



**HAL**  
open science

## Vers un contrôleur d'écran graphique VLSI

Philippe Matherat

► **To cite this version:**

Philippe Matherat. Vers un contrôleur d'écran graphique VLSI. Revue des Sciences et Technologies de l'Information - Série TSI : Technique et Science Informatiques, 1984, 3 (2), pp 129-141. <hal-00194017>

**HAL Id: hal-00194017**

**<https://hal.science/hal-00194017v1>**

Submitted on 5 Dec 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

VLSI  
architecture des ordinateurs  
circuits intégrés  
visualisation graphique

VLSI  
computer architecture  
integrated circuits  
graphic display

# Vers un contrôleur d'écran graphique VLSI

Towards a graphic video controller VLSI chip



Philippe MATHERAT

Chargé de recherches au CNRS, ERA 1042

Philippe Matherat a soutenu une thèse de 3<sup>e</sup> cycle en 1978 sur l'étude du circuit intégré EF 9365, premier circuit mondial contrôleur de visualisation graphique. Ses recherches actuelles portent sur l'étude des architectures de circuits VLSI, en particulier de visualisation graphique, dans le but de réaliser un micro-ordinateur VLSI.

École Normale Supérieure, laboratoire d'informatique expérimentale, 45, rue d'Ulm, 75230 Paris Cedex 05

## PRÉSENTATION

A l'heure où un Projet National français et deux projets de la CEE développent de nouveaux outils de CAO pour les circuits VLSI, il est paradoxal de constater que très rares sont les équipes de recherche qui, en France, se penchent sur la conception et l'architecture des circuits en question.

L'article de Philippe Matherat, présentant une activité de conception de circuit, nous permet de mieux comprendre en quoi consiste ce type de recherche : si le ton peut en apparaître parfois un peu vif, il traduit bien les préoccupations de ces chercheurs qui sont soumis à des contraintes très particulières et qui se posent des questions sur leur rôle dans le développement d'une micro-électronique française.

Encore ne faut-il pas oublier que l'architecture des VLSI englobe aussi des aspects plus macroscopiques, plus théoriques ou plus proches de l'algorithmique parallèle, et sur lesquels travaillent de nombreuses équipes universitaires.

Guy Mazaré

## COMMENTARY

*With a French National Project and two EEC programmes working on new CAD tools for VLSI circuits, it is something of a paradox that in France there are very few research teams studying the design and architecture of such circuits.*

*Philippe Matherat's article describes circuit design work, giving us a clearer idea of what this kind of research entails ; if the tone of his discussion is on occasion a little sharp, it accurately reflects the concerns of these scientists, who are subjected to particularly harsh constraints and are beginning to wonder what role the future of French micro-electronics holds for them.*

*It should also not be forgotten that VLSI architecture has aspects which are more macroscopic, more theoretical and closer to parallel algorithm work, on which a number of university teams are working.*

Guy Mazaré

## TABLE DES MATIÈRES

### Présentation

#### 1. Étudier les architectures des circuits intégrés à l'Université. Pourquoi et comment ?

- 1.1 Aspects économiques de l'intégration des circuits. Problèmes français
- 1.2 Rôle de la recherche universitaire. Différence entre la France et les États-Unis
- 1.3 Approche de l'équipe du laboratoire de l'ENS

#### 2. Étude d'une architecture de contrôleur d'écran graphique

- 2.1 Introduction
- 2.2 Rappels : architecture d'un contrôleur d'écran graphique
- 2.3 Étude des parties critiques

#### 3. Conclusion

#### Bibliographie

## Présentation

Le but de cet article est de préciser comment se posent, à notre avis, les problèmes de recherche dans l'intégration des circuits, de présenter notre méthodologie, et de l'illustrer par le cas concret de l'étude de circuits pour la visualisation graphique.

Le passé récent nous a en effet montré que si les aspects physiques et chimiques de l'intégration sont largement perçus par la communauté scientifique, si les problèmes d'étude des outils de dessin des masques commencent à être bien attaqués par quelques équipes, le formidable besoin d'étudier les architectures des circuits VLSI dans leurs fonctionnalités est moins bien compris.

Ces problèmes sont étudiés dans la première partie. La deuxième partie, elle, est consacrée à l'étude d'une architecture de contrôleur d'écrans graphiques. Cette partie est destinée à indiquer l'état d'avancement de notre projet. Nous verrons que l'intégration d'une famille de circuits est dès aujourd'hui techniquement et économiquement envisageable sous forme d'une famille de quatre VLSI permettant de réaliser une visu graphique performante en sept circuits intégrés, mais que les difficultés pour aboutir à cette intégration nous poussent plutôt à continuer cette étude vers des performances plus élevées.

### 1. Étudier les architectures des circuits intégrés à l'Université. Pourquoi et comment ?

Dans tout cet article, nous ne parlerons que des circuits intégrés digitaux. Les problèmes se posent assez différemment pour les circuits analogiques.

#### 1.1. ASPECTS ÉCONOMIQUES DE L'INTÉGRATION DES CIRCUITS. PROBLÈMES FRANÇAIS

##### 1.1.1. *L'intégration est autant un processus économique que technologique*

Si on regarde ce qu'apporte l'augmentation des densités d'intégration pour les diverses fonctions réalisées par les circuits actuels, on constate que très peu de ces fonctions demandent impérativement des densités LSI ou VLSI. C'est le cas des circuits pour lesquels l'encombrement et/ou le poids sont problématiques (circuits embarqués dans l'espace, « pacemakers », etc...), ou des circuits pour lesquels la vitesse de fonctionnement ne peut pas être atteinte autrement, ou pour lesquels la consommation électrique doit être faible (appareils fonctionnant sur piles ou cellules solaires).

Par contre la majorité des fonctions sont réalisables avec des circuits MSI et SSI. Dans ce cas, l'intégration LSI ou VLSI est éventuellement intéressante sur le plan économique puisque l'on a montré que le prix de revient d'un appareil électronique est davantage fonction du nombre de ses composants que de leur nature [Sutherland 77]. Mais cette intégration coûte fort cher et ne peut être décidée que sur des motifs de rentabilité liés au cycle de la figure 1 [Noyce 77].

Les évolutions des densités, des vitesses et des consommations sont des retombées de ce cycle, qui interviennent d'une part comme facteurs accélérateurs de celui-ci, et qui d'autre part rendent possible l'intégration de

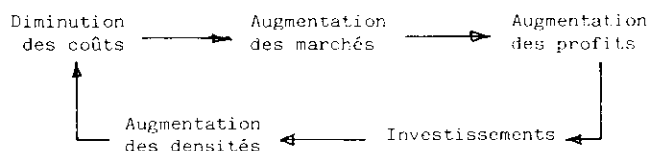


Figure 1. — Le cycle du progrès technologique.

fonctions nouvelles, à partir d'un certain seuil de performances.

Maîtriser les circuits intégrés, c'est autant maîtriser ces aspects économiques que les technologies de réalisation.

##### 1.1.2. *Potentialités en France*

Cette évolution, à la fois technique et économique, a été dominée par l'industrie électronique américaine. Notre pays ne présente que peu de retard sur le plan technique. Son réel problème se situe dans la maîtrise de ces cycles économiques. En effet, si l'on parcourt les catalogues de circuits LSI des grands constructeurs américains et japonais, (en éliminant les secondes sources et les circuits ne présentant que peu de variantes), on est frappé par la faiblesse du nombre de circuits réellement différents [EDN 83] :

- une dizaine de microprocesseurs,
- une dizaine de mémoires,
- une dizaine de contrôleurs de périphériques divers,
- quelques opérateurs arithmétiques,
- quelques autres circuits spécialisés.

D'autre part, il n'y a encore que très peu de circuits comprenant de l'ordre de 100 000 transistors (moins de 10 d'intérêt général dans le monde) : quelques microprocesseurs 32 bits, les mémoires 64 K. Tous les autres sont de l'ordre de 10 000 transistors ou moins.

Or, si on regarde les possibilités techniques en France, on constate qu'une filière technologique permettant de réaliser des circuits de 10 000 transistors est disponible en France depuis 1977 (NMOS à SESCOSEM); et 100 000 transistors est possible depuis 1981 (microprocesseur 68000 par EFCIS).

Pourquoi alors ne sommes nous pas mieux placés sur le marché mondial ? Probablement parce qu'il ne suffit pas de posséder la technologie pour réaliser le chiffre d'affaires. Encore faut-il sortir le bon circuit au bon moment.

##### 1.1.3. *Qu'est ce qu'un « bon » circuit intégré ?*

Prenons l'exemple des coupleurs de disques souples de Western Digital. Les disques souples ont existé avant ces coupleurs intégrés car la fonction des coupleurs est bien réalisable en circuits MSI. Le contrôleur LSI comprend moins de 10 000 transistors.

A l'heure actuelle, aucun ingénieur au monde ne s'amuse à construire un contrôleur sans cette « puce » et aucun étudiant n'apprend en détail les problèmes liés à ce type de contrôleurs. Tous les utilisateurs se contentent de recopier le schéma d'utilisation fourni avec la notice, ce qui assure un confortable chiffre d'affaires à Western Digital (et à ses secondes sources !).

En effet, cette société a commercialisé le premier circuit assurant cette fonction à une époque (fin 77) correspondant à l'augmentation de l'utilisation de ces disques et où le prix de l'électronique MSI devenait trop élevé par rapport à celui du reste de ces appareils.

La compétence nécessaire pour faire ce « bon circuit » est loin de n'être que « Marketing ». En effet, il faut avoir étudié tous les schémas d'application du circuit ainsi que tous les logiciels associés et posséder la compétence de tous les utilisateurs du circuit réunis.

Pour beaucoup de circuits, le problème est de maîtriser cette connaissance architecturale au moment où les considérations économiques rendent le circuit viable. Ensuite, comme les utilisateurs ne réétudient pas l'« intérieur de la puce », si celui-ci est une réussite commerciale, cette connaissance assure une suprématie technique aux créateurs du circuit.

A notre avis, outre les critères économiques, l'intérêt d'un circuit intégré peut être jugé en s'aidant des critères suivants :

1) Le fait qu'il permette l'existence d'une fonction non réalisable autrement (ou très difficilement) ;

2) Le fait qu'il repousse les limites pour les fonctions existantes (vitesse, capacité mémoire, consommation, ...) ;

Mais surtout, comme dans le cas du contrôleur de disques souples :

3) Le fait qu'il réalise une fonction « opportune » assurant une suprématie technique.

## 1.2. RÔLE DE LA RECHERCHE UNIVERSITAIRE. DIFFÉRENCE ENTRE LA FRANCE ET LES ÉTATS-UNIS

### 1.2.1 MPC aux États-Unis et CMP en France

Aux États-Unis, les industriels ont été les acteurs de cette révolution technologique en réalisant des circuits intéressants, chacun pour au moins un des 3 critères ci-dessus. En outre, les universitaires ont senti le besoin de disposer d'un outil permettant de travailler sur des architectures n'ayant pas forcément d'intérêt commercial immédiat. Ceci a débouché sur MPC (Multi-Project Chip) qui diminue les coûts de fabrication de circuits d'étude en rassemblant plusieurs sur une même tranche de Silicium.

Cette facilité permet :

- La formation des ingénieurs et des chercheurs dans les universités ;

- L'étude de circuits spécifiquement « recherche » servant à valider des architectures ou des algorithmes pour des projets à long terme.

Ceci est possible grâce au fait que les densités d'intégration actuelles sont bien supérieures à ce qui est nécessaire pour un grand nombre de circuits intéressants. Cet éloignement des contraintes technologiques permet de « prendre ses aises » en optimisant moins le dessin des masques, tant du point de vue électrique que topologique. On peut ainsi se focaliser sur des aspects plus algorithmiques [Mead 80]. Les circuits intéressants mis au point à l'université peuvent ensuite facilement faire l'objet d'un échange avec l'industrie.

En France, les mêmes arguments ont conduit à la même solution (depuis 1981), sous la forme de CMP mis sur pied grâce aux efforts de F. Anceau. Mais ici, le contexte est beaucoup moins favorable. Les industriels hésitent à lancer sur le marché un circuit totalement nouveau à vocation informatique. Dans ce cadre, il est difficile d'industrialiser un circuit universitaire, et la tentation est grande de faire à l'université des circuits sans penser aux applications, ce qui, par certains côtés, est paradoxal quand on utilise une technologie autant justifiée par ses aspects économiques.

Pour résoudre ce problème français, l'attitude généralement admise consiste à lancer d'ambitieux projets pour mettre au point tous les outils nécessaires pour la réalisation future de circuits très complexes. Ceci est bien sûr important, mais pendant ce temps, l'industrie doit vivre avec peu de produits nouveaux à grand marché. Nous croyons qu'il faut penser au court terme en faisant des circuits avec les outils existants. Ces circuits auraient en outre l'avantage de valider les évolutions de ces outils.

### 1.2.2. Possibilités offertes par le CMP

Le CMP permet de réaliser des circuits de performances suivantes :

- technologie NMOS (Grille de 5  $\mu$ ),
- densité jusqu'à 30 000 transistors,
- temps de traversée d'un inverseur de moins de 10 ns,
- superficie jusqu'à 50 mm<sup>2</sup>.

Ceci signifie que l'on peut réaliser des fonctions déjà très complexes, et descendre à des temps de cycle de l'ordre de 100 ns. Or, de nombreuses fonctions intéressantes ne nécessitent pas des performances aussi élevées. On peut donc utiliser des hypothèses simplificatrices :

- inutilité d'optimiser le dessin des masques,
- inutilité d'effectuer de larges simulations électriques, dans la mesure où des considérations d'entrance et de sortie suffisent pour estimer si la vitesse est suffisante.

A l'opposé, on peut aussi valider de la recherche sur des outils de CAO, dans la mesure où l'on peut tout aussi bien réaliser des circuits plus proches des limites de cette technologie et nécessitant des simulations électriques et logiques, ou des optimisations de dessin.

## 1.3. APPROCHE DE L'ÉQUIPE DU LABORATOIRE DE L'ENS

### 1.3.1. Expérience des circuits 364 et 365

Jusqu'à maintenant, notre équipe s'est plutôt focalisée sur les problèmes architecturaux mentionnés en 1.1.3.

La première réalisation est le circuit 364 sorti en 1977 grâce à une coopération avec la société SESCOSEM (Contrôleur d'écran alphanumérique EF 9364) [Gastinel 77b].

La technologie utilisée (NMOS-Grille Si de 6  $\mu$ ) a permis sans difficultés ce circuit de 3 000 transistors.

Les outils disponibles à l'époque (simulateur électrique jusqu'à 50 transistors, pas de simulateur logique, pas d'extracteur, pas de vérificateur de gardes, possibilité d'utiliser un éditeur graphique interactif pour des corrections de masques) nous ont conduits à adopter des systématisations simplificatrices : [Gastinel 77a]

- logique entièrement statique et synchrone,
- simulation logique réalisée sur maquette MSI et SSI,

- définition de briques de base réutilisées systématiquement pour minimiser les besoins de simulation électrique.

L'utilisation de maquettes de simulation (fig. 2) a permis en outre l'étude simultanée des montages d'application et du test du futur circuit. Ce travail a pris 1,5 an, suivi par 6 mois de dessin des masques.

Sur ces bases, nous avons pu démarrer en 1976 l'étude du circuit 365 (Contrôleur d'écran graphique) avec les mêmes outils. Ce circuit est plus complexe (10 000 tran-

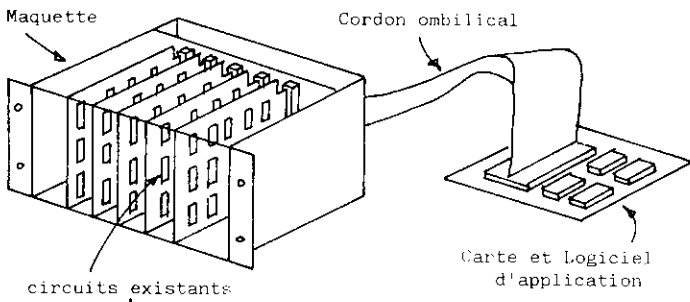


Figure 2. — Utilisation de maquettes de simulation.

sistors) et a posé des problèmes supplémentaires (au niveau logiciel principalement, testé sur les maquettes), mais la méthodologie a été la même, avec simplement une systématisation encore plus poussée des hypothèses simplificatrices comme la minimisation du nombre de briques de base de la bibliothèque [Matherat 78a] :

- un seul type de bascules (pour registres, compteurs, etc.),
- un seul type d'additionneur,
- etc.

Cela a permis de réaliser 95 % du circuit avec seulement 13 briques de base, limitant ainsi énormément la simulation électrique. Pour ce circuit, nous avons défini la bibliothèque de ces briques (logiquement et électriquement), l'arborescence de leur interconnexion, et leur disposition sur la puce [Matherat 80].

L'aspect topologique global de la puce (fig 3) ressemble énormément à ce qui se fait maintenant grâce aux outils de CAO évolués qui utilisent fatalement les mêmes hypothèses simplificatrices et la même hiérarchisation de description des masques. Ceci conduit à la même perte de densité par rapport à l'optimisation « manuelle ».

qui est inévitable si l'on veut changer l'ordre de grandeur des complexités atteintes.

Ce circuit est sorti commercialement en 80 (3 ans d'étude architecturale, 1 an de dessin des masques) sous la référence EF 9365.

Ces deux circuits ont été chacun à leur époque les circuits catalogues les plus complexes entièrement conçus en France. Le 365 n'a vu apparaître un concurrent, que six mois plus tard, sous la forme du circuit japonais NEC  $\mu$ P 7220 [Oguchi 81], [NEC 82].

### 1.3.2. Méthodologie

Ces deux réussites techniques nous ont montré le bien-fondé de notre démarche et de la simulation sur maquettes. Bien entendu, cela est lié au fait que ces circuits ne sont pas en limite de la technologie : perte de vitesse et de densité possibles dans une proportion suffisante. Mais ceci est d'autant plus vrai aujourd'hui alors que la technologie permet de réaliser des circuits de 500 000 transistors et de temps de transfert par porte de 1 ns.

La simulation logique sur maquette est à notre avis indispensable pour les circuits suivants :

- circuits très liés à une interface temps réel : visualisation d'une image, mais aussi synchronisation sur la rotation d'un disque ou sur l'horloge d'une ligne de transmission, etc.
- circuits implémentant un algorithme complexe dans le but de tester les performances globales dans l'environnement logiciel final.

Cette méthode est d'autant plus intéressante que la complexité logique augmente. La maîtrise de ce qui se passe dans une fonction de 100 000 transistors est plus délicate que pour 1 000 transistors ! Elle a en outre l'avantage de permettre la conception simultanée des montages et des logiciels d'application, ainsi que les problèmes de test du futur circuit.

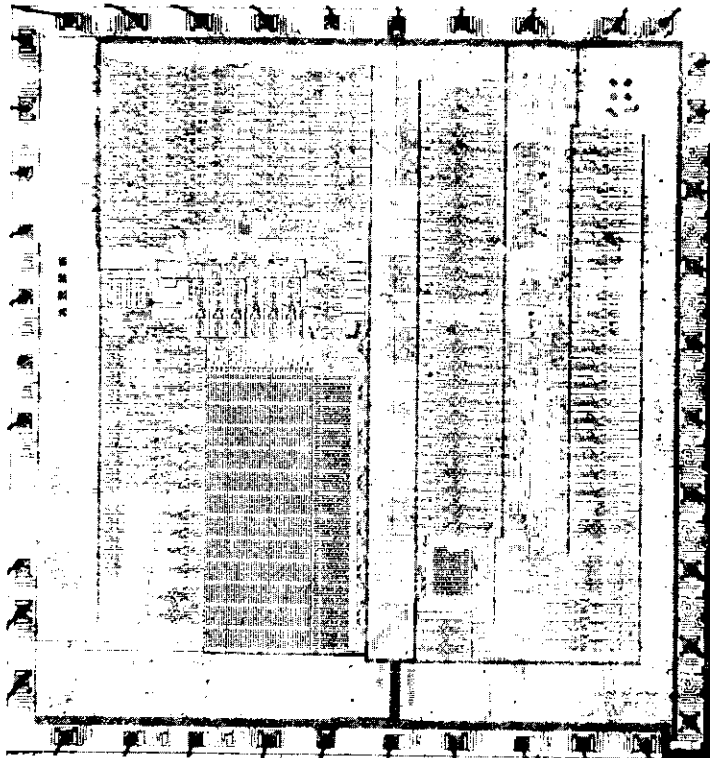


Figure 3. — Le circuit EF 9365.

Ces « maquettes » ne traduisent pas forcément la volonté de créer une « copie logique » des futurs circuits. Pour tester l'efficacité d'un algorithme de génération d'objets graphiques, il peut suffire de ne simuler que le débit d'écriture pour l'appliquer en temps réel à des dessins complexes permettant ainsi de tester l'ensemble « générateur + logiciel d'activation » qui est ce qui nous intéresse réellement. Un tel essai peut montrer que certaines parties doivent être accélérées, alors que d'autres sont très largement assez efficaces.

Cette démarche « expérimentale » nécessite de nombreux aller et retour entre la conception du schéma logique et le test sur maquettes. Ainsi, pour l'étude du circuit 365, quatre maquettes très différentes ont été successivement réalisées, occupant trois ans de travail. Nous pensons qu'aucun outil de CAO ne pourra diminuer ce temps pour ce type de circuits. Le seul gain de temps d'étude possible est dans la minimisation du nombre des maquettes par un choix judicieux des problèmes testés : mais il est hors de question de faire l'économie d'une validation sur des problèmes réels, dans l'environnement de l'utilisateur.

Bien entendu, il est des circuits pour lesquels il est impossible de réaliser des maquettes. C'est par exemple le cas des matrices de circuits systoliques de traitement du signal qui conduiraient à des montages énormes. Dans ce cas, il est fondamental de développer des outils logiciels de simulation de leur fonctionnement. On pourrait donc dire que les outils de simulation développés à cette occasion serviraient aussi pour des circuits (peut-être moins complexes) de couplage de périphériques, en appliquant la formule « Qui peut le plus peut le moins ». Mais nous pensons que la simulation sur maquettes, dans les cas où elle est possible, apporte davantage que la simulation logicielle, en particulier la possibilité de simulation en temps réel qui permet de travailler sur les applications du circuit avant son existence, et d'intégrer très tôt les remarques des utilisateurs.

## 2. Étude d'une architecture de contrôleur d'écran graphique

Cette deuxième partie décrit l'état d'avancement d'une expérimentation sur maquette dans le but de définir les fonctionnalités d'une future famille de circuits pour la visualisation graphique. Il s'agit d'une première étape dans laquelle nous n'avons pas recherché la performance pure (résolution, vitesse de tracé, nombre de couleurs élevé), mais la constitution d'une base pour expérimenter : d'une part le transport de logiciels existants, d'autre part certains problèmes matériels tels que l'incidence du nombre de bits par pixel.

### 2.1. INTRODUCTION

#### 2.1.1. Intérêt de l'intégration

Le thème de recherche de notre équipe est l'étude d'un micro-ordinateur VLSI. Pour ne pas être qu'un gadget, un tel appareil doit être très performant. Il doit en outre être petit et bon marché, ce qui impose de très grandes densités d'intégration. L'écran graphique, organe essentiel de sortie vers l'opérateur (messages alphanu-

mériques, dessins en couleur, ...) doit faire l'objet d'un soin particulier. Les performances minimales nécessaires (résolution, nombre de couleurs, et surtout vitesse d'affichage), placent la barre très haut. A l'heure actuelle, malgré la très rapide évolution de ce domaine (due justement aux VLSI), un contrôleur d'écran graphique performant est encore d'un coût bien trop élevé (300 kF) pour être compatible avec une utilisation individuelle très répandue.

#### 2.1.2. Travail à réaliser

Notre premier circuit graphique, ainsi que le NEC 7220, d'une complexité de l'ordre de 10 000 transistors, permettent la réalisation de contrôleurs graphiques de bonne qualité pour un coût guère supérieur à celui de contrôleurs simplement alphanumériques, ce qui les rend très intéressants pour les options graphiques ajoutées aux ordinateurs individuels actuels.

Mais l'ordinateur individuel des années 90 devra posséder un affichage bien plus performant que ce que l'on peut réaliser avec seulement un petit nombre de circuits autour d'un de ces LSI. L'architecture de ces futurs contrôleurs d'écrans n'est pas encore maîtrisée. Tout d'abord, les performances souhaitées pour un tel objet ne sont pas toujours claires. Que veut-on afficher, avec quelle précision et quelle vitesse ? Et ensuite, quelles architectures permettront ces performances ?

Comme pour les circuits précédents conçus au laboratoire, nous avons choisi d'étudier ces problèmes sur des « maquettes » câblées avec des circuits existants.

#### 2.1.3. Aspects logiciels

Depuis une dizaine d'années, de nombreux efforts sont faits pour développer des programmes graphiques indépendants du dispositif de sortie (écran vecteur ou « raster-scan », traceur incrémental, imprimante laser, etc...). La méthode générale consiste à compiler les programmes en langage de haut niveau vers un langage intermédiaire indépendant du périphérique puis d'interpréter (ou de compiler) celui-ci d'une façon différente pour chaque périphérique.

D'autre part, des normes commencent à se fixer pour le langage de haut niveau employé par les utilisateurs [Status 79], [ISO 82]. En effet, la technologie évoluant très vite, et les performances avec, la durée de vie des programmes devient très courte et quelquefois comparable au temps de réécriture de ces logiciels souvent très complexes (outils de CAO par exemple). L'absence de normalisation à ce niveau serait un frein énorme au développement de tout le domaine graphique.

Jusqu'à maintenant, ces normalisations n'ont pas influencé les architectures matérielles des dispositifs graphiques, mais cherchaient uniquement à concilier leurs performances disparates. Il est clair que ceci va changer dans la mesure où il existera peu de VLSI graphiques. Seuls 1 ou 2 circuits contrôleurs d'écrans emporteront le marché et deviendront la norme, tant logicielle que matérielle, comme il en a été des contrôleurs de disques et de communication dans un proche passé [Lewitt 83].

Il est donc important de maîtriser les relations entre norme logicielle et architecture matérielle sous les aspects de performance globale si l'on souhaite concevoir un circuit contrôleur d'écran qui ait une chance d'être largement utilisé.

## 2.2. RAPPELS : ARCHITECTURE D'UN CONTRÔLEUR D'ÉCRAN GRAPHIQUE

Dans un écran graphique à balayage de trame, l'image est rafraîchie périodiquement, à 25 Hz en télévision, à 30 ou 40 Hz ou davantage pour obtenir des images exemptes de scintillement. Ce rafraîchissement est obtenu grâce à un faisceau d'électrons qui balaye systématiquement toute la surface de l'écran, ligne par ligne, de haut en bas. Les informations de modulation du faisceau d'électrons (ou des 3 faisceaux Rouge, Vert, Bleu en couleur) sont issues d'une mémoire d'image qui contient la description totale, point par point, de l'image affichée.

Pour une image de 1000 \* 1000 points où chaque point est codé sur 8 bits, il faut une mémoire d'image de 1 Moctet. On comprend ainsi aisément pourquoi ce type d'écran « à balayage de trame » ne s'est vraiment développé qu'à partir de 1976, époque de sortie commerciale des mémoires RAMs de 4 k \* 1 bits permettant un rapport densité/coût suffisant.

Pour assurer l'entretien de l'image sur l'écran, il est nécessaire de lire la totalité de cette mémoire d'image à la fréquence de rafraîchissement de l'écran. Parallèlement à cette visualisation, il faut écrire dans cette mémoire les modifications des points que l'on peut classer en :

- tracés de caractères alphanumériques,
- tracés de segments de droites ou d'arcs de courbes, remplissages de polygones,
- copies d'une zone de l'écran de ou vers l'extérieur ou intérieurement à la mémoire.

Ces opérations d'écriture doivent être rapides. Il faut remplir une page de texte presque instantanément, dessiner un tracé complexe sans demander une heure d'attente, etc. Il faut pouvoir animer une partie de l'image, c'est-à-dire l'effacer et la réécrire à la fréquence des trames. Pour fixer les idées, disons que le débit d'écriture doit être égal à celui de visualisation : débit qui est réellement utilisé pour une animation totale de l'écran (dans ce cas, la mémoire d'image est utilisée en « tampon intermédiaire » et doit être doublée).

Nous voyons ainsi qu'une caractéristique essentielle de cette mémoire d'image, outre sa capacité, est son débit. Dans le cas de l'écran ci-dessus, s'il est rafraîchi à 40 Hz, le débit sera :

$$D = 2 * 40 = 80 \text{ Moctets/s} = 640 \text{ Mbits/s}$$

Et ce débit est utilisé pour la moitié à une tâche hautement prioritaire : le rafraîchissement de l'écran. Cette caractéristique fait qu'en général, cette mémoire n'est pas la mémoire centrale de l'ordinateur hôte, mais une mémoire spécialisée (même si elle est souvent plus grande et plus rapide que la mémoire centrale). Néanmoins il est intéressant de la situer dans l'espace d'adressage de l'Unité Centrale (UC) pour profiter des instructions de calcul directes, ainsi que des Accès Directs à la Mémoire (ADM) (accès disque, communications, etc.) [Baecker 79].

Suivant les performances recherchées (vitesse d'écriture), on ajoutera éventuellement des opérateurs de tracé spécialisés qui pour certaines tâches iront beaucoup plus vite que ne le peuvent des programmes exécutés par l'UC (vecteurs, caractères, etc.).

Un autre opérateur qui, lui, est plus proche de la visualisation, est le remplissage de la mémoire avec une image vidéo (caméra, magnétoscope, etc.) et digitalisée.

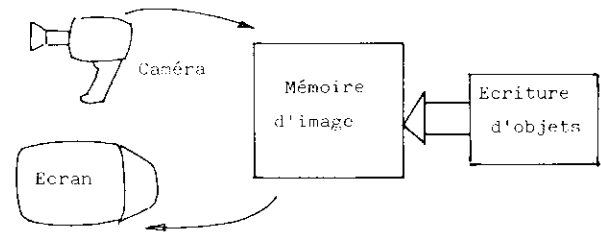


Figure 4. — Schéma d'une visu graphique.

## 2.3. ETUDE DES PARTIES CRITIQUES

Dans cette partie, nous exposons un certain nombre de choix faits pour notre maquette. Rappelons le cadre fixé : réalisation d'une maquette pour expérimenter des problèmes matériels et logiciels, sans rechercher la performance pure, et en simplifiant la réalisation à l'aide de LSI existants. Dans ce cadre, l'architecture est largement imposée par le contrôleur utilisé (EF 9365). Nous utiliserons donc une présentation ascendante de cette architecture, dans la mesure où l'expérience passée nous a montré quels choix initiaux se répercutaient sur l'ensemble de la maquette.

### 2.3.1. Choix de la dimension et du débit de la mémoire pour cette étude

Dans le but d'étudier les problèmes d'organisation de la mémoire d'image, et de tous les chemins de données y accédant, nous avons dû choisir un format d'image et une fréquence de balayage, ces deux paramètres influençant beaucoup le nombre de circuits nécessaires pour la réalisation des maquettes. Nous avons choisi le format 512 \* 512 balayé à 25 images/s. Le nombre de bits par point de l'image (par « pixel ») a été choisi multiple de 4 bits. Nous avons choisi de doubler la mémoire d'image pour pouvoir fonctionner en « tampons alternés » dans les applications dynamiques.

Les raisons de ces choix apparaîtront dans les développements qui vont suivre. Disons tout de même qu'un gros argument est la possibilité de réaliser une partie des maquettes par le circuit 365 lui-même. Les parties qui nous intéressent en effet dans cette étude ne sont pas celles prises en charge par ce circuit, puisqu'elles concernent davantage l'étude des chemins de données qui entourent la mémoire et leur influence sur l'architecture globale, que les générateurs d'adresses de visualisation et d'écriture des vecteurs et des caractères. En outre ce choix permet déjà de très bonnes performances et est bien adapté pour un ordinateur individuel.

Le nombre de 4 bits par point a été choisi car il est bien adapté aux écrans informatiques : traitement de textes, CAO, tracé de courbes, de surfaces, etc... puisqu'il permet d'afficher 16 couleurs ou niveaux de gris simultanés. Nous avons étudié ce qui est particulier à cette valeur, mais aussi ce qui est à dupliquer si on veut augmenter ce nombre (jusqu'à 32 bits).

Le débit de visualisation (pour 4 bits par pixels) est de :

$$25 * 512 * 512 * 4 = 25 \text{ Mbits/s}$$

Ceci est le débit moyen puisque la totalité du signal ne peut pas être support d'information image. Il faut ménager les temps de retour du spot en fin de ligne et en fin de trame. En fait, lors des périodes de visualisation, ce

débit est 56 Mbits/s. Le temps des marges est utilisé pour écrire de nouveaux objets (vecteurs, caractères) dans la mémoire d'image.

Ces signaux sont portés par 4 fils indiquant l'état des points successifs de l'écran (à  $56/4 = 14$  MHz). Il y a donc 4 blocs de mémoires en parallèle à 14 MHz. Pour atteindre ce débit, nous avons choisi de découper chacun de ces 4 blocs en 8 boîtiers mémoires de 64 K fonctionnant en parallèle (fig. 5).

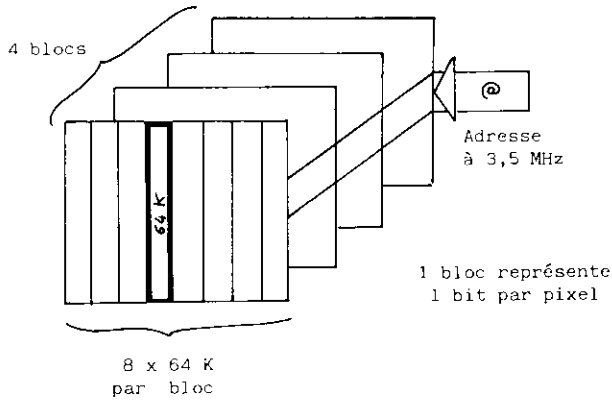


Figure 5. — Découpage de la mémoire.

La fréquence des accès mémoire est alors de  $14/8 = 1,75$  MHz. Nous avons utilisé des mémoires deux fois plus rapides et nous permettant donc un accès supplémentaire à 1,75 MHz (en sus des accès de visualisation normaux et des cycles disponibles pour l'écriture lors des marges). Le débit de cette mémoire est donc :

$$4 * 8 * 3,5 = 112 \text{ Mbits/s} = 14 \text{ Moctets/s} .$$

Sa capacité est :

$$4 * 512 * 512 * 2 = 2 \text{ Mbits} = 256 \text{ Koctets} .$$

Pour tout ce qui suit, nous ne considérerons qu'un seul bloc puisque tous fonctionnent en parallèle de façon identique.

### 2.3.2. Les opérateurs de visualisation et d'entrée vidéo

Les opérateurs de visualisation et d'entrée vidéo sont simples : le générateur d'adresse doit fournir, à la fréquence de 1,75 MHz une adresse en phase avec la position du spot sur l'écran, c'est-à-dire avec le signal de synchronisation. Cette tâche est généralement effec-

tuée par un compteur, des reconnaisseurs sur les valeurs de celui-ci servant alors à générer les tops de synchronisation.

Pour cette étude, tous les problèmes de génération d'adresses ont été reportés sur le 365 et n'ont donc pas été réétudiés.

Du point de vue des données, la fréquence des pixels étant 8 fois plus rapide, il faut utiliser des registres à décalage faisant passer de 8 bits en parallèle à 1 bit à 14 MHz à 1 bit à 14 MHz (ou réciproquement pour l'entrée vidéo) (fig. 6).

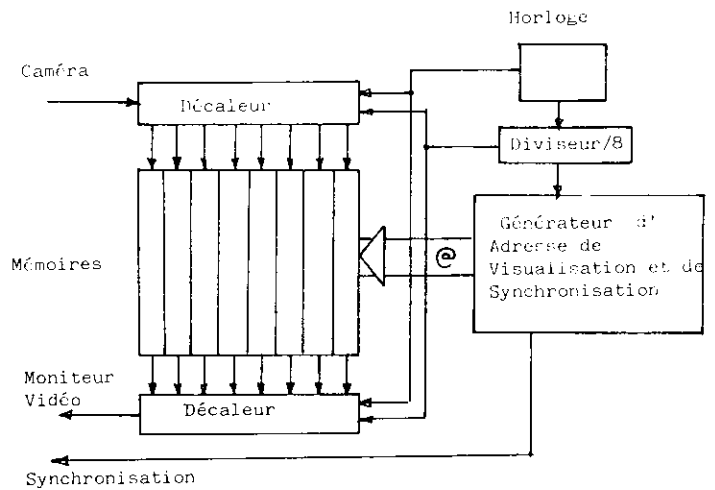


Figure 6. — Les décalages.

On peut mettre en commun l'entrée et la sortie et confondre les deux registres à décalage, mais nous verrons plus loin les inconvénients de cette architecture.

### 2.3.3. Ecriture d'un pixel

En ce qui concerne les opérateurs d'écriture de nouveaux objets (segments de droite, caractères, remplissage de zone, coniques, etc...), nous avons choisi délibérément pour cette étude une hypothèse simplificatrice : ces opérateurs fourniront au maximum 1 point par cycle mémoire (ce qui permet déjà 3.5 Mpixels/s). Cela signifie que lors de ces cycles mémoire, un seul boîtier sur les 8 est sélectionné (fig. 7).

Il est souvent intéressant d'écrire une donnée qui est une fonction de l'état précédent du pixel. On peut ainsi faire un OU exclusif, etc. Dans ce cas, le cycle mémoire est un cycle de lecture-modification-écriture

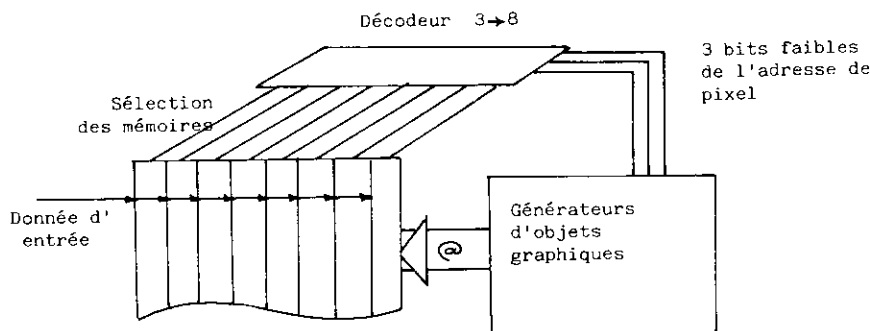


Figure 7. — Ecriture d'un pixel.

(LME). Le schéma de transfert des données est alors celui de la figure 8.

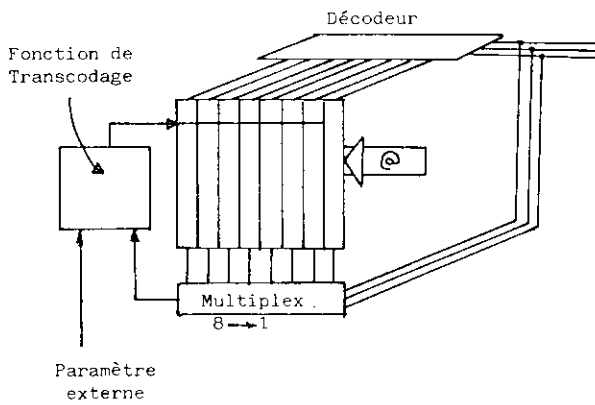


Figure 8. — Lecture-modification-écriture.

Mais pour ce type de transcodage, il est bien plus intéressant de considérer tous les blocs mémoire puisque les fonctions ainsi permises sont plus nombreuses (opérations arithmétiques par exemple (fig. 9)).

Cette fonction de transcodage peut être arbitrairement complexe, pas forcément combinatoire. Elle peut a priori utiliser un nombre de cycles quelconque pour son calcul. Dans la pratique, on utilise de simples fonctions combinatoires, soit sans paramètre (simple table de transcodage par exemple), soit avec un paramètre externe fourni par l'UC, de même largeur (UAL  $8 * 8 \rightarrow 8$  par exemple).

Dans notre cas, pour tester les fonctions de transcodage intéressantes, et comme dans un premier temps, nous nous sommes limités à 4 bits par pixel, nous avons réalisé une table de transcodage par une mémoire de 16 mots de 4 bits modifiable par l'UC. Pour des applications de remplissage de contour ou de détection d'objet, pour effectuer rapidement (à la vitesse de tracé des vecteurs) un test sur la valeur du pixel en mémoire d'image, nous avons doublé cette table de transcodage avec la possibilité de commuter dynamiquement d'une table à l'autre en

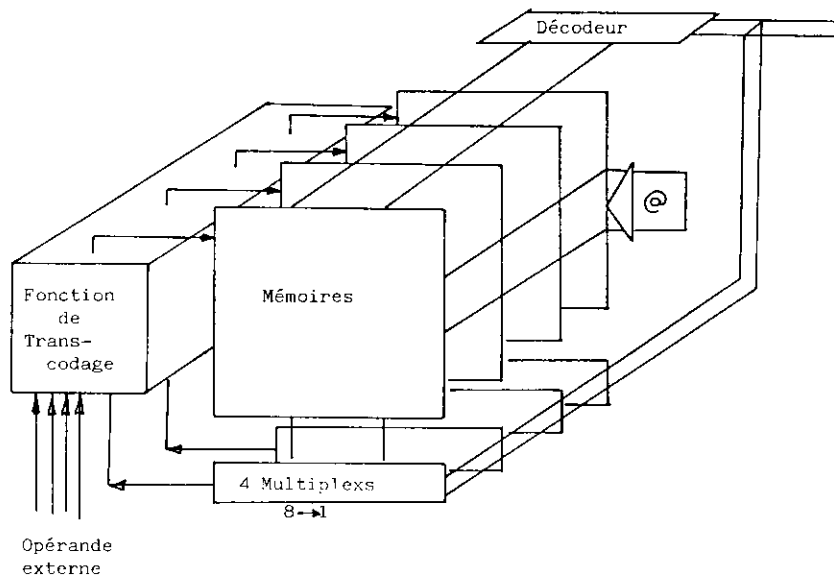


Figure 9. — Extension à tous les blocs mémoire.

fonction de l'état ancien du pixel dans la mémoire d'image.

### 2.3.4. Accès à la mémoire d'image par l'UC

En ce qui concerne les accès à la mémoire d'image par l'UC, ceux-ci peuvent être de deux types :

- soit pixel par pixel, dans le cas de :
  - algorithmes de tracés d'objets,
  - traitements d'image.

On sélectionne alors une seule mémoire par bloc, et les données utilisent la même voie que ci-dessus (1 fil commun en entrée, 1 multiplexeur en sortie, voire même l'opérateur de LME) :

— soit par segment dans un bloc. On accède alors aux 8 mémoires d'un bloc simultanément (comme en visualisation). Ceci est utile lorsque l'on souhaite un accès très rapide (par un ADM par ex.) mais pas sur tous les bits d'un pixel. Dans ce cas, il faut disposer de registres ou de portes supplémentaires (fig. 10).

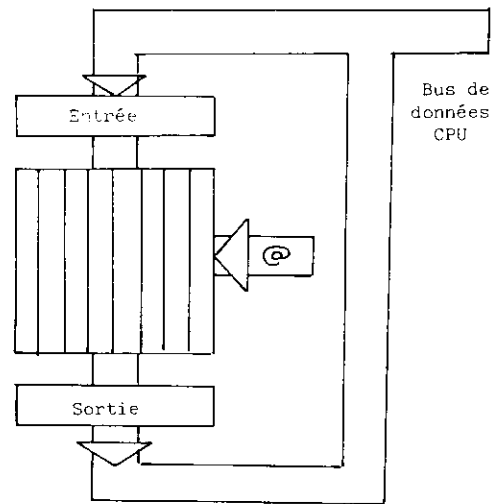


Figure 10. — Accès à la mémoire par l'UC.

2.3.5. Chemins des données

Il nous reste à fondre ensemble tous ces ports d'accès à la mémoire et à multiplexer les adresses mémoires en fonction du type de cycle envisagé. Ce qui va complètement orienter les choix suivants est la décision de relier ou non les fils d'entrée et de sortie de chaque mémoire. Ceci simplifierait les bus de données mais pose un problème pour les cycles de LME qui font simultanément une entrée et une sortie. Veut-on séparer les actions d'entrée et de sortie en deux cycles distincts ? Ou bien veut-on les séparer au milieu du cycle ? Dans les 2 cas, il faut ajouter un registre intermédiaire pour assurer la stabilité de la donnée de sortie.

Nous avons choisi de faire un LME en un seul cycle pour réaliser cette opération plus vite, et nous n'avons pas mis de registre, pour profiter au mieux du temps d'accès des mémoires. Ceci nous conduit vers des entrées-sorties séparées.

Étudions d'abord les sorties : elles sont de trois sortes :

- sortie série 14 MHz par registre à décalage :
- sortie 1 bit par multiplexeur (pour LME et lecture pixel par UC) :
- sortie 8 bits vers UC par 1 registre ou une barrière (fig. 11).

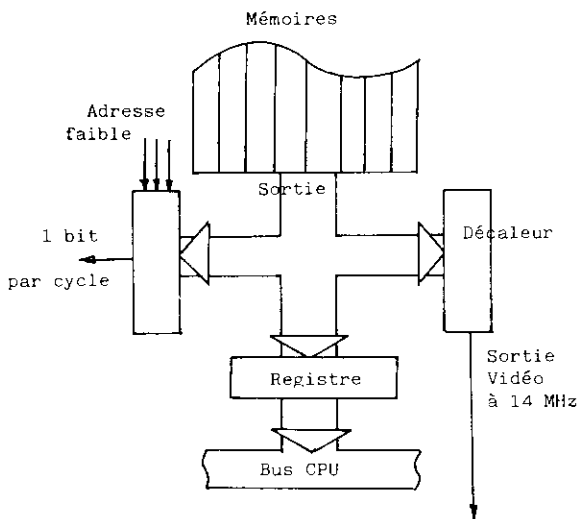


Figure 11. — Les sorties.

Étudions les entrées : il faut de même trois entrées qui peuvent être sélectionnées par un multiplexeur ou une barrière à trois états pour :

- entrée série 14 MHz par registre à décalage (entrée vidéo) :
- entrée 1 bit par bloc et par cycle pour une écriture pixel par générateur d'objet ou UC :
- entrée 8 bits par UC (fig. 12).

2.3.6. Utilisation de la sortie vidéo

De chaque bloc de mémoire en parallèle sort un signal vidéo à 14 MHz. Comme il y a  $n$  ( $1 \leq n \leq 8$ ) blocs, la vidéo est codée sur 4 à 32 bits. Il est intéressant de faire en sorte que, au niveau de la mémoire d'image, le codage des pixels soit banalisé, c'est-à-dire que ces  $4 * n$  fils sortant la vidéo n'attaquent le moniteur que par l'intermédiaire d'une fonction de transcoding facilement paramétrable. Nous avons par exemple utilisé le montage de la figure 13 pour  $n = 1$ .

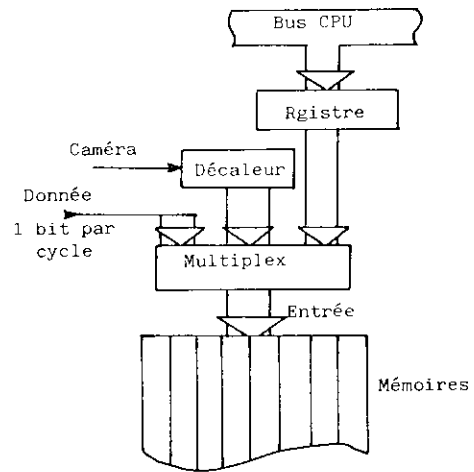


Figure 12. — Les entrées.

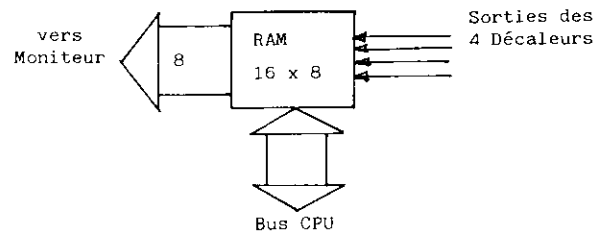


Figure 13. — Montage pour  $n = 1$ .

On intercale une mémoire de 16 mots de 8 bits qui constituera une table de transcoding modifiable par l'UC. On a ainsi une « palette » de 16 couleurs parmi 256. On envoie ensuite ces 8 bits sur 3 convertisseurs Digitaux/Analogiques générant les signaux Rouge/Vert/Bleu pour le moniteur. On peut par exemple envoyer 3 bits sur le convertisseur Rouge, 3 sur le vert, 2 sur le bleu (fig. 14).

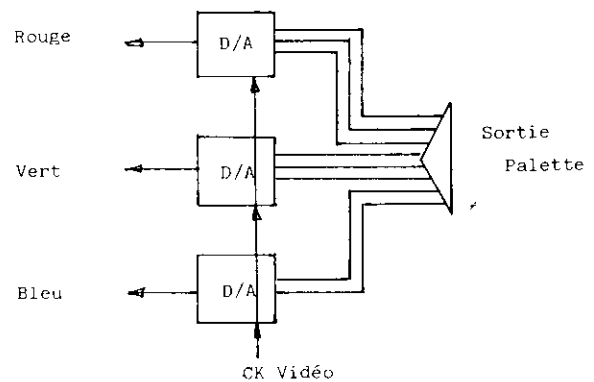


Figure 14. — Codage de la couleur.

Comme il est intéressant aussi de disposer de niveaux de gris simultanément avec les couleurs, sur la même image, nous avons choisi un codage un peu plus complexe où le bit faible indique si les 7 autres bits doivent être envoyés identiques aux 3 DACs ou bien séparés en 3 champs de 2 bits, 3 bits, 2 bits. On dispose ainsi de 16 « états » choisis dans une palette de 128 couleurs et 128 niveaux de gris (fig. 15).

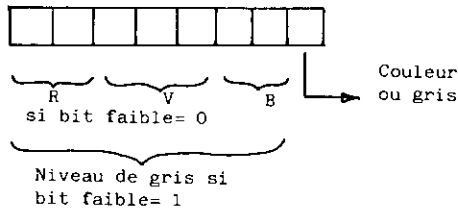


Figure 15. — Couleur et niveaux de gris.

Cette fonction combinatoire simple peut être réalisée grâce à des circuits programmables par fusibles (PALs de MMI par exemple). Ceci permet d'ailleurs une reprogrammation pour d'autres types d'affectations (fig. 16).

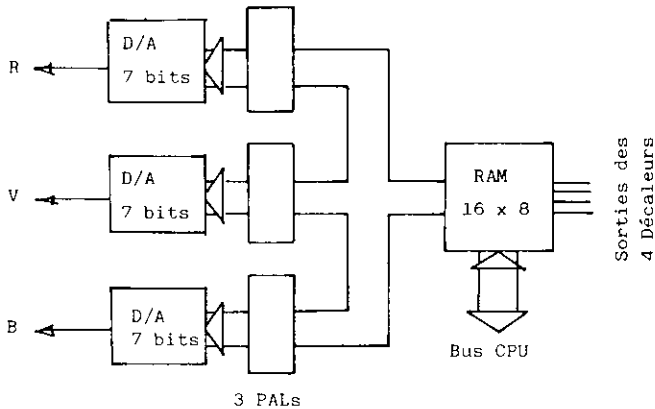


Figure 16. — Utilisation de circuits programmables par fusibles.

Que se passe-t-il quand  $n$  augmente ? Jusqu'à  $n = 3$ , on peut encore réaliser ce genre de montage car il est possible de trouver des RAMs très rapides (à la vitesse de la vidéo !) d'une capacité de  $2^{4n} = 4096$  mots (de 8 bits par exemple). Ensuite, il est impossible de garder une banalisation complète des données en mémoire d'image. On résout classiquement ce problème en mettant plusieurs tables de palette sur des portions de l'ensemble des bits sortant des registres à décalage, ces bits étant éventuellement permutable grâce à une circuiterie combinatoire de type « crossbar ».

Tous ces systèmes de transcodage vidéo peuvent être utilisés sur les signaux provenant de la digitalisation de l'entrée vidéo.

2.3.7. Plan d'ensemble

On peut ainsi (fig. 17) rassembler les différentes parties vues précédemment. Les adresses mémoire sont issues des différents générateurs (par l'intermédiaire d'un opérateur de translation effectuant des additions sur les adresses). L'opérateur « accès par UC » réalise uniquement une adaptation des adresses de l'UC aux adresses mémoire d'image.

2.3.8. Découpage fonctionnel

Du point de vue architectural, on peut découper cet ensemble de la façon suivante (correspondant aux pointillés de la fig. 17) :

A) Partie indépendante du nombre de bits par pixel

Cette partie se découpe elle-même en deux :

A1) Une partie à la fréquence des accès mémoire, qui regroupe tous les générateurs d'adresses :

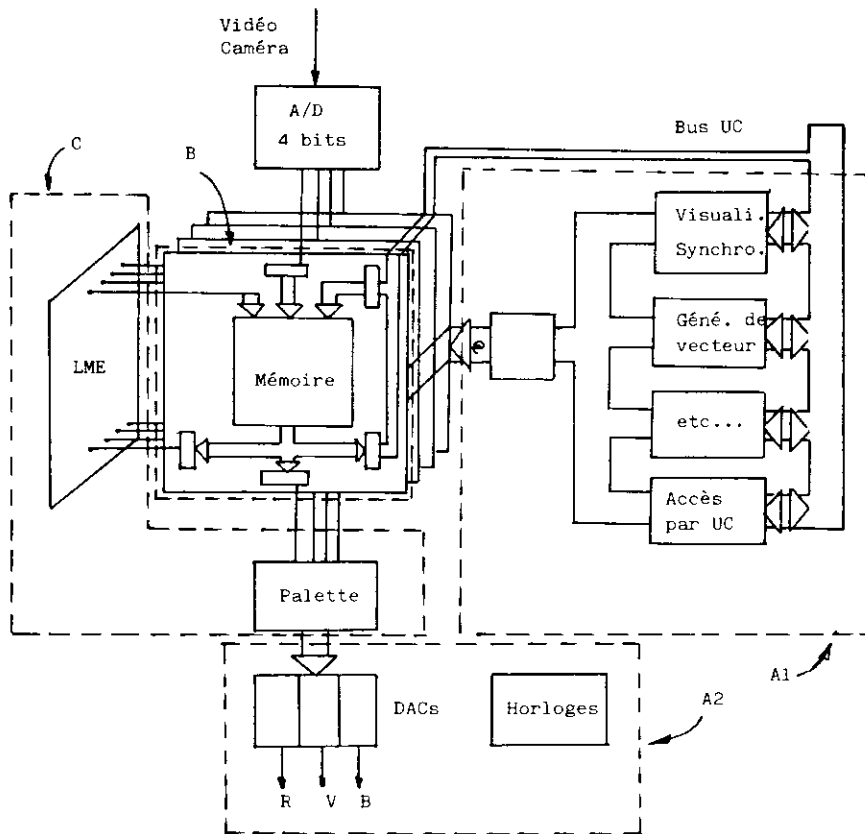


Figure 17. — Plan d'ensemble.

- visualisation et synchronisation,
- tracés d'objet : vecteurs, caractères, etc.,
- accès par UC,
- allocation de la mémoire,
- opérateurs de scroll,
- etc.

A2) Une partie à la fréquence de la vidéo, qui regroupe :

- les générateurs d'horloge,
- les convertisseurs A/D et D/A.

**B) Partie de complexité proportionnelle au nombre de bits par pixel.**

C'est la mémoire d'image et toute la circuiterie d'accès à cette mémoire :

- registres à décalage,
- registres d'accès par UC,
- circuiterie d'entrée et de sortie à 1 bit par cycle.

**C) Partie très liée au nombre de bits par pixel**

Il reste ici deux choses :

- la fonction de LME,
- la palette de sortie vers les DACs.

2.3.9. Définition d'une famille de VLSI possibles

Le découpage précédent permet de dégager des fonctions assez indépendantes les unes des autres, communiquant entre elles par des bus de taille raisonnable. Ces fonctions ont une grande généralité et ceci suggère la définition d'une famille de VLSI organisés ainsi :

- La partie A1 pourrait être réalisée par un circuit VLSI en technologie HMOS. Toutes ses opérations sont effectuées à la cadence des cycles mémoire. Il réaliserait l'interface entre un bus microprocesseur et l'adresse de la mémoire d'image (fig. 18). On pourrait l'appeler « Contrôleur de mémoire graphique » (CMG). Ce circuit pourrait être un VLSI très dense si les générateurs d'objet sont complexes et rapides. Pour l'instant, dans notre maquette, c'est le circuit 365 avec quelques MSI autour.

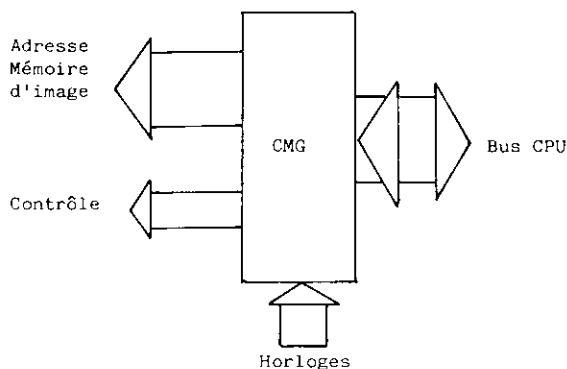


Figure 18. — Circuit contrôleur d'adresses mémoires.

Les signaux de contrôle indiqueraient à la mémoire d'image la nature du cycle.

- La partie A2 pourrait être réalisée par un LSI devant fonctionner à la cadence des pixels. Circuit « Horloges et Conversions » (H/C) (fig. 19).

- La partie B est une mémoire spécialisée puisque équipée d'une périphérie adaptée au graphique (MG) (fig. 20).

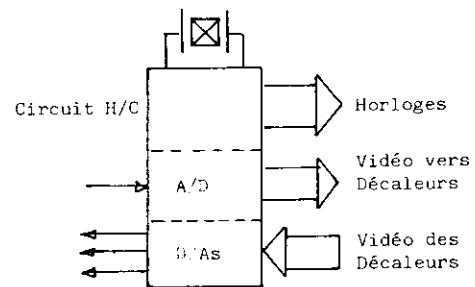


Figure 19. — Circuit horloge et conversions.

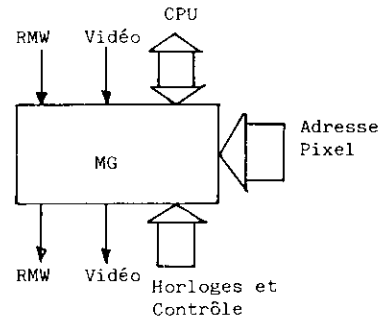


Figure 20. — Circuit mémoire.

La capacité mémoire est 512 Kbits. On peut chercher à réaliser ceci en 2 parties de 256 K ou tenter un circuit de 512 K en demandant davantage à la technologie. Une petite partie du circuit (les registres à décalage) fonctionne à la vitesse vidéo et devra être optimisée localement.

- La partie C peut être réalisée pour  $n = 1$  puisque un ensemble de 4 bits par pixel est largement utilisable pour la CAO et le traitement de texte. Elle doit être réalisée en bipolaire puisqu'elle travaille à la vitesse de la vidéo (fig. 21).

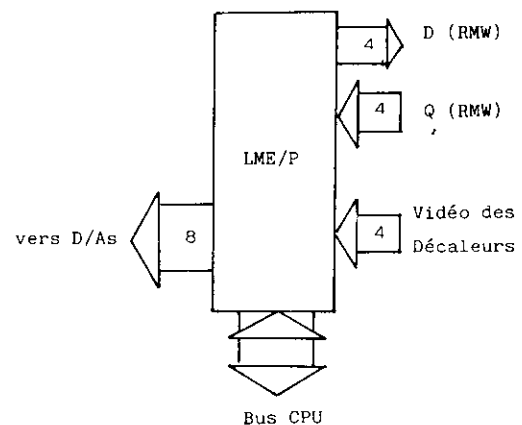


Figure 21. — Partie C.

En définissant ainsi cette famille de circuits, on peut réaliser un contrôleur d'écran 512 \* 512 et 16 couleurs en 7 circuits intégrés tenant sur une petite carte (fig. 22).

2.3.10. Architecture de la maquette de simulation actuelle

Nous avons implémenté cette architecture avec des composants existants. Une partie du circuit CMG est

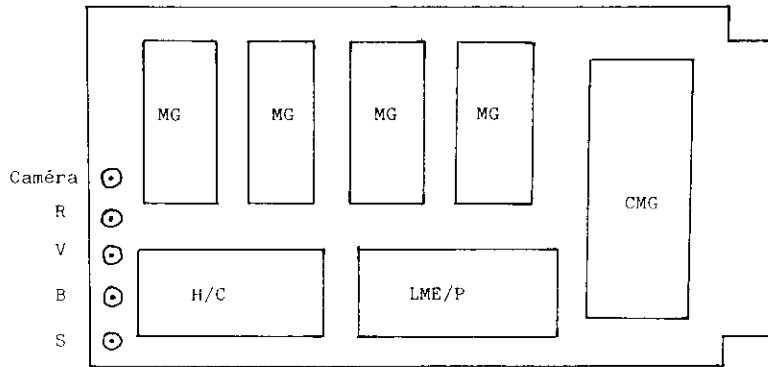


Figure 22. — L'ensemble sur une carte.

réalisée avec le circuit 365. Les mémoires sont des  $64 k * 1$  bit, il en faut donc 32. Pour garder la modularité liée au découpage précédent, nous avons découpé l'ensemble en 3 cartes distinctes :

- une carte mère rassemblant toute la partie A (circuits CMG et H/C),
- une carte fille mémoire rassemble les 4 plans mémoire avec leur périphérie,
- une carte fille supplémentaire réalise le circuit LME/P.

Les cartes mémoire sont empilables de 1 à 8 grâce à un connecteur spécial : la carte LME/P doit être changée lorsque l'on modifie le nombre de cartes mémoire. La carte mère contient environ 100 circuits intégrés. La carte mémoire comporte 60 circuits ; la carte LME/P contient 30 circuits (fig. 23).

### 2.3.11. Limitations et évolutions possibles

Le choix d'un affichage  $512 * 512$  n'amène pas de restrictions. Tout ce que nous avons appris avec ce format est transposable à des résolutions plus élevées ou des formats rectangulaires. Ce choix a eu l'énorme avantage de permettre d'utiliser le circuit 365 dont l'architecture se prête mieux que le NEC à ce découpage fonctionnel. Ceci permet de remplacer 300 circuits par un seul dans la maquette. En outre, cette maquette, qui nous a appris beaucoup sur l'architecture de ces futurs VLSI, réalise en elle-même une carte graphique très intégrée, de rapport performance/coût inégalé sur le marché actuel, ce qui n'est pas très étonnant vu le souci de densité d'intégration qui a présidé à cette étude.

Les limitations de cette architecture résident dans les deux points suivants :

1) A notre avis, le débit d'écriture de 1 pixel par cycle mémoire pour la génération d'objets est trop faible, en particulier pour un opérateur de recopie câblée de zones rectangulaires dont une application essentielle est la gestion de « fenêtres » sur l'écran ainsi que l'utilisation de fontes alphanumériques variées.

2) Toutes les opérations sur un octet complet dans un bloc mémoire (visualisation par exemple) se heurtent à une frontière d'octets rigide. En particulier, une translation horizontale câblée de précision inférieure à 8 pixels est difficile à réaliser.

Ces 2 problèmes se rejoignent. Il faut organiser l'accès mémoire plus souplesment : pouvoir accéder à des mots mémoire non forcément cadrés d'une seule manière horizontalement. Mais cela aurait beaucoup alourdi notre maquette actuelle.

### 2.3.12. Plan de travail pour l'avenir

Cette maquette actuelle n'est pas, pour nous, la maquette d'une famille de circuits intégrés tels quels. Elle est d'une part un outil, et d'autre part une étape d'acquisition de connaissances sur ce genre d'architectures.

Sur ces bases, nous pouvons réfléchir aux fonctionnalités d'une future famille de circuits, qui, en reprenant ce découpage fonctionnel, résoudra les problèmes mentionnés plus haut. Il faudra réaliser d'autres maquettes pour tester la validité des choix que nous ferons.

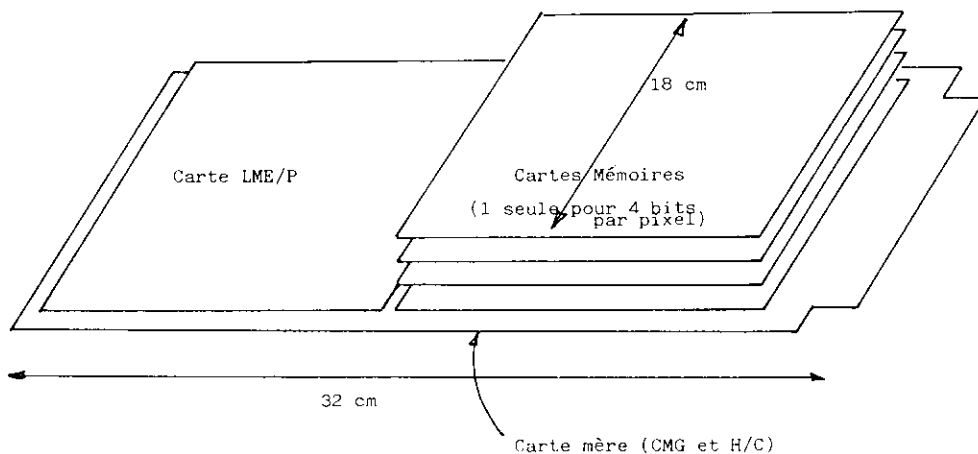


Figure 23. — Maquette actuelle.

Parallèlement, nous pouvons tout de même intégrer des parties de la maquette actuelle pour la condenser. Outre l'intérêt de la faire exister sous une forme plus condensée et dissipant moins de chaleur, cela nous permettra de mettre en œuvre les outils de CAO existants et la technologie de réalisation, et ainsi d'en déduire des enseignements pour les circuits à plus long terme.

### 3. Conclusion

Nous avons exposé, sur l'exemple de l'architecture d'un contrôleur d'écran graphique, comment nous menons parallèlement l'étude de la fonction de futurs VLSI et l'expérimentation sur des maquettes émulant ces fonctions. Le stade actuel de cette étude est concrétisé par la maquette qui est un produit intéressant en soi, et que nous utilisons quotidiennement comme outil.

Nous avons proposé une famille de VLSI permettant de remplacer les 200 circuits de cette maquette par 7 boîtiers. Les technologies disponibles en France actuellement permettent d'envisager leur réalisation. Ces circuits pourraient s'incorporer à n'importe quel ordinateur individuel ou à des postes de travail de CAO. Le temps nécessaire pour réaliser des circuits (de l'ordre de deux ans) est, à notre avis, compatible avec le besoin croissant de tels écrans dans les années à venir.

Malgré cela, il nous semble que l'investissement nécessaire pour créer les masques de cette famille n'est pas encore justifié dans la mesure où les performances pourraient encore être nettement augmentées, sans grande augmentation de la complexité des circuits. Nous pensons que d'ici peu de temps, l'ensemble des performances demandées à une visu graphique va évoluer vers un standard qui sera plus performant que ces circuits (résolution et vitesse d'écriture supérieure), en particulier en liaison avec une utilisation plus large à domicile d'appareils photo et vidéo numériques, et qu'à ce moment-là, il sera plus opportun de créer cette famille. Nous pensons qu'il faut mener en parallèle l'étude de cette famille plus performante et l'intégration (à moins grande échelle) des fonctions de la carte actuelle.

### BIBLIOGRAPHIE

- [Baecker 79] R. BAECKER : *Digital Video Display Systems and Dynamic Graphics* : Conférence SIGGRAPH'79.
- [EDN 83] Electronic Design News Magazine. 20 janvier 1983.
- [Gastinel 77a] J. GASTINEL : *Conception et Intégration d'un Terminal Alphanumérique* : Thèse de 3<sup>e</sup> cycle, Université Pierre-et-Marie-Curie, 20 janvier 1977.
- [Gastinel 77b] J. GASTINEL et A. BERNARDY : *The Design of a Low-Cost TTY Replacement* : Euromicro 1977.
- [ISO 82] *Graphical Kernel System (GKS). Functional Description* : Draft International Standard ISO/DIS7942, novembre 1982.
- [Lewitt 83] M. LEWITT et B. COHEN (Intel Corp.) : *Computer Graphics needs Standards as a Foundation for Future Growth* : Electronics, 24 février 1983, 124.
- [Matherat 78a] P. MATHERAT : *Conception d'un circuit intégré pour la Visualisation Graphique* : Thèse de 3<sup>e</sup> cycle, Université Pierre-et-Marie-Curie, 19 mai 1978.
- [Matherat 78b] P. MATHERAT : *A Chip for Low-Cost Raster-Scan Graphic Display* : Computer Graphics, **12** (3), août 1978, 181.
- [Matherat 80] P. MATHERAT, D. BOUTEAUD, N. FORGET, J. LEFRUN, J. P. MOREAU : *A High-Performance Integrated True Graphic Processor* : ESSCIRC 80, Grenoble.
- [Mead 80] C. MEAD et L. CONWAY : *Introduction to VLSI Systems* : Addison-Wesley, 1980.
- [NEC 82] NEC 1982 Catalog, 515.
- [Newman 80] W. M. NEWMAN et R. F. SPROULL : *Principles of Interactive Computer Graphics* : McGraw-Hill, 1979.
- [Noyce 77] R. NOYCE : *La Microélectronique* : La Révolution Microélectronique, Edition spéciale de Pour La Science, édition française de Scientific American, Diffusion Belin, 1979, 6.
- [Oguchi 81] TETSUJI OGUCHI, MISAO HIGUCHI, TAKASHI UNO, MICHINORI KAMAYA et MUNEKAZU SUZUKI (Nippon Electric Corp.) : *A Single-Chip Graphic Display Controller* : IEEE International Solid-State Circuits Conference, Digest of Technical Papers, 1981, 170.
- [Status 79] *Status Report of the Graphic Standards Planning Committee* : Computer Graphics, **13** (3), août 1979.
- [Sutherland 77] I. SUTHERLAND et C. MEAD : *La Microélectronique et la Théorie des Ordinateurs* : La Révolution Microélectronique, Edition spéciale de Pour La Science, édition française de Scientific American, Diffusion Belin, 1979, 112.