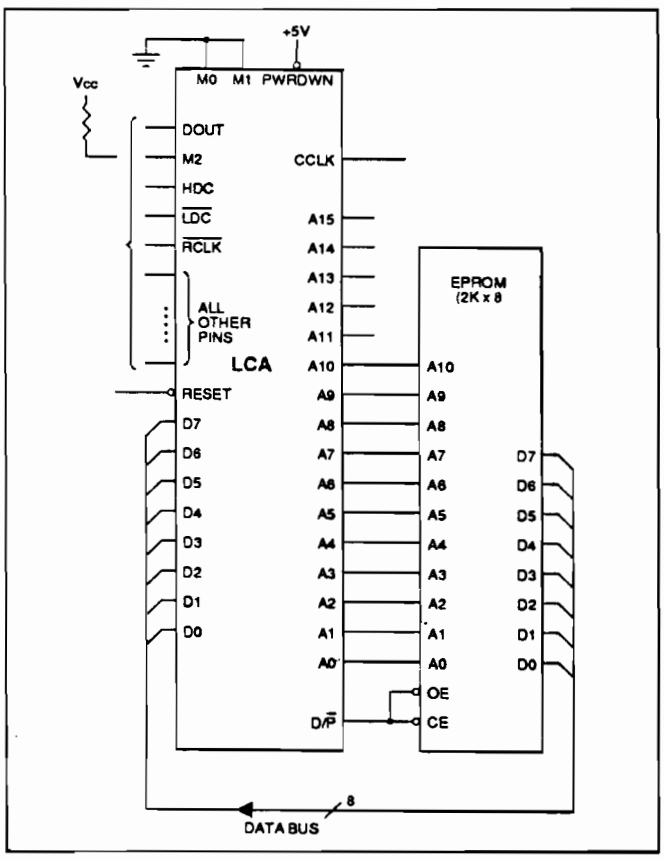


Fig. 7. – Méthode de programmation d'un LCA par recopie du contenu d'une PROM.

Bien sûr, il faut un IBM PC ou compatible pour les programmer et, surtout, il faut le programme adéquat. Ce dernier est vendu à l'heure actuelle aux environs de 25 000 F. Un tel prix, qui peut sembler élevé aux amateurs que vous êtes, est remarquablement bas vu les performances du produit. L'investissement IBM PC + programme est de toute façon plus de cent fois inférieur à celui qu'il faut consentir pour pouvoir faire des Gate Arrays classiques.
Le LCA est donc la solution idéale pour le concepteur de circuits logiques un tant soit peu complexes qui souhaite réduire les coûts de fabrication, diminuer l'encombrement du matériel et en augmenter la fiabilité. Le développement spectaculaire rencontré par ces produits depuis leur mise sur le marché en est la meilleure preuve. L'auteur de ces lignes ne désespère d'ailleurs pas de vous proposer une réalisation à base de LCA dans quelque temps...

CONCLUSION

Nous en resterons là pour cette présentation des divers circuits logiques programmables. Il y aurait encore beaucoup à dire, bien sûr. Mais, dans le cadre de cette initiation à la micro-informatique, nous pensons avoir assez abordé ce sujet pour pouvoir, dès le mois prochain, nous consacrer à autre chose.

C. TAVERNIER

Nota : Les figures qui illustrent cet article sont extraites du manuel technique de présentation des LCA édité par Xilinx.

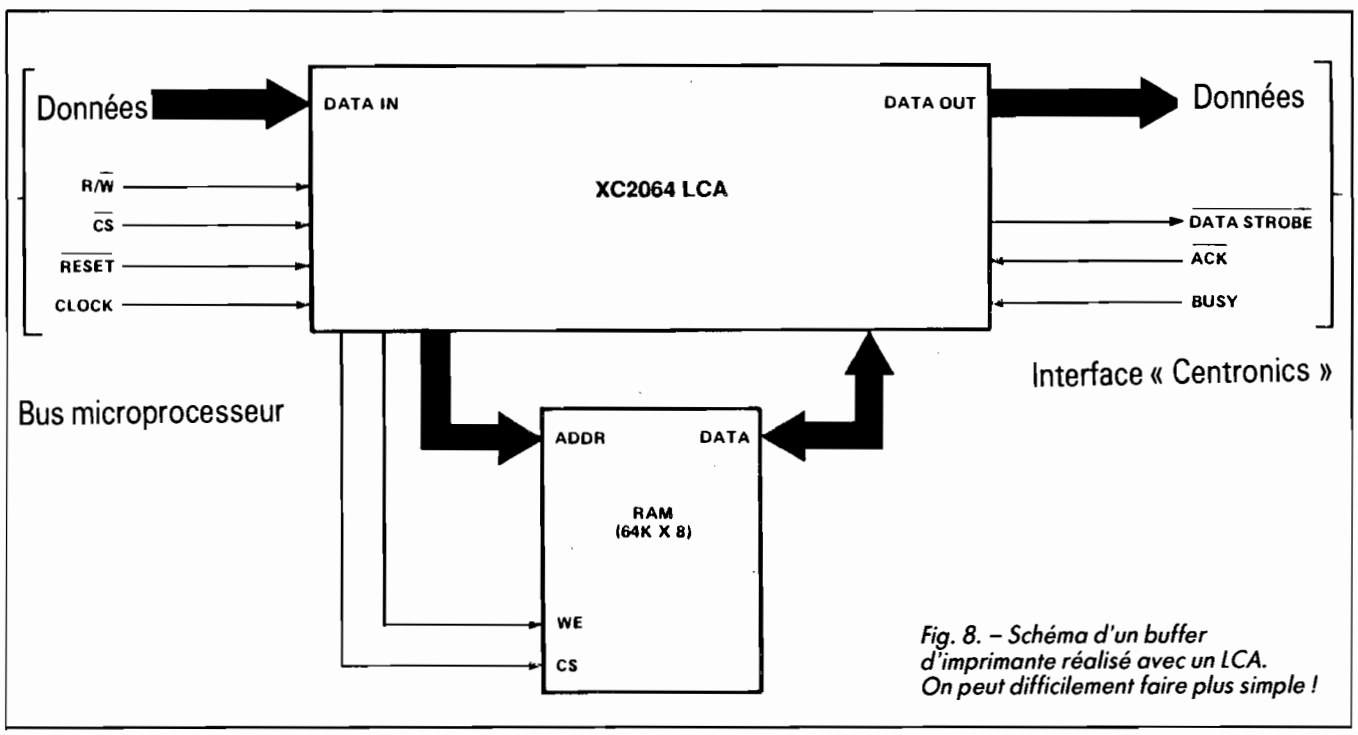


Fig. 8. – Schéma d'un buffer d'imprimante réalisé avec un LCA. On peut difficilement faire plus simple !