



**HAL**  
open science

# Etudes et évaluations d'architectures de pré-synthétiseurs d'images réalistes : HELIOS / GETRIS

K. Chibane

► **To cite this version:**

K. Chibane. Etudes et évaluations d'architectures de pré-synthétiseurs d'images réalistes : HELIOS / GETRIS. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 1986. Français. NNT: . tel-00322005

**HAL Id: tel-00322005**

**<https://theses.hal.science/tel-00322005>**

Submitted on 16 Sep 2008

**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.

# THESE

Présentée à

**L'INSTITUT NATIONAL POLYTECHNIQUE  
DE GRENOBLE**

pour obtenir le titre de  
docteur ingénieur

"informatique"

par

**Karim CHIBANE**

OOOOO

**ETUDES ET EVALUATIONS D'ARCHITECTURES DE  
PRE-SYNTHETISEURS D'IMAGES REALISTES.  
HELIOS / GETRIS**

OOOOO

Thèse soutenue le 13 Novembre 1986 devant la commission d'examen.

**L. BOLLIET**

**Président**

**R. CAUBET**

**C. LAUGIER**

**J. MERMET**

**F. MARTINEZ**

**Examineurs**



# INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : Daniel BLOCH  
Vice-Présidents : B. BAUDELET  
H. CHERADAME  
R. CARRE  
J.M. PIERRARD

Année universitaire 1984-1985

## Professeurs des Universités

### E.N.S.E.E.G.

BESSON	Jean	LOUCHET	François
BONNETAIN	Lucien	PARIAUD	Jean-Charles
BONNIER	Etienne	RAMEAU	Jean-Jacques
DURAND	François	SOHM	Jean-Claude
GUYOT	Pierre	SOUQUET	Jean-Louis

### E.N.S.E.R.G.

BARIBAUD	Michel	GENTY	Pierre
BLIMAN	Samuel	GUERIN	Bernard
BUYLE BODIN	Maurice	POUPOT	Christian
CHENEVIER	Pierre	SERMET	Pierre
COHEN	Joseph	ZADWORNÝ	François
COUMES	André		

### E.N.S.I.E.G.

BARRAUD	Alain	JOUBERT	Jean-Claude
BAUDELET	Bernard	JOURDAIN	Geneviève
BLOCH	Daniel	LACOUME	Jean-Louis
BRISSONNEAU	Pierre	LONGUEUE	Jean-Pierre
CAVAIGNAC	Jean-François	MASSELOT	Christian
CHARTIER	Germain	MORET	Roger
CHERUY	Arlette	PAUTHIENET	René
DURAND	Jean-Louis	PERRET	René
FELICI	Noël	PERRET	Robert
FOULARD	Claude	POLOUJADOFF	Michel
GAUBERT	Claude	SABONNADIÈRE	Jean-Claude
IVANES	Marcel	SCHILENKER	Claire
JALINIER	Jean-Michel	SCHILENKER	Michel
JAUSSAUD	Pierre		

### E.N.S.H.G.

BOIS	Philippe	LESPINARD	Georges
BOUVARD	Maurice	MOREAU	René
LESIEUR	Marcel	PIAU	Jean-Michel

E.N.S.I.M.A.G.

ANCEAU	François	MOSSIERE	Jacques
FONLUPT	Jean	ROBERT	François
LATOMBE	Jean-Claude	SAUCIER	Gabrielle
MAZARE	Guy	VEILLON	Gérard

U.E.R.M.C.P.P.

CHERADAME	Hervé	RENAUD	Maurice
CHIAVERINA	Jean	ROBERT	André
GANDINI	Alessandro	SILVY	Jacques

Professeurs Associés

BLACKWELDER	Ronald	ENSHG
HAYASHI	Hirashi	ENSIEG
PURDY	Gary	ENSEEG

Professeurs à l'Université des Sciences Sociales (Grenoble II)

BOLLIET	Louis
CHATELIN	Françoise

Chercheurs du C.N.R.S.

Directeurs de recherche :

CARRE	René
FRUCHARD	Robert
JORRAND	Philippe
VACHAUD	Georges

Maître de recherche :

ALLIBERT	Michel	JOURD	Jean-Charles
ANSARA	Ibrahim	KAMARINOS	Georges
ARMAND	Michel	KLEITZ	Michel
BINDER	Gilbert	LANDAU	Ioan-Dore
BORNARD	Guy	LASJAUNIAS	Jean-Claude
DAVID	René	MERMET	Jean
DESORTES	Jacques	MUNIER	Jacques
DRIOLE	Jean	PIAU	Monique
GIGNOUX	Damien	PORTESEIL	Jean-Louis
GIVORD	Dominique	THOLENCE	Jean-Louis
GUELIN	Pierre	VERDILLON	André
HOPFINGER	Emile	SUERY	Michel

Personnalités habilitées à diriger des travaux de recherche  
(Décision du conseil scientifique)

E.N.S.E.E.G.

ALLIBERT	Colette	HAMMOU	Abdelkader
BERNARD	Claude	MALMEJAC	Yves (CENG)
BONNET	Roland	MARTIN GARIN	Régina
CAILLET	Marcel	NGUYEN TRUONG	Bernadette
CHATILLON	Catherine	RAVAINE	Denis
CHATILLON	Christian	SAINFORT	(CENG)
COULON	Michel	SARRAZIN	Pierre
DIARD	Jean-Paul	SIMON	Jean-Paul
EUSTATHOPOULOS	Nicolas	TOUZAIN	Philippe
FOSTER	Panayotis	URBAIN	Georges(ODEILLO)
GALERIE	Alain		

E.N.S.E.R.G.

BARIBAUD	Michel	DOLMAZON	Jean-Marc
BOREL	Joseph	HERAULT	Jeanny
CHOVET	Alain	MONLLOR	Christian
CHEHIKIAN	Alain		

E.N.S.I.E.G.

BORNARD	Guy	LEJEUNE	Gérard
DESCHIZEAUX	Pierre	MAZUER	Jean
GLANGEAUD	François	PERARD	Jacques
KOFMAN	Walter	REINISCH	Raymond

E.N.S.H.G.

ALEMANY	Antoine	OBLED	Charles
BOIS	Daniel	ROWE	Alain
DARVE	Félix	VAUCLIN	Michel
MICHEL	Jean-Marie	WACK	Bernard

E.N.S.I.M.A.G.

BERT	Didier	DELLA DORA	Jean
CALMET	Jacques	FONLUPT	Jean
COURTIN	Jacques	SIFAKIS	Joseph
COURTOIS	Bernard		

U.E.R.M.C.P.P.

CHARUEL	Robert		
---------	--------	--	--

C.E.N.G.

CADET	Jean	NIFENECKER	Hervé
COEURE	Philippe (LETI)	PERROUD	Paul
DELIAYE	Jean-Marc (STT)	PEUZIN	Jean-Claude(LETI)
DUPUY	Michel (LETI)	TAIEB	Maurice
JOUBE	Hubert (LETI)	VINCENDON	Marc
NICOLAU	Yvan (LETI)		

Laboratoires extérieurs

C.N.E.T.

DEMOULIN  
DEVINE  
GERBER

Eric  
R.A.B.  
Roland

MERCKEL  
PAULEAU

Gérard  
Yves

I.N.S.A. Lyon

GAUBERT

C.

\*\*\*\*\*

DEDICACE :

-----

A LA MEMOIRE DE SMAIL





REMERCIEMENTS :

Je tiens tout d'abord a remercier Monsieur Louis Bolliet, Professeur a l'universite de Grenoble, d'avoir si gentillemeent accepte de me faire l'honneur de presider ce jury.

Je voudrais exprimer ma profonde gratitude a l'egard de Monsieur Jean Mermet, Directeur du laboratoire ARTEMIS pour avoir accepte de juger mon travail et de sieger dans ce jury.

J'aimerais remercier Monsieur Francis Martinez, professeur d'universite et responsable du theme Images et Communications Graphiques Interactives au sein du laboratoire ARTEMIS; de m'avoir accueilli, propose ce sujet de these et guide avec beaucoup de patience, durant la redaction de ce document.

J'aimerais remercier Monsieur Rene Caubet, Maitre de conference a l'universite Paul Sabatier de Toulouse; pour avoir accepte de juger techniquement cette etude. Ses conseils et ses critiques ont permis une amelioration finale du document.

J'aimerais remercier Monsieur Christian Laugier, Ingenieur I.N.R.I.A pour avoir si gentillemeent accepte de participer a ce jury et d'evaluer techniquement cette etude.

Enfin, je tiens a exprimer ma gratitude aux membres du laboratoire ARTEMIS pour leur gentillesse et pour ces annees passees ensemble dans un climat de camaraderie et de bonne entente, en particulier Messieurs J.F Grabowieky, P. Genoud, G. Kuntz, P. Uvietta et bien d'autres ...

ATTENTIONS PARTICULIERES :

Je tiens a remercier Monsieur Mohamed Arrada, Ingenieur d'etudes de la societe CORI et specialiste en telecommunications; pour m'avoir soutenu moralement et techniquement pendant toute la duree de ma these.

J'aimerais exprimer mes sincereres amities aux membres du laboratoire de micro-informatique pour leur soutien durant le developpement des maquettes de test des pre-synthetiseurs; tout particulierement Messieurs P. Collovati, J.P SERPAGGI, R BOUTTAZ, P DIGIACOMO, J.L DE MARINIS et bien d'autres ...



## INTRODUCTION

Aujourd'hui l'image de par la quantité d'informations qu'elle renferme, peut être définie comme l'un des supports de communication les plus puissants. Les progrès des sciences de l'informatique ont permis au fil des années de construire, de manipuler et de visualiser des objets graphiques de plus en plus complexes, et dotés de caractéristiques de visualisation de plus en plus sophistiquées appelées attributs, tels que la couleur, l'éclairage, la réflexion, les ombres portées, les textures etc.

De même les algorithmes de traitement de scènes aussi bien bi-dimensionnelles que tri-dimensionnelles ont permis de synthétiser des images de plus en plus cohérentes, de moins en moins imparfaites. (Découpage bi et tridimensionnel, l'élimination des parties cachées, etc) En un mot, on synthétise actuellement des images dites de meilleure qualité, ou encore à haut degré de réalisme, l'utilisateur recherchant des terminaux graphiques de plus en plus performants et offrant des fonctions évoluées.

Au cours de cette évolution, le concept de terminal graphique interactif s'est élargi, les frontières entre l'application c'est-à-dire l'utilisateur, et les fonctions offertes par le périphérique sont désormais difficiles à délimiter. Ainsi la répartition des différentes tâches liées à la synthèse d'images varie d'un constructeur à un autre. Cependant cette surcharge de traitement au niveau du périphérique implique une évolution de l'architecture, utilisant des processeurs puissants et une "intelligence" complexe, adaptée à la synthèse d'image quel que soit le degré de réalisme considéré. Il est nécessaire de redéfinir ce qu'est un terminal graphique interactif.

Du point de vue philosophique E.Green le considère comme un ensemble de techniques spécifiques orienté vers le dialogue entre l'homme et l'ordinateur par le biais d'un dispositif d'affichage. Topologiquement, il s'agit d'une chaîne de traitement dont une extrémité est constituée par l'ordinateur et l'autre par l'unité d'affichage. C'est donc l'unité intermédiaire qui est à redéfinir. ( MOL 76 )

En effet, il s'agit maintenant d'une chaîne complète dotée d'une "intelligence" microprogrammée ou câblée, capable de gérer les différentes fonctions offertes par le terminal, ainsi que l'interaction avec l'utilisateur. Plusieurs entités caractéristiques sont définies :

- Les chaînes de traitement : pré-synthèse et post-synthèse,
- Les extrémités : le calculateur hôte et le dispositif d'affichage,
- Le noyau : la mémoire d'image représentant l'écran.

La pre-synthese opere en "amont" de la memoire d'image, la post-synthese opere en "aval" de la memoire d'image. L'interaction est definie comme la communication possible entre le calculateur et le dispositif d'affichage, a travers et au moyen de ces deux sous-chaines de traitement.

Cette these etudie principalement les problemes lies a la pre-synthese d'images, en essayant de definir des architectures de pre-synthetiseurs, a base de processeurs et des operateurs de calcul performants. De plus des modeles de communication entre la pre-synthese et le calculateur hote d'une part et entre la pre-synthese et la post-synthese d'autre part sont a determiner. Dans cette etude le lecteur pourra decouvrir l'evolution du concept de pre-synthese a travers un exemple particulier le terminal graphique interactif HELIOS jusqu'a sa version commerciale GETRIS.

Le chapitre I presente l'aspect theorique de l'approche d'une conception d'un systeme graphique interactif a partir des concepts de hierarchisation et d'ordonnancement des processus de synthese d'images.

Le chapitre II expose une proposition de nouvelle architecture pour la phase de pre-synthese et met l'accent sur les differents points nevralgiques de cette architecture.

Le chapitre III decrit une realisation de pre-synthetiseur bi-dimensionnel et monoprocesseur pour le terminal graphique interactif HELIOS.

Le chapitre IV montre l'evolution possible au niveau de la pre-synthese et decrit une architecture de pre-synthetiseur tri-dimensionnel et monoprocesseur.

Le chapitre V decrit les tendances actuelles de la pre-synthese, c'est-a-dire la mise en oeuvre d'architectures paralleles et multi-processeurs aussi bien au niveau operatoire, qu'au niveau processus de synthese, et ceci en s'interessant particulierement aux problemes de performances de tels systemes par rapport aux contraintes de temps-reel.

Enfin le chapitre VI decrit les options retenues pour le terminal graphique interactif HELIOS quant a la version commercialisee par la societe grenobloise GETRIS-IMAGES. Une etude comparative avec d'autres stations de travail du marche actuel, sera presentee.

L'ambition de cette etude n'est pas de remettre en question les differents concepts de synthese d'images ou les differentes architectures proposees par de nombreux constructeurs, mais plutot de mieux apprecier les problemes lies a la synthese d'images (pre/post-synthese) afin de definir et de realiser des architectures de terminaux graphiques interactifs, modulaires et mieux adaptees aux differents domaines d'application.

## SOMMAIRE

CHAPITRE I	ARCHITECTURES DE SYSTEMES DE SYNTHÈSE D'IMAGES :	
I.1	APPROCHE HIERARCHISEE DE LA CONCEPTION :	I-2
I.1.1	DEFINITION ET TERMINOLOGIE :	I-2
I.1.2	STRUCTURE D'UN SYSTEME GRAPHIQUE :	I-4
I.1.2.1	METHODE DE CONCEPTION :	I-4
I.1.2.2	ORDONNANCEMENT DU PROCESSUS :	I-6
I.2	ARCHITECTURES A "N" SYNTHETISEURS :	I-8
I.2.1	CONFIGURATION MINIMALE :	I-9
I.2.2	CONFIGURATION EVOLUEE :	I-9
I.2.3	CONNEXION A UN CALCULATEUR :	I-10
I.2.4	CONFIGURATION MULTIPLE :	I-11
I.2.5	DIFFICULTES DE LA CONCEPTION :	I-12
I.2.5.1	REALISATION D'UN SYNTHETISEUR :	I-13
I.2.6	ETUDE DE CAS :	I-13
I.3	NOTION DE PARALLELISME :	I-15
I.3.1	ORDONNANCEMENT DU PROCESSUS :	I-15
I.3.2	SYNTHETISEURS MONOPROCESSUS :	I-16
I.3.2.1	DIFFERENTS TYPES D'ARCHITECTURE :	I-17
I.3.2.2	ARCHITECTURE MULTIPROCESSUS :	I-17
I.3.2.3	REDUCTION DU NOMBRE D'OPERATEURS :	I-18
I.3.2.4	STRATEGIES DE CONCEPTION :	I-19
I.4	PRE-SYNTHÈSE D'IMAGES ET REALISME :	I-22
I.4.1	PERFORMANCES D'UN SYSTEME :	I-22
I.4.2	ROLE PRIMORDIAL DE LA PRE-SYNTHÈSE :	I-24
I.4.2.1	ARCHITECTURE MINIMALE :	I-24
I.4.2.2	DEGRE DE REALISME :	I-24
I.4.2.3	REPARTITION DES TACHES :	I-25
I.5	DIFFERENTS TYPES DE PRE-SYNTHÈSE :	I-27
I.5.1	PRE-SYNTHÈSE DE BASE :	I-27
I.5.2	PRE-SYNTHÈSE EVOLUEE :	I-30

CHAPITRE II

LA CONFIGURATION "CALCULATEUR SATELLITE" :

II.1	CARACTERISTIQUES D'UN T.G.I :	II-2
II.1.1	PRESENTATION GENERALE :	II-3
II.1.2	PROCESSUS DE VISUALISATION :	II-3
II.1.3	ORDONNANCEMENT DU PROCESSUS :	II-4
II.1.4	ARCHITECTURE HIERARCHISEE :	II-5
II.1.5	DESCRIPTION DES PROCESSEURS :	II-7
II.2	INTERFACE EN AMONT :	II-10
II.2.1	LE "CALCULATEUR SATELLITE" :	II-10
II.2.2	MODES D'UTILISATION DU T.G.I :	II-11
II.2.3	DIALOGUE ET ECHANGES D'INFORMATIONS :	II-12
II.3	INTERFACE EN AVAL :	II-12
II.3.1	MODULE DE COMMUNICATION :	II-12
II.3.2	DESCRIPTION DES REGISTRES :	II-13
II.3.3	PROTOCOLE D'ECHANGE :	II-15
II.3.4	EVALUATION DES ACCES :	II-15
II.4	PRINCIPALES FONCTIONS-HELIOS :	II-18
II.4.1	ATTRIBUTION ET CONSULTATION :	II-19
II.4.2	DESCRIPTION ET VISUALISATION :	II-20
II.4.2.1	GENERATION DE VECTEURS :	II-21
II.4.2.2	GENERATION DE CARACTERES :	II-23
II.5	REPARTITION DES TACHES :	II-25
II.5.1	REPLISSAGE DE TACHES POLYGONALES :	II-25
II.5.2	TYPE DE PRE-SYNTHESE :	II-28
II.6	DIFFERENTS TYPES D'INTERACTIVITES :	II-29
II.6.1	DISPOSITIFS DE DIALOGUE :	II-29
II.6.2	FONCTIONS INTERACTIVES :	II-29

CHAPITRE III

ARCHITECTURE A PRE-SYNTHESEUR 2.D :

III.1	PRE-SYNTHESEUR MICROPROGRAMME :	III-3
III.1.1	CARACTERISTIQUES TECHNIQUES :	III-4
III.1.1.1	GESTION DES ENTREES/SORTIES :	III-4
III.1.1.2	MEMOIRES LOCALES :	III-5
III.1.1.3	INTERFACE D'ACCES AU M.C :	III-6
III.1.2	GESTION DU PRE-SYNTHESEUR :	III-7
III.2	MODES DE COMMUNICATION :	III-9
III.2.1	PROTOCOLE D'ECHANGE :	III-10
III.2.2	FORMAT ET CODAGE DES COMMANDES :	III-13
III.2.2.1	GESTION DES PARAMETRES :	III-15
III.2.2.2	OPTIMISATION DE LA COMMUNICATION :	III-18
III.3	CONTROLE DE LA POST-SYNTHESE :	III-21
III.3.1	LE MODULE DE COMMUNICATION :	III-21
III.3.1.1	DESCRIPTION DES REGISTRES :	III-22
III.3.1.2	REGISTRES SUPPLEMENTAIRES :	III-25
III.3.2	EVALUATION DES ECHANGES :	III-28
III.3.2.1	MECANISME DE LECTURE :	III-28
III.3.2.2	MECANISME D'ECRITURE :	III-29
III.4	ETUDE DES PERFORMANCES DU T.G.I :	III-30
III.4.1	REPertoire DE COMMANDES :	III-30
III.4.1.1	GENERATION DE VECTEURS :	III-32
III.4.1.2	GENERATION DE POLYGONES :	III-35
III.4.1.3	GENERATION DE CARACTERES :	III-38
III.5	TYPES D'INTERACTIVITE DU T.G.I :	III-40
III.5.1	DIALOGUE ET CONSULTATION :	III-41
III.5.2	INTERACTIONS SPECIALES :	III-42



CHAPITRE IV

REALISATION D'UNE MAQUETTE D'EVALUATION 3.D :

IV.1	CONCEPTS LIES AU TRAITEMENT D'UNE SCENE 3.D :	IV-2
IV.1.1	TRANSFORMATIONS GEOMETRIQUES :	IV-4
IV.1.2	PROJECTION EN PERSPECTIVE :	IV-6
IV.1.3	DECOUPAGE TRI-DIMENSIONNEL :	IV-8
IV.1.4	ELIMINATION DES PARTIES CACHEES :	IV-11
IV.1.5	PASSAGE DANS LE REPERE-ECRAN :	IV-13
IV.2	FONCTIONNALITES DU PRE-SYNTHETISEUR :	IV-14
IV.2.1	GESTION DE L'ESPACE MEMOIRE :	IV-16
IV.2.2	GESTION DE LA COMMUNICATION :	IV-17
IV.2.3	FONCTIONS DU PRE-SYNTHETISEUR :	IV-20
IV.3	ARCHITECTURE DU PRE-SYNTHETISEUR 3.D :	IV-21
IV.3.1	CHOIX DU PROCESSEURS 16/32 BITS :	IV-22
IV.3.2	CARACTERISTIQUES TECHNIQUES :	IV-24
IV.3.2.1	UNITES D'ENTREES / SORTIES :	IV-25
IV.3.2.2	MEMOIRES LOCALES :	IV-26
IV.3.2.3	INTERFACE SPECIALISEE :	IV-28
IV.3.2.4	EVALUATION DES ECHANGES :	IV-29
IV.4	SOLUTION MICRO-PROGRAMMEE :	IV-30
IV.4.1	STRUCTURE DU MICROLOGICIEL 3.D :	IV-33
IV.4.2	METHODE D'EVALUATION :	IV-34
IV.4.3	PERFORMANCES DU MICROLOGICIEL :	IV-36

CHAPITRE V

EVALUATIONS D'ARCHITECTURES PARALLELES :

V.1	MULTI-PROCESSEUR 3.D CENTRALISE :	V-2
V.1.1	CO-PROCESSEURS ARITHMETIQUES :	V-2
V.1.2	MULTIPLIEURS / DIVISEURS RAPIDES :	V-4
V.2	OPERATEURS A BASE DE MULTIPLIEURS :	V-7
V.2.1	OPERATEUR MATRICIEL PARTIEL : O.M.P :	V-7
V.2.2	OPERATEUR MATRICIEL COMPLET : O.M.C :	V-8
V.2.3	ETUDE DES PERFORMANCES :	V-10
V.3	OPERATEURS SPECIFIQUES DE TRANSFORMATION :	V-11
V.3.1	TRANSFORMATION-ECRAN :	V-11
V.3.1.1	DIFFERENTES PROPOSITIONS :	V-11
V.3.1.2	ETUDE DES PERFORMANCES :	V-14
V.3.2	PROJECTION EN PERSPECTIVE :	V-15
V.3.2.1	DIFFERENTES PROPOSITIONS :	V-16
V.3.2.2	ETUDE DES PERFORMANCES :	V-19
V.3.3	PHASE DE DECOUPAGE TRI-DIMENSIONNEL :	V-19
V.4	PROCESSUS DE VISUALISATION 3.D AMELIORE :	V-21
V.4.1	PERFORMANCES RELATIVE A CHAQUE PHASE :	V-23
V.4.2	MULTI-PROCESSEUR 3.D DECENTRALISE :	V-25
V.4.2.1	PROCESSEURS SYSTOLIQUES :	V-26
V.4.2.2	ETUDE DE PROCESSEURS SPECIFIQUES :	V-28
V.4.2.3	CAS DES PROCESSEURS EN TRANCHES :	V-30

CHAPITRE VI

LE CONCEPT DE LA STATION DE TRAVAIL :

VI.1	LA NOUVELLE TENDANCE : . . . . .	VI-2
VI.1.1	EVOLUTION DES ARCHITECTURES : . . . . .	VI-2
VI.1.2	SYSTEMES MULTI-PROCESSEUR : . . . . .	VI-5
VI.2	STATION GRAPHIQUE GETRIS-3.D : . . . . .	VI-6
VI.2.1	LA PARTIE CALCULATEUR : . . . . .	VI-6
VI.2.2	PARTIE GENERATEUR D'IMAGES : . . . . .	VI-7
VI.2.3	ETUDE DES PERFORMANCES : . . . . .	VI-10
VI.3	STATION GRAPHIQUE METHEUS : . . . . .	VI-10
VI.3.1	PARTIE CALCULATEUR : . . . . .	VI-11
VI.3.2	PARTIE GENERATEUR D'IMAGES : . . . . .	VI-11
VI.3.3	ETUDE DES PERFORMANCES : . . . . .	VI-12
VI.4	STATION GRAPHIQUE CHROMATICS : . . . . .	VI-13
VI.4.1	PARTIE CALCULATEUR : . . . . .	VI-13
VI.4.2	PARTIE GENERATEUR D'IMAGES : . . . . .	VI-13
VI.4.3	ETUDES DES PERFORMANCES : . . . . .	VI-13
VI.5	STATION GRAPHIQUE : SILICON-GRAPHS . . . . .	VI-15
VI.5.1	LE COMPOSANT "GEOMETRY-ENGINE" : . . . . .	VI-15
VI.5.2	PARTIE CALCULATEUR : . . . . .	VI-16
VI.5.3	PARTIE GENERATEUR D'IMAGES : . . . . .	VI-17
VI.5.4	ETUDE DES PERFORMANCES : . . . . .	VI-18
VI.6	STATION GRAPHIQUE : APOLLO-DN.550-660 . . . . .	VI-18
VI.6.1	PARTIE CALCULATEUR : . . . . .	VI-18
VI.6.2	PARTIE GENERATEUR D'IMAGES : . . . . .	VI-19
VI.6.3	ETUDE DES PERFORMANCES : . . . . .	VI-20

CHAPITRE I

ARCHITECTURES DE SYSTEMES DE SYNTHESE D'IMAGES :

REFERENCES BIBLIOGRAPHIQUES :

( FER 81 ), ( GRA 80 ), ( GAU 85 ), ( LAG 85 ),  
( MAT 78 ), ( MAR 82 ), ( MAR 84 ), ( MOL 76 ),  
( NES 79 ), ( RTS 83 ), ( CIG 84 ).



## I.1 APPROCHE HIERARCHISEE DE LA CONCEPTION :

## I.1.1 DEFINITION ET TERMINOLOGIE :

## LES ENTITES DE BASE :

La conception d'un terminal graphique interactif s'appuie sur la notion de chaîne de traitement. Comme l'indique la figure I.1, une extrémité est le calculateur hôte de l'application, l'autre extrémité est le dispositif d'affichage en l'occurrence, le moniteur TV de visualisation.

Chaque sous-chaîne de l'architecture est chargée d'effectuer une tâche spécifique. On considère deux catégories principales de tâches. La première concerne les tâches relatives à un pré-traitement graphique asynchrone.

Il s'agit de la PRE-SYNTHESE qui assure les échanges avec le calculateur hôte; La seconde concerne les tâches relatives à un post-traitement graphique synchrone du rythme vidéo. Il s'agit de la POST-SYNTHESE qui assure le pilotage du moniteur T.V de visualisation. Ce découpage de la chaîne de traitement est indiqué dans la figure suivante.

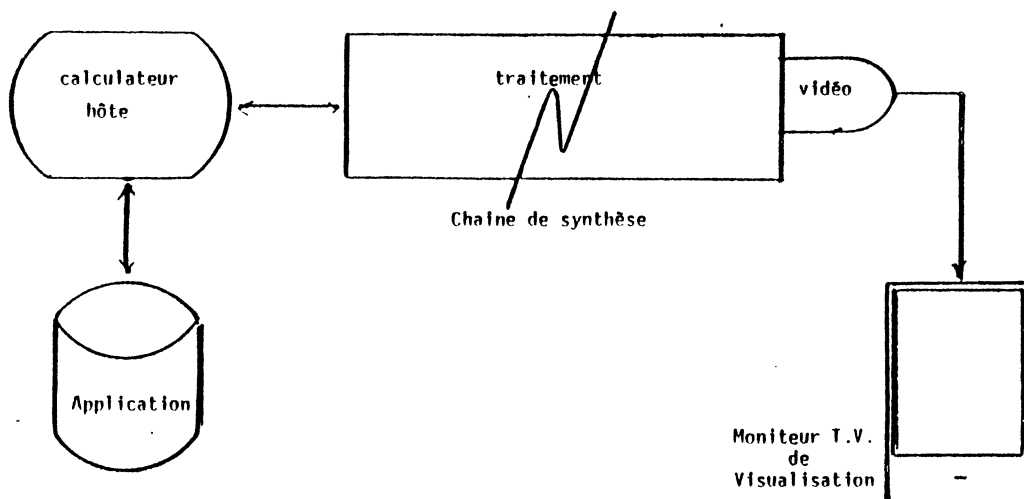


Figure No I.1

De plus, on définit le synthétiseur comme un étage d'une des deux sous-chaînes de traitement. La pré-synthèse est constituée d'un groupe de pré-synthétiseurs, il en est de même pour les post-synthétiseurs de la post-synthèse. L'architecture d'un synthétiseur quelconque est constituée d'un ou de plusieurs processeurs. Un processeur est une entité physique capable d'exécuter une fonction donnée.

Ce processeur peut posséder des ressources locales :

- Mémoires de stockages,
- Bus d'échanges internes,
- Supports de communication avec l'extérieur.

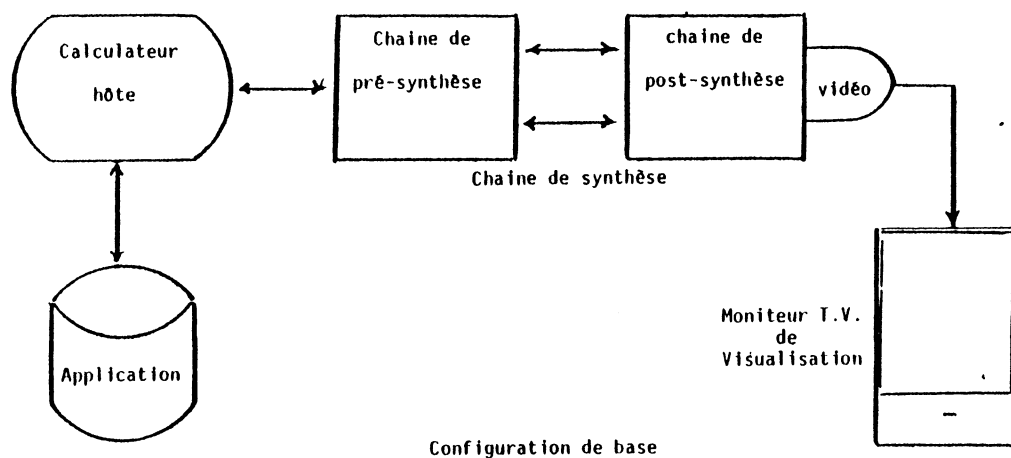


Figure No 1.2

Ainsi cette terminologie permettra dans la suite de l'étude d'identifier et d'analyser chaque entité de base au sein du terminal graphique interactif HELIOS. De plus, la performance globale d'un système dépend directement de deux critères de conception. Le premier est fonction de l'ordonnancement des tâches et des étapes de traitement associées. Il s'agit de la répartition des traitements entre les différents synthétiseurs d'une des deux sous-chaines de traitement. Le second découle de la détermination et de l'adaptation des différents processeurs nécessaires, en particulier de la méthode de réalisation de ces processeurs au sein d'une architecture. L'objectif est de développer des architectures répondant de façon optimale à ces deux critères. ( MAR 82 )

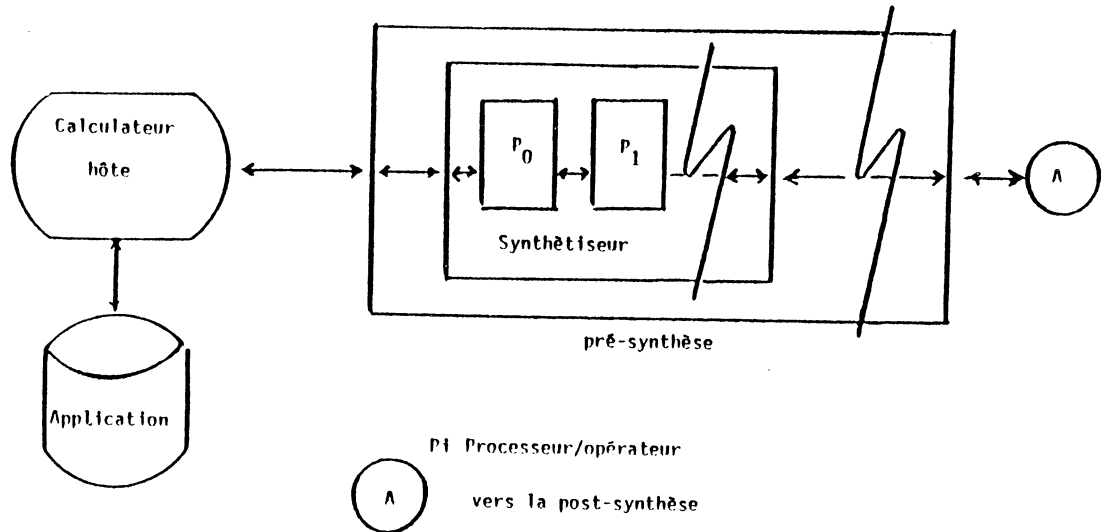


Figure No I.3

Enfin l'interactivité est définie comme étant la faculté de designer et d'identifier une information relative à une entité graphique visible sur l'écran de visualisation. (le point dans la plupart des cas) La qualité de l'interaction d'un système graphique est fonction de la qualité des dispositifs d'interaction. De plus, la possibilité d'accéder à des informations disponibles au niveau d'un synthétiseur quelconque et ceci quelle que soit la sous-chaine considérée, garantit la cohérence et la symétrie de la conception.

### I.1.2 STRUCTURE D'UN SYSTEME GRAPHIQUE :

#### I.1.2.1 METHODE DE CONCEPTION : -

L'approche la plus courante de conception d'un système graphique est une approche hiérarchisée ou partitionnement en couches. Cette approche est inhérente au processus de synthèse d'images. La complexité de ce processus est déterminée par le degré de réalisme des images considérées. Ainsi le système est partitionné en un ensemble de couches. Une couche "i" est considérée comme un synthétiseur de traitement intermédiaire pour la couche de rang supérieur "i+1". L'analogie entre la chaîne de traitement décrite dans l'introduction et le partitionnement en couches, est immédiate, une couche correspond à un maillon de la chaîne, cette correspondance est représentée dans la figure I.4.



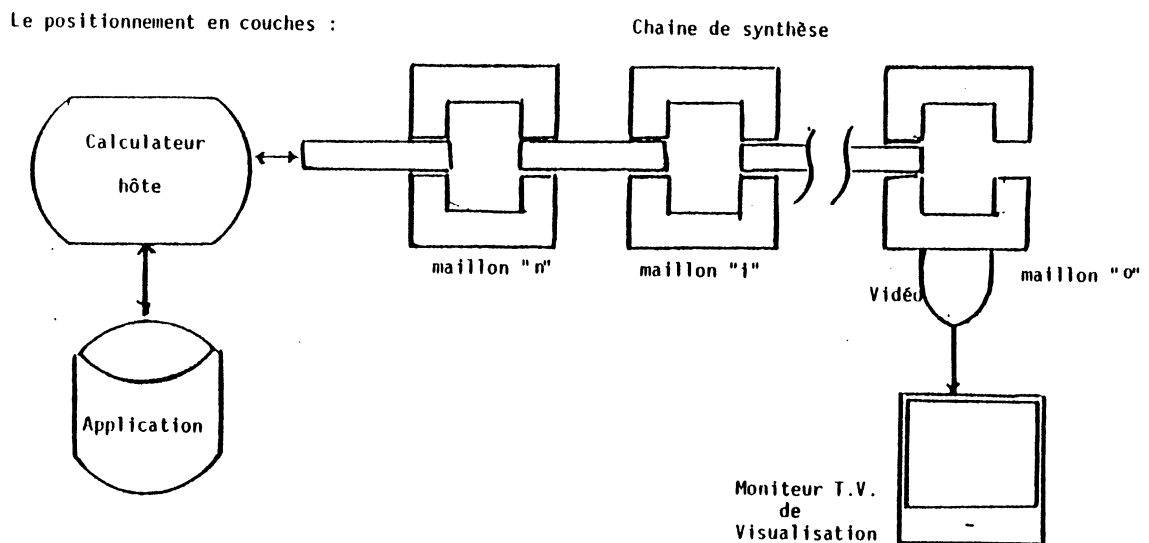


Figure No I.4

La couche 'n' correspond au synthetiseur de niveau le plus bas. Il s'agit de la logique cablee de pilotage du moniteur T.V de visualisation. Chaque couche est caracterisee par sa propre architecture. Elle realise un processus de synthese intermediaire specifique. Ce processus est fonction de la localisation du synthetiseur, (de la couche) dans la chaine de traitement. La complexite d'un niveau de l'architecture correspond :

- aux types d'informations acceptees par le synthetiseur,
- aux types d'informations produites par le synthetiseur,
- a la vitesse d'execution des operations associees au synthetiseur.

Le synthetiseur est constitue d'un ensemble d'operateurs contribuant a la realisation du procesus qui lui est associe, il peut necessiter plusieurs processeurs selon la solution adoptee par le concepteur.

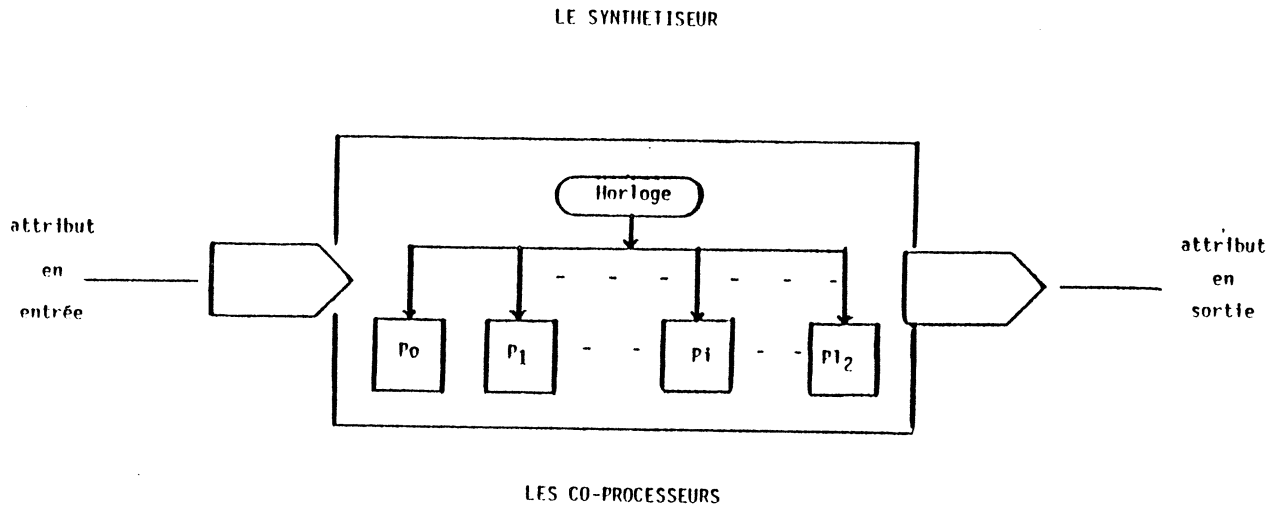


Figure No I.5

Le second concept est la hiérarchisation d'un tel système signifiant l'ordonnement du processus de synthèse. Il est nécessaire d'étudier les différents processus coopérants à la synthèse finale d'une image pour un degré de réalisme donné. La démarche de conception consiste à effectuer une décomposition en trois phases :

- L'ordonnement des opérations liées au processus de synthèse,
- La répartition des opérations au niveau des différents synthétiseurs,
- Le choix d'une méthode de réalisation des différents opérateurs.

La complexité de cette étude réside dans le fait que les applications graphiques produisent des résultats variés et innombrables. L'idéal serait le système graphique GENERAL et UNIVERSEL aux performances satisfaisant un grand nombre d'utilisateurs. (temps réel, réalisme, animation, applications interactives.) Cependant des gammes de terminaux sont proposées pour des domaines d'applications spécifiques tel que la C.A.O., C.F.A.O., Animation. La flexibilité de ce type d'équipement est nettement amoindrie.

#### 1.1.2.2 ORDONNEMENT DU PROCESSUS : -

L'ordonnement du processus de synthèse découle directement du type d'images considérées. L'image renferme une classe d'informations synthétiques qui sont :

- L'identite
- La morphologie
- L'aspect
- La geometrie
- L'eclairage
- La structure

La visualisation est l'etape finale du processus de synthese. Le resultat est obtenu apres interpretation de ces informations a differents niveaux de la pre-synthese et de la post-synthese. Les types d'informations possibles sont :

- Semantiques : nature de l'information,
- Syntaxiques : modelisation de l'information,
- Structurels : classe d'entites de base de l'information.

L'ordonnement du processus consiste a definir des synthetiseurs manipulant les classes d'informations. Le synthetiseur est considere comme une unite de traitement indivisible. Cette unite est capable de gerer quatre taches principales :

- L'attribution,
- La consultation,
- La description,
- La visualisation.

Par exemple, dans le cas d'un balayage de trame, le processus d'attribution est le balayage TV, le processus de visualisation est la luminescence de l'ecran enfin la description et la consultation sont determinees par le dispositif d'interaction associe au moniteur TV. (un reticule)

Chaque tache constitue un processus intermediaire contribuant au processus de synthese general. Le processus general de synthese est une composition des differents processus au niveau des differents synthetiseurs de la chaine de traitement du systeme graphique. Le concepteur determine la composition des processus cites. Il peut privilegier une information synthetique donnee. Par exemple, pour une image realiste il privilegie l'aspect. (eclairage, reflexion, couleur, texture.)

## REMARQUE :

Generalement, les processus d'attribution et de consultation caracterisent le dialogue entre le systeme et le programme d'application. Les processus de description et de visualisation caracterisent le dialogue entre le systeme et l'utilisateur. Le synthetiseur accepte en entree des types d'information qui constituent son UNIVERS INITIAL. Il produit en sortie des types d'information qui constituent son UNIVERS FINAL.

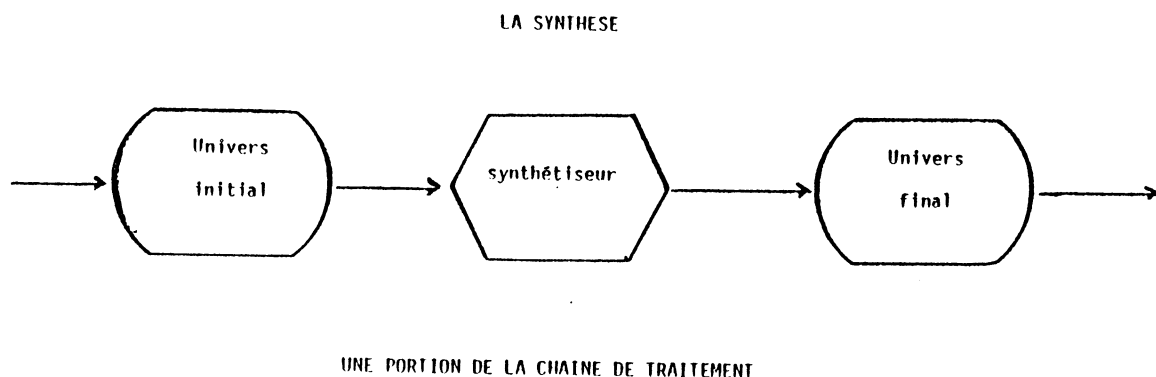


Figure No I.6

L'univers final d'un synthetiseur "i" constitue l'univers initial du synthetiseur "i+1". L'univers initial du systeme graphique est constitue de l'ensemble des types d'objets graphiques manipules. L'univers final du systeme graphique est constitue de l'ensemble des points de l'ecran du moniteur TV de visualisation, il est produit par le dernier synthetiseur de la chaine, c'est-a-dire le module de generation video.

## RECAPITULATIF :

Un systeme graphique est un ensemble de synthetiseurs. Chaque synthetiseur est constitue d'un ensemble d'operateurs. Le synthetiseur de depart est l'application programme implantee au niveaux du calculateur hote. Le synthetiseur final est le moniteur TV de visualisation. Les quatre processus fondamentaux doivent etre geres, soit directement par l'application, soit indirectement par l'intermediaire d'un logiciel specifique a partir de primitives accessibles en langage evolue de type PASCAL ou C. Ce point fait l'objet d'une these au sein de l'equipe communication graphique du Laboratoire ARTEMIS ( GEG 85 ).

## I.2 ARCHITECTURES A "N" SYNTHETISEURS :

## I.2.1 CONFIGURATION MINIMALE :

Cette architecture est constituee de deux synthetiseurs, le synthetiseur initial le calculateur hôte, Le synthetiseur final la console graphique. Le modele de communication inter-synthetiseurs est le plus souvent une ligne de communications asynchrone (1200 - 9600 bauds). La chaine de post-synthese est constituee d'un synthetiseur; le module de generation du signal video qui realise la visualisation et le rafraichissement de la memoire d'images appelee aussi memoire de trames. La chaine de pre-synthese est constituee d'un synthetiseur; l'application programme.

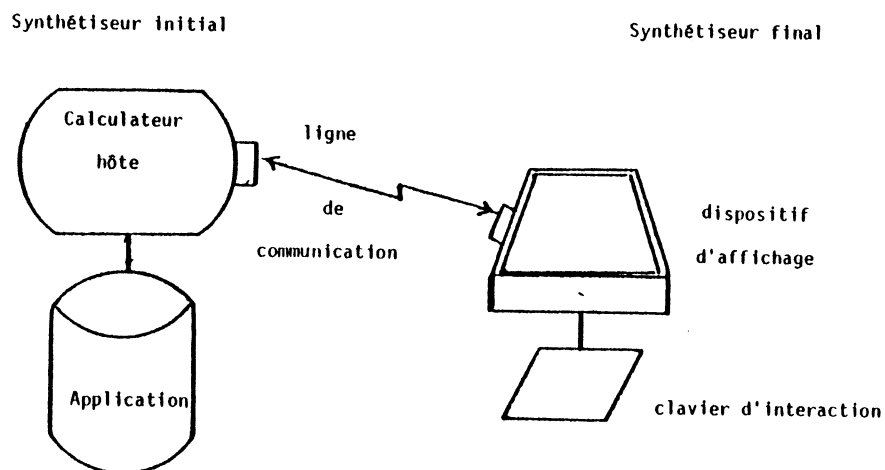


Figure No I.7

Les processus d'attribution et de visualisation de points, vecteurs, caracteres etc, sont geres au niveau du logiciel de l'application. Les processus de description et de consultation sont aussi a la charge du logiciel graphique de l'application. Dans cette configuration, l'application est asservie aux caracteristiques de la console graphique, en particulier aux possibilites offertes par la post-synthese.

## I.2.2 CONFIGURATION EVOLUEE :

Cette architecture est dotee d'un synthetiseur intermediaire qui assure un ensemble de taches permettant de decharger l'application. Il gere les quatre processus fondamentaux par l'intermediaire de commandes que peut lui transmettre le calculateur hôte. Ce synthetiseur intermediaire est constitue d'un ensemble d'entites constituant son architecture :

- L'unite de traitement : un processeur V.L.S.I quelconque,

- La memoire locale,
- Les supports de communication : serie ou parallele,
- Le micrologiciel charge de l'attribution, la description, la visualisation et la consultation.

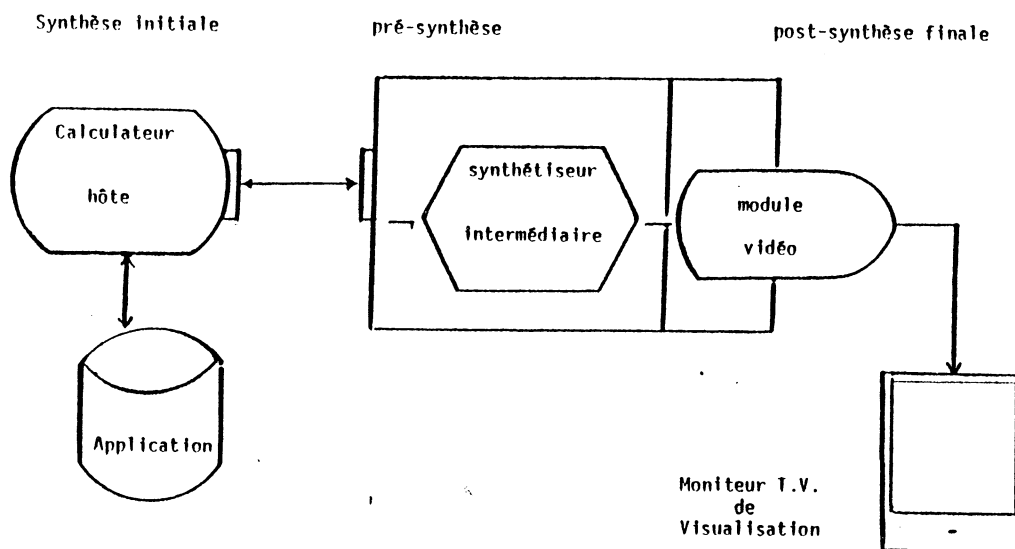


Figure No I.8

Dans cette architecture a trois synthetiseurs, l'interet fondamental est d'asservir le terminal graphique a l'application par le biais du synthetiseur intermediaire et non l'inverse.

### I.2.3 CONNEXION A UN CALCULATEUR :

C'est une architecture a trois synthetiseurs. le synthetiseur intermediaire est un calculateur independant integre a la chaine de synthese. On l'appelle calculateur satellite, il appartient a la chaine de pre-synthese. Il est independant de la console graphique constituant la chaine de post-synthese.

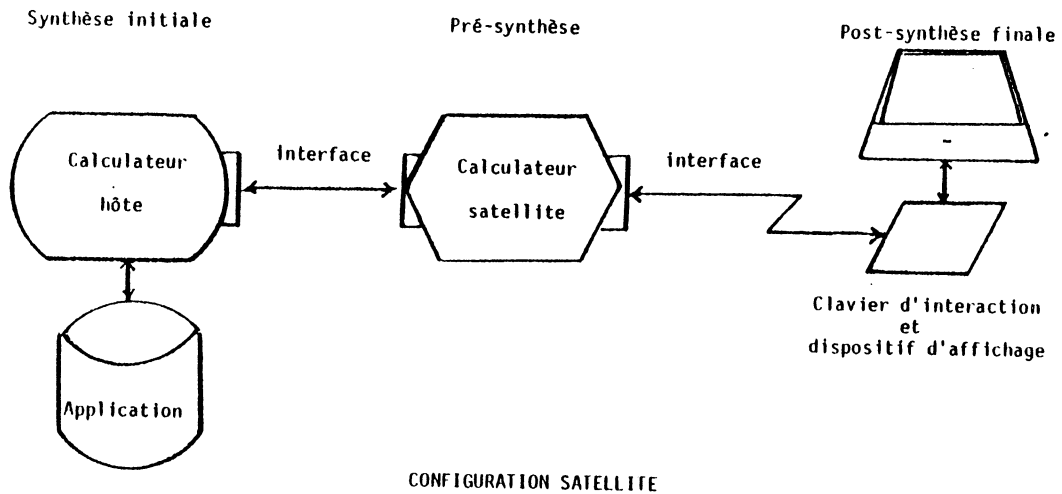


Figure No I.9

La structure de ce synthetiseur intermediaire est plus complete puisque l'application dispose des ressources du calculateur satellite, en particulier des memoires de masse qui permettent un archivage local. L'independance entre le programme application et la console grahique est accentuee. Ainsi, le jeu de commande accepte par le calculateur satellite peut etre de plus en plus evolue. Les quatre processus fondamentaux sont a la charge du calculateur satellite.

#### I.2.4 CONFIGURATION MULTIPLE :

Dans ce cas, la fonctionnalite de l'architecture est differente. Le synthetiseur intermediaire se presente comme un frontal auquel est connecte un ensemble de consoles graphiques. L'application implantee au niveau du calculateur hote pilote une console graphique simulee par une couche logicielle.

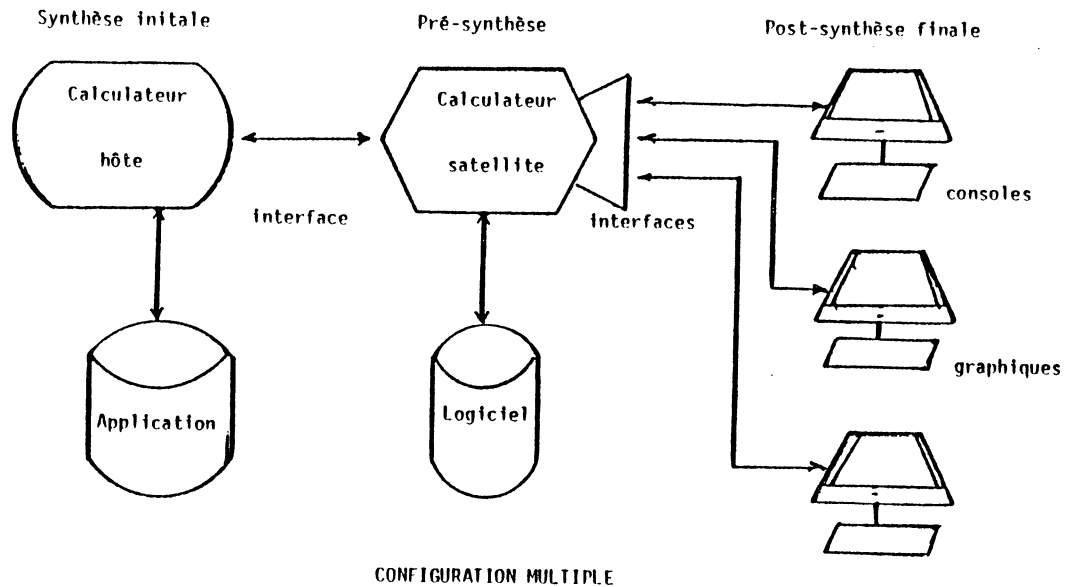


Figure No I.10

On peut entrevoir dans ce type de configuration la notion de terminal graphique interactif virtuel independant des caracteristiques materielles de la chaine de post-synthese d'un terminal donne. Ce type de configuration ne constitue pas notre pole d'interet.

#### I.2.5 DIFFICULTES DE LA CONCEPTION :

##### COHERENCE ET SYMETRIE

La qualite de la fonctionnalite d'un terminal graphique interactif ne reside pas uniquement dans la puissance des commandes graphiques qu'il propose, mais plutot dans la coherence de l'ordonnement du processus de synthese. Ainsi un terminal qui accepte d'executer les transformations geometriques 2.D et 3.D est inutilisable car l'etape d'elimination des parties cachees consecutive a cette transformation n'est pas disponible au sein du materiel. Il s'agit la d'une erreur dans l'ordonnement du processus de synthese.

La qualite de l'interactivite du terminal graphique attachee aux processus de description et de consultation, constitue le second point nevralgique de la conception. Ainsi un terminal capable de realiser un remplissage de taches polygonales avec une texture quelconque, est incapable de restituer vers l'application les informations de description de cette entite graphique. En effet, la synthese prematuree d'un attribut provoque



generalement l'impossibilite de sa consultation par la suite, de meme pour sa description qui est inexistante. Il s'agit la de la disymetrie.

#### I.2.5.1 REALISATION D'UN SYNTHETISEUR : -

La definition de l'architecture d'un systeme graphique permet de determiner les synthetiseurs de la chaine de synthese, puis les fonctions respectives et les operateurs qui leur sont associes. Enfin le concepteur determine leur nombre et leur ordonnancement en tenant compte des criteres de coherence et de symetrie.

Il reste le choix d'une methode de realisation d'un synthetiseur quelconque. La premiere solution est la mise en oeuvre logicielle. Il s'agit d'une bibliotheque de primitives accessibles a partir d'un langage evolue assurant la gestion des quatre processus fondamentaux. Cette methode est la plus courante pour la realisation du synthetiseur initial hote de l'application. La seconde solution est la mise en oeuvre materielle. Il s'agit de fonctions specifiques activees au niveau du terminal graphique et executees :

- Par un micrologiciel implante sur une carte a microprocesseur,
- Par un "nanologiciel" implante sur une carte a processeurs en tranches,
- Par un operateur realise en logique cablee TTL.
- Par un operateur integre V.L.S.I specifique ou general.

#### REMARQUE :

La difference entre un micrologiciel et un "nanologiciel" n'est pas une difference "physique" mais plutot une difference au niveau de la flexibilite aussi bien des architectures que des logiciels associes a ces architectures.

#### I.2.6 ETUDE DE CAS :

##### LA CONSOLE EVOLUEE RADIANCE 320

Le terminal graphique interactif RADIANCE 320 est commercialise par la Societe francaise GIXI. L'architecture est dotee de deux synthetiseurs de base, la configuration est celle d'une console evoluee. La chaine de pre-synthese est constituee d'un synthetiseur unique architecture autour d'un microprocesseur 8 bits, le MC.6809 (l'horloge de base est a 2 MHz) et

d'un co-processeur graphique V.L.S.I, le EF.9367.

Par contre, la chaine de post-synthèse est constituée d'une mémoire d'image représentant la matrice-écran et d'un module de generation du signal video qui effectue a travers des registres a decalage, la serialisation de l'information graphique et la conversion numerique-analogique. Cette architecture constitue le second synthetiseur. On fait abstraction du moniteur T.V de visualisation pour ce decoupage en deux synthetiseurs :

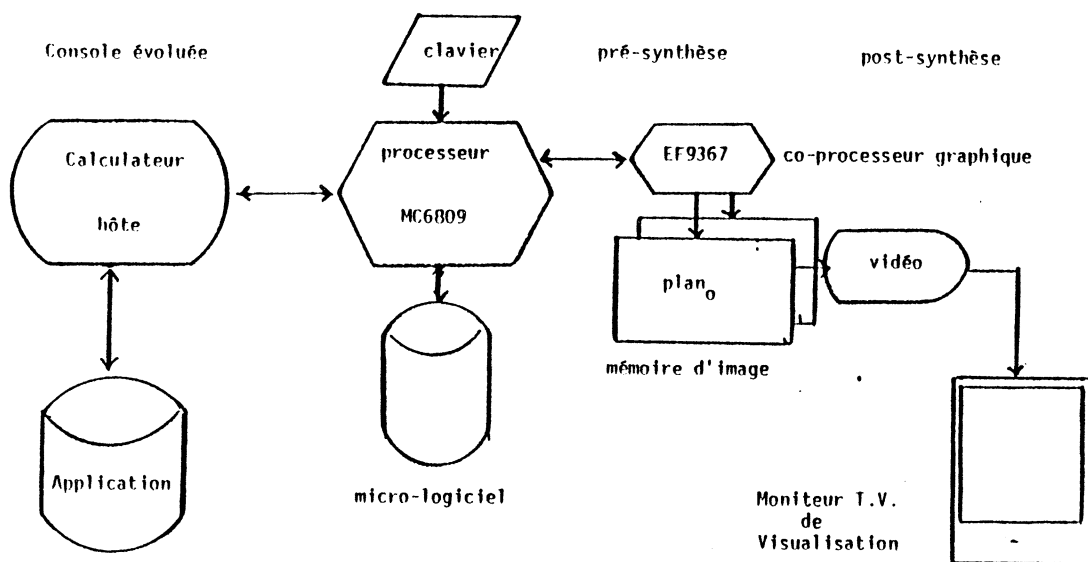


Figure No I.11

La qualite de conception de ce systeme est mediocre. L'incoherence existe entre la gestion de l'affichage de caracteres et la generation de vecteurs, en effet une transformation geometrique est applicable aux vecteurs mais pas aux caracteres. La disymetrie est inherente aux types d'entites graphiques acceptees par la post-synthèse. Le co-processeur n'assure aucune notion de type d'objet. La seule information accessible de part et d'autre de la memoire d'image est le point de l'ecran. Les processus de description et de consultation sont quasiment impossibles pour les autres entites graphiques. (vecteurs, caracteres ...)

Ainsi, l'identification d'un contour par une interaction doit etre geree par les deux synthetiseurs en amont. L'identification d'une figure sur l'ecran passe par l'appariement d'un ensemble de points a cette figure au niveau du processeur MC.6809. L'utilisation du co-processeur graphique implique une rigidite dans le processus de synthèse, en particulier au niveau de la post-synthèse qui est figee. Les processus d'attribution et de visualisation sont realises a partir d'operateurs cables integres au co-processeur graphique :

- Texture de trace, Police des caracteres,
- Traces de vecteurs.

Le processus de consultation se resume a la designation d'un point de l'ecran, donc de la memoire d'images par l'intermediaire d'un dispositif d'interaction, tel que le clavier associe a un reticule, ou le photostyle.

### I.3 NOTION DE PARALLELISME :

L'architecture de base consiste a mettre en oeuvre des synthetiseurs monoprocessus de synthese. Un synthetiseur mono-processus est defini par :

- Un univers initial ne comportant qu'un seul type d'elements a synthetiser.
- Un univers terminal unique pour le prochain synthetiseur de la chaine de synthese.

les processus que l'on peut associer a un synthetiseur sont la description, l'attribution et la visualisation. la visualisation est le processus de synthese final susceptible de traiter tous les attributs relatifs a une image. Cette architecture mono-processus propose des performances dependantes de celles du synthetiseur final de visualisation. Ainsi, le processus de synthese est entierement sequentiel pour la pre-synthese et la post-synthese.

#### I.3.1 ORDONNANCEMENT DU PROCESSUS :

Differents attributs sont affectes aux entites graphiques constituant une scene. Au niveau du synthetiseur final l'element considere est le point de l'ecran dont il faut traiter tous les attributs. La visualisation est le resultat de ce traitement. On definit l'influence d'un attribut comme le nombre d'operations elementaires qu'il faut effectuer au sein d'un ou de plusieurs synthetiseurs. Selon l'ordonnancement du processus, l'influence d'un attribut sera importante. Cette influence n'est pas une grandeur qui permet de determiner la complexite ou la performance des differents synthetiseurs. Le choix de l'ordonnancement depend du type d'applications auquel pretend repondre le terminal. L'operateur privilegie sera :

- Pour une simulation en temps-reel : le calcul de la prise de vue.
- Pour du dessin au trait : la generation de vecteurs.

- Pour la synthese d'images : le remplissage de taches et les textures.

la reduction de l'influence d'un attribut privilegie se fait a l'aide de deux methodes complementaires, la memorisation des attributs synthetises et la penetration des attributs non-synthetises. Cette memorisation et cette penetration des attributs a travers les differents synthetiseurs assurent la coherence et la symetrie du systeme. ( MAR 82 )

### I.3.2 SYNTHETISEURS MONOPROCESSUS :

Il s'agit de mettre en evidence un parallelisme intra et inter-processus au sein de la chaine de synthese. Des synthetiseurs peuvent fonctionner en parallele et produire des resultats de facon synchrone pour les synthetiseurs suivants de la chaine. On parle dans ce cas de "co-synthese". Il s'agit d'un parallelisme inter-synthetiseurs.

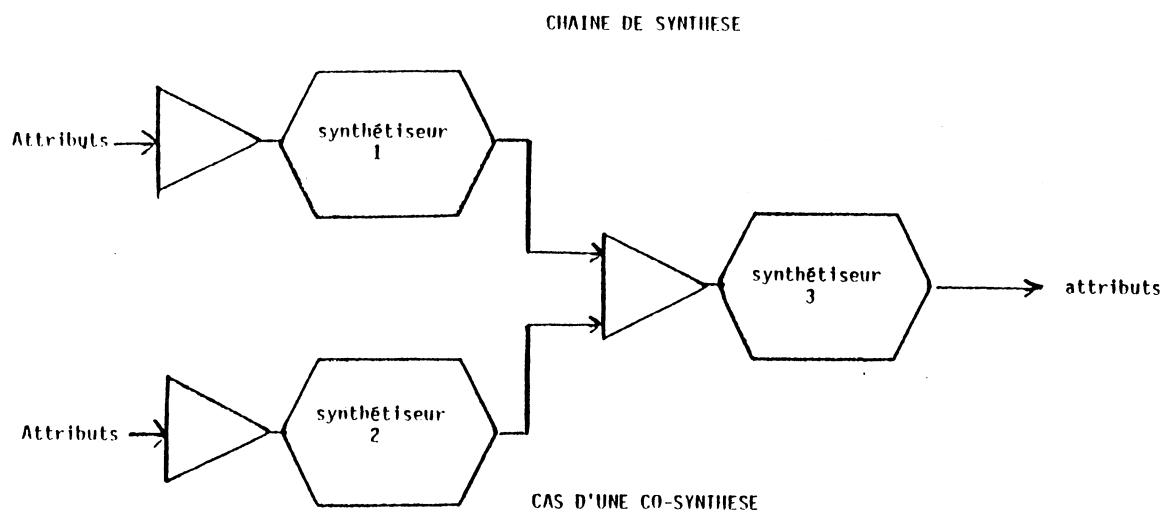


Figure No I.12

De plus, au sein d'un synthetiseur pour un processus donne, plusieurs operateurs fonctionnent en parallele et produisent un resultat final de facon asynchrone pour les synthetiseurs suivants de la chaine. Ce parallelisme evoque le probleme de mise en oeuvre de plusieurs processeurs.

- La solution minimale : 1 processeur pour tous les processus,
- La solution maximale : 1 processeur, soit un operateur par processus.

## I.3.2.1 DIFFERENTS TYPES D'ARCHITECTURE : -

La premiere possibilite est appelee "SEQUENTIEL INTEGRAL". On affecte un processeur a un processus de synthese. Ce processus est effectue pour une entite graphique donnee et determinee par les attributs qui lui sont associes.

La seconde possibilite est appelee "PIPELINE". Un processus donne est subdivise en un ou plusieurs sous-processus sequentiels. On affecte un processeur a un sous-processus donne. La synchronisation entre les differents processeurs peut etre realisee de deux facons :

- Le "PIPELINE" synchrone : la synchronisation se fait par l'intermediaire d'une horloge commune,
- Le "PIPELINE" asynchrone : la synchronisation se fait par rapport a la periode la plus longue. Le processus le plus court se dote d'un mecanisme de temporisation afin de resorber le decalage temporel.

Ce mecanisme de temporisation peut etre mis en oeuvre soit par attente active, soit par une memoire-tampon.

## I.3.2.2 ARCHITECTURE MULTIPROCESSUS : -

L'association d'un processus a un synthetiseur determine les attributs pris en compte pour une entite graphique definie. L'association de plusieurs processus a un synthetiseur unique implique une synchronisation entre les differents processus. Ainsi une homogeneite des differentes entites graphiques permettrait de definir des architectures coherentes. Cette homogeneite n'est possible que pour des cas particuliers d'applications, la C.A.O de pieces mecaniques par exemple. (Dessin industriel) La solution consiste a determiner un point de convergence, c'est-a-dire de synchronisation des operations. Ce point de convergence s'appuie sur trois criteres :

- La suppression des disparites : il faut determiner un modele de l'entite graphique de base. Par exemple, le terminal HELIOS manipule des faces planes polygonales.
- La convergence au niveau du pixel : tout processus doit produire comme element terminal un pixel accompagne eventuellement d'informations supplementaires, la profondeur par exemple,
- La convergence intermediaire : C'est un compromis entre les deux criteres precedents.

Cependant ce nouveau concept de multiprocessus au sein d'un synthetiseur provoque une multiplication des operateurs donc des processeurs au niveau de l'architecture. Les solutions obtenues deviennent trop couteuses. Il est preferable de reduire le nombre des operateurs afin d'obtenir des chaines de synthese adaptees aux applications graphiques les plus courantes. ( MAR 82 )

#### 1.3.2.3 REDUCTION DU NOMBRE D'OPERATEURS : -

Le synthetiseur general accepte en entree une classe d'applications issue d'un ensemble de synthetiseurs en amont de la chaine. Il produit en sortie une classe d'entites graphiques acceptee par les synthetiseurs en aval constituant l'interface de visualisation. Cette constatation est valable pour les deux sous-chaines de pre-synthese et de post-synthese. Le synthetiseur specifique considere un type d'application. Dans le cas d'une approche multiprocessus on definit :

- N : le nombre de processus,
- K : le nombre moyen des operateurs par processus.

Dans ce cas, Le nombre total d'operateurs est :  $T\text{-opr} = N * K$ . Plus generalement si l'on considere les parametres suivants

- NC : nombre de classes d'applications a satisfaire,
- NM : nombre de materiels disponibles.

Le nombre d'operateurs, dans cette nouvelle configuration de synthetiseur general et universel est  $T\text{-op} = NC * NM * ( N * K )$ . Ce nombre prohibitif d'operateurs demontre la necessite de reviser l'approche des differentes architectures des synthetiseurs proposes. Une premiere possibilite consiste a definir une classe d'applications et un materiel donne. Une unite intermediaire est notee  $U(C,M)$ .

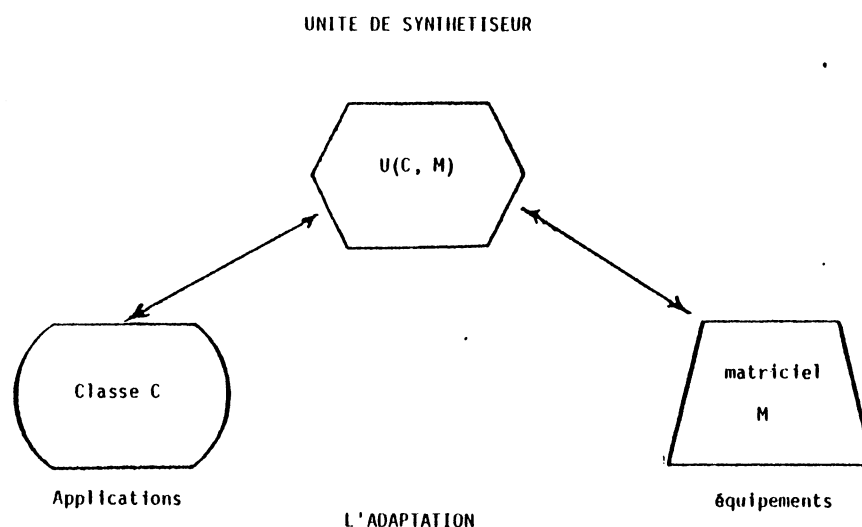


Figure No I.13

$U(C, M)$  est l'unité du synthetiseur permettant à la classe C d'utiliser le matériel M. Un synthetiseur de ce type possède  $(NC * NM)$  unités. La figure I.13 représente la relation existant entre l'unité, le matériel et la classe des applications.

#### I.3.2.4 STRATEGIES DE CONCEPTION : -

##### PARTITIONNEMENT EN COUCHES ET HIERARCHISATION :

On considère maintenant deux groupes d'opérateurs, les opérateurs de synthèse de construction et de composition. Ces opérateurs sont relatifs aux processus de base de description et de visualisation. La deuxième catégorie est celle des opérateurs de mémorisation, ils mémorisent les structures de données et gèrent la communication avec les opérateurs de synthèse. Une unité de contrôle se charge de l'ordonnement des processus et du contrôle de leur déroulement. Elle assure les tâches suivantes :

- La gestion de la banque de processus vis à vis des synthetiseurs amonts,
- La communication avec le synthetiseur en aval,
- Les échanges entre les deux unités associées aux processus de description et de visualisation, et l'unité de communication.

L'unité de communication gère la structure de données nécessaire aux opérateurs de mémorisation. L'unité de description et de consultation propose des opérations sur cette structure de données.

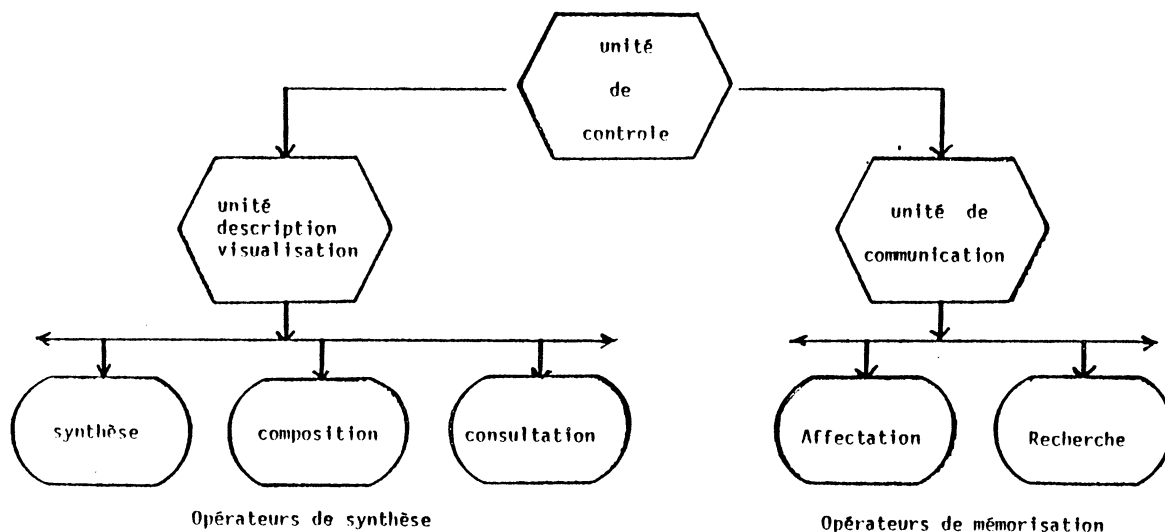


Figure No I.14

L'intérêt de ce type d'organisation hiérarchique est multiple. Tout d'abord, une indépendance vis à vis de l'application de niveau supérieur. L'unité de contrôle propose une interface par le biais d'une bibliothèque de primitives standard. En second lieu une indépendance vis à vis du dispositif de saisie et d'affichage, rendue possible par la mise en œuvre de l'unité de description et de visualisation. Enfin une cohérence et une banalisation des structures de données accessibles par l'unité de communication. Ce point permet de valider la qualité de la conception c'est-à-dire la cohérence et la symétrie du système graphique. De plus la réduction des opérateurs donc des processeurs au sein d'un synthétiseur en découle.

Cependant, cette solution soulève deux types de problèmes. Le premier est la définition et la réalisation d'une base de données banalisée répondant aux différentes structures des synthétiseurs. Le second est la gestion versatile des processus en fonction des types d'attributs manipulés et du matériel utilisé pour un synthétiseur donné. Ainsi, le partitionnement consiste généralement à constituer deux couches principales par l'entremise d'un univers intermédiaire entre l'univers initial et l'univers final d'un synthétiseur. On définit deux unités correspondantes :

- U(C,S) : l'unité indépendante du matériel mais dépendante de l'application,



- $U(S,M)$  : l'unité indépendante de l'application mais dépendante du matériel.

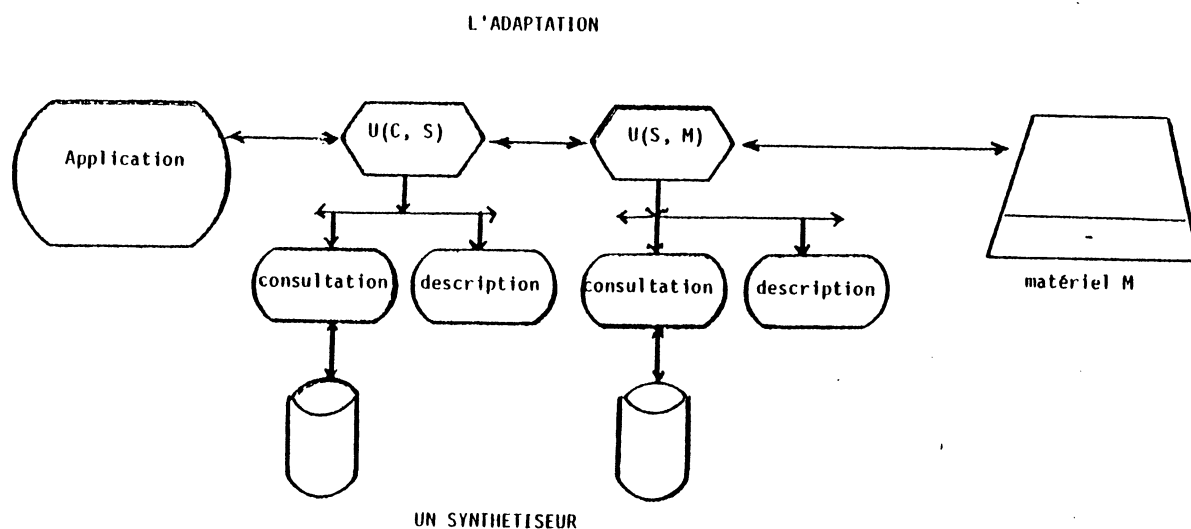


Figure No I.15

Ce découpage est caractéristique des synthétiseurs de la sous-chaine de pre-synthèse. Les deductions faites à partir de cette organisation sont :

- La performance globale dépend étroitement de la couche la moins importante,
- La nécessité de mémorisation et de gestion des informations intermédiaires au niveau du synthétiseur.
- La mise en œuvre de supports efficaces de communication étant donné le volume d'informations circulant dans la chaîne de synthèse. Ce volume découle de la multiplication des univers intermédiaires.

La pre-synthèse est plus particulièrement la couche la moins performante dans ce type d'organisation. Elle gère les structures de données les plus complexes et les plus volumineuses. Elle assure la communication avec l'application dont elle dépend. Enfin elle pilote la post-synthèse du système graphique. Elle constitue dans la plupart des architectures de constructeurs ayant retenu le partitionnement en couches le principal goulot d'étranglement de par le volume d'informations qui y transite, et de par la complexité des algorithmes qui y sont exécutés :

- Acquisition des attributs relatifs à une scène complète,

- Gestion et memorisation des structures de donnees,
- Execution des algorithmes associes aux quatre processus fondamentaux :
  - . Description d'une sphere ou d'un disque,
  - . Attribution d'une couleur ou d'une texture,
  - . Visualisation avec decoupage, etc.

**REMARQUE :**

La pre-synthese manipule des objets graphiques structures par opposition a la post-synthese qui ne manipule que des points dotes d'attributs specifiques tels que la couleur ou peut etre la profondeur.

**1.4 PRE-SYNTHESE D'IMAGES ET REALISME :****1.4.1 PERFORMANCES D'UN SYSTEME :**

Les performances d'un systeme dependent essentiellement du compromis choisi entre trois criteres qui sont la qualite de l'image, la rapidite ou temps de reponse des differents synthetiseurs et la complexite de la scene. Pour une application dont les contraintes de temps-reel sont primordiales, la qualite de l'image sera moyenne. (animation d'objets constitues de faces planes dotees de couleurs)

Le processus de synthese sera volontairement simplifie. En fonction des objectifs a atteindre, dependant des classes d'applications, le concepteur fera un choix :

- De l'ordonnancement du processus de synthese,
- Des types d'attributs manipules,
- De l'organisation de l'architecture : sequentielle ou parallele,
- De la methode de realisation des divers operateurs d'un synthetiseur,
- De la communication et des supports d'echanges inter-synthetiseurs.

Enfin, la puissance de l'interaction en matiere de description et de consultation, sera conditionnee par la coherence et la symetrie de la pre-synthese. Les concepts de memorisation intermediaire et de penetration des attributs, doivent etre respectes. L'indépendance et l'adaptabilité

d'un système conçu de façon modulaire doivent permettre de proposer des systèmes généraux aux performances moyennes, et des systèmes spécifiques aux performances adéquates.

Le partitionnement en couches plaide en faveur de la première proposition qui semble convenir à la plupart des constructeurs. Cependant, le critère prépondérant dans le choix d'un système graphique est certainement le coût de développement. Ce critère sanctionne la plupart des qualités exposées ci-dessus. ( MGE 84 ) ( MAR 82 )

Ainsi la limitation du coût de développement peut se faire :

- En limitant l'investissement total : le partitionnement en couches avec aucune possibilité d'extension future.
- En limitant l'investissement initial : La réalisation d'une organisation hiérarchique à partir d'un noyau minimal.

La courbe présentée dans la suite de l'étude indique le coût moyen d'un système graphique actuel dans une configuration dite de station de travail. Cette configuration sera particulièrement étudiée dans le chapitre VI.

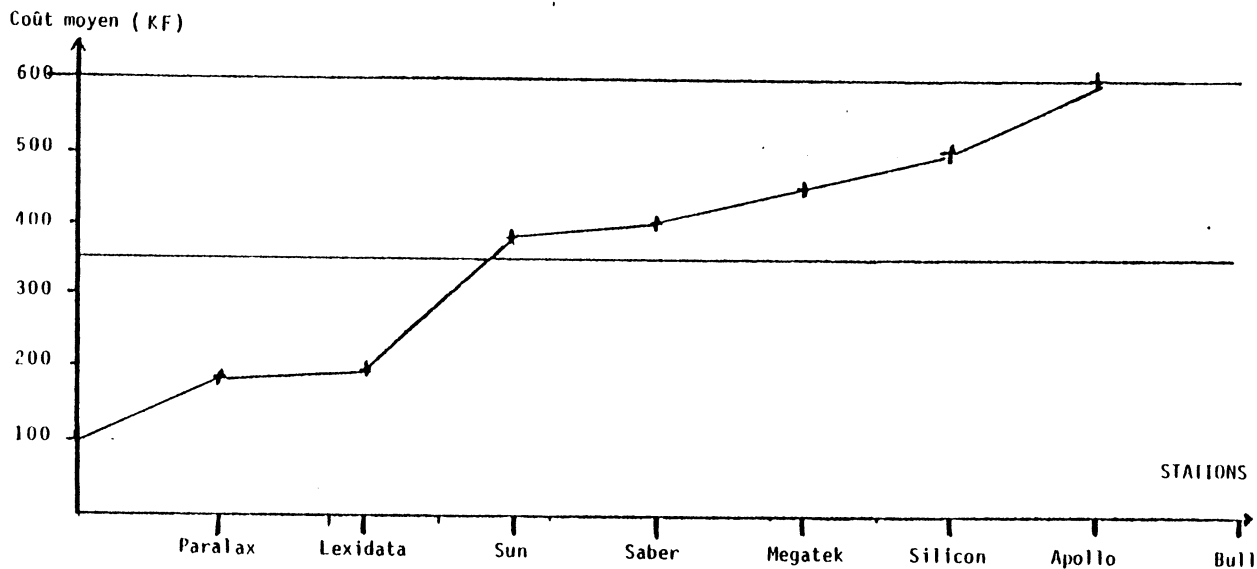


Figure No I.16

Ces coûts sont des indications moyennes pour des configurations relativement comparables du point de vue des caractéristiques de l'architecture interne et des performances. Il s'agit d'architectures multiprocesseurs pour des images 3.D de haute résolution (1024 X 1024 points). Le système est en version de base monoposte, de nombreuses extensions des différentes ressources (mémoires, périphériques...) sont proposées par les constructeurs. ( GAU et LAG 85 ).

## I.4.2 RÔLE PRIMORDIAL DE LA PRE-SYNTHESE :

La pre-synthèse constitue l'étape cruciale du processus de synthèse complet, pour une organisation hiérarchique ou un partitionnement en couches, la communication fonctionnelle entre le calculateur hôte et le terminal est assurée par la pre-synthèse.

## I.4.2.1 ARCHITECTURE MINIMALE : -

Généralement le dispositif d'affichage est doté d'un processeur d'entretien câble étant donné la contrainte de temps-reel du rythme vidéo. Un processeur d'échange avec le calculateur hôte gère les fonctionnalités de ce type de terminal. Ce processeur assure les processus fondamentaux décrits précédemment. Il s'agit d'une configuration analogue à celle du terminal Radiance 320 :

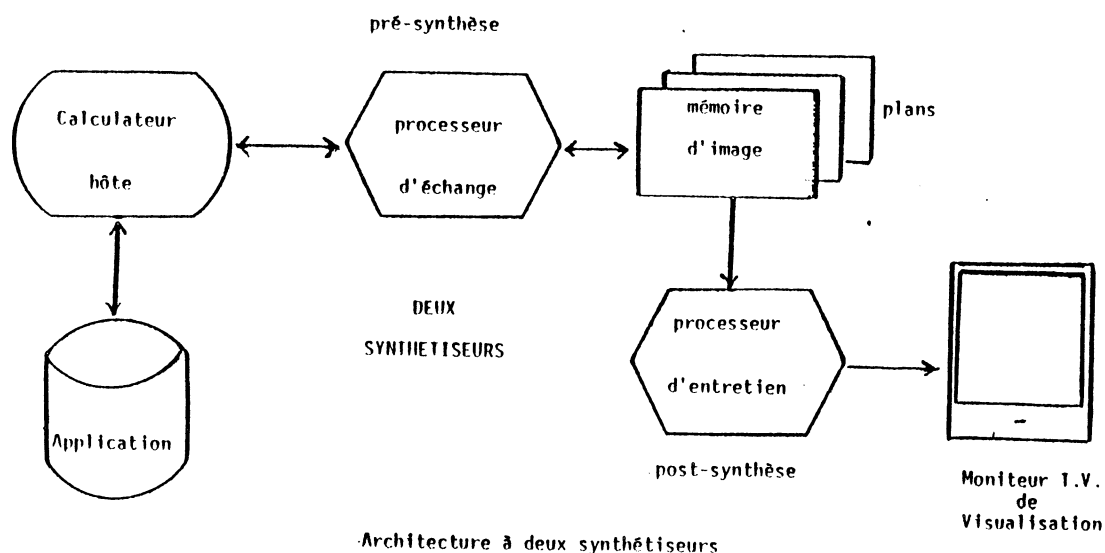


Figure No I.17

## I.4.2.2 DEGRÉ DE RÉALISME : -

La proportion des tâches effectuées au niveau de la pre-synthèse est souvent fonction du réalisme de l'image à synthétiser. Dans le cas d'une synthèse d'image sans réalisme, la pre-synthèse se limite à la génération d'un ensemble fini d'objets graphiques dotés éventuellement d'attributs simples tels que la couleur, les textures de trace, le clignotement etc... La mémorisation et la pénétration des attributs sont limitées. L'interaction sera à la charge de l'application. Les objets graphiques manipulés sont :

- Les points, les vecteurs,
- Les polygones, les cercles,
- Les caracteres.

Par contre, pour des images realistes, le volume des attributs est plus important. La pre-synthese est chargee de leur gestion et de l'execution des operations qui leurs sont associees. Une memorisation intermediaire est necessaire afin d'assurer la symetrie de l'architecture, les processus mis en oeuvre doivent etre structures afin de permettre une coherence de l'ensemble du systeme. Ainsi le terminal HELIOS par l'intermediaire de la table des faces peut restituer pour un point de l'ecran tous les attributs qui lui sont associes.

#### I.4.2.3 REPARTITION DES TACHES : -

Cette repartition se fait entre la pre-synthese et la post-synthese. Elle est determinee par les differences de temps de reponse des deux sous-chaines de traitement. La pre-synthese opere en amont du processus, l'execution des taches est ralentie par :

- La communication avec le calculateur hote,
- Le volume d'informations manipule a l'initialisation d'un processus de synthese,
- La complexite des algorithmes mis en oeuvre,

Le remplissage d'une tache polygonale illustre bien cette difficulte de repartition optimale entre les deux chaines. L'application delimite un trapeze dont la morphologie est transmise a la chaine de pre-synthese. Le trapeze est considere comme la region elementaire coherencee acceptee par le terminal :

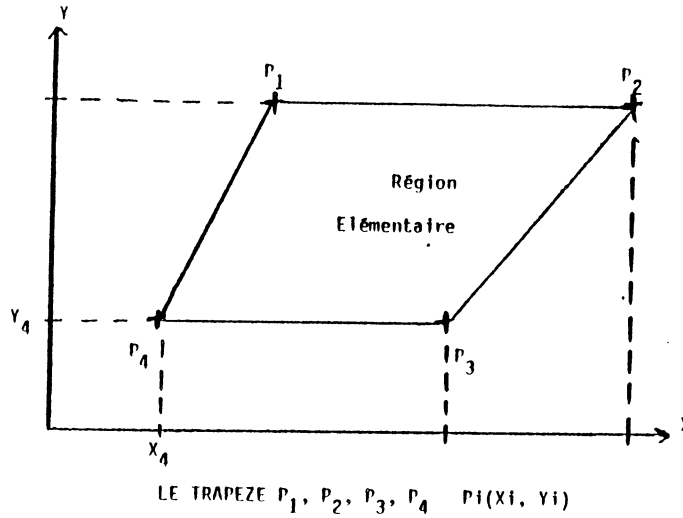


Figure No I.18

La pre-synthèse effectue une interpolation de la ligne courante pour piloter le remplissage de la tache delimitée par le trapeze. La logique de generation de vecteurs visualise chaque point appartenant aux segments de droite delimités par les demi-droites  $(P_1, P_4)$  et  $(P_2, P_3)$ . Cette tache est attribuée à la pre-synthèse. La generation d'une ligne horizontale se fait au rythme video, soit un point toutes les soixantes dix nanosecondes. Par contre l'interpolation d'une ligne se fait au rythme de l'horloge du processeur de la pre-synthèse.

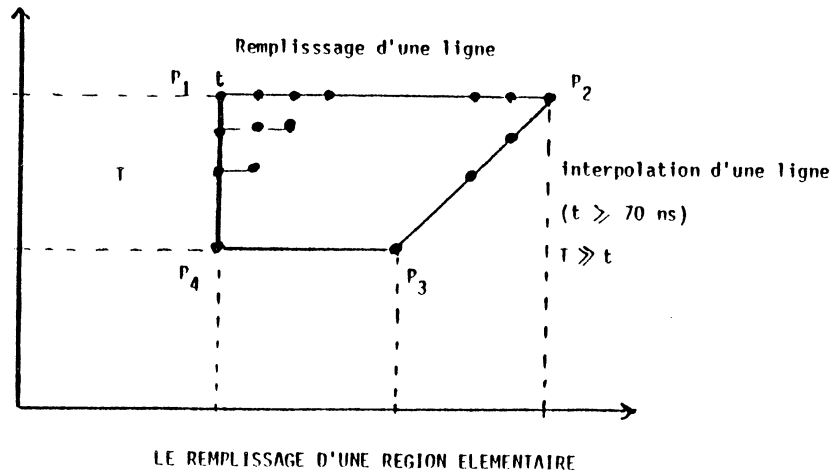


Figure No I.19

Il est possible d'attribuer le découpage de polygones en trapèzes à un autre synthétiseur de la chaîne de pré-synthèse. Dans ce cas l'application se charge de découper des polyèdres de l'espace en une suite de polygones. La pré-synthèse peut être progressivement surchargée, la disproportion des temps de réponse par rapport à la post-synthèse s'accroît. Il est nécessaire d'envisager des architectures évolutives de la chaîne de pré-synthèse afin d'exploiter au mieux le parallélisme de la logique de visualisation associée à la post-synthèse. Cette évolution consiste à mettre en œuvre plusieurs synthétiseurs dans des configurations multi-processeur. Cette option dépend des performances requises par l'utilisateur d'un tel système. On définit ainsi un cheminement à travers une architecture hiérarchisée :

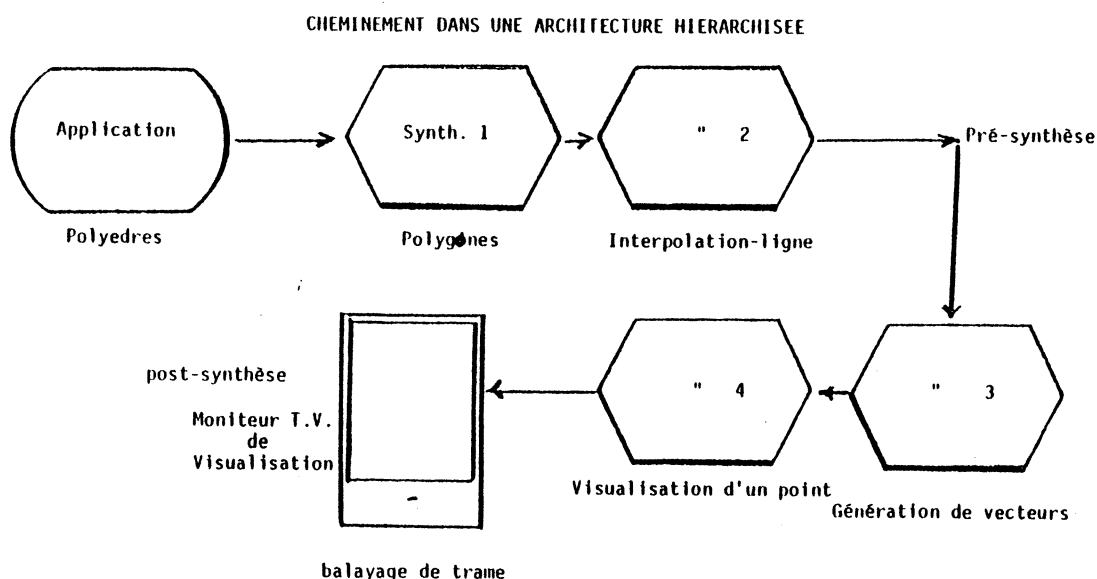


Figure No I.20

## I.5 DIFFÉRENTS TYPES DE PRÉ-SYNTHÈSE :

### I.5.1 PRÉ-SYNTHÈSE DE BASE :

La pré-synthèse de base s'inspire de l'architecture d'une configuration minimale présentée dans le paragraphe précédent. La tendance actuelle est la multiplication des processeurs au niveau de la pré-synthèse, comme l'indique la figure I.21. Ce type d'architecture privilégie le parallélisme qu'il soit synchrone ou asynchrone, au niveau de la pré-synthèse.

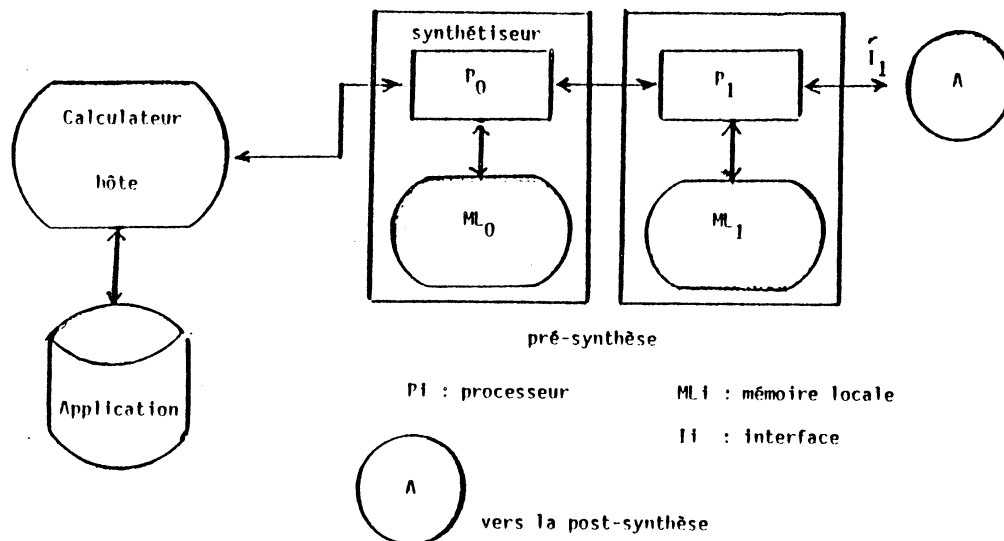


Figure No I.21

La conception d'une pré-synthèse de base consiste à définir des processeurs adaptés, de moyenne performance pour les synthétiseurs d'échange et d'entretien de la configuration de base. Le processeur d'échange peut être réalisé à partir d'un microprocesseur standard 8, 16 ou 32 bits doté de son environnement. Selon les performances requises, le processeur standard peut être remplacé par des microprocesseurs en tranches dont l'horloge de base est plus rapide. Le processeur d'entretien peut être réalisé à partir d'opérateurs câblés en logique discrète ou des processeurs V.L.S.I spécialisés dans la gestion d'une mémoire d'images.

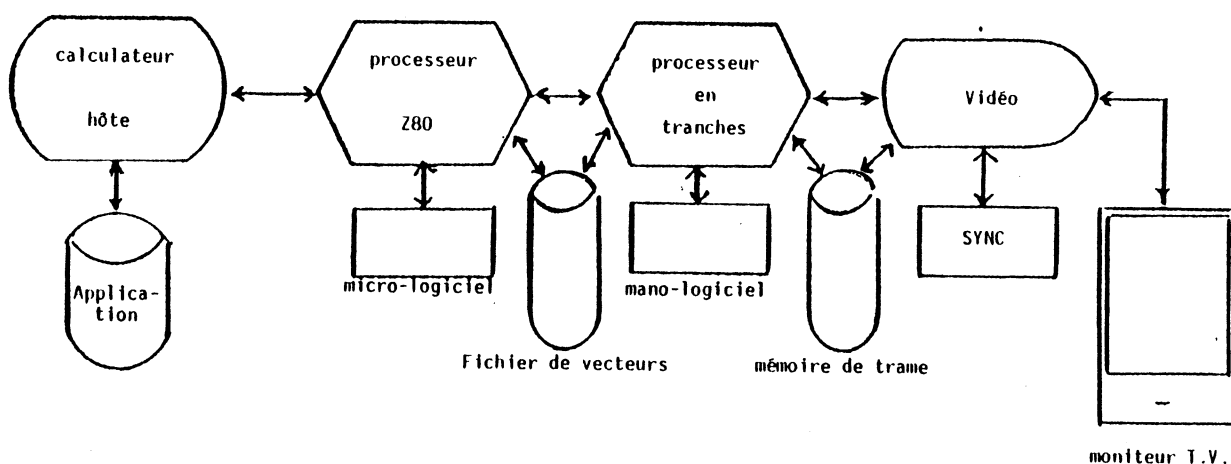
Ces architectures dépendent souvent de la technologie du dispositif d'affichage qui de plus détermine les types de traitements associés aux chaînes de pré-synthèse et de post-synthèse. Le cas le plus répandu est l'utilisation d'un co-processeur V.L.S.I aux fonctions évoluées gérant les processus suivants :

- La visualisation et le rafraîchissement de l'écran,
- La description de base,
- La consultation et l'attribution de base.

Ces processus concernent généralement des entités graphiques simples enregistrées dans une mémoire d'images. (les points, les vecteurs et les caractères.) Les attributs associés à ces entités graphiques simples sont la couleur, une texture géométrique de trace, la taille et l'orientation, la visibilité, et le clignotement. Dans cette configuration, la chaîne de pré-synthèse de base se contente d'assurer une communication relativement



souple avec le calculateur hôte. De plus elle pilote le processeur d'entretien. La pre-synthese de base est l'illustration ideale de l'organisation hierarchique, l'evolution naturelle est le partitionnement en couches avec une possibilite de multiprocesseur au niveau de la pre-synthese. Le role de la post-synthese est de realiser une serialisation des memoires d'images, une conversion digitale-analogique et un melange de l'information avec les differents signaux de synchronisation du signal video. (retour-ligne et retour-trame). Des tables de couleur peuvent s'interposer entre la memoire de trames et l'interface video. L'interaction est possible mais limitee. Elle est a la charge de la pre-synthese, le mecanisme de base de la designation etant gere par la post-synthese. L'entite graphique accessible est le point de l'ecran. La premiere illustration de ce type de configuration est le terminal RAMTEK-9400.



LE TERMINAL RAMTEK-9400

Figure No I.22

Cette pre-synthese de base consiste a elaborer une methode de conception initiale pour la realisation des synthetiseurs. Generalement, l'architecture est constituee d'un synthetiseur unique qui est le processeur d'echanges. Cependant l'evolution actuelle consiste a mettre en oeuvre des architectures plus completes. Le partitionnement en couches permet d'envisager cette possibilite. La tendance actuelle est de concevoir des chaines de pre-synthese multiprocesseurs dans une architecture parallele asynchrone. Une seconde illustration de ce type de configuration evolutive est le terminal RASTER-TECHNOLOGIE.

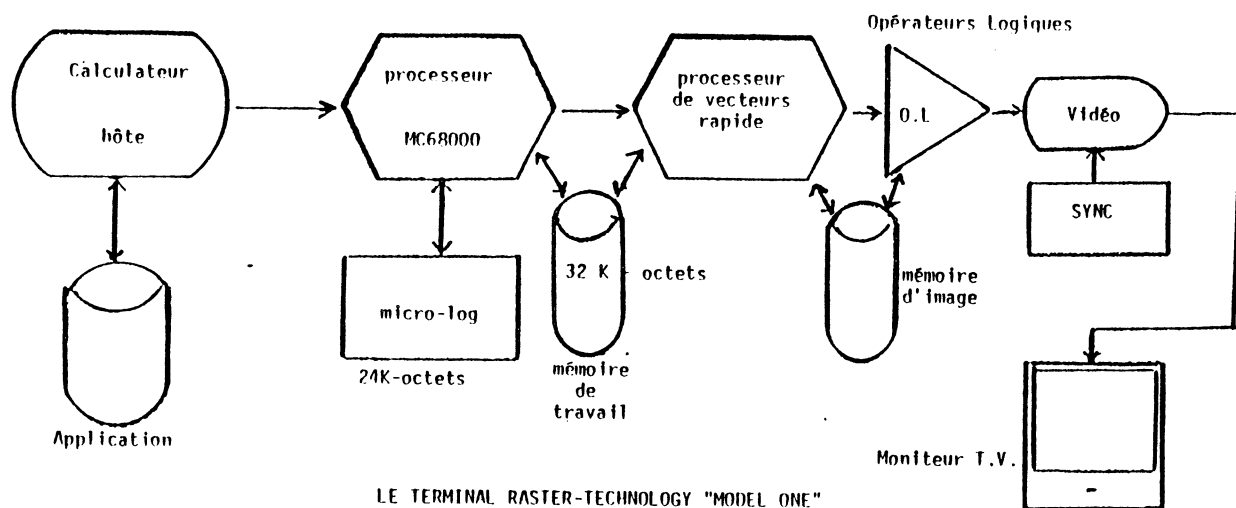


Figure No I.23

## I.5.2 PRE-SYNTHESE EVOLUEE :

La première constatation consiste à dire qu'il est possible de décharger de plus en plus le calculateur hôte en intégrant des fonctions de plus en plus complexes au sein du terminal. Il ne s'agit pas seulement de réduire le volume du programme-application. La pré-synthèse doit être capable d'effectuer des tâches complexes avec des temps de réponse acceptables. L'objectif majeur de cette approche est de mettre en œuvre des processus de synthèse évolués concernant plus particulièrement les attributs suivants :

- M : la morphologie,
- G : la géométrie,
- S : la structure,
- I : l'identité.

Ces processus sont par exemple la manipulation d'une scène 3.D avec possibilité d'attribution, de consultation d'objets graphiques complexes (S,I); la transformation géométrique et le calcul du point de vue (G)(M). Le volume des traitements de la pré-synthèse est déterminé d'une part par les fonctions intrinsèques offertes par les synthétiseurs de la post-synthèse, d'autre part par le degré de réalisme considéré et par l'interaction disponible.

L'architecture du systeme que nous allons brievement decrire illustre le partitionnement en couches de la pre-synthese. Cette tendance est accentuee par les couts toujours en regression :

- Des circuits memoires,
- Des processeurs V.L.S.I standards,
- Des processeurs V.L.S.I specialises.

Il s'agit d'un terminal developpe par la societe GRAPHICA COMPUTER CORPORATION. Ce systeme graphique tri-dimensionnel possede une architecture de type "pipeline asynchrone". Il est capable de visualiser 10000 polygones par seconde dotes d'attributs de couleur, d'ombres portees, de transparence et de visibilite, avec eliminations des surfaces cachees. ( FPI 84 ) Le processus de synthese est decompose en une suite d'etapes. Chaque etape est associee a un synthetiseur.

La chaine de pre-synthese est constituee d'un processeur principal capable d'accéder aux differentes memoires de l'architecture. Il s'agit d'un MC.68000 qui gere la liste de polygones d'une scene 3.D. Il peut de plus initialiser la memoire de profondeur de texture et la table de correspondance de la post-synthese.

Les autres synthetiseurs bases sur des processeurs TMS.320 fonctionnent cinq fois plus vite que le processeur MC.68000. La memoire principale permet de stocker les differents attributs associes a l'entite graphique de base qui est le polygone 3.D. Les vecteurs, les rectangles, les courbes constituent des contours quels qu'ils soient, ouverts ou fermes. Chaque sommet d'un polygone est caracterise par :

- Ses coordonnees X, Y, Z,
- Sa normale : N,
- La profondeur,

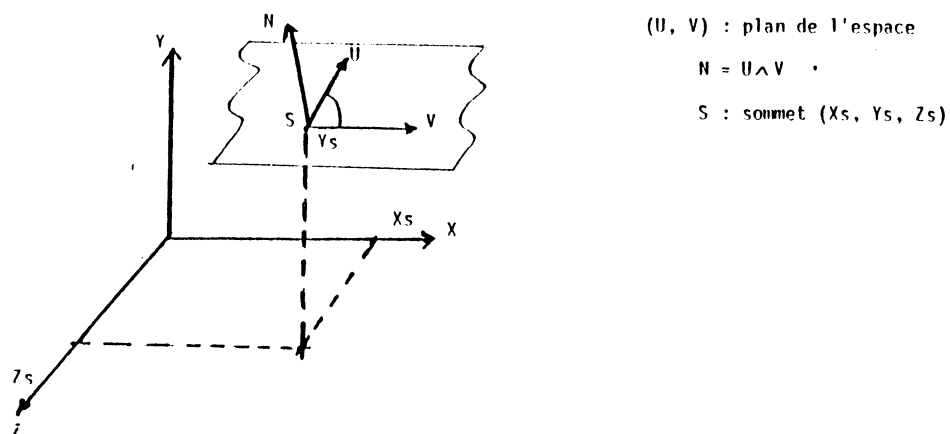


Figure No I.24

La structure d'une scene 3.D est localement conservee. Le synthetiseur initial (MC.68000) gere cette structure et alimente la chaine "pipeline". Le premier synthetiseur se charge des transformations geometriques courantes :

- Les translations, les homotheties,
- Les rotations, les Symetries,
- Les Projections en perspective.

Il effectue le calcul de l'eclairage de la reflexion et de l'ombre portee selon des algorithmes d'interpolation de la couleur ou de la normale en chaque point. Ce synthetiseur sous-traite une partie des calculs lies aux transformations geometriques a un co-processeur rapide. (opérateur avec table de division).

Le second synthetiseur effectue le decoupage d'un polygone de "n" cotes en "n-2" triangles. C'est la phase de triangularisation. Il sous-traite aussi une partie des calculs au co-processeur decrit ci-dessus.

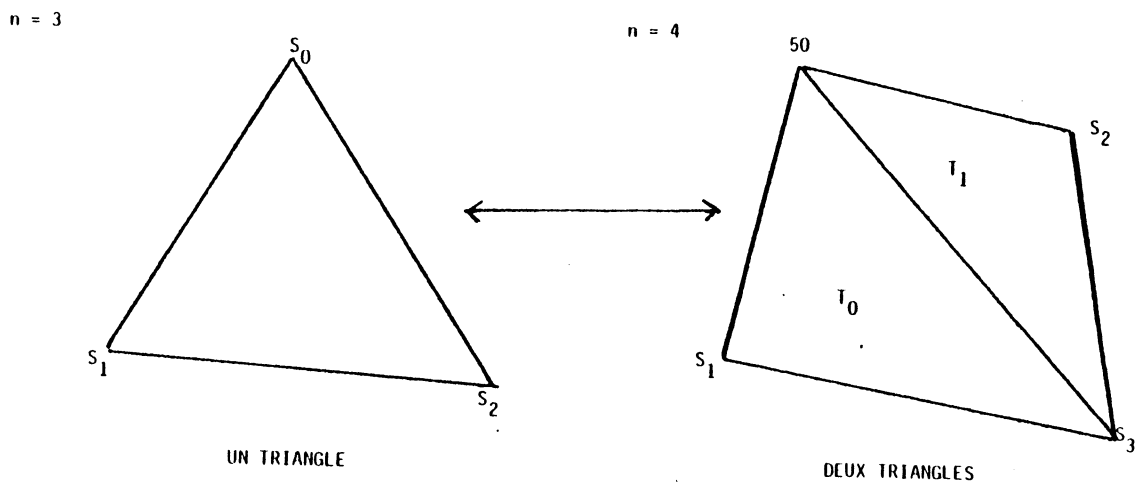


Figure No I.25

Le troisieme synthetiseur se charge du remplissage de taches pour le triangle courant. Il effectue l'initialisation du module d'acces a la memoire de trame. En fait le remplissage de taches est effectue par ce module specifique realise en technologie ECL (rapidite de commutation). Il s'agit d'un operateur capable d'accéder a la memoire de trame en ecriture pendant les cycles de rafraichissement.

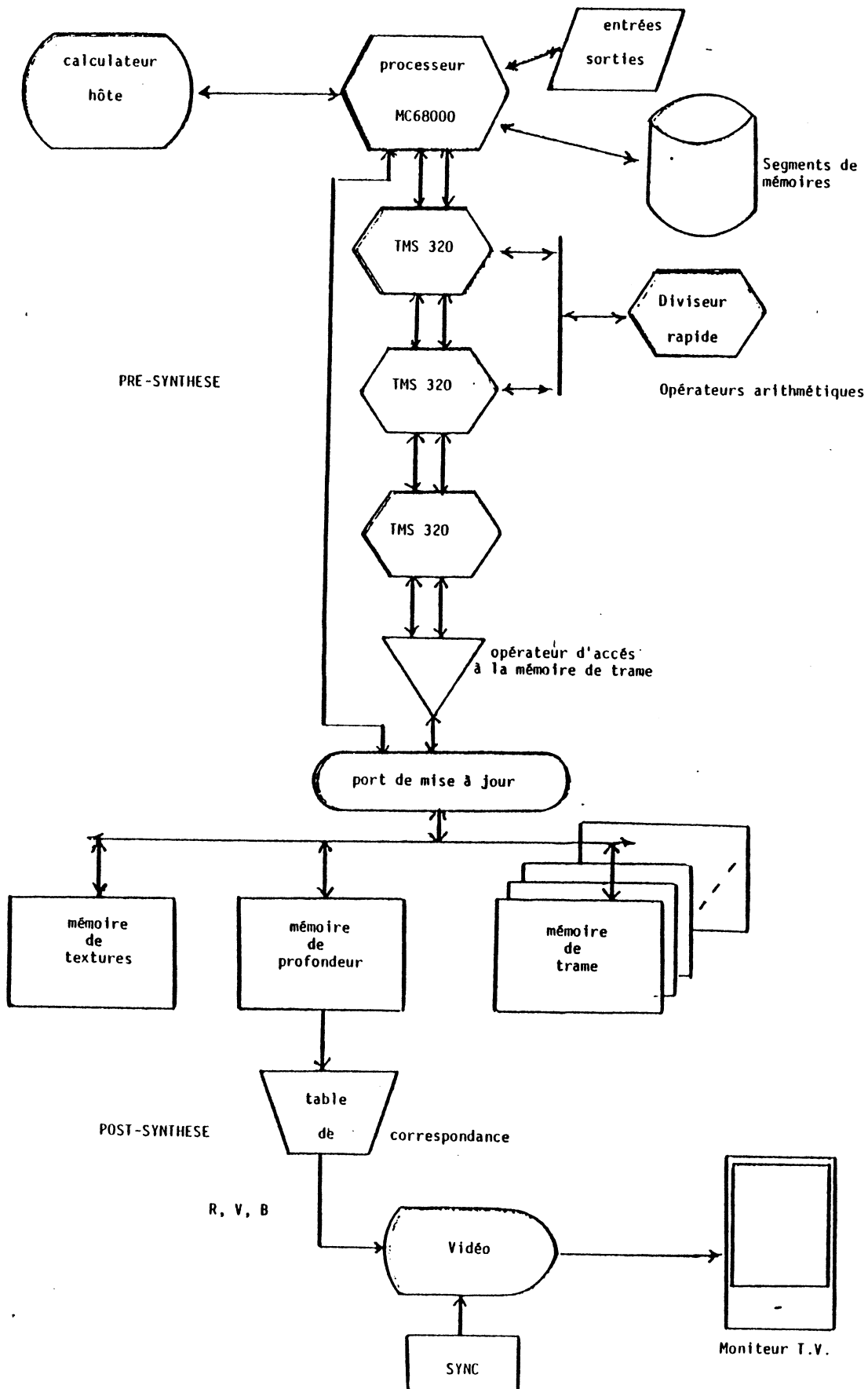
Cette pre-synthese sophistiquee et multiprocesseur est adaptee a la synthese d'images realistes avec des performances qui sont remarquables. Il apparait que l'architecture de la post-synthese est aussi consequence. Un port de mise a jour qui participe au processus de visualisation, permet de realiser certaines taches dites de "bas niveau" :

- Le decoupage cable : suppression des informations hors de la cloture d'ecran,
- Le test et la comparaison de la profondeur Z en chaque point a celle du plan de decoupage necessaire a l'elimination des surfaces cachees.
- La mise a jour des memoires de trame, de profondeur et de texture.

Le synthetiseur de generation video lit continuellement la memoire de trame. Il effectue a partir de la table de correspondance, la conversion de l'intensite en composantes couleur Rouge, Vert, Bleu. Il pilote le dispositif d'affichage. L'architecture de cette chaine de synthese est presentee dans la figure suivante. ( FPI 84 )

Figure : I.26

LA CONFIGURATION : LE TERMINAL TEMPS-REEL



L'interet de ce type d'architecture est de demontrer la possibilite de decomposition du processus de synthese tri-dimensionnel. Le cheminement des informations sur le "pipeline" est defini comme une interdependance des differents etages tenant compte de l'ordonnement retenu.

- L'etage "i" recoit une information de l'etage "i-1",
- L'etage "i" effectue le traitement necessaire,
- L'etage "i" transmet le resultat intermediaire a l'etage "i+1".

De plus, il apparait que la mise en oeuvre de fonctions evoluees au niveau de la pre-synthese ne nuit pas obligatoirement aux performances globales du systeme, a condition de prevoir une architecture adequate. L'application residente au niveau du calculateur hote se contente de parametrier les prises de vue et de consulter eventuellement de facon interactive les entites graphiques d'une image synthetisee. ( CIG 84 )

REMARQUE :

La pre-synthese constitue en particulier, l'etape ou de nombreux traitements affectant la structure, la morphologie et la geometrie sont primordiaux :

- Le decoupage des polygones,
- Les transformations geometriques,
- Le decoupage tri-dimensionnel,
- L'elimination de surfaces cachees, etc ...

## CONCLUSION :

La tendance actuelle est de mettre en oeuvre des architectures "pipeline" pour la chaine de synthese complete d'un systeme graphique. La repartition des differentes taches d'un processus au sein de cette chaine permet d'identifier les ressources necessaires.

- Le nombre de synthetiseurs,
- La methode de realisation : cablee, microprogrammee ou autre,
- La memoire necessaire,
- Les supports d'echanges et de communications.

Cette tendance concerne essentiellement les systemes de haut gamme, synthetisant des images realistes avec des performances en temps-reel acceptables. Cependant, la demarche de la conception permet de determiner une configuration initiale, l'architecture etant modulaire et evolutive.

La premiere partie de l'etude consiste a definir une chaine de pre-synthese pour le terminal graphique interactif HELIOS. Ce systeme de synthese d'images realistes developpe au Laboratoire ARTEMIS de l'Institut I.M.A.G (Equipe "Communication Graphique Interactive") est dote d'une chaine de post-synthese particulierement puissante.

Une premiere configuration de type "calculateur satellite" sera decrite et evaluee du point de vue des performances dans le chapitre suivant. Le but de cette analyse est de degager les points faibles de la configuration afin de proposer des architectures plus performantes par la suite.





## CHAPITRE II

### LA CONFIGURATION "CALCULATEUR SATELLITE" :

#### REFERENCES BIBLIOGRAPHIQUES :

( BRE 65 ), ( BLZ 84 ), ( CON 82 ), ( FER 81 ),  
( MOL 76 ), ( MAR 82 ), ( MAR 84 ), ( MOL 76 ),  
( SAR 82 ).

## LE TERMINAL GRAPHIQUE INTERACTIF HELIOS : T.G.I

## II.1 CARACTERISTIQUES D'UN T.G.I :

Le terminal graphique interactif HELIOS a été initialement mis en oeuvre dans une configuration dite "Calculateur satellite". L'option retenue lors de cette réalisation était de proposer une architecture hautement câblée afin d'obtenir des performances temps-reel au niveau des différents opérateurs de la chaîne de post-synthèse. Cette réalisation a été effectuée au laboratoire ARTEMIS. C'est le résultat d'une recherche qui a débuté en 1981 sous la direction de F.MARTINEZ. L'objectif de ce chapitre est de présenter ce système graphique afin d'évaluer les performances de cette première architecture. L'évolution de la configuration HELIOS sera mise en évidence tout au long de l'étude.

HELIOS possède une architecture originale à forte pénétration des attributs avec mémorisation intermédiaire. Ce terminal permet de synthétiser aussi bien des images réalistes que du dessin au trait. Le synoptique de son architecture est présentée dans la figure suivante :

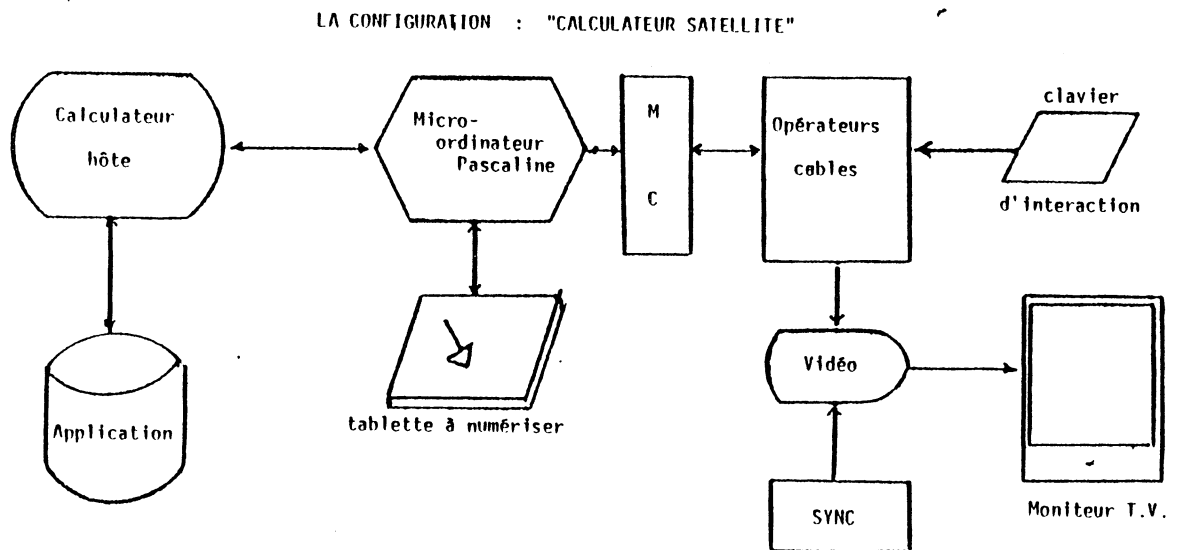


Figure No II.1

La caractéristique principale est l'intégration d'opérateurs en logique câblée discrète, au niveau de la post-synthèse. La chaîne de pré-synthèse est architecturée autour d'un micro-ordinateur 16 bits la Pascaline, les processus associés sont macro-programmés en langage Pascal.

La communication avec la chaine de post-synthese se fait a travers un port parallele de 3 X 8 bits. La communication avec le calculateur hote se fait a travers une ligne serie asynchrone a 9600 bauds.

( FER 81 )( SAR 82 )

#### II.1.1 PRESENTATION GENERALE :

La chaine de traitement est constituee de synthetiseurs essentiellement cables. Les processus de synthese associes acceptent et manipulent des attributs evolues. Les fonctions offertes par le terminal sont :

- La modification en temps-reel :
  - Des attributs d'aspect (texture, brillance),
  - . Des attributs d'eclairage (position, couleur et intensite de la source lumineuse, couleur de la lumiere ambiante),
  - . Des attributs geometriques (cloture d'ecran),
- L'identification instantanee d'un objet sur l'ecran a l'aide d'un reticule integre,
- Calcul en temps-reel de la projection des textures et de la reflexion,
- Compression des donnees echangees avec la pre-synthese ou le calculateur hote,
- Prise en compte d'une partie de l'elimination des parties cachees.

Les domaines d'application sont principalement la C.A.O, l'E.A.O, l'animation, la simulation graphique, l'art, l'audiovisuel, etc.

#### II.1.2 PROCESSUS DE VISUALISATION :

Le terminal HELIOS est un synthetiseur monoprocesseur dans la configuration "calculateur satellite". Il constitue la couche terminale du processus de visualisation. L'entite graphique de base est appelee FACE. Il s'agit d'un ensemble homogene de points possedant la meme identite, le meme aspect et les memes attributs geometriques.

Les attributs d'aspect sont la texture, la reflexion, la visibilite (invisibilite) et la transparence (opacite). Les attributs geometriques sont exprimes dans un repere 3.D associe a l'ecran.

On définit deux vecteurs  $U, V$  qui sont les vecteurs de base du plan de la face et un vecteur  $N$  qui est le vecteur normal au plan de la face.

LA CONFIGURATION : "CONSOLE EVOLUEE"

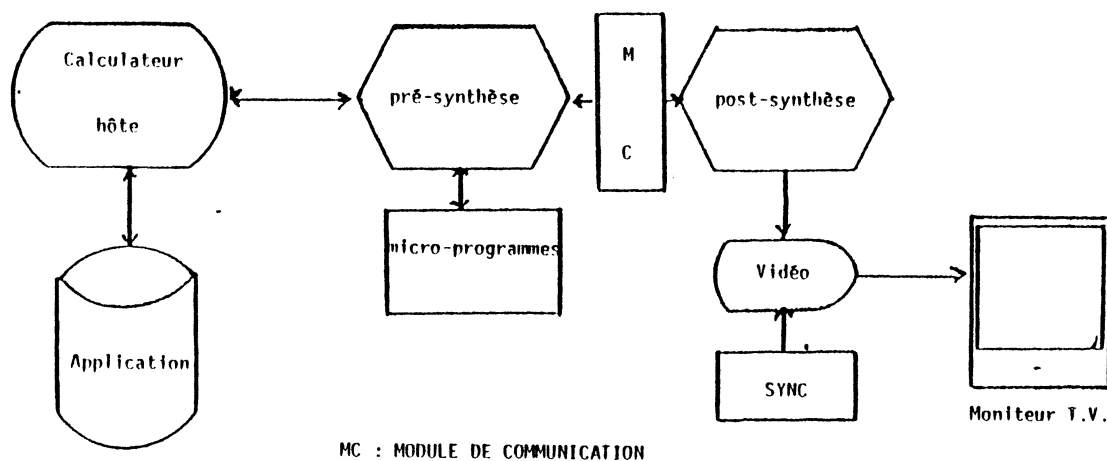


Figure No II.2

Dans sa première version, HELIOS acceptait jusqu'à 1024 faces différentes. L'identification d'une face correspond à un numéro interne code sur 10 bits connu de l'utilisateur. Ce numéro peut être récupéré après une interaction à l'aide du réticule ou de la tablette à numériser.

### II.1.3 ORDONNANCEMENT DU PROCESSUS :

L'ordonnement de ce processus privilégie la modification en temps-réel des attributs d'aspect. Les types d'attributs possibles sont repartis selon les classes suivantes :

- I, M, G, Gv : L'identification et la morphologie de la face, la géométrie de la scène et la géométrie de la prise de vue,
- E : La description de la source lumineuse relative à l'éclairage couleur, la lumière ambiante et la direction,
- Ga : La géométrie de l'affichage, la description de la fenêtre à projeter sur l'écran.

Les processus fondamentaux sont disponibles au niveau de la chaîne de post-synthèse. Il s'agit de l'attribution et la consultation de tous les attributs et de la description et la consultation des informations visibles sur l'écran par l'intermédiaire de dispositifs interactifs.

Le processus implicite et continu est la visualisation en temps-reel de l'ensemble des faces enregistrees. La premiere etape concerne la synthese des attributs geometriques. Un operateur effectue la transformation matricielle du repere de la face, dans le repere de la vue. La classe des attributs geometriques est constituee des informations suivantes :

- G : La position des faces les unes par rapport aux autres,
- Gv : La position de la prise de vue.

La seconde etape concerne la synthese des attributs morphologiques. L'attribut morphologique est unique. Il s'agit de M, le contour des faces. Un operateur effectue la projection des faces dans le repere de la vue avec le coupage si cela est necessaire. Le repere de la vue est defini a partir du point d'observation et du point de visee de l'utilisateur. Enfin l'etape de la visualisation caracterisee par le remplissage de la memoire de trame des points interieurs a une face. Chaque point est dote d'un numero d'identification de 0 a 1024 enregistre dans le plan memoire d'identification.

La post-synthese est constituee de quatre synthetiseurs. Chaque synthetiseur est dote d'un processeur specialise. Il s'agit des processeurs de visibilite, des textures, de reflexion et d'eclairage. Les processeurs des textures et de reflexion possedent des memoires locales qui constituent respectivement la banque des textures et la banque de reflexion. Un parallelisme peut etre mis en evidence au niveau des synthetiseurs de la post-synthese. Le parallelisme inter-processus concerne la multitude des plans d'identification (huit au maximum) qui peuvent alimenter en meme temps les operateurs traitant les attributs de la geometrie de l'affichage.

#### II.1.4 ARCHITECTURE HIERARCHISEE :

Cette architecture est caracterisee par trois unites principales. La premiere est l'unite de controle constituee d'un processeur qui assure les echanges avec le calculateur hote. De plus il selectionne l'ordonnancement du processus du systeme. Il distribue les signaux de synchronisation aux autres processeurs. Cette unite concerne particulierement les synthetiseurs de la pre-synthese. La seconde est l'unite de communication constituee d'un processeur qui assure l'attribution et la consultation des differentes memoires associees aux synthetiseurs de la chaine de post-synthese. Il effectue le rafraichissement des memoires, gere les conflits d'accès, le stockage et la restitution des attributs memorises. Enfin la troisieme est l'unite de description et de visualisation. Il s'agit du regroupement de quatre processeurs cites dans le paragraphe precedent.

On reconnaît le découpage en unités fonctionnelles de la chaîne de post-synthèse. Une quatrième unité spécifique est l'unité d'interaction. Elle gère des dispositifs qui sont le réticule, le pupitre d'interaction et la tablette à numériser. Elle assure les processus de désignation, de collecte de coordonnées et de consultation d'attributs des faces.

Comme l'indique la figure suivante, Les informations manipulées par les différents processeurs sont réparties dans quatre mémoires locales distinctes qui sont la mémoire de trame ou plans d'identification, la table des faces, les banques de textures et de réflexions.

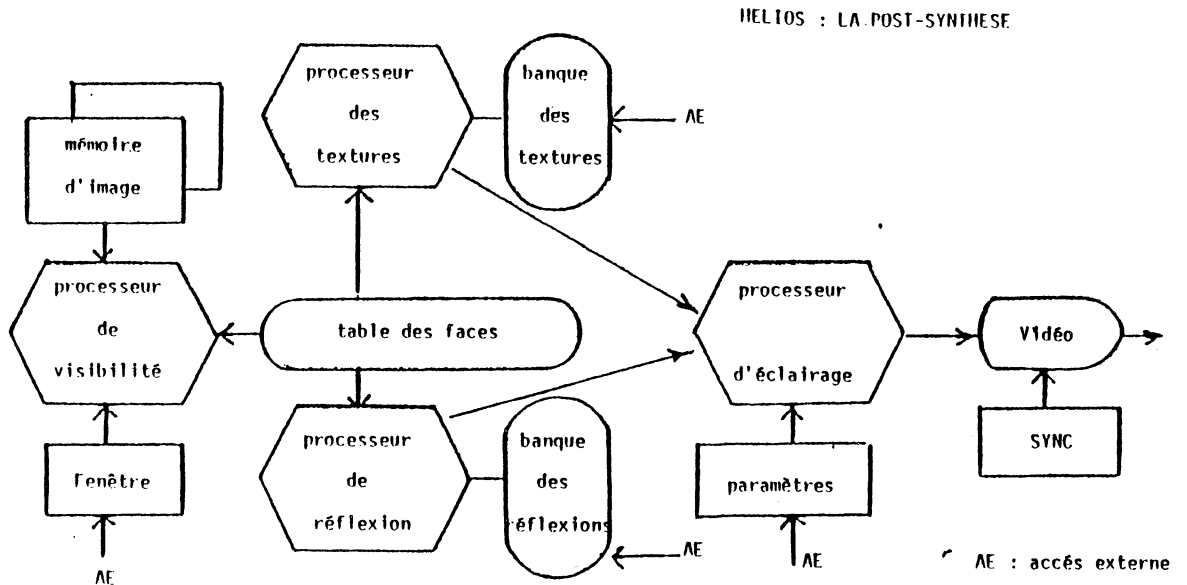


Figure No II.3

Certains paramètres de fonctionnement sont mémorisés dans des registres internes tels que les paramètres de l'éclairage et les paramètres de fenêtrage. Ces informations correspondent aux types d'attributs acceptés par le terminal HELIOS. Ainsi la table des faces est une ressource commune aux différents processeurs. Pour chaque face enregistrée elle fournit les attributs suivants :

- La visibilité, la réflexion, la transparence et la texture
- Le Vecteur normal  $N$  et les vecteurs de base  $(U, V)$ .

L'ensemble des memoires associees a chaque processeur est initialise de facon externe par la pre-synthese. Cette initialisation se fait par l'intermediaire d'un module de communication. Ce module sera decrit dans la suite du chapitre. Reciproquement, ces memoires peuvent etre consultees si une interaction est prise en compte au niveau de la pre-synthese qui declenchera des lectures des memoires de la post-synthese via le module de communication. On constate ainsi que Les concepts de coherence et de symetrie sont respectes.

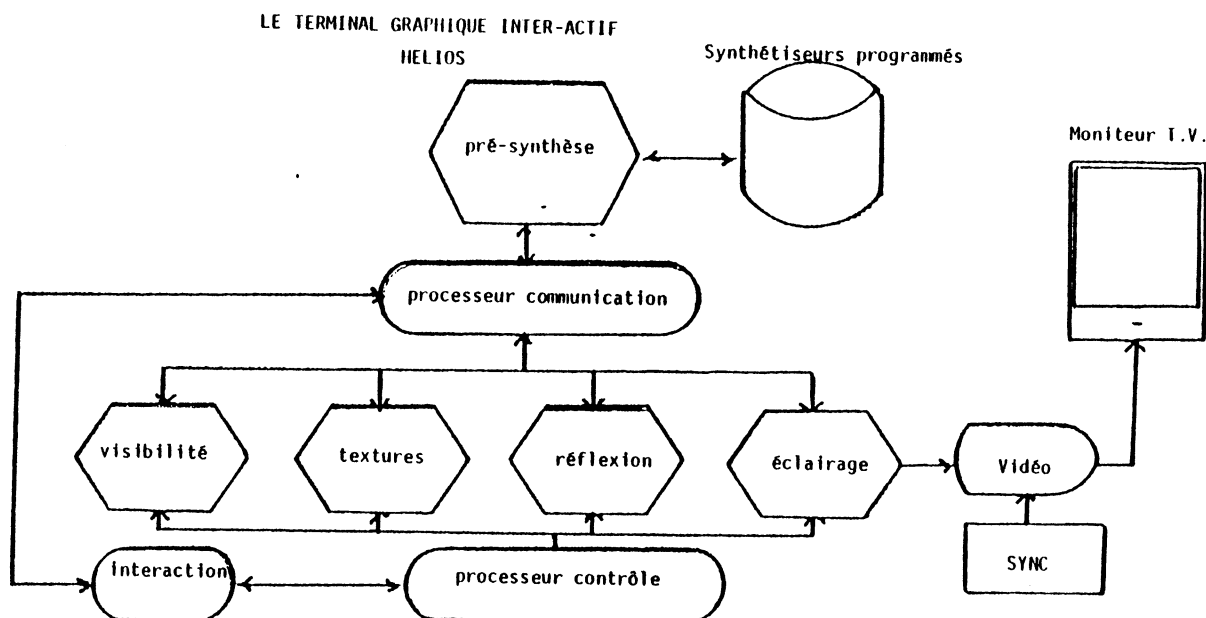


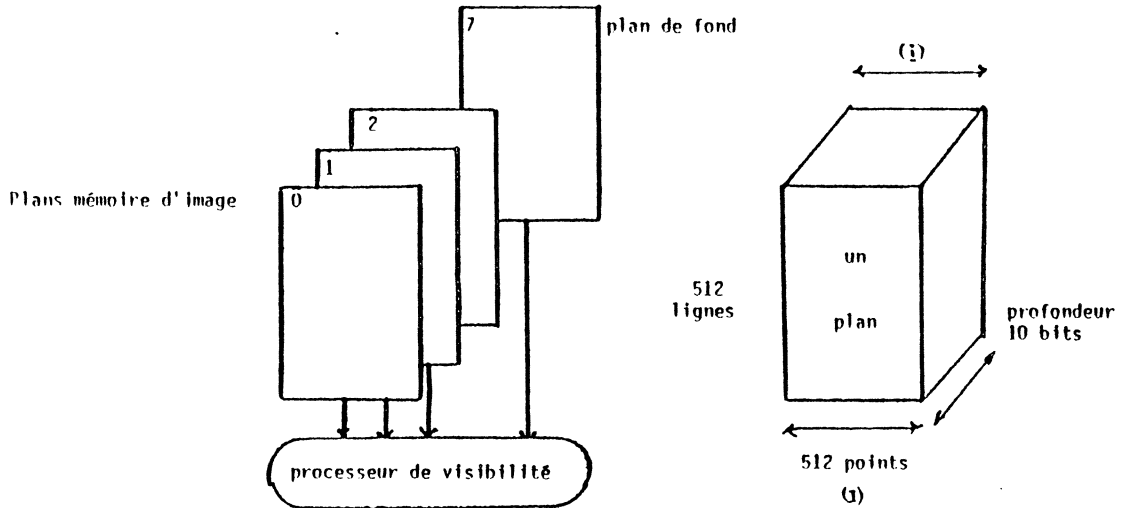
Figure No II.4

La complexite de la chaine de post-synthese (quadri-processeurs) implique la mise en oeuvre d'une pre-synthese adaptee et performante. La pre-synthese dispose d'un processeur unique qui est l'unité centrale du micro-ordinateur Pascaline.

#### II.1.5 DESCRIPTION DES PROCESSEURS :

Le processeur de visibilite constitue le synthetiseur initial de la post-synthese. Il accepte les attributs d'identite, de morphologie et de geometrie : I, M, G. Il gere l'affectation et la consultation des huit plans d'identification. Chaque plan a une capacite de 512 X 512 points avec 10 bits de profondeur par point. Un point peut appartenir a une face numerotee de 0 a 1023. Les plans sont numerotes de 0 a 7. Le plan 7 est appele plan de fond. Il ne memorise qu'un numero de face visible pour tous les points.





LES PLANS D'IDENTIFICATIONS

Figure No II.5

Le processeur de visibilité recupere pour chaque plan les parametres de fenestration qui sont le coin superieur gauche, la taille de la fenetre et le mode de decoupage de la vue a l'interieur de la fenetre. ( FER 81 ) La table des faces peut contenir 1 indicateur de visibilité par plan pour chaque face, soit 7 indicateurs au total. L'attribut de visibilité peut etre modifie en temps-reel pour une face ou un groupe de face.

Le processeur des textures effectue deux operations associees a cet attribut, la projection de la texture dans le plan de la face concernee et le pavage de la texture. La matrice de projection axonometrique est memorisee dans la table des faces. (point de vue a l'infini) Une texture est caracterisee par sa couleur intrinseque associee a une face. Cette couleur est exprimee en trichromie a partir des composantes fondamentales Rouge, Vert et Bleu codees sur quatre bits chacune. La texture est definie dans le plan de la face associee. Elle constitue une simulation visuelle de l'aspect d'un materiau. La banque des textures peut contenir jusqu'a 64 K-points codes sur 12 bits. De plus on peut associer a une face ou a un groupe de faces un pointeur unique dans la banque des textures afin de simuler un materiau quelconque. ( FER 81 )

Le processeur de reflexion determine les coefficients en chaque point de l'ecran. Les vecteurs exprimes en coordonnees spheriques sont la normale de la face  $N$ , la direction de la source lumineuse et la direction mediane entre la source et l'observateur. Les coordonnees spheriques s'expriment en fonction de deux angles  $\alpha$  et  $\beta$ . L'observateur dans ce repere est considere comme etant a l'infini sur l'axe  $Oz$ .

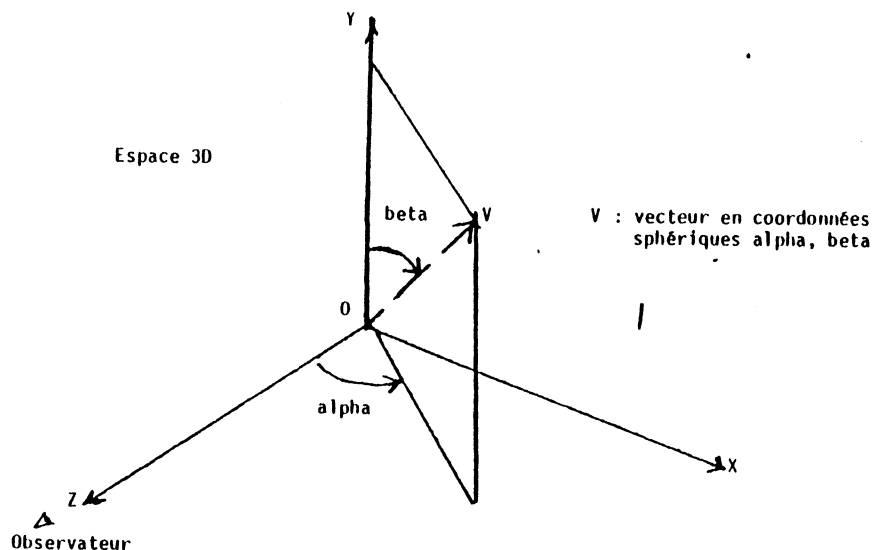


Figure No II.6

Le calcul de la reflexion tient compte de la lumiere ambiante qui est codee sur 4 bits. Les angles alpha et beta sont codes sur 5 bits chacun, soit 1024 directions differentes, dans le demi-espace avant de l'ecran. Ce processeur calcule deux composantes qui sont la reflexion speculaire et la reflexion diffuse. Ce processeur possede une banque des modeles de reflexion, chaque face peut etre associee a un modele par l'intermediaire de la table des faces. Le calcul explicite des composantes de la reflexion est decrit et developpe dans d'autres publications de l'equipe.

Enfin le processeur d'eclairage constitue l'operation finale de la chaine de post-synthese. Il compose les attributs d'aspect et d'eclairage. Il calcule la couleur en chaque point de l'ecran en tenant compte de l'attribut d'aspect c'est-a-dire la couleur issue du processeur des textures et les coefficients de reflexion issus du processeur de reflexion, Il tient compte aussi de l'attribut d'eclairage qui est la couleur de la source lumineuse et l'intensite de la lumiere ambiante. La couleur de la source lumineuse est memorisee dans un registre interne du processeur. Le modele du calcul de l'eclairage est decrit en detail dans d'autres publications de l'equipe.

Le processeur reticule est essentiellement attache au pupitre d'interaction, ce pupitre dote de six touches permet de gerer les déplacements du reticule. Les attributs caracterisant le reticule sont la couleur, la visibilite, et la position (X,Y) dans l'espace-ecran. La morphologie est caracteristique. Il s'agit d'une croix constituee de deux lignes de l'ecran, l'une horizontale et l'autre verticale. L'intersection des deux lignes permet de designer une face sur l'ecran, de collecter des coordonnees X, Y d'un point de l'ecran et d'affecter ou de consulter des attributs.

Le processeur de communication assure l'interface entre le bus de données d'HELIOS (24 bits) et le bus de données du calculateur satellite. Il permet à la chaîne de pré-synthèse d'initialiser les différentes mémoires locales associées aux processeurs de la chaîne de post-synthèse. Une fonction de contrôle assure la synchronisation et délivre les différentes horloges nécessaires aux autres processeurs de la post-synthèse. Il apparaît comme une batterie de registres adressables par le calculateur satellite. Ces registres indiquent :

- La nature de l'accès : lecture / écriture,
- La destination de l'accès : mémoire concernée,
- Le processeur mis en jeu.

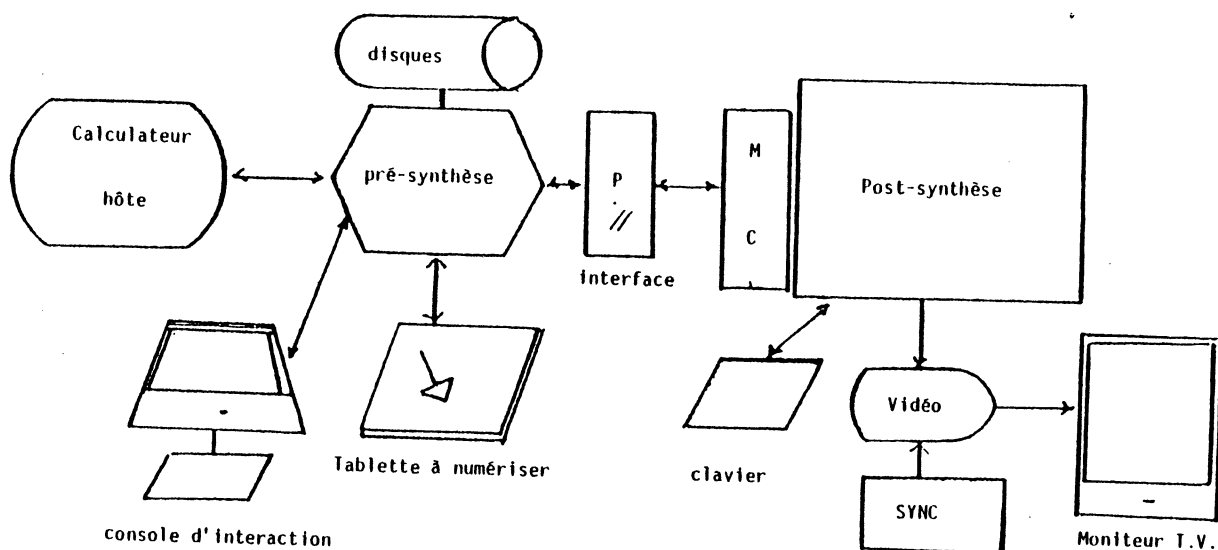
En conclusion, cette architecture permet de mettre en évidence une chaîne de post-synthèse évoluée traitant des attributs sophistiqués en temps-reel en raison du parallélisme de la chaîne et de la méthode de réalisation choisie. La chaîne de pré-synthèse mise en œuvre dans la configuration satellite permet de spécifier les différentes commandes à programmer afin de décharger le calculateur hôte des fastidieux échanges avec le module de communication.

## II.2 INTERFACE EN AMONT :

### LE CALCULATEUR HOTE ET LE T.G.I :

#### II.2.1 LE "CALCULATEUR SATELLITE" :

Le premier prototype d'HELIOS était uniquement constitué de la chaîne de post-synthèse. La chaîne de pré-synthèse à l'exception du module de communication, est entièrement réalisée de manière programmée. Les échanges avec le module de communication se font à travers un port parallèle d'entrées/sorties de la Pascaline. Cette architecture peut être connectée à un calculateur hôte à travers une ligne asynchrone. (V.24-RS.232)



LA CONFIGURATION INITIALE

Figure No II.7

### II.2.2 MODES D'UTILISATION DU T.G.I :

Le pre-synthetiseur propose trois types d'utilisation du terminal.

**LE MODE TRANSPARENT :** Ce mode permet un dialogue direct entre le calculateur hôte et l'opérateur. Les synthetiseurs cables de la post-synthèse ne sont pas sollicités.

**LE MODE LOCAL :** Ce mode permet de piloter la chaîne de post-synthèse localement sans prendre en compte une communication éventuelle avec le calculateur hôte. Ce pre-synthetiseur engendre les différents accès au module de communication.

**LE MODE OPERATION :** Ce mode de fonctionnement en ligne avec le calculateur hôte, permet de recevoir, de decoder et d'exécuter les commandes envoyées depuis le calculateur par une application programmée.

Le logiciel pilote donne accès aux fonctionnalités du terminal qui sont l'affectation, la consultation et la description des mémoires d'attributs, l'interaction, l'identification et la collecte de coordonnées, et enfin la visualisation d'entités graphiques en tenant compte de leurs attributs respectifs.

## II.2.3 DIALOGUE ET ECHANGES D'INFORMATIONS :

Dans la configuration "calculateur satellite", la communication se fait via une ligne serie asynchrone. Il est necessaire de decrire brievement la nature des informations echangees entre le calculateur hote et le pre-synthetiseur, ainsi que leurs types et leurs codages. Ces informations sont essentiellement les attributs initialises dans les memoires des differents processeurs de la post-synthese :

TYPE D'INFORMATION -----	TAILLE -----
Le numero de face	10 bits
La couleur	12 bits
La direction de la source lumineuse	10 bits
Les coordonnees :	2 X 10 bits

Le premier objectif d'optimisation des echanges entre le calculateur hote et le pre-synthetiseur programme implique le choix d'un modele de communication performant et d'un codage et d'une representation de l'information simples. Le second objectif consiste a choisir un port de communication adapte entre le pre-synthetiseur et le module de communication de la post-synthese repondant aux memes criteres. Dans la configuration "calculateur satellite" l'interface parallele de la Pascaline permet un debit de 2 M-bits par seconde. Cependant, certaines actions de la pre-synthese via le module de communication, ne permettent pas un tel debit d'echanges avec le pre-synthetiseur programme.

## II.3 INTERFACE EN AVAL :

## LA LOGIQUE CABLEE ET LE PRE-SYNTHETISEUR :

## II.3.1 MODULE DE COMMUNICATION :

Cette interface se presente sous l'aspect d'une memoire locale. Il s'agit d'un groupe de huit registres specialises. L'accès a ces registres se fait de maniere externe a travers le port parallele du micro-ordinateur Pascaline.

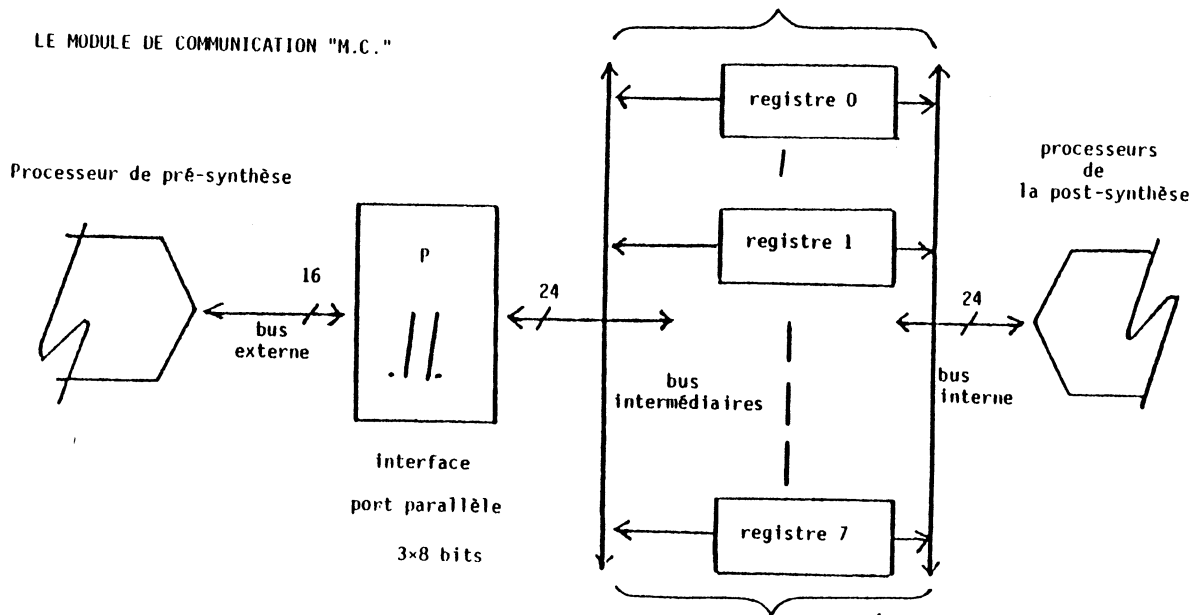


Figure No II.8

Le bus intermédiaire est un bus classique d'accès à une mémoire locale. Il est constitué des lignes d'échange suivantes :

- Un bus d'adresse : A0,A1,A2 (Adressage des registres),
- Une ligne de lecture / écriture : L / E,
- Une ligne de sélection : HELIOS,
- Un bus de données : D.0 à D.23 (24 bits).

Ces registres réalisent l'adaptation entre le bus externe du processeur de la pré-synthèse et les différents bus internes des processeurs de la post-synthèse. Un protocole par demande et acquittement est mis en œuvre par l'intermédiaire d'un ensemble de registres de contrôle et d'état de la communication. Ainsi plusieurs accès sont nécessaires pour lire ou écrire vers HELIOS.

### II.3.2 DESCRIPTION DES REGISTRES :

#### LE REGISTRE D'ETAT :

Ce registre est accessible en lecture seulement, il indique la disponibilité du registre d'entrée-HELIOS, la disponibilité du registre de sortie-HELIOS, et l'état de la synchronisation-video. (retour-trame et retour-ligne)

#### LE REGISTRE D'ENTREE :

Ce registre permet d'écrire vers une mémoire interne de la chaîne de post-synthèse. L'information est définie dans un format compatible avec l'organisation de la mémoire concernée.

#### LE REGISTRE DE SORTIE :

Ce registre permet de lire une mémoire interne de la chaîne de post-synthèse. Cette lecture est effectuée de manière anticipée des que le registre d'adresse est positionné.

#### LE REGISTRE D'ADRESSE : X et Y

Ce registre contient l'adresse d'une mémoire mono-dimensionnelle ou bi-dimensionnelle de la chaîne de post-synthèse. Il s'incrémente automatiquement. L'accès se fait sur la position mémoire suivante. La table des faces est une mémoire mono-dimensionnelle, un plan d'identification est une mémoire bi-dimensionnelle.

#### LE REGISTRE MODULE :

Ce registre est initialisé avant chaque accès à la post-synthèse. Il indique le processeur concerné par l'accès parmi les quatre possibles, au sein d'un processeur, il précise la nature de l'accès, lecture/écriture et la mémoire sélectionnée.

#### LE REGISTRE MODE :

Il permet de configurer le mode d'accès en fonction de l'organisation interne de la mémoire concernée. Il indique la synchronisation des accès avec le balayage vidéo.

#### LE REGISTRE REPETITION :

Ce registre permet d'effectuer des transferts compacts de données en écriture vers les mémoires de la post-synthèse. Il contient une valeur de 0 à 511. C'est le nombre de fois qu'une action doit être déclenchée par le module de communication. Cette option permet d'obtenir des débits d'informations vers la post-synthèse optimaux, en particulier pour les opérations de remplissage des plans d'identification et des mémoires de trame. L'optimisation réside dans le fait qu'une seule séquence d'accès au module de communication est nécessaire pour l'écriture consécutive de plusieurs informations.

## II.3.3 PROTOCOLE D'ECHANGE :

Il s'agit de séquencer des accès à travers le module de communication. La première opération concerne le mécanisme de lecture. Elle correspond à un processus matériel active par une séquence logicielle du pré-synthétiseur. Ce processus s'exécute au niveau du module de communication. Le registre d'état indique si la communication est possible, à cet instant l'information contenue dans le registre de sortie correspond à une donnée lue dans un format correspondant au processeur concerne de la post-synthèse. On note le parallélisme entre le processus matériel et la séquence logicielle, le registre d'adresse étant incrémenté automatiquement. Il existe un asynchronisme entre le mécanisme d'accès du module de communication et la séquence d'échantillonnage des informations à partir du registre-sortie.

La seconde opération concerne le mécanisme d'écriture. Ce processus matériel est active au niveau du module de communication par une séquence logicielle du pré-synthétiseur. L'écriture vers une mémoire de la post-synthèse, tient compte du registre de répétition qui indique le facteur de compactage des données. Ce facteur indique le nombre d'emplacements consécutifs d'une mémoire dans lesquels l'information doit être écrite. L'information à écrire est contenue dans le registre-entrée du module de communication, dans un format compatible avec celui de la mémoire accédée. Le registre d'adresse est initialisé pour indiquer le premier emplacement en écriture. à la fin de l'opération d'écriture ce registre contient la valeur de l'adresse de départ plus la valeur contenue dans le registre répétition. Ce processus d'écriture est initialisé par le pré-synthétiseur. Il s'exécute en parallèle et de façon asynchrone.

Des que l'information contenue dans le registre-entrée est lue par le processus matériel, l'état "entrée-libre" est "vrai", une autre écriture pourra être initialisée de façon anticipée et déclenchée lorsque l'état "communication-libre" sera "vrai". Le logiciel du pré-synthétiseur synchronise ces accès au module de communication à partir de ces deux états fondamentaux :

- Communication-libre : processus de communication libre ou occupé,
- Entrée-libre : registre d'entrée libre ou occupé.

Le pré-synthétiseur programme scrute systématiquement le registre d'état du module de communication avant chaque accès. Le registre de répétition permet de minimiser les échanges avec la pré-synthèse, tout en ayant un débit important d'informations vers la post-synthèse. Ces opérations seront évaluées dans le paragraphe II.4.3.

## II.3.4 EVALUATION DES ACCES :

Le séquencement des accès au module de communication permet d'effectuer



des lectures ou des écritures des mémoires de la post-synthèse. Il est nécessaire d'initialiser convenablement les différents registres adresse X et Y, compression, entrée / sortie et module. Ces initialisations se font par l'intermédiaire d'un ensemble de procédures. Une séquence d'accès vers HELIOS dépend d'un protocole d'échange qui consiste à :

- Choisir le module,
- Adresser la mémoire spécifique à l'intérieur du module,
- Sélectionner la nature de l'accès : lecture / écriture,
- Définir le mode d'accès : déclenchement effectif,
- Initialiser la compression de données,
- Définir l'information à écrire ou vérifier l'information lue.

Ce protocole s'exécute à travers l'interface parallèle de la Pascaline. Le temps de réponse de ces primitives correspond au débit de cette interface. En effet, la possibilité de compression des données à destination de la post-synthèse permet d'atteindre des débits de l'ordre de quelques dizaines M-bits par seconde. Le pré-synthetiseur programme détermine la performance globale des primitives. L'horloge de base du processeur de la Pascaline est de 3,0 MHz.

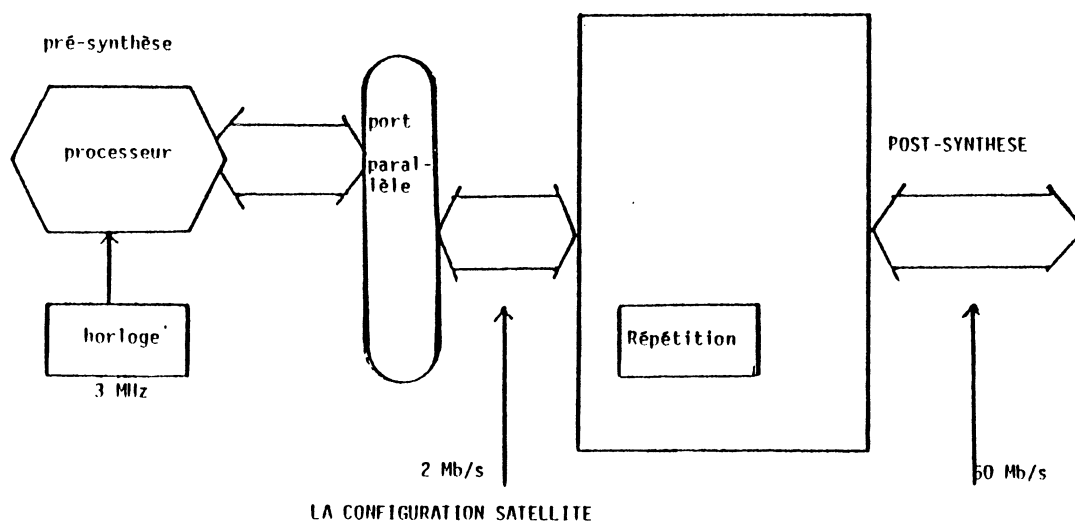


Figure No II.9

La procedure "out" permet de transferer une information de taille variable vers un registre du module de communication :

```
procedure out(adresse , donnee)
debut
  port.//.B := adresse ;
  port.//.B := donnee par tranche de 8 bits ;
fin
```

Cette procedure necessite deux fois 10 acces en memoire pour son execution, soit 20 acces au total. Le temps de reponse est de l'ordre de 6 a 7 microsecondes. Elle permet de transferer une information sur 8 bits vers la memoire selectionnee. Ainsi l'initialisation du registre-repetition implante a l'adresse '7' se fait comme suit :

```
procedure compact (nb : entier) ;
debut
  out(7 , nb);
  (* '7' adresse registre repetition *)
fin
```

L'evaluation de "out" represente environ 20 acces a la memoire. L'initialisation du registre repetition se fera en 30 acces dont un en memoire, soit environ 10 microsecondes. Cependant, une lecture ou une ecriture vers la post-synthese est une combinaison d'accès de base aux differents registres du module de communication.

Un acces a un registre specifique implique une sequence d'initialisation des ports A et B de l'interface parallele de la Pascaline. Le transfert d'information peut se faire sur 8, 16 et 24 bits. L'evaluation des operations de base indique les temps de reponse suivants :

- La lecture : un maximum de 8 acces, soit environ 26,4 microsecondes,
- L'ecriture : un maximum de 6 acces, soit environ 20 microsecondes

## ALGORITHMES :

```

-----
procedure ecriture; (* 8,16 ou 24 bits consecutifs *)
debut
  pour i:=1 jusqu'a nbr-octets
  faire
    out(adr+i-1, data[i]);
  ffaire
fin

```

```

procedure lecture;
debut
  port.//.B := 0 ;
  strob-lecture vers port.//;
  pour i:= 1 jusqu'a nbr-octets
  faire
    port.//.B := adr+i-1;
    data[i] := port.//.A;
  ffaire;
  invalider strob-lecture sur port.//;
fin

```

Le declenchement de la prise en compte de l'operation a lieu lors de l'ecriture dans le registre module qui necessite une sequence d'accès supplementaire a travers le port parallele. Soit au total :

- La lecture-HELIOS : 33,1 microsecondes,
- L'ecriture-HELIOS : 26,7 microsecondes.

En conclusion, on remarque que l'accès au module de communication est sensiblement ralenti par l'interface parallele. Une amelioration immediate serait de pouvoir acceder directement aux differents registres a partir du bus interne de la pascaline. De plus dans cette configuration la repartition des taches est limitee par les performances du pre-synthetiseur macro-programme. Ce point de l'etude est presente dans le paragraphe suivant.

## II.4 PRINCIPALES FONCTIONS-HELIOS :

Ces fonctions ou commandes sont envoyees au pre-synthetiseur a travers une ligne asynchrone. La performance globale du systeme se degrade etant donne le volume d'informations echange entre le calculateur hote et le terminal. Une etape intermediaire consiste a mettre en oeuvre une memoire

auxiliaire pour le pre-synthetiseur dans la configuration satellite. Cette memoire peut etre l'unite de disques souples de la Pascaline dans laquelle on effectue un archivage local :

## LA CONFIGURATION SATELLITE DOTE E D'UNE PRE-SYNTHESE PROGRAMMEE

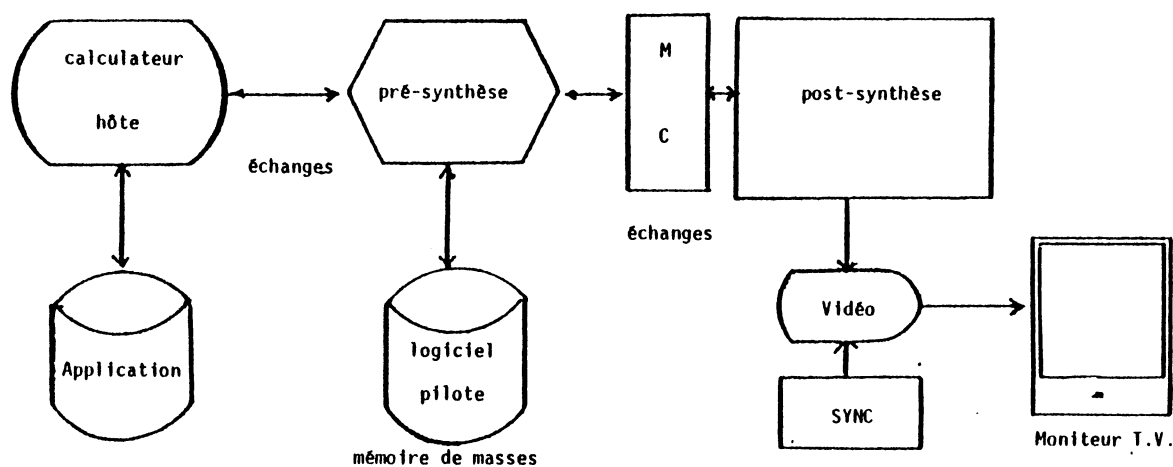


Figure No II.10

Un fichier graphique et des banques de texture et de reflexion sont geres localement par le pre-synthetiseur. Cette amelioration permet un rapport de performances superieur a 10. L'accès aux fichiers stockes dans la memoire auxiliaire se fait par accès direct. Il s'agit dans ce cas d'une memorisation intermediaire qui permet l'economie d'echanges d'informations volumineuses entre le calculateur hôte et le terminal HELIOS, a travers la ligne serie. ( SAR 82 )

## II.4.1 ATTRIBUTION ET CONSULTATION :

## ATTRIBUTION D'UNE FACE :

Ce processus concerne les attributs associes a l'entite graphique de base manipulee par le terminal, la face.

## LA MORPHOLOGIE :

Une primitive permet de definir un segment de droite dans un plan courant. Ce segment est identifie par un numero de face. Le segment est horizontal.

- segment (No plan, No Face, Y, X gauche, X droit)

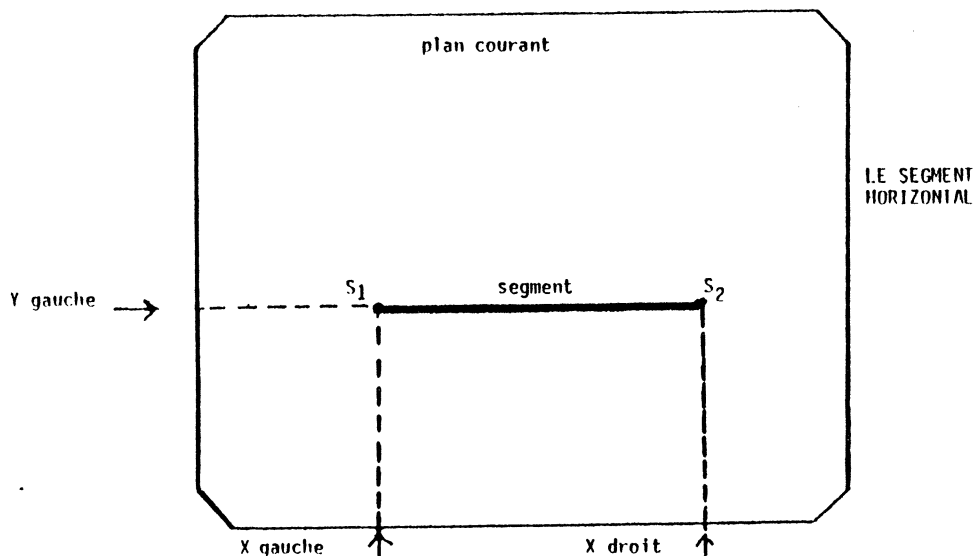


Figure No II.11

## L'ASPECT :

Il s'agit d'affecter une texture a un intervalle de faces predefini. Cette primitive affecte la texture choisie dans la banque du terminal. De plus une reflexion choisie dans la banque correspondante peut etre affectee de la meme facon. Enfin cette primitive attribue la visibilite ou la transparence au groupe de faces selectionne. Ce premier groupe de primitives affecte les attributs de geometrie, de morphologie, d'aspect et de visibilite d'une face. L'eclairage et la geometrie de l'affichage sont des attributs concernant la scene complete.

## CONSULTATION D'UNE FACE :

La consultation se fait a l'aide des primitives duales de celles existant pour l'attribution. L'ensemble des attributs est memorise dans les differentes memoires internes de la post-synthese. La consultation concerne les classes d'attributs suivantes qui sont l'identification, l'aspect, la geometrie et la geometrie de l'affichage.

## II.4.2 DESCRIPTION ET VISUALISATION :

Dans la configuration "calculateur satellite" aucun processus de description n'est disponible au niveau du terminal. La gestion de ce processus est entierement a la charge du pre-synthetiseur programme. Le logiciel pilote offre un premier niveau de description. Les primitives de description sont associees aux differents dispositifs de dialogue et d'interaction.

La visualisation constitue une séquence d'accès effectuée par l'intermédiaire du module de communication. L'option de compression de données est fortement sollicitée lors des opérations d'écriture dans les mémoires de trame. Par contre, les primitives de visualisation d'entités graphiques de base nécessitent l'exécution d'algorithmes spécifiques utilisant le parallélisme possible entre la pré-synthèse et la post-synthèse, en particulier l'incrémentation automatique du registre-adresse. Les entités graphiques de base sont :

- Les vecteurs, les lignes brisées,
- Les trapèzes avec remplissage éventuel,
- Les cercles,
- Les caractères, etc

L'ensemble des attributs décrits précédemment peut être affecté à ces entités. On évaluera deux algorithmes de génération d'entités graphiques par le pré-synthétiseur programme. Il s'agit de la génération de vecteurs, et de la génération de caractères.

#### II.4.2.1 GENERATION DE VECTEURS : -

L'algorithme choisi tient compte des possibilités offertes par le module de communication. Il s'appuie essentiellement sur la primitive de base de génération d'un segment horizontal. Un vecteur est défini par  $(x_d, y_d)$ ;  $(x_g, y_g)$ .

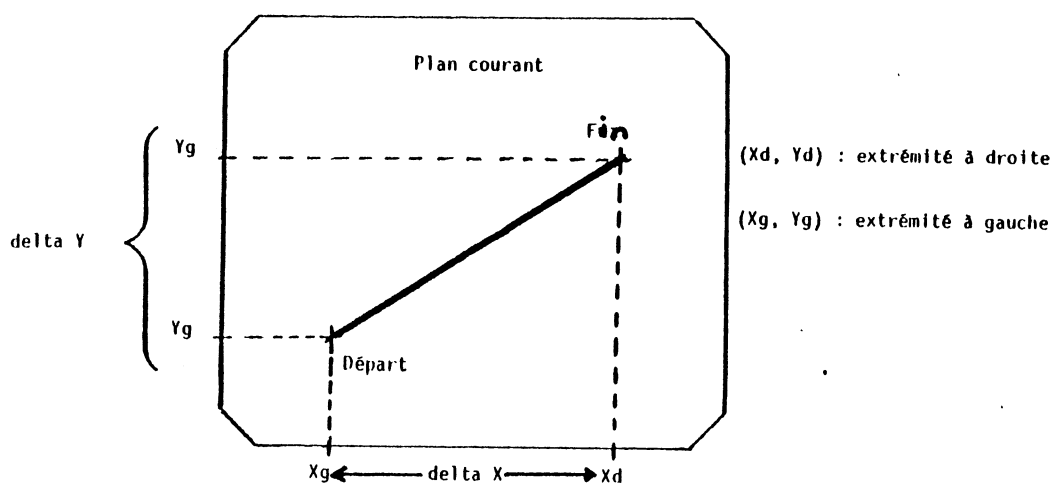


Figure No II.12

Le vecteur est approxime par une suite de portions de segments horizontaux. Pour chaque ligne d'affichage on effectue :

```

Debut
  Pour Y <- ydeb a yfin
  faire
    calculer ( origine, nombre de points )
    afficher ( segment-horizontale )
  finfaire
fin

```

Une primitive "NUMERISE" determine pour chaque ligne de balayage, le segment horizontal a afficher. Un cas particulier est isole dans l'algorithme, il s'agit du vecteur horizontal dont la generation se ramene a un appel a la primitive "segment" du logiciel pilote.

#### ALGORITHME SIMPLIFIE :

```

-----
procedure segment(xd,xg:entier);
debut
  tantque valeur de repetition <> 0
  faire
    positionner(xd,xg)
    initialiser(repetition)
    initialiser(entree)
    Calculer(valeur de repetition)
  finfaire
fin

```

La primitive "segment" effectue une moyenne de 12 acces a travers le port parallele afin d'initialiser le module de communication. (1 acces a 3 MHz) Si on considere que les delais d'entrees/sorties sont preponderants par rapport au temps effectif de calcul de la procedure "REMPVEC". La vitesse de traces moyenne d'un vecteur est :

$$V = ( V_{min} + V_{max} ) / 2$$

Resolution : 256 x 256 points.

V<sub>min</sub> : vecteur horizontal de 256 points,

V<sub>max</sub> : vecteur vertical de 256 points.

Soit environ 1,2 millions de points par seconde en vitesse de pointe et environ 0,6 millions de points par seconde en vitesse moyenne. Des valeurs intermediaires de taille et d'orientation de vecteurs ont ete evaluees du point de vue de la vitesse de generation pour obtenir cette performance moyenne. (vecteurs horizontal, vertical et diagonal pour des projections

multiple de 1/8 de la largeur de l'ecran.) La procedure "rempvec" effectue la generation de vecteurs :

ALGORITHME :

-----

```

procedure rempvec;
debut
  si yf <> yd alors
  faire
    si yf < yd alors permutation finsi
    initialiser (dx, dy, ix, xdeb, xfin, h);
    pour j:=yd jusqu'a yf
    faire
      si xgauche <= borne sup de l'ecran alors
      si xdroit > borne sup de l'ecran alors
        xdroit := borne sup de l'ecran finsi
      segment(j, xgauche, xdroit);
    finsi;
  ffaire
  ffaire
  sinon ecriture segment horizontal;
  si xf > borne sup de l'ecran alors xf := borne sup de l'ecran
  segment(xdroit, xgauche, xf);
  finsi
fin

```

En conclusion, la vitesse de generation est appreciable, cependant l'accès au port parallele introduit un delai de propagation des informations du bus interne de la Pascaline au bus intermediaire du module de communication.

REMARQUE :

La procedure "SEGMENT" est optimisee car elle n'accède pas au module de communication par les primitives standard "LECTURE" et "ECRIURE" decrites precedemment. Les acces se font directement a travers le port B du circuit d'interface parallele de la Pascaline.

#### II.4.2.2 GENERATION DE CARACTERES : -

La technique de generation des caracteres s'appuie sur une representation matricielle classique. La taille de la matrice est de neuf lignes par sept colonnes, Soit 63 bits d'information necessaires pour visualiser un caractere.



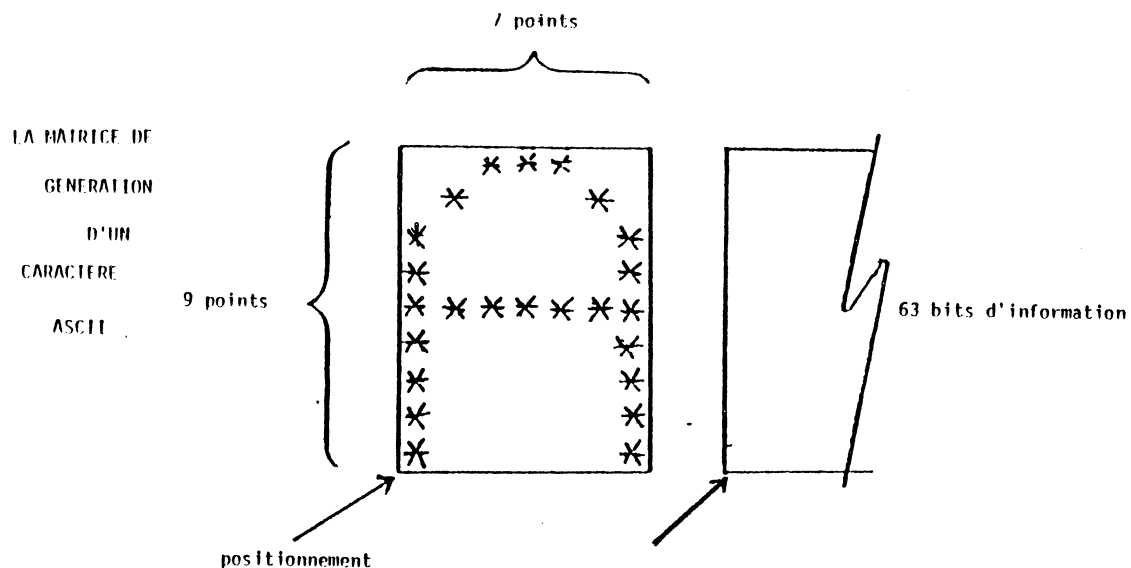


Figure No II.13

L'algorithme utilise les possibilites de compression de donnees pour le remplissage d'une ligne de la matrice. La primitive "SEGMENT" est appelee neuf fois successivement pour un caractere donne, le nombre de points allumes pour une ligne etant determine prealablement. Une fonction permet de determiner le coin superieur gauche a partir duquel se fait la generation. L'algorithme effectue l'affichage d'un caractere a la fois:

## ALGORITHME :

```

-----
procedure remp7X9 (car, x, y, no-plan, no-face);
debut
  si car E ensemble des caracteres ASCII alors
  faire
    pour j:=1 jusqu'a 9
    faire
      n:=0;
      xx:=x;
      pour i:=1 jusqu'a 7
      faire
        si tabc7X9[ord(car), i, j] alors n:=n-1
        sinon faire
          si n<>0 ecriture de n points
            alors segment(y+j-1, xx, xx+n-1) n=0;
            sinon xx:=xx+1
          finsi
        ffaire
      ffaire
    si n<>0 alors segment(y+j-1, xx, xx+n-1)
  ffaire
finsi
fin

```

Une phase d'initialisation permet de se positionner dans la memoire d'image (registre d'adresse X, Y). Le caractere sera affiche a la position courante (Plan, Face, position). Globalement l'affichage d'un caractere se fait en :  $T.C = 9 \times T.SEGMENT$ . T-SEGMENT represente le temps d'execution de la primitive "SEGMENT" qui effectue 12 acces a travers le port parallele de la Pascaline soit environ 39,6 microsecondes.

Le temps total de visualisation est  $T-CARACTERE = 356$  microsecondes. C'est-a-dire environ 3000 caracteres par seconde, ce debit correspond approximativement a la vitesse de transmission de 250 K-bits/s. Cette performance met en evidence la puissance du compression de donnees du module de communication malgre des delais d'accès non negligables a travers le port parallele.

#### II.5 REPARTITION DES TACHES :

Il s'agit d'effectuer un partage equitable des traitements entre le calculateur hote et la chaine de synthese. Le pre-synthetiseur programme, materialise par le logiciel pilote, propose des primitives d'accès aux fonctionnalites de la chaine de post-synthese, dans une configuration "satellite". Les evaluations ont montre que cette configuration penalise les performances intrinseques de la post-synthese. Toutes les taches associees aux differents processus sont a la charge du logiciel pilote :

- Generation d'un vecteur,
- Generation d'un caractere,
- initialisation d'une memoire ou d'un registre de la post-synthese.

Les points faibles de cette configuration sont mis en evidence dans le cadre du probleme du remplissage de taches ou la necessite d'un processeur intermediaire se fait sentir...

##### II.5.1 REMPLISSAGE DE TACHES POLYGONALES :

Le probleme de remplissage de taches polygonales illustre bien la necessite d'une repartition des operations entre les differentes chaines de synthese. Une tache est par definition un polygone quelconque constitue d'un nombre fini de sommets :

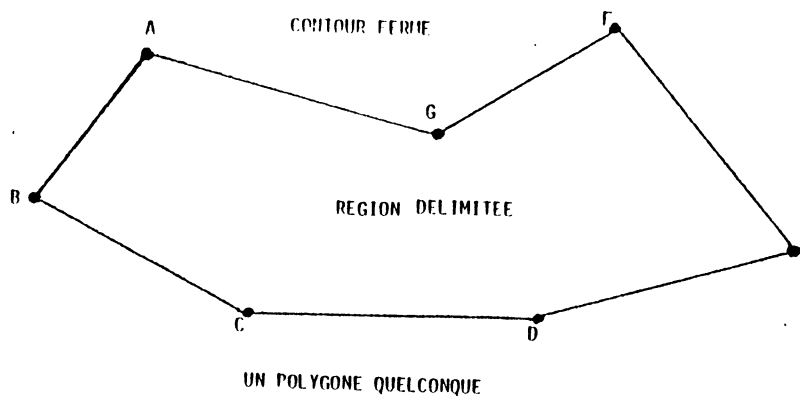


Figure No II.14

Dans la configuration "Calculateur satellite" le logiciel pilote se charge de quatre operations consecutives. Le remplissage se fait ligne a ligne. Cet algorithme est particulierement adapte aux possibilites d'ecriture dans la memoire de trame avec compression de donnees. Pour chaque ligne horizontale il faut :

- Determiner l'ensemble des aretes l'intersectant,
- Trier les aretes de gauche a droite (sens du remplissage),
- Calculer les points d'intersection,
- Afficher les vecteurs horizontaux reliant les points d'intersection deux a deux.

Le processus identifie les aretes concernees : FE, GF, AG, BA, puis effectue le tri : BA, AG, GF, FE, determine les intersections : P1, P2, P3, P4, et sequence la visualisation : P1-P2, P3-P4.

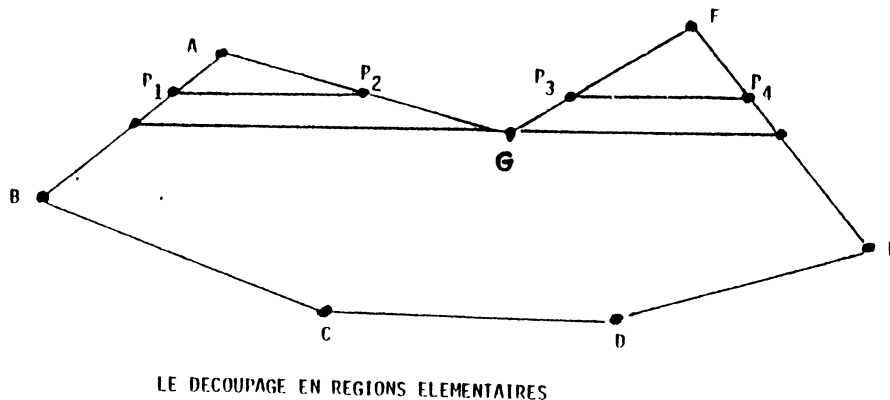


Figure No II.15

L'operation de visualisation represente la proportion de temps la plus faible pour une ligne. Les acces au module de communication concernent les registres suivants :

- L'initialisation du registre de repetition,
- L'initialisation des registres Entree, X, Y apres avoir positionne le registre module,
- Le remplissage du plan d'identification au rythme d'un point toutes les 120 nanosecondes.

Dans cette configuration, le parallelisme est negligeable, le pre-synthetiseur programme ne recupere que le temps du remplissage des plans d'identification, cette fonction etant a la charge du module de communication. Le remplissage d'une face necessite pres de 460 millisecondes. Le temps de reponse de cet algorithme pour un objet d'une cinquantaine de faces est de l'ordre de 23 secondes. ( SAR 82 ) Une nouvelle approche consiste a repartir les taches entre l'application et le logiciel pilote :

- L'application delimite des regions coherentes dans le polygone,
- Le logiciel pilote effectue le remplissage d'une region coherente qui peut etre un trapeze ou un triangle.

Le parallelisme est meilleur mais la communication via la ligne asynchrone de la morphologie d'une region coherente va de nouveau penaliser la performance globale du terminal. Cependant, cette decomposition permet une repartition plus equitable du traitement :

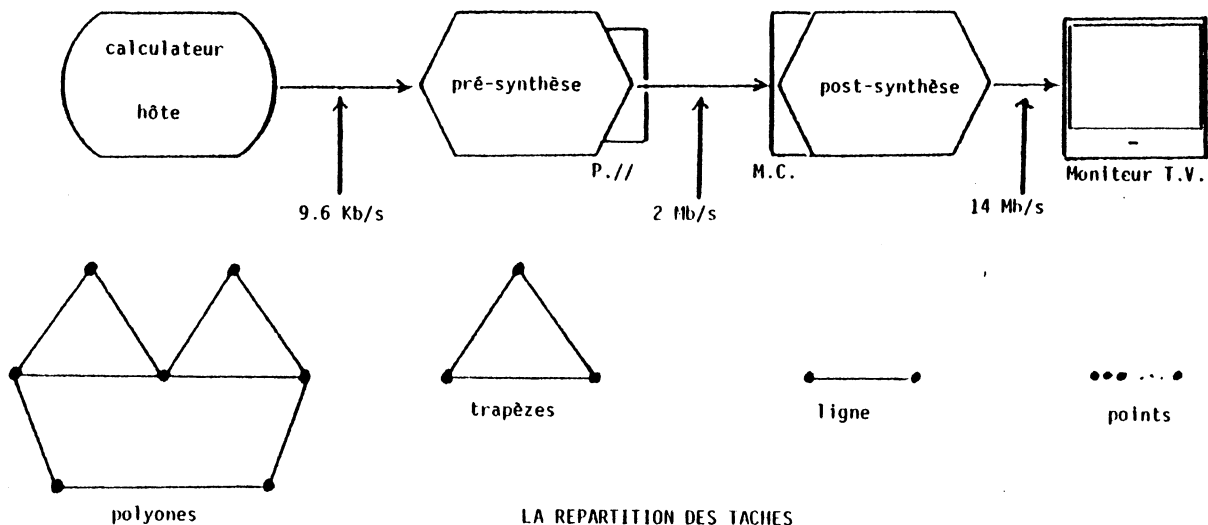


Figure No II.16

La recherche d'un parallelisme optimal implique l'accroissement de la puissance de calcul au niveau de la pre-synthese par l'adjonction d'un processeur intermediaire. Une communication meilleure entre le module de communication et la pre-synthese doit etre mise en oeuvre. La recherche d'algorithmes mixtes pour des processus de visualisation donnees permettrait de repartir les taches associees entre les deux sous-chaines de traitement. L'objectif est de minimiser les temporisations entre les differents etages :

APPLICATION => PRE-SYNTHESE => POST-SYNTHESE.

-----

#### II.5.2 TYPE DE PRE-SYNTHESE :

On peut considerer au depart deux types de pre-synthese s'appuyant sur une architecture de base ou evoluee. Il s'agit essentiellement d'accroitre la puissance de calcul de la pre-synthese en interposant entre le calculateur hote et la post-synthese un ou plusieurs processeurs capables de prendre en charge certaines taches et d'avoir un debit d'informations proche de celui du module de communication. Ce choix d'une architecture parallele illustre de plus le concept de partitionnement en couches decrit dans le chapitre I. (Definition d'univers intermediaires)

La chaine de post-synthese est coherente et symetrique. La chaine de pre-synthese conserve ces caracteristiques. De plus elle definit un univers initial pour l'application et produit un univers terminal intermediaire pour les processeurs de la post-synthese. L'indpendance de l'application vis a vis du materiel est renforcee. L'application peut manipuler des entites graphiques complexes et travailler a des niveaux de description, d'attribution et de consultation superieurs. Le compromis fixant l'architecture de la pre-synthese est fonction :

- Des fonctionnalites des primitives de bases (le logiciel pilote peut constituer un cahier des charges),
- Des performances requises par l'application,
- De la modularite de l'architecture,
- Du cout du ou des processeurs mis en oeuvre.

Ces criteres ont ete appliques dans le cadre de la realisation d'un pre-synthetiseur 2.D mono-processeur qui sera presente dans le chapitre III. La chaine de pre-synthese de base se resume a l'adjonction d'une interface microprogramme dotee de ressources locales, memoires, Entrees/sorties... et d'un microprocesseur en l'occurrence le MC.6809 qui est un pseudo 16 bits.

## II.6 DIFFERENTS TYPES D'INTERACTIVITES :

## II.6.1 DISPOSITIFS DE DIALOGUE :

Ce paragraphe decrit rapidement les fonctions interactives disponibles au niveau du pre-synthetiseur programme. Elles ne feront pas l'objet d'une evaluation. Le pre-synthetiseur programme gere trois dispositifs de dialogue qui sont la console-operateur, le reticule-integre et la tablette a numeriser. Ces dispositifs sont accessibles a partir de primitives du logiciel pilote. Le reticule peut etre pilote a partir de la pre-synthese via le module de communication. Cependant les differentes actions associees aux touches du pupitre d'interaction sont gerees par la post-synthese. Ce processus est entierement cable.

## II.6.2 FONCTIONS INTERACTIVES :

La fonction de base concerne l'attribution. Ce processus manipule essentiellement le reticule integre. La morphologie du reticule est implicite, ainsi que la geometrie. On dispose d'une fonction de positionnement du reticule dans l'espace ecran. L'aspect du reticule est defini par une primitive supplementaire. Cette primitive affecte les attributs d'aspect du reticule qui sont la visibilite et la couleur.

la seconde fonction est la consultation. Ce processus concerne en premier lieu l'attribut de geometrie. Une primitive permet de consulter la position courante du dispositif d'interaction. En second lieu, ce processus s'interesse a l'attribut d'aspect, une primitive permet de consulter l'attribut de visibilite du reticule. Cette information permet la validation d'une collecte de coordonnees.

Enfin la description constitue essentiellement une operation de collecte de coordonnees. Ces primitives de gestion des dispositifs d'interaction sont suffisantes pour des applications graphiques de base. Des fonctions interactives plus evoluees peuvent etre mises en oeuvre, au niveau de la pre-synthese. Cette amelioration sera presentee dans le chapitre suivant, dans le cadre de la nouvelle configuration.

En conclusion, le dispositif principal d'interaction est le reticule. Il est gere comme une entite specifique dote des attributs de morphologie, de geometrie et d'aspect. Les quatre processus fondamentaux sont associes a cette entite specifique. La visualisation est implicite. L'ensemble des primitives de consultation d'attribution et d'identification de faces peut etre utilise de maniere interactive et en temps-reel par l'intermediaire du reticule et du pupitre d'interaction ou encore a partir de la tablette a numeriser :

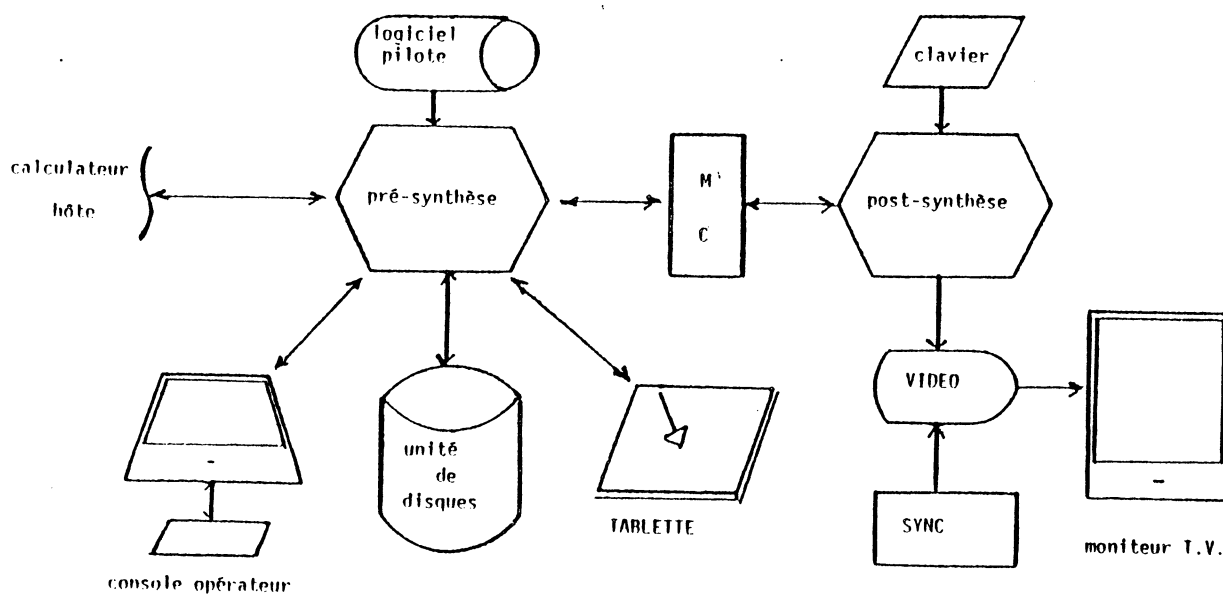


Figure No II.17

De plus, des fonctions locales sont exécutées par le logiciel pilote à la mise sous tension du terminal. Il s'agit d'initialiser l'ensemble des registres et mémoire de la post-synthèse. Des options standards ont été retenues. Cette initialisation globale du terminal peut être relancée localement à partir d'une touche-spécifique prise en compte par le pré-synthétiseur programme. Le détail des opérations d'initialisation est disponible dans d'autres publications. ( FER 81 ) ( SAR 82 ) ( MAR 82 )

Des commandes de configuration et d'archivage local sont disponibles au niveau du logiciel pilote. Dans la configuration satellite, cette option est possible par l'utilisation des mémoires de masse pré-existantes au niveau de la Pascaline.

## CONCLUSION :

Les insuffisances de cette configuration sont essentiellement les supports d'echange utilises par la pre-synthese. Le port parallele du micro-ordinateur Pascaline ralentit considerablement les acces au module de communication. De plus la communication avec le calculateur hote se fait par une ligne serie sans protocole specifique. Enfin les algorithmes de generation d'objets graphiques elementaires sont executes au niveau de la Pascaline dont les performances U.C sont mediocres. (Horloge de base a 3 MHz et cycle memoire superieur a 500 ns)

Dans le cadre de la definition d'une nouvelle configuration, il est necessaire de determiner d'une part les taches maintenues au niveau d'un logiciel pilote ou de l'application; d'autre part les taches deportees au niveau de la pre-synthese de base, prealablement a la charge de l'application ou du logiciel pilote.

Des ameliorations ont ete apportees au niveau de l'architecture de la pre-synthese et de la post-synthese, en vue d'optimiser les echanges d'informations entre les differentes entites. De plus il est necessaire de se doter d'une puissance de calcul accrue integree au terminal presente dans une nouvelle configuration : "la console evoluee".

Elle sera presentee dans le chapitre suivant. Il s'agit du second prototype HELIOS II qui a ete developpe durant l'annee 1983 pour le laboratoire de robotique (actuel LIFIA). Sa realisation a ete en partie financee par un contrat A.D.I.





CHAPITRE III

ARCHITECTURE A PRE-SYNTHESEUR 2.D :

REFERENCES BIBLIOGRAPHIQUES :

|

( BRE 65 ), ( CON 82 ), ( FER 81 ), ( GRA 80 ),  
( MAR 82 ), ( MER 79 ), ( MOL 76 ), ( NES 79 ),  
( SAR 82 ).

LE PRE-SYNTHESEUR HELIOS-09 MONO-PROCESSEUR :

Cette nouvelle configuration dite "la console evoluee" est dotee d'un pre-synthesiseur microprogramme et integre au terminal HELIOS. Le chapitre III fait une description detaillee de ce pre-synthesiseur. Dans le cadre du developpement d'une architecture de synthesiseur d'images realistes de seconde generation, cette premiere realisation a permis de valider les concepts de pre-synthese de base. Elle est representee dans la figure suivante.

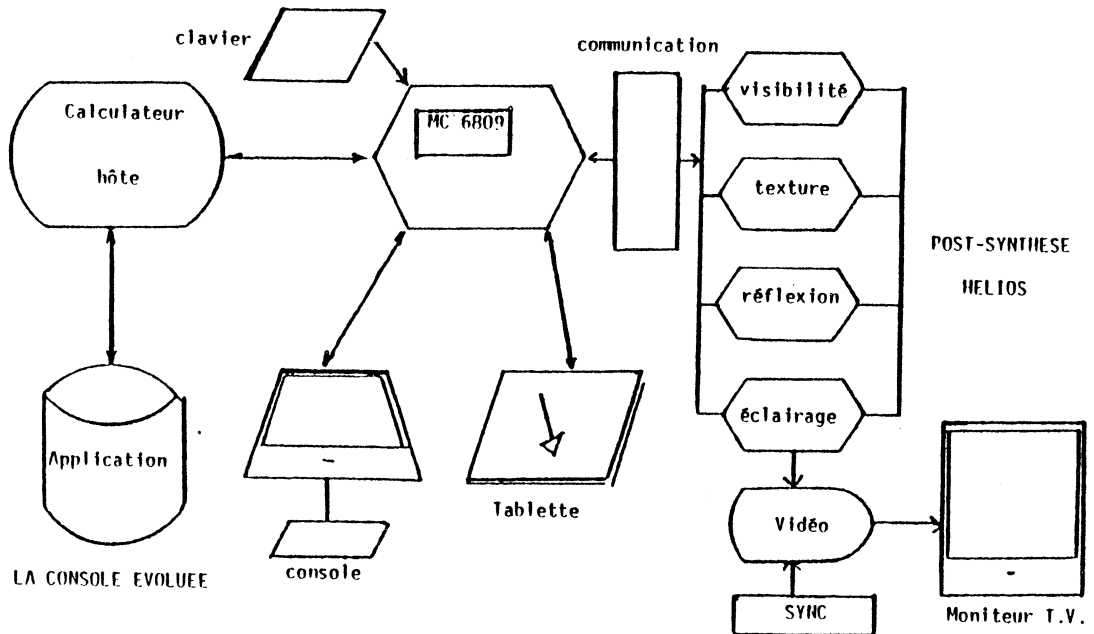


Figure No III.1

La realisation d'un second prototype HELIOS II, caracterise la configuration de "console evoluee". La notion de calculateur satellite disparaît. Le processeur de la pre-synthese se charge de la gestion des echanges avec le calculateur hôte, il pilote directement la post-synthese a travers le module de communication.

Les premieres ameliorations sont la simplification des echanges avec le calculateur hôte, l'augmentation du debit d'informations vers le module de communication. (le port parallele etant supprime) La mise en oeuvre d'un logiciel resident, modulaire et extensible, au sein du terminal a permis d'integrer au terminal des fonctionnalites meilleures.

## III.1 PRE-SYNTHETISEUR MICROPROGRAMME :

Le pre-synthetiseur programme de la configuration "calculateur satellite" a permis de definir un jeu de primitives evoluees permettant d'accéder aux fonctionnalités du terminal. Ce jeu de primitives constitue un cahier des charges initial des différentes tâches destinées à prendre place au niveau du pre-synthetiseur "microprogramme". Ce choix s'avère nécessaire, l'objectif étant de recadrer les performances globales du système dans une nouvelle configuration.

Le prototype mis en oeuvre et réalisé durant l'année 1983, était destiné au laboratoire de robotique LIFIA désirant développer des applications spécifiques tri-dimensionnelles et en couleur... Toutes les fonctions offertes par la post-synthèse sont accessibles par l'intermédiaire d'un jeu de commandes évoluées issue d'un calculateur-hôte. L'architecture présentée dans la figure suivante constitue la version II du synthetiseur d'images réalistes HELIOS.

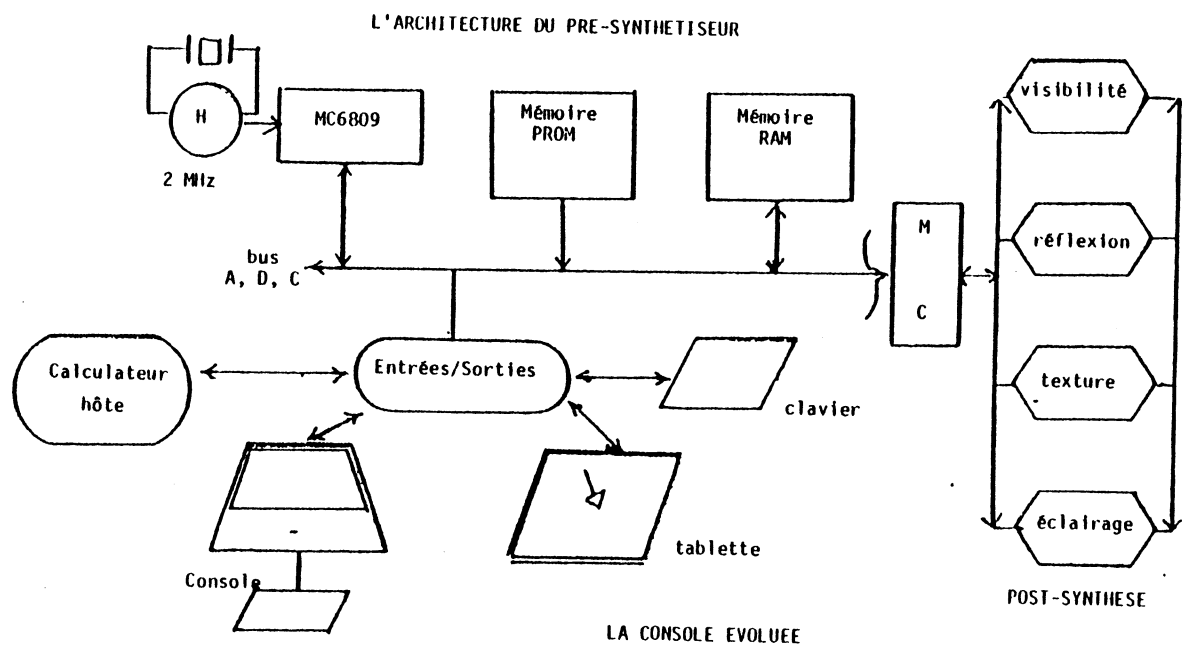


Figure No III.2

Le pre-synthetiseur microprogramme est constitué d'un microprocesseur et de son environnement classique. Des interfaces spécifiques ont été mises en oeuvre. Les tâches qui incombent au pre-synthetiseur concernent l'exécution des algorithmes associés au processus de visualisation :

- Le remplissage de tâches,
- La génération d'entités graphiques diverses : droites, lignes brisées, cercles, disques, sphères...

- La generation de caracteres : taille, orientation, police...

Les processus d'attribution et de consultation sont realises en tenant compte des caracteristiques de la post-synthese. Le processus de description concerne les diverses interactions et la collecte d'informations. Une gestion des differentes interfaces de communication a ete realisee. Il s'agit du controle des differentes lignes de dialogue avec le calculateur hote, l'operateur, les dispositifs d'interactions et les echanges avec le module de communication.

Ce recadrage des performances dans la nouvelle configuration a implique, d'une part l'optimisation des echanges entre la pre-synthese et la post-synthese, d'autre part la normalisation du dialogue entre le calculateur hote et la pre-synthese ainsi que la facilite de connexion. De plus une banalisation des primitives d'utilisation du terminal a ete realisee.

### III.1.1 CARACTERISTIQUES TECHNIQUES :

Cette architecture specifique est basee sur un microprocesseur pseudo-16 bits le MC.6809 de la famille Motorola. Le choix de ce composant s'explique facilement. C'est le processeur le plus puissant du marche dans sa categorie. (Bus de donnees de 8 bits). De plus, la simplicité de mise en oeuvre de ce composant est a signaler. Ses principales caracteristiques sont :

- L'horloge de base a 2 MHz soit un cycle de base de 500 nanosecondes,
- L'unite arithmetique et logique 16 bits avec multiplication 8 x 8 bits,
- Six possibilites d'interruption dont 3 par logiciel.

#### III.1.1.1 GESTION DES ENTREES/SORTIES : -

Le processeur MC.6809 gere trois lignes asynchrones de type V.24-RS.232. Cette gestion est faite par interruption afin d'optimiser la communication. Les interfaces des lignes de communication sont realisees a partir de circuits classiques de type ACIA - MC.6850. Les caracteristiques techniques des differentes lignes ainsi que la vitesse de transmission peuvent etre consultees et modifiees par l'intermediaire d'un module de configuration (SET-UP). Ce module est mis en oeuvre a partir d'un circuit d'interface parallele de type PIA-MC.6820 et d'une logique de multiplexage des vitesses de transmission.

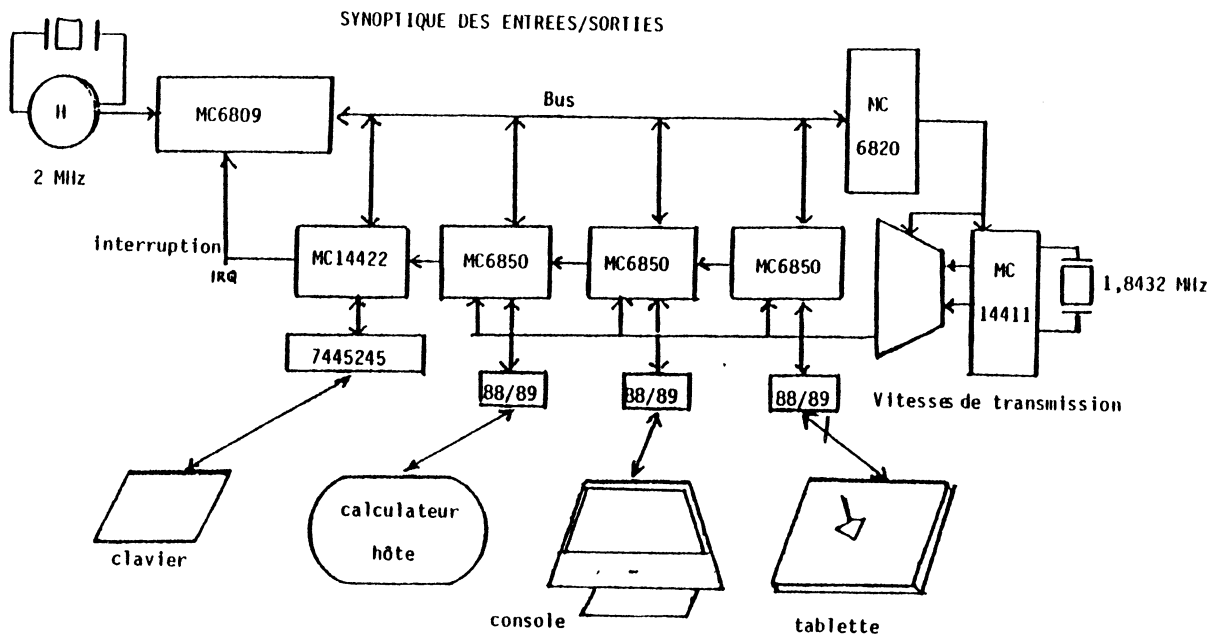


Figure No III.3

L'affectation des trois lignes se fait comme suit : Le port A est affecté à la console opérateur, le port B à la Liaison calculateur hôte et le port C à la tablette à numériser. Enfin le processeur gère une interface spécialisée de type parallèle pour le pupitre d'interaction constitué de 16 touches. Cette interface est constituée d'un circuit encodeur de clavier doté d'un circuit d'amplification. Deux états sont fournis au processeur par l'intermédiaire de deux bascules, l'état "touche enfoncée" et l'état "touche maintenue enfoncée". (répétition) La gestion de ce clavier se fait par scrutation systématique lors d'une session d'utilisation. Il peut être affecté à un niveau d'interruption.

#### III.1.1.2 MEMOIRES LOCALES : -

La mémoire de travail est constituée de trois pages de 2 K-octets chacune, soit une capacité totale de 6 K-octets. Elle permet la sauvegarde d'un ensemble d'informations utiles. En particulier les tampons associés aux lignes de communication, les variables locales du système ainsi que la pile. Une partie de l'espace mémoire est réservée aux paramètres accompagnant les commandes issues du calculateur hôte. La mémoire de programmes est constituée de cinq pages mémoire de 2 K-octets chacune, soit une capacité totale de 10 K-octets. Elle permet de stocker l'ensemble du micrologiciel résident chargé des tâches suivantes :

- La gestion des lignes de communication,
- La gestion du pupitre d'interaction,
- La gestion du module de communication,
- La recuperation, le decodage et l'execution des commandes issues du calculateur hote.

La memoire locale du pre-synthesiseur microprogramme a une capacite totale de 16 K-octets, ce qui represente le quart de la capacite d'adressage du processeur MC.6809.

#### III.1.1.3 INTERFACE D'ACCES AU M.C : -

Il s'agit d'une interface constituee d'un systeme de decodage et de lecture/ecriture de huit positions memoire. Ces registres organises en memoire locale, ont une taille variable. Le delai d'acces est fonction de la largeur de l'information :

- 1 octet en un cycle-memoire soit 500 nanosecondes,
- 2 octets en deux cycles-memoire consecutifs soit 1 microseconde.

La logique d'acces acquite les fins de transfert en tenant compte de la taille de la donnee. Les informations qui transitent du pre-synthesiseur vers la post-synthese ont une taille variable. Ainsi l'interface d'acces au module de communication est capable de reconnaitre deux largeurs de transfert qui sont 8 ou 16 bits. Cette interface possede :

- Un registre d'ecriture sur 16 bits,
- Un registre de lecture sur 16 bits,
- Un systeme de decodage de l'adresse du registre selectionne,
- Un systeme de decodage du sens du transfert,
- Un systeme d'acquieement en fonction de la taille de l'information (8/16 bits).

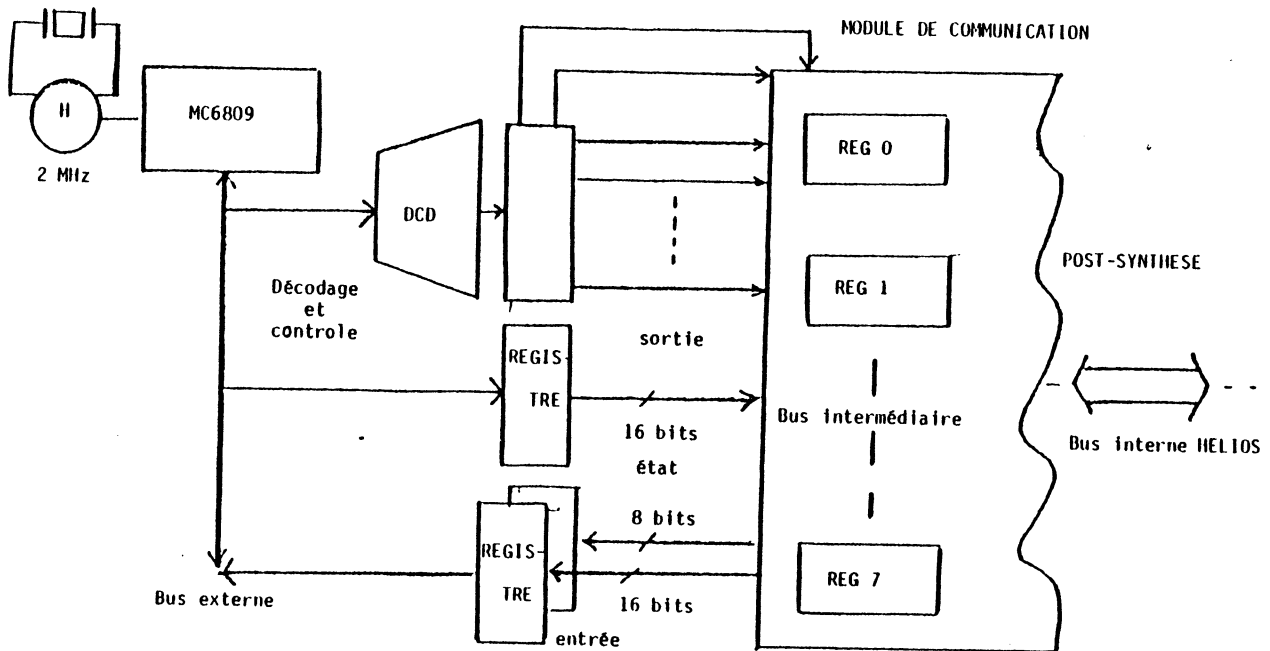


Figure No III.4

REMARQUE :

Le registre "ETAT" du module de communication est accessible directement en lecture et sur 8 bits par le processeur MC.6809. Ce registre est géré de façon spécifique car il indique si les lectures ou les écritures vers les autres registres sont possibles.

III.1.2 GESTION DU PRE-SYNTHESEUR :

Il s'agit de l'ensemble des microprogrammes résidents au niveau du pré-synthétiseur. Cet ensemble est constitué de quatre modules essentiels. Ces modules se chargent de la gestion des lignes de communication et de l'interface spécialisée d'accès à la post-synthèse HELIOS II. Ces modules proposent des primitives d'accès aux différentes interfaces décrites précédemment.

Le premier module gère le contrôle et le fonctionnement des deux lignes asynchrones du pré-synthétiseur. Il permet trois modes d'utilisation du terminal identiques à ceux de la précédente configuration. (les modes opération, local et transparent.) Il offre la possibilité de choisir une configuration des différentes caractéristiques des lignes par une action programmée, appelée "SET-UP". Il assure la connexion au calculateur-hôte et le dialogue utilisateur via la console opérateur. Les deux lignes sont attachées à des tampons de stockage et gérées en mode interruption par le processeur MC.6809.



Un second module est active en scrutation du pupitre si une commande interactive ou une collecte d'informations est prise en compte par le pre-synthetiseur. Le pupitre d'interaction possede 16 touches codees de 0 a 15. Ce module est dote d'une action de temporisation capable de detecter l'enfoncement d'une touche et le maintien d'une touche enfoncee.

Un troisieme module est aussi active en scrutation si une commande interactive ou une collecte d'informations est prise en compte par le pre-synthetiseur. Il s'agit d'un outil supplementaire d'interaction au niveau du terminal. Le pupitre ou la tablette a numeriser sont utililises simultanement avec la visualisation d'un reticule sur le moniteur TV du terminal.

Le dernier module gere la communication avec les quatre processeurs de la post-synthese. Il offre des primitives de base d'ecriture et de lecture d'une memoire d'un processeur quelconque de la post-synthese. Il assure les initialisations des registres adresse, entree/sortie, repetition et module. Ces registres sont identiques a ceux decrits dans la configuration "calculateur satellite". Ce module effectue les taches suivantes :

- L'initialisation des plans d'identification et des attributs associes, (table des faces)
- L'initialisation des banques de modeles de texture et de reflexion,
- L'initialisation des registres parametres : lumiere ambiante et couleur de la source lumineuse,
- L'initialisation des parametres du reticule : couleur, visibilite et position.

Un module de niveau superieur est charge essentiellement de la prise en compte des commandes graphiques destinees au terminal. Ce module est constitue d'une action d'interpretation des commandes en provenance du calculateur-hote ou emises localement. Il recupere le code-operatoire et les parametres associes d'une commande puis les communique au module d'accès a la post-synthese qui effectue l'execution en sequencant les transferts a travers le module de communication. Une seconde action se charge de l'initialisation du terminal, c'est-a-dire des deux chaines de pre- et de post-synthese. Cette initialisation consiste a :

- Configurer les differentes interfaces d'entree/sortie,
- Initialiser les processeurs de la post-synthese,
- Initialiser les variables locales, les parametres et la pile du micrologiciel du pre-synthetiseur.

La structure modulaire du micrologiciel est representee dans la figure suivante.

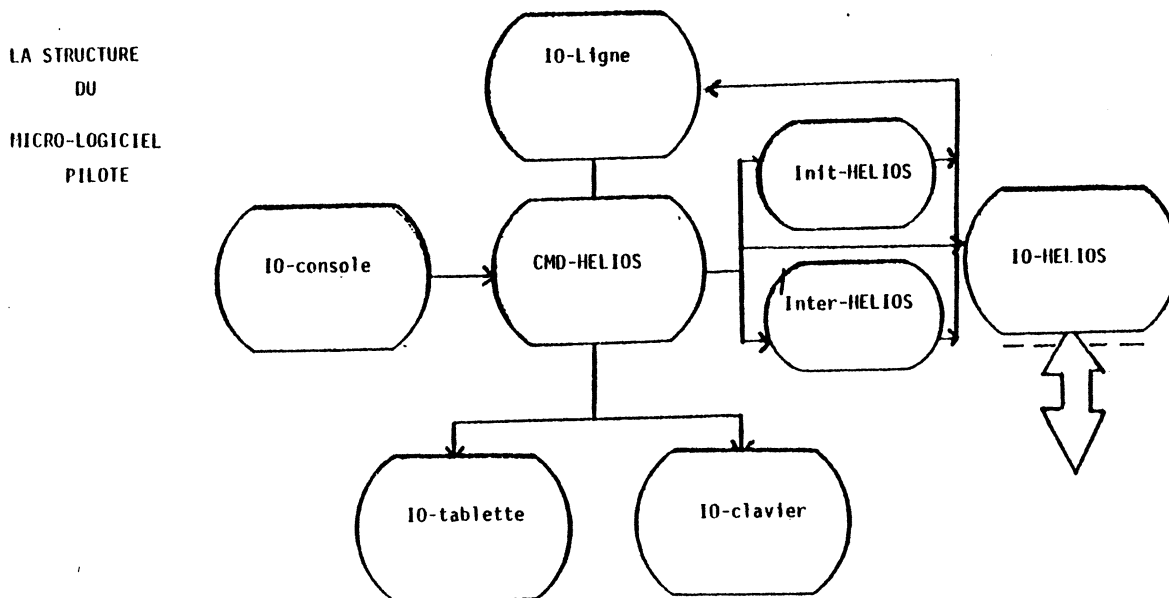


Figure No III.5

### III.2 MODES DE COMMUNICATION :

#### LE CALCULATEUR HOTE ET LA PRE-SYNTHESE :

La description des differents modes d'echange depend de la nature des informations circulant sur les deux lignes de communication principales qui sont la ligne console-operateur et la ligne calculateurs-hote. le protocole d'echange est defini selon un modele classique du type PRODUCTEUR - CONSOMMATEUR. Les deux lignes sont affectees au niveau d'interruption "IRQ" du processeur.

- La ligne calculateur-hote est associee a un tampon memoire de N positions, (N = 1024 octets)
- La ligne console-operateur est associee a un tampon memoire d'une position.

## III.2.1 PROTOCOLE D'ECHANGE :

Le tampon associe a la ligne "calculateur-hote" permet de stocker des commandes en attente de traitement. Le processus de production assure le remplissage du tampon. Le caractere special <escape> indique que les informations disponibles sur la ligne sont destinees au pre-synthesiseur.

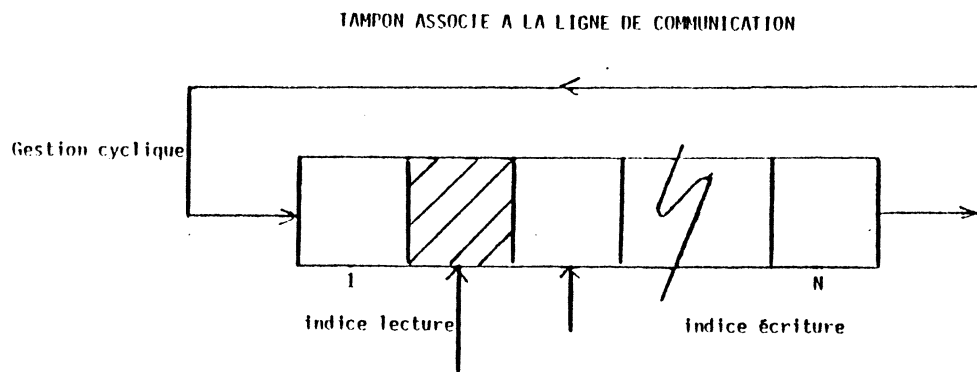


Figure No III.6

L'indice d'écriture est gère de façon cyclique. Le tampon est ensuite lu par le processus de consommation, il s'agit d'une réception différée des commandes issues du calculateur hôte. La lecture d'un caractère se fait de manière ininterrompue et provoque la libération d'un emplacement du tampon associé à la ligne de communication. Le contrôle du débordement du tampon se fait à partir de la condition fixée des indices de lecture et d'écriture. L'indice d'écriture ne peut dépasser l'indice de lecture plus d'une fois.

Pour la plupart des commandes le temps de transmission est largement supérieur au temps d'exécution du pré-synthesiseur. Pour certaines commandes "longues" cette condition n'est pas vérifiée et le risque de débordement du tampon est inévitable. Dans ce cas deux solutions sont proposées par le protocole d'échange afin de contrôler la communication :

- Un mécanisme d'acquiescement,
- Une procédure : "X.ON-X.OFF".

Le mécanisme d'acquiescement peut être systématique ou réservé aux commandes "longues" provenant du calculateur. La procédure "X.ON-X.OFF" peut être sélectionnée au niveau du "SET-UP" du pré-synthesiseur à condition qu'elle soit acceptée et gérée par le calculateur hôte.

En conclusion l'option de communication via une ligne asynchrone de type V.24-RS.232 procure une facilité de connexion avec de nombreux calculateurs dans une configuration "console évoluée". Par contre les performances du système sont dépendantes de la vitesse de transmission

selectionnee. Dans le cas d'une vitesse de l'ordre de 9600 bauds, le temps de transmission d'une commande a ete evaluee, les delais caracteristiques sont :

- T-min = 2 millisecondes,
- T-max = 533 millisecondes,
- T-moy = 268 millisecondes.

Cette evaluation a ete realisee a partir des syntaxes des differentes commandes acceptees par HELIOS II, la moyenne a ete calculee en fonction de la longueur de la commande, c'est-a-dire en fonction du nombre de caracteres transmis constituant la commande (code operatoire et parametres) et des vitesses de transmission standard. Ce point est detaille dans le paragraphe III.2.3. D'autres types d'interface peuvent ameliorer les performances globales du systeme, une interface parallele de largeur 8 bits par exemple. (Debit : 2 M-bits par seconde) Ce debit peut etre ameliore en definissant un bus de communication sur 16 bits. Dans ce cas les temps d'execution et de communication peuvent etre du meme ordre de grandeur. Des protocoles de type demande-acquitement, sont generalement mis en oeuvre. (D.M.A entre le calculateur hote et le terminal)

La console operateur est attachee au meme niveau d'interruption "IRQ" que la ligne "calculateur hote". Cependant, lors de l'occurrence d'une requete d'interruption, un sous-programme de reconnaissance de la source teste en priorite si la reception d'un caractere provient de la console-operateur. Ce sous-programme assure les taches suivantes :

- Le test du caractere recu,
- L'activation de la sequence de configuration des lignes, SET-UP si cette action est selectionnee par l'envoi d'un caractere special.
- L'activation de la sequence de connexion au calculateur hote si l'operateur selectionne le mode operation, le cas de MULTICS implique l'envoi de la sequence : "Break, return, return"
- Le deroutement de l'ensemble des caracteres recuperes a la console operateur vers le calculateur hote tant que le mode local n'est pas selectionne.

On notera que seule la reception est geree par interruption qu'il s'agisse de la ligne "calculateur - hote" ou de la "console - operateur". L'emission est par contre geree de facon classique au niveau de l'interface de communication serie, c'est-a-dire par scrutation systematique de la disponibilite d'un registre d'emission.

Les différents modes de fonctionnement mis en oeuvre sont identiques à ceux de la configuration "calculateur satellite", cependant le mode transparent est supprimé.

#### LE MODE LOCAL :

Ce mode est sélectionné par une commande spéciale issue de la console-opérateur. Dans cette configuration, le pré-synthétiseur microprogramme récupère les commandes directement, introduites par l'intermédiaire de la console. Les commandes sont interprétées et exécutées immédiatement, ce mode permet d'effectuer un test rapide du terminal et de déclencher une opération ponctuelle sur une image. Le dialogue avec le calculateur hôte est suspendu pendant toute la session d'utilisation en mode local. Cependant l'ensemble des commandes provenant de l'application est mémorisé. L'exécution de ces commandes se fera en différé, une fois le mode local abandonné. La mémorisation des commandes est possible jusqu'à concurrence de la capacité du tampon associé à cette ligne de communication.

#### LE MODE OPERATION :

Ce mode permet l'utilisation du terminal dans la configuration "la console évoluée". Les commandes émises par le calculateur hôte sont réceptionnées, interprétées et exécutées au niveau de l'interface microprogrammée. Le pré-synthétiseur reconnaît les commandes destinées à HELIOS par le caractère <escape> qui précède chaque commande. Les informations circulant sur la ligne de communication et ne constituant pas des commandes sont repercutées vers la console-opérateur. Cette option permet à l'application de faire des entrées/sorties standard à partir de la console opérateur sans précaution particulière. La figure suivante récapitule les différents modes de fonctionnement.

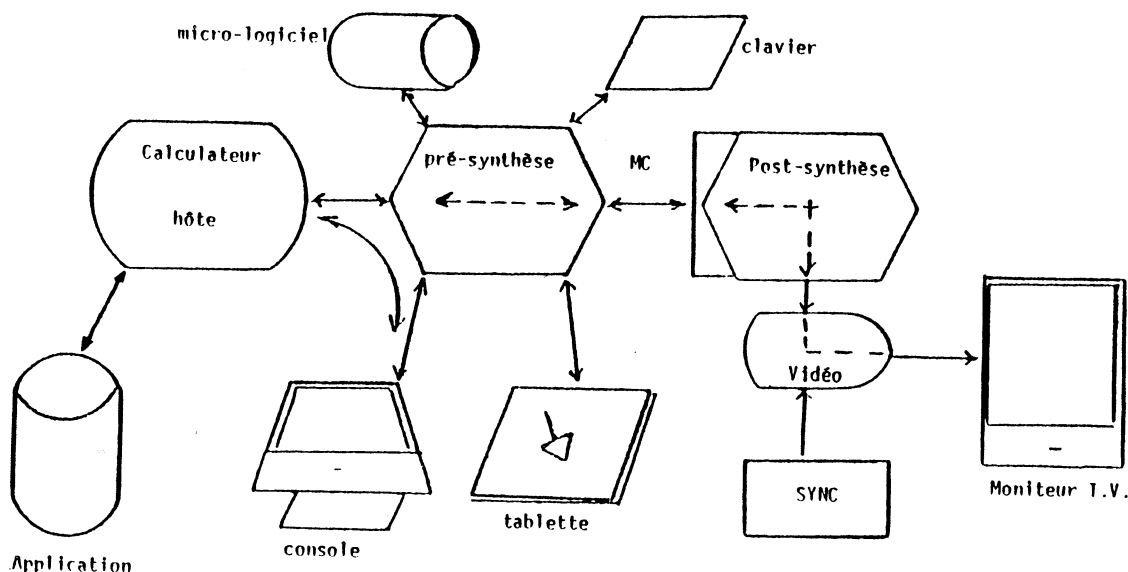


Figure No III.7

ALGORITHME :

```

-----
Debut
  si info sur PB alors
  faire
    si info = '<ecs>' alors action - commande fsi;
  finfaire;
  sinon info vers PA
  fsi;

  si info sur PA alors
  faire
    si info = '<local>' alors action - mode-local
    si info = '<set-up>' alors action - mode-set-up
    (* connexion calculateur hote : MULTICS *)
    si info = '<break>' alors action - break
    sinon info vers PB fsi;
  finfaire;
  fsi;
fin;
    
```

III.2.2 FORMAT ET CODAGE DES COMMANDES :

Comme l'indique la figure suivante, le format d'une commande comporte trois champs qui sont le champ caractere special ou debut de la commande, le caractere "code" ou code operation de la commande et une suite de caracteres constituant les parametres.

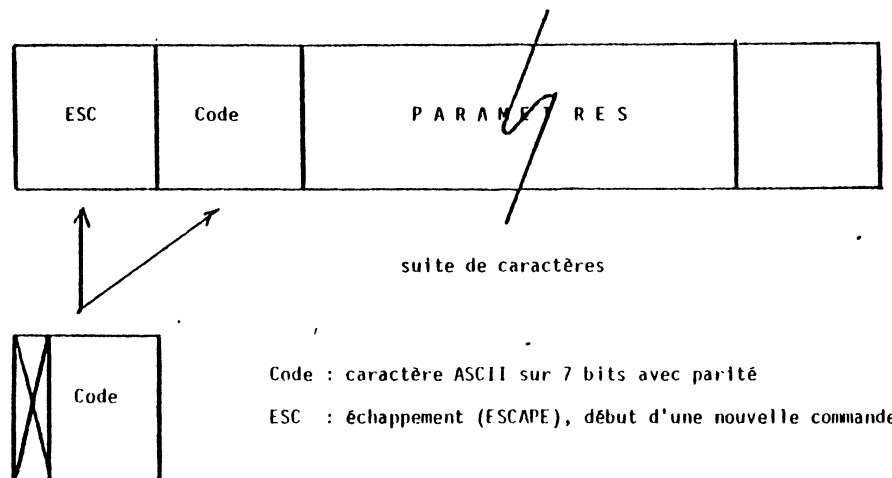


Figure No III.8

le calculateur hôte envoie un groupe de commande vers le terminal. Pour chaque commande reçue, le terminal peut éventuellement émettre une commande d'acquiescement, le format de l'acquiescement est :

- <ACK> : code ASCII d'acquiescement (04H),
- Des paramètres éventuels,
- <RC> : code ASCII du retour chariot (0DH).

FORMAT GENERAL D'UN ACQUITEMENT DE COMMANDE



ACK : caractère ASCII sur 7 bits avec parité d'acquiescement

RC : caractère ASCII sur 7 bits avec parité de retour-chariot

Figure No III.9

## REMARQUE :

Par défaut, le caractère de différenciation des informations circulant sur la ligne est <escape>. Cependant si le calculateur hôte utilise ce code, il peut être redéfini en activant l'action "SET-UP". Cette option permet une adaptation des codes des caractères de contrôle en fonction du calculateur hôte connecté au terminal graphique. (idem pour les caractères X-ON, X-OFF, Break, <ESC>, <ACK> et <RC>)

## III.2.2.1 GESTION DES PARAMETRES : -

Les commandes issues du calculateur hôte sont codées sous la forme de caractères ASCII. Ces paramètres constituent les différents attributs acceptés par la chaîne de post-synthèse. Les types de paramètres sont :

TYPE -----	NOMBRE DE CARACTERES -----	SYMBOLE -----
Entier court/caractere	1	< entier >
Entier long	1 / 2	<< entier >>
Numero de face	1 / 2	<< face >>
Couleur	1 / 2	<< couleur >>
Direction spatiale	2	< alpha > < beta >
Coordonnees X, Y	1, 2, 3 ou 4	(X, Y)

Généralement certains attributs sont associés à une ou plusieurs faces. Selon le code-opération de la commande, un nombre déterminé de paramètres est attendu.

L'entier court exprime une valeur positive sur 7 bits :

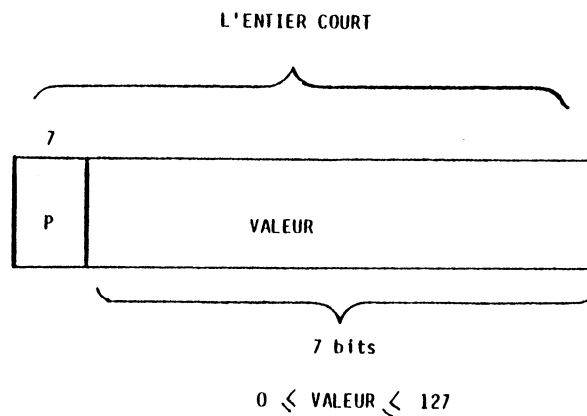


Figure No III.10

L'entier long exprime une valeur positive codée sur 12 bits :



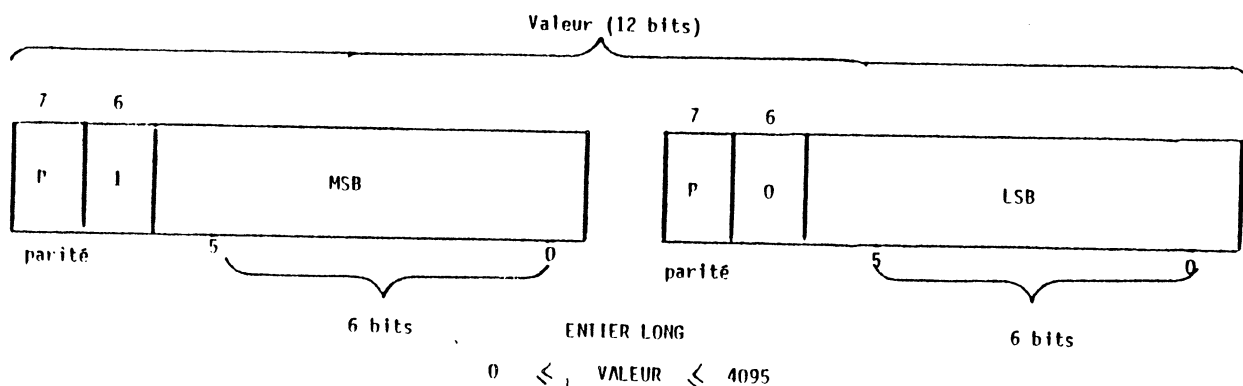


Figure No III.11

Le bit No 6 indique s'il s'agit des poids forts ou des poids faibles. Les poids faibles sont toujours transmis. L'absence des poids forts implique la conservation et la prise en compte des poids forts du dernier entier long transmis. Cette option vise à minimiser la transmission d'informations entre le calculateur hôte et le terminal.

Le numero de face exprime une valeur positive codee sur 10 bits :

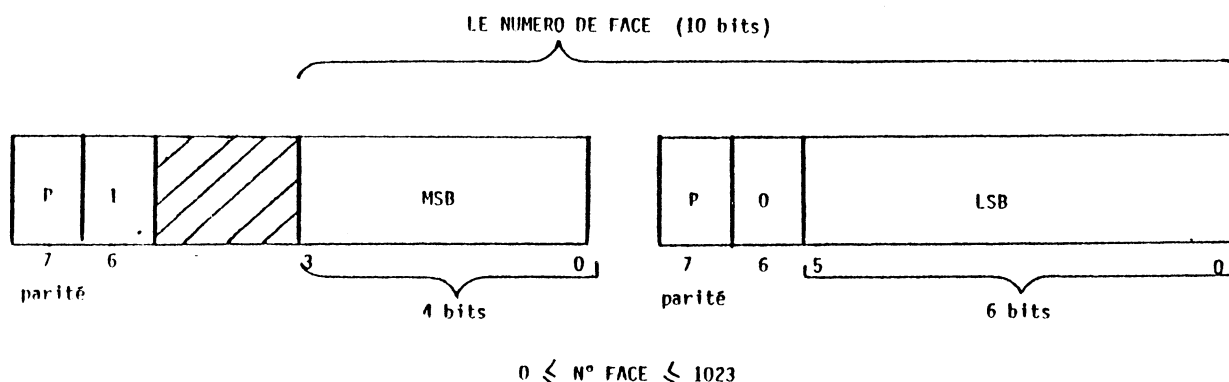


Figure No III.12

L'absence du premier caractere implique la reconduction des forts poids du dernier numero de face transmis. Le but de ce type de codage est d'economiser des echanges inutiles entre le calculateur hôte et le terminal. Cette remarque est aussi valable pour l'attribut suivant.

La couleur est une valeur positive exprimee sur 12 bits. Elle est constituee de trois composantes : R. V. B. Chaque composante est codee sur 4 bits.

LA COULEUR (12 bits)

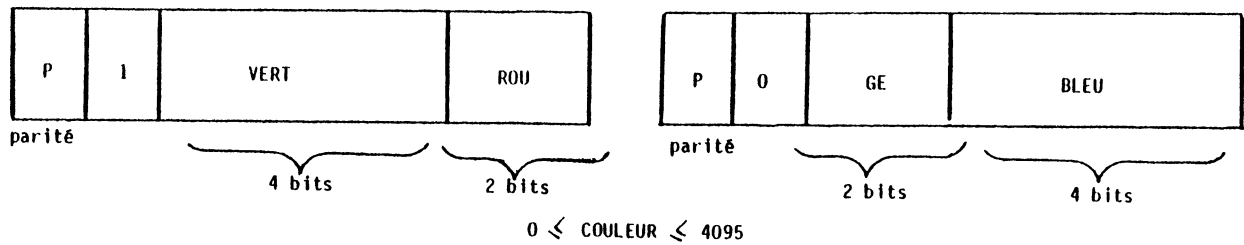


Figure No III.13

La direction spatiale s'exprime en fonction de deux angles alpha et beta. (cf CH II). Chaque angle est code sur 6 bits plus un bit de signe. La progression s'exprime en  $1/32$  ieme de  $\pi$ .

LA DIRECTION (12 bits)

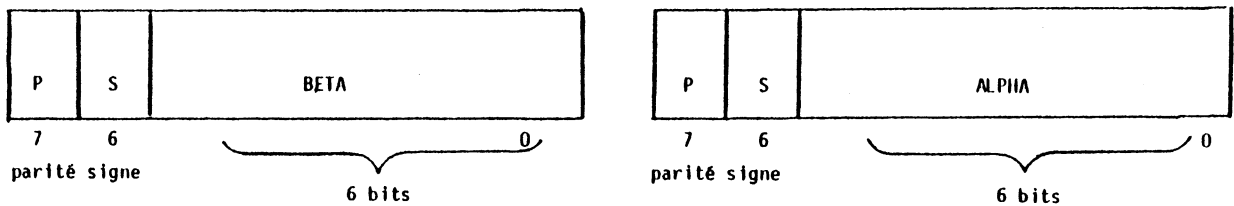


Figure No III.14

Dans ce cas les angles constituant la direction spatiale sont toujours transmis. L'angle alpha varie entre  $-\pi$  et  $+\pi$ , l'angle beta varie entre  $-\pi/2$  et  $+\pi/2$ .

Les coordonnees (X, Y) sont codees sur 10 bits chacune. Leur codage necessite la transmission de 4 caracteres :

LA COORDONNEE (X, Y) (2 X 10 bits)

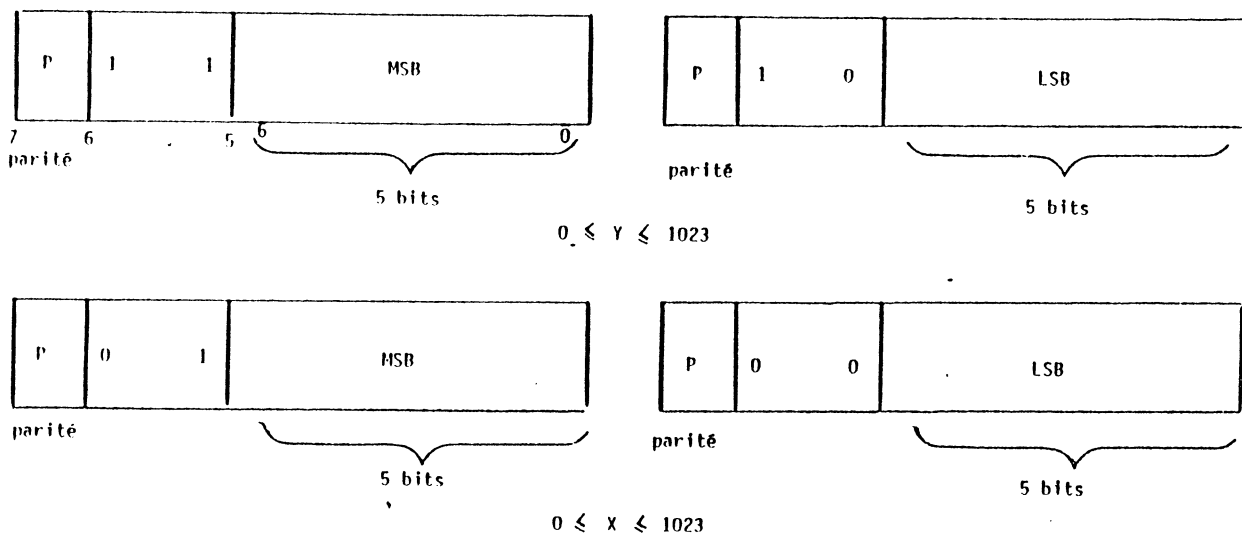


Figure No III.15

Dans le cas de commandes utilisant des coordonnees negatives la representation se fait modulo 1024. Les poids faibles de l'abscisse sont toujours transmis (X-LSB). L'absence des autres informations implique la reconduction des valeurs correspondant a la derniere coordonnee transmise. Dans ce cas l'optimisation est de l'ordre de 75 % etant donne la proximite des coordonnees d'une figure geometrique a visualiser. Ce point sera detaille dans le paragraphe suivant.

III.2.2.2 OPTIMISATION DE LA COMMUNICATION : -

Il faut rappeler que l'objectif d'une configuration "console evoluee" est d'ameliorer les performances globales du terminal graphique. Il s'agit de minimiser les echanges d'informations entre le calculateur hote et la pre-synthese d'une part, la post-synthese et la pre-synthese d'autre part.

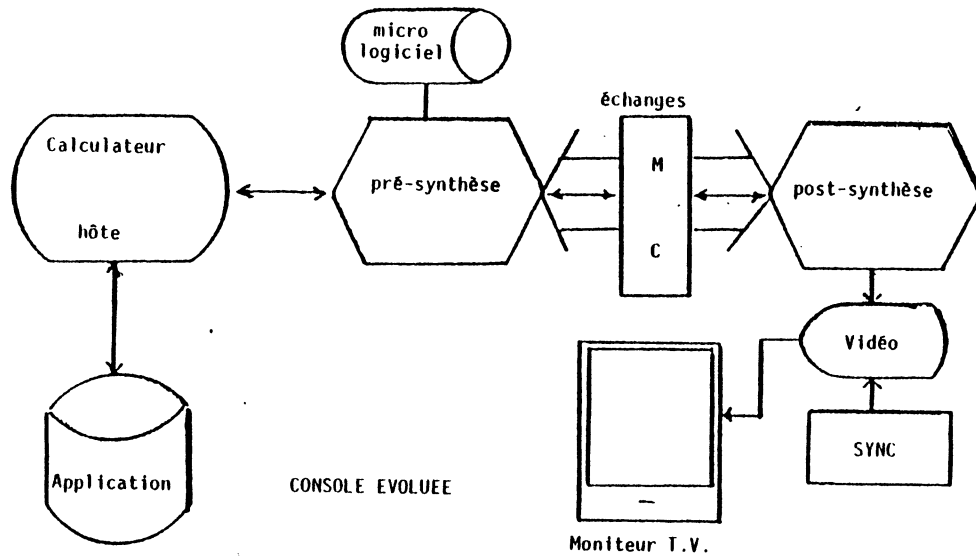


Figure No III.16

Le debit maximum de la ligne serie est de 9,6 K-bits / s. La gestion de la ligne associee a un tampon de memorisation intermediaire permet d'assurer un quasi-parallelisme entre la tache de recuperation des commandes et la tache d'execution. De plus le codage des differents attributs permet d'economiser des transmissions redondantes de l'information. Nous presentons dans ce paragraphe quelques exemples de commandes illustrant cette optimisation de la communication. Les parametres suivants ne sont retransmis que si l'une de leurs composantes respectives est a redefinir au niveau de la pre-synthese :

TYPE ----	CODAGE -----
L'entier long	2 caracteres ASCII,
Le numero de face	2 caracteres ASCII,
La couleur	2 caracteres ASCII,
Les coordonnees	4 caracteres ASCII.

La commande d'identification d'une face dont le format est indique par la suite constitue un exemple de gain en transmission entre le calculateur hote et la pre-synthese. En effet le numero de face dans un sous-ensemble de la scene est relativement invariant au niveau des poids forts d'ou l'inutilite de leur retransmission.

< esc > < 1 > << numface >>

< esc > < 2 > << numfacedeb >> << numfacefin >>

(Intervalle de faces)

Cette commande indique les faces concernees par les commandes graphiques suivantes concernant une affectation ou une consultation. Le temps de transmission de la commande sans optimisation est :

$$- T = ( T- < 1 > + T- < 2 > ) / 2 = 5,20 \text{ millisecondes.}$$

Le temps de transmission avec optimisation c'est-a-dire sans retransmission des poids forts est  $T\text{-optim} = 3,64$  millisecondes. Cette amelioration est obtenue de la meme maniere pour la transmission d'une couleur ou d'un entier long, Les commandes de visualisation d'entites graphiques de base necessitent le transfert de coordonnees. Par exemple la commande de generation de vecteurs a le format suivant :

< esc > < - > ( Xdeb, Ydeb ) ( Xfin, Yfin ).

Le temps de transmission sans optimisation est de 10,40 millisecondes. Le temps de transmission avec optimisation c'est-a-dire sans l'envoi systematique de toutes les coordonnees est  $T\text{-optim} = 4,16$  millisecondes. Soit une amelioration 60 % par rapport au temps de transmission sans optimisation. En effet la visualisation d'un trapeze ne necessite pas la retransmission des coordonnees completes des quatre sommets. Il suffit de transmettre les variations des faibles poids sur X ou sur Y par rapport aux coordonnees d'un sommet choisi et constituant une base pour les trois sommets restants. Il s'agit la d'un cas assez courant de proximite des coordonnees de plusieurs entites graphiques. La commande de generation d'un polygone a le format suivant :

< esc > < / > < nb-point > ( X, Y )... ( Xn, Yn ).

Considerons un polygone de quatre sommets. Le temps de transmission sans optimisation est de 19,80 millisecondes. De la meme maniere, le temps de transmission avec optimisation est :

$T\text{-optim} = 7,3$  millisecondes en temps absolu,

$T\text{-optim} = 13,54$  millisecondes en temps moyen.

On suppose que d'un sommet a un autre les faibles poids de Y et de X sont a retransmettre (la proximite des sommets, les uns par rapport aux autres ne necessite pas la transmission des forts poids). Dans ce cas le gain maximum est de 63 %, le gain moyen est de 31 %. L'optimisation des

echanges est a la charge de l'application qui ne doit transmettre des informations vers la pre-synthese que si elles sont a redefinir. Globalement le codage des caracteres constituant les attributs des differentes commandes, permet un gain moyen en transmission sur la ligne de l'ordre de 46 %, soit une economie de pres de la moitie d'un temps de transmission normal.

**REMARQUE :**

Un module d'acquisition des parametres accompagnant les commandes se charge de deux actions de conversion. La premiere correspondant a la conversion ASCII -> BINAIRE d'un parametre, la seconde correspondant a la conversion BINAIRE -> ASCII d'un parametre. La seconde tache concerne les commandes interactives du terminal ou la communication se fait de la pre-synthese vers la calculateur hote. Le protocole d'echanges des parametres est symetrique. L'optimisation a lieu dans les deux sens, seules les informations modifiees sont retransmises.

**III.3 CONTROLE DE LA POST-SYNTHESE :****LA PRE-SYNTHESE ET LA LOGIQUE CABLEE :****III.3.1 LE MODULE DE COMMUNICATION :**

L'interface reservee a l'accès au module de communication, permet d'effectuer des transferts de taille variable vers les differents registres. Cette possibilite decoule directement des caracteristiques du processeur utilise, c'est-a-dire le MC.6809. Le module de communication se presente comme une memoire locale de huit registres de 16 bits chacun soit 16 positions memoire consecutives.

Le mecanisme de decodage et de controle acquie les fins de transfert d'information en fonction de l'operation commandee au module de communication. L'architecture du module de communication est sensiblement identique a celle de la configuration "calculateur satellite". Cependant certaines fonctionnalites sont rajoutees ainsi que des registres supplementaires. L'objectif de ces ameliorations est encore une fois l'optimisation des echanges d'informations entre le pre-synthesiseur microprogramme et la post-synthese du terminal graphique HELIOS.

## III.3.1.1 DESCRIPTION DES REGISTRES : -

## REGISTRE MODULE :

( ecriture : 16 bits )

La fonction assuree par ce registre est sensiblement la meme que celle de la configuration decrite dans le chapitre II. Il indique le processeur de la post-synthese concerne par les commandes a venir, la memoire interne selectionnee au niveau du processeur et le sens de transfert (lecture/ecriture).

L.E. REGISTRE MODULE

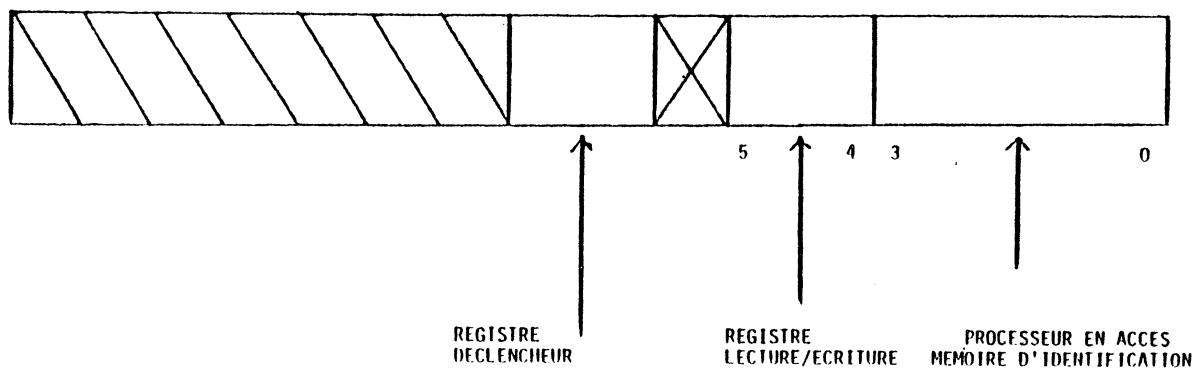


Figure No III.17

L'amelioration essentielle mise en oeuvre au niveau de ce registre est de pouvoir declencher la prise en compte et l'execution d'une operation de transfert sur un acces a un registre banalise. La premiere configuration necessitait un acces systematique au registre module afin de declencher l'operation de transfert. Le registre declencheur de l'operation de transfert peut a present etre :

- Le registre de donnees,
- Le registre d'adresse X, Y,
- Le registre repetition.

Lors du processus d'ecriture vers un processeur de la post-synthese, cette possibilite permet d'economiser systematiquement un acces en double largeur soit au minimum une microseconde, si un registre interne du MC.6809 est prealablement initialise.

## REGISTRE DE DONNEES :

( lecture / ecriture : 16 bits )

Ce registre permet un transfert bi-directionnel d'informations sur une largeur de 12 bits significatifs. Les informations peuvent etre :

TYPE D'INFORMATION -----	CODAGE -----
Un numero de face	( 10 bits ),
Une couleur	( 12 bits ),
Une abscisse X	( 10 bits ),
Une ordonnee Y	( 10 bits ),
Un coefficient de reflexion	( 8 bits ),
Un indicateur de visibilite	( 1 a 2 bits ),
Un facteur de grossissement	( 8 bits ).

Ces attributs de taille variable sont destines au processeur specifique de la post-synthese, selectionne par le registre module.

## REGISTRE DE REPETITION :

( ecriture 12 bits )

Ce registre permet d'effectuer des transferts vers la post-synthese avec une compression de donnees. Cette operation concerne les ecritures consecutives dans une memoire specifique :

- Affectation d'une couleur a N faces,
- Affectation d'une normale a N faces,
- Remplissage des plans d'identification, etc.

La taille de ce registre a ete augmentee (12 bits), le facteur de compression est quadruple, 4095 est la valeur maximale (256 dans la la configuration precedente)



## REGISTRE D'ADRESSE X, Y :

( ecriture : 2 X 16 bits )

Ce registre contient une adresse significative sur 20 bits au maximum. Cette adresse concerne une memoire d'un processeur de la chaine de post-synthese :

- La memoire est bi-dimensionnelle : X et Y sont significatifs soit 20 bits, (Exemple : un plan d'identification)
- La memoire est mono-dimensionnelle : X est significatif soit 10 bits, (Exemple : une table des faces)
- La memoire est un registre interne specifique : X et Y sont non significatifs. (Exemple : La lumiere ambiante)

## REGISTRE-ADRESSE (X, Y)

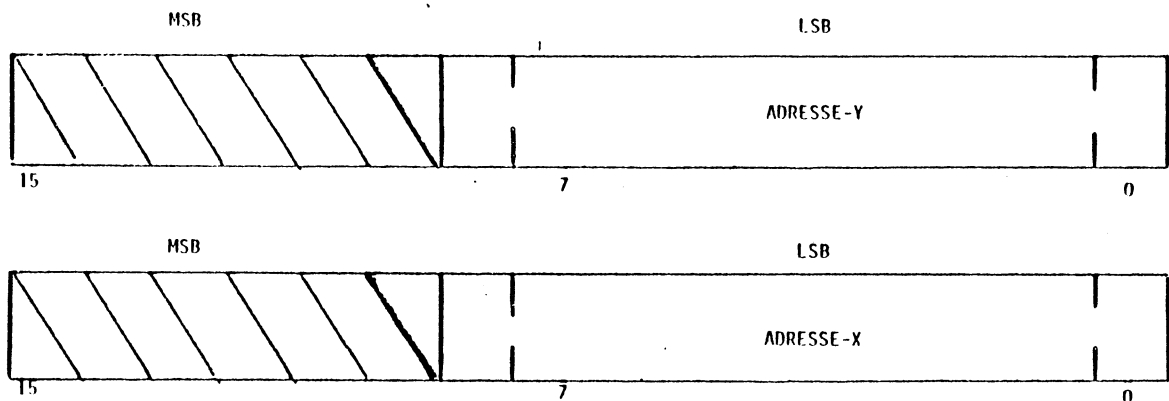


Figure No III.18

L'amélioration mise en oeuvre pour ce double-registre X, Y est l'automodification. (post-incrementation de X et/ou Y; et post-decrementation de X et/ou Y) Cette possibilite permet d'optimiser en particulier le remplissage des plans d'identification pour les algorithmes de generation de vecteurs, de cercles et de polygones et pour le remplissage de taches. Une evaluation des performances sera presentee dans le paragraphe III.4. L'objectif est d'economiser des acces au registre d'adresse X, Y qui est en double largeur. Son initialisation necessite :

- Pour une memoire bi-dimensionnelle : 10 cycles soit 5 microsecondes,
- Pour une memoire mono-dimensionnelle : 5 cycles soit 2,5 microsecondes.

On suppose dans cette evaluation que les registres internes du processeur MC.6809 sont prealablement initialises, dans le cas contraire il faut pratiquement doubler ces temps d'accès...

III.3.1.2 REGISTRES SUPPLEMENTAIRES : -

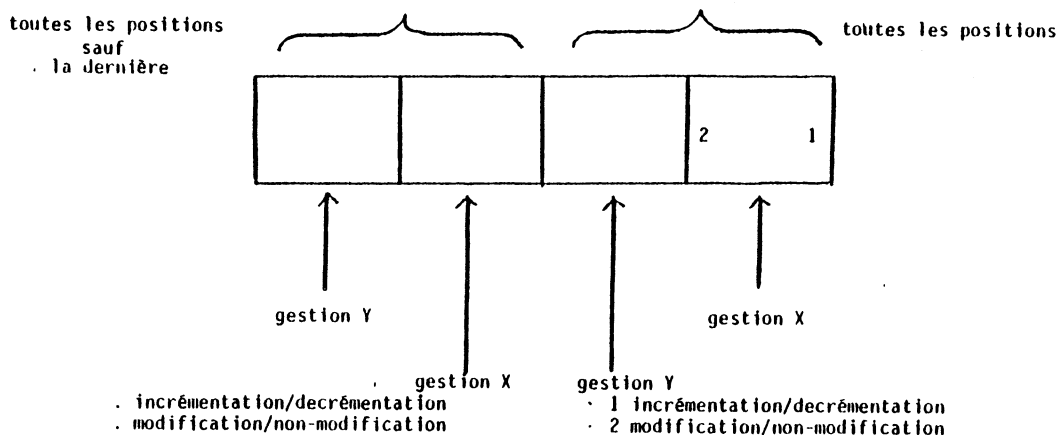
REGISTRE INCREMENTATION :

( ecriture 8 bits )

Ce registre controle l'operation d'auto-modification des registres d'adresse X, Y. Pour une adresse initialement memorisee dans le registre X, Y, il permet de valider la progression de l'abscisse ou de l'ordonnee et de selectionner le sens de la progression. (l'incrementation ou la decrementation.) Ces deux fonctions sont redefinies a chaque initialisation du registre incrementation. Cette initialisation se fait en 4 cycle memoire soit 2 microsecondes. La mise a jour du registre adresse X, Y peut se faire systematiquement :

- Apres chaque lecture / ecriture d'une position memoire,
- Apres chaque lecture / ecriture d'un groupe de positions memoire defini par le registre repetition.

Un codage par champs est mis en oeuvre :



LE REGISTRE INCREMENTATION

Figure No III.19

De plus la derniere position memoire fait l'objet d'un traitement particulier. (cas d'un groupe de positions) Le choix du registre de declenchement couple a l'utilisation des registres d'incrementation et de

repetition optimise considerablement le nombre d'accès necessaires pour executer les algorithmes cites precedemment.

### REGISTRE MASQUE :

( ecriture 16 bits )

Ce registre permet de definir des textures geometriques de traces. En fait le remplissage des plans d'identification est conditionne par la valeur binaire memorisee dans le registre masque. Cette facilite permet une generation cablee, au rythme de l'horloge du module de communication :

- De vecteurs a trait continu, pointille, mixte...
- De caracteres selon une police versatile,
- De textures geometriques pour le remplissage de taches.

Elle dispense donc le pre-synthesiseur microprogramme de developper des algorithmes complexes et couteux en temps. De plus le registre masque contient un facteur de grossissement de la texture memorisee (texture geometrique). L'ecriture de 4 points consecutifs dans la memoire de trame se fait toutes les 400 nanosecondes, le registre masque permet une ecriture conditionnelle dans la memoire de trame, au rythme video.

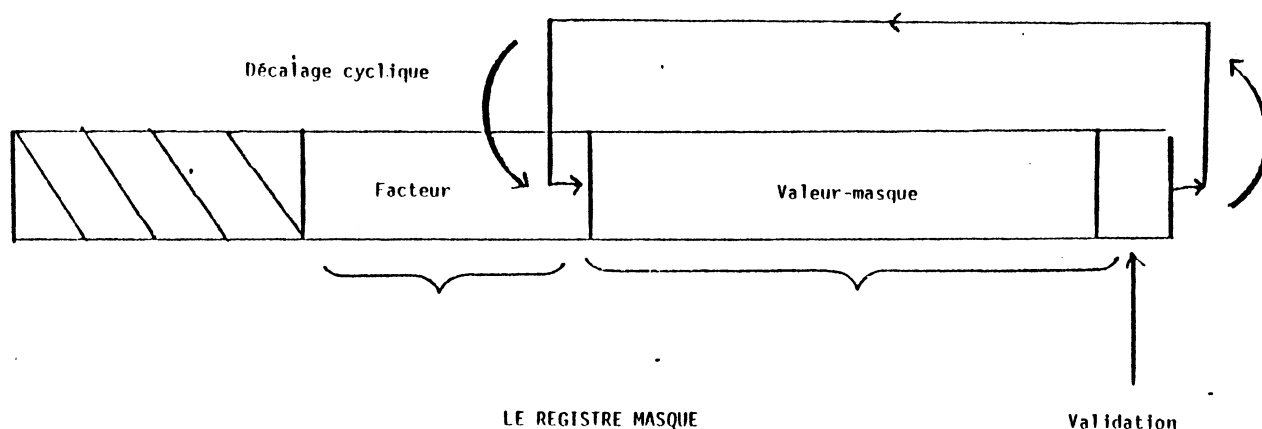


Figure No III.20

L'utilisation du registre-masque se deroule de la facon suivante. Pour chaque position candidate a l'ecriture, le bit '0' du masque valide l'ecriture s'il est egal a '1'. On effectue une permutation circulaire vers la droite pour la position suivante, si le facteur "fact" est nul la permutation est faite pour chaque ecriture, sinon la permutation est effectuee toutes les "fact + 1" positions.

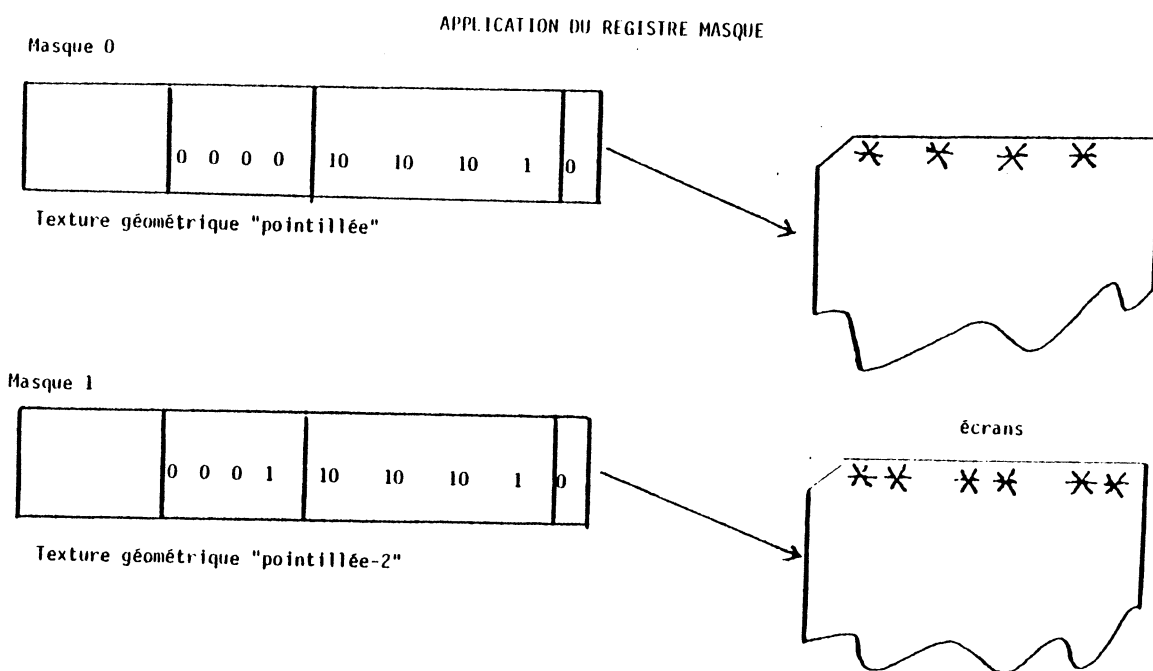


Figure No III.21

Comme l'indique la figure précédente, le rôle du facteur de grossissement est de définir des textures géométriques de taille variable et de pré-calculer des matrices de caractères de taille variable.

## LE REGISTRE-ETAT :

( lecture : 8 bits )

Le registre d'état de cette configuration constitue une concaténation d'informations utiles contenues initialement dans deux registres de la configuration précédente qui sont le registre-etat et le registre-mode. Ce dernier est supprimé dans la nouvelle architecture du module de communication. Le registre d'état peut être systématiquement testé avant d'initialiser un accès vers les processeurs de la post-synthèse. Il indique au presynthétiseur :

- La présence d'informations dans le registre de données, dans le cas d'une lecture,
- La disponibilité du registre de données, dans le cas d'une écriture,
- La disponibilité de l'ensemble des registres du module de communication pour le pré-synthétiseur, dans le cas de l'envoi d'une nouvelle commande.

De plus il fournit certains signaux de synchronisation video afin de choisir les instants ou les acces vers la post-synthese auront lieu. Deux possibilites sont offertes au niveau de la pre-synthese. La premiere permet d'effectuer des acces en permanence, dans ce cas les performances sont optimales. L'evaluation est presentee dans le paragraphe suivant. La seconde ne permet les acces que pendant les retours-trame ou ligne. (toutes les 64 microsecondes)

### III.3.2 EVALUATION DES ECHANGES :

Le processeur MC.6809 du pre-synthesiseur microprogramme effectue des acces directs aux differents registres du module de communication. L'adresse d'un registre est exprimee sur un octet, l'ensemble des registres est implante en page zero de l'espace memoire du processeur. La lecture d'un registre sur 8 bits se fait en quatre cycles processeur soit environ 2 microsecondes. La lecture d'un registre sur 16 bits se fait en cinq a six cycles processeur soit environ 2,5 a 3 microsecondes. Ces temps d'accès sont identiques pour une operation d'écriture.

Ainsi, le gain de la nouvelle configuration est appreciable, il n'est plus necessaire d'initialiser un port parallele emulant un bus d'accès pour la lecture ou l'écriture d'un registre du module de communication. Dans la configuration "calculateur satellite" l'accès a un registre implique :

- L'accès au port parallele : adresse, sens de transfert,
- L'accès au port parallele : validation-HELIOS,
- L'accès au port parallele : lecture/écriture d'une donnée (8 bits maximum).

Soit environ dix microsecondes pour une lecture/écriture sur une largeur de bus de données de 8 bits (13,3 microsecondes pour 16 bits). Etant donné les évaluations des échanges pour la nouvelle configuration, le gain en temps d'accès est de l'ordre de 80 %.

#### REMARQUE :

Une suite d'accès a l'ensemble des registres du module de communication se fait en 20 microsecondes environ. Cette evaluation suppose que les registres internes du processeur MC.6809 sont initialises aux valeurs adequates. Dans le cas contraire il faut pratiquement doubler ce temps.

### III.3.2.1 MECANISME DE LECTURE : -

La lecture d'une memoire de la chaine de post-synthese s'effectue selon le processus suivant :

- La scrutation du registre d'etat : test de disponibilite,
- L'initialisation du registre module (16 bits),
- L'initialisation du registre adresse X, Y (2 X 16 bits),
- La lecture du registre de donnees (16 bits).

Dans ce cas, le registre de donnees declenche l'operation de lecture, ceci est indique dans le registre-module. Pour un nombre de lecture successives, le registre incrementation est initialise, le registre de declenchement est le registre d'adresse. Le temps d'execution du processus de lecture est :

$$T\text{-lect} = T\text{-scrute} + T\text{-acces.}$$

T-scrute : temps de scrutation minimum : 4,5 microsecondes,

T-acces : moyenne pour une memoire : 10 microsecondes.

(la memoire peut etre mono ou bi-dimensionnelle)

$$T\text{-lect} = 15 \text{ microsecondes.}$$

Le gain obtenu par rapport a la configuration "satellite" est de l'ordre de 54 %. Une recapitulation de ces evaluations indique que les delais d'echange entre le calculateur hote et la pre-synthese ont ete reduits de moitie. Il en est de meme des delais d'echange entre la pre-synthese et le module de communication.

### III.3.2.2 MECANISME D'ECRITURE : -

Les ecritures vers une memoire de la post-synthese s'effectuent selon le processus suivant :

- La scrutation du registre d'etat,
- L'initialisation du registre module (16 bits),
- L'initialisation du registre d'adresse X, Y (2 X 16 bits),
- L'ecriture dans le registre de donnees (16 bits),

- L'initialisation du registre de repetition (16 bits).

Dans ce cas le registre de declenchement est le registre de repetition. On suppose que les registres "incrementation" et "masque" sont prealablement initialises. Dans le cas contraire huit cycles supplementaires sont necessaires soit 4 microsecondes. La compression de donnees peut atteindre un facteur de 4095 ecritures successives au rythme d'une ecriture toutes les 100 nanosecondes. Le temps d'execution du processus d'ecriture est :

$$T\text{-ecr} = T\text{-scrute} + T\text{-acces} : \text{soit } 16 \text{ microsecondes.}$$

(moyenne pour une memoire bi ou mono-dimensionnelle.

On note que le temps d'accès a la memoire de trame se reduit de 20 microsecondes par rapport a celui de la configuration precedente... Le gain est de l'ordre de 44 %. Cette evaluation permet de mettre en evidence une amelioration des echanges au niveau de la pre-synthese de l'ordre de 62 %. (Pour la lecture et l'ecriture) Cependant il faut aussi tenir compte des fonctionnalites apportees par les registres supplementaires du module de communication.

#### III.4 ETUDE DES PERFORMANCES DU T.G.I :

##### III.4.1 REPERTOIRE DE COMMANDES :

Ce repertoire decoule de la definition et de la realisation d'un jeu de primitives au niveau du pre-synthesiseur programme mise en oeuvre dans la configuration "calculateur satellite". Dans une premiere phase, il est necessaire de pouvoir configurer le terminal graphique. Il s'agit de choisir :

- Le mode de fonctionnement : operation / local / emulation,
- Le mode de communication : caracteristiques des lignes,
- Les droits d'accès aux memoires de trames.

Une premiere categorie de commande permet de definir des attributs statiques ou permanents. Ces attributs restent valides jusqu'a la prochaine definition communiquee au terminal. Il s'agit des attributs suivants :

- La face courante ou la sequence courante (intervalle de faces),
- Le plan courant,

- L'aspect graphique :
  - . La Police de caracteres et ses caracteristiques,
  - . La texture de trace des vecteurs
  - . La texture de remplissage des taches.

Une deuxieme categorie permet de definir des attributs dynamiques qui sont pris en compte en temps reel par le terminal. Cette attribution porte sur la ou les faces courantes dans le plan courant selectionne. Ces attributs sont :

- La couleur d'une texture : 4096 couleurs possibles,
- Le modele de reflexion : brillant / satine / mat / neutre,
- La visibilite : visible / invisible / clignotant,
- La normale N : (Par rapport au repere de l'ecran),
- Le repere de la face : (U, V), par rapport au repere ecran.

Une troisieme categorie permet de definir dynamiquement les attributs concernant la source lumineuse et le reticule. Cette attribution est prise en compte en temps-reel. Pour la source lumineuse, il s'agit de La couleur et de la direction. Pour le reticule, il s'agit de la couleur et de la position (X, Y).

Une quatrieme categorie permet la definition dynamique de l'espace-ecran en temps-reel, en particulier l'origine d'affichage des plans, l'origine relative du repere associe au plan et un "zoom" applique au plan. (16 niveaux possibles)

L'avant-derniere categorie concerne essentiellement les possibilites d'interaction du terminal graphique. Ces commandes permettent l'echange d'informations entre le calculateur hote et la post-synthese via le pre-synthetiseur microprogramme. Ces commandes concernent essentiellement la collectes de coordonnees, La designation de faces et la recuperation de numeros de fonctions a partir du pupitre d'interaction. De plus l'utilisation du dialogue implicite, permet la consultation d'attributs de faces. Des commandes speciales mettent en oeuvre des moyens de synchronisation externe. (macro-fonctions synchrones du signal video ou d'un dispositif externe).

La derniere categorie concerne les fonctions d'affichage graphique ou alphanumerique. La visualisation se fait dans le plan courant. Un numero de face unique est attribue a ce plan. Ces commandes permettent d'utiliser le terminal HELIOS comme chaine de synthese d'entites graphiques de base



dans le plan. Ces commandes sont l'affichage d'un point, la generation d'un segment de droite, la generation d'un polygone convexe avec remplissage eventuel, la generation de cercles avec remplissage eventuel, l'affichage d'un texte alphanumerique et eventuellement l'effacement total de l'ecran.

Cette classification demontre que toutes les fonctionnalites de la chaine de post-synthese sont accessibles. De plus le pre-synthetiseur microprogramme offre un ensemble de commandes concernant le mode d'affichage, il s'agit de la derniere categorie. Le processus de visualisation necessite l'execution d'algorithmes specifiques au niveau du pre-synthetiseur microprogramme, en utilisant le parallelisme possible entre la pre-synthese et la post-synthese. Les ameliorations apportees au niveau du module de communication vont permettre d'optimiser les echanges entre la logique cablee et le pre-synthetiseur microprogramme. Ces ameliorations sont :

- Le choix du registre de declenchement,
- L'utilisation du registre-incrementation,
- L'utilisation du registre-masque.

Le cycles d'accès a la memoire de trames est passe de 120 nanosecondes a 100 nanosecondes par pixel. Enfin l'optimisation de la communication entre le calculateur hote et le terminal graphique implique une progression globale des performances du systeme. Certaines commandes ont fait l'objet d'une evaluation, elle est presentee dans les paragraphes suivants. ( MAR 82 )

#### III.4.1.1 GENERATION DE VECTEURS : -

La syntaxe de la commande :

< esc > < - > ( xdeb, ydeb ) ( xfin, yfin )

La semantique de la commande : les parametres sont les deux extremités du segment de droite a visualiser dans le plan courant.

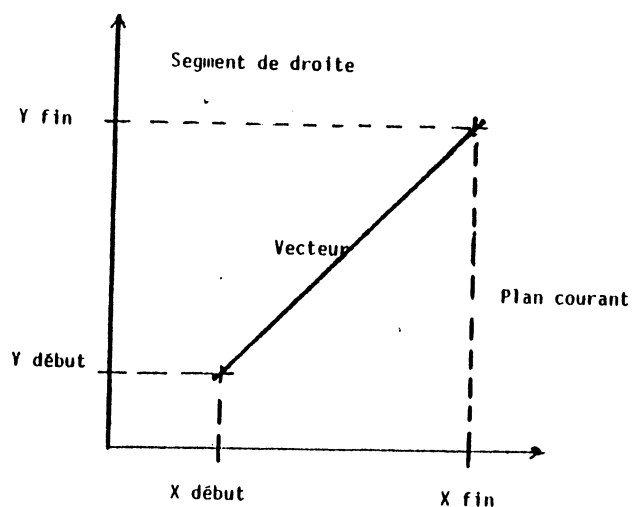


Figure No III.22

L'algorithme de generation de vecteur est constitue de deux parties. Une premiere partie microprogrammee est executee par le processeur MC.6809. Elle determine la progression en X ou en Y en fonction des valeurs des coordonnees des deux extremités du vecteur. Quatre progressions sont possibles. Pour chaque pas de l'algorithme, elle determine :

- La portion de segment horizontal,
- Le nombre de points a afficher,
- La progression pour le pas suivant. (X ou Y)

la seconde partie cablee est sequencee par le module de communication. Cet algorithme utilise les fonctionalites des registres incrementation et repetition, ce dernier etant le registre de declenchement de l'operation. Le registre-masque contient la texture de trace.

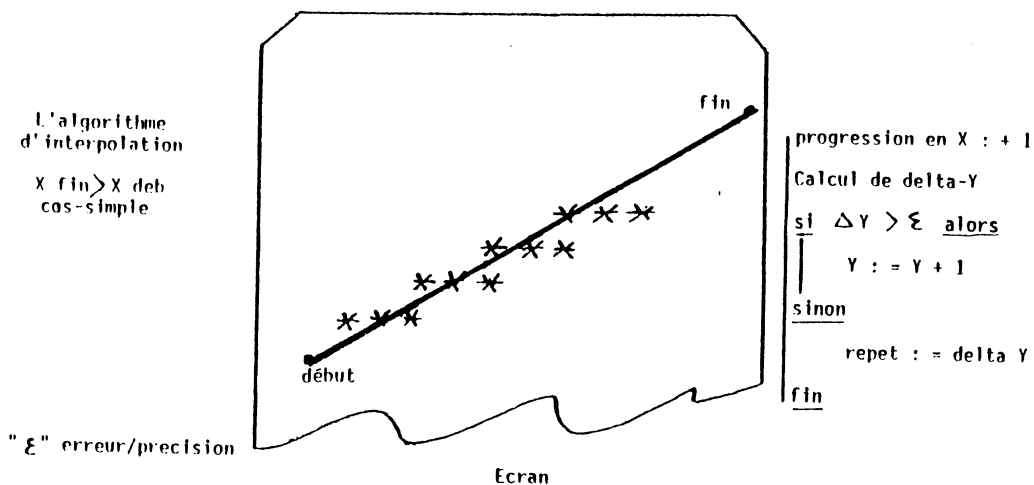


Figure No III.23

( BRE 65 )( CON 82 )

ALGORITHMME SIMPLIFIE:

```

-----
vecteur(xdeb, ydeb, xfin, yfin)
(* le vecteur horizontal est teste,
   c'est un cas particulier *)
debut
(* le pas peut etre en x ou en y *)
determiner(dx, dy, pasxy, err)
xdroit=xdeb
pour j=xydeb jusqu'a xyfin
faire
calculer(xgauche)
test-cloture-ecran (* decoupage aux frontieres de l'ecran *)
(* postsynthese *)
afficher-segment-horizontal, xdroit, nb-point) (* repetition *)
calculer(progression-xy, err) (* incrementation *)
maj(xdroit)
finfaire
fin
    
```

L'evaluation du temps d'execution du sous-programme de generation de vecteurs a ete faite dans trois configurations. Les deux cas extremes sont le vecteur horizontal et le vecteur vertical. Le cas intermediaire est le vecteur diagonal avec differentes projections en X et en Y. Le dimensionnement de la taille du vecteur permet de tester les vitesses de trace sur toute la largeur de l'ecran. (les valeurs de projection pour les

trois cas sont 1/8, 1/4, 3/8, 1/2, 5/8, 3/4, 7/8 et 1 de la largeur de l'ecran) Les resultats obtenus sont les suivants :

- T-max = 160 microsecondes,
- T-min = 110 microsecondes,
- T-moy = 135 microsecondes.

Pour le cas d'un vecteur horizontal la vitesse maximum est de l'ordre de 3,8 Millions de points par seconde. Pour le cas d'un vecteur vertical la vitesse minimum est de l'ordre de 80 000 points par seconde. Dans le cas le plus favorable la vitesse de generation est de l'ordre de 4,7 millions de points par seconde.

La configuration "calculateur satellite" possede une vitesse moyenne de 1,2 millions de points par seconde. Soit une amelioration de l'ordre de 75 % pour la configuration "console evoluee". Globalement et au vue des ameliorations obtenues aux differents niveaux de la chaine de synthese, la performance globale est quadruplee par rapport a la configuration "calculateur satellite".

#### III.4.1.2 GENERATION DE POLYGONES : -

La syntaxe de la commande :

< esc > < = > ( xig, yig ) ( xsd, ysd )

Ce premier cas correspond a la visualisation d'un rectangle. Le remplissage interieur est possible. Cette option est precisee dans la commande d'aspect graphique.

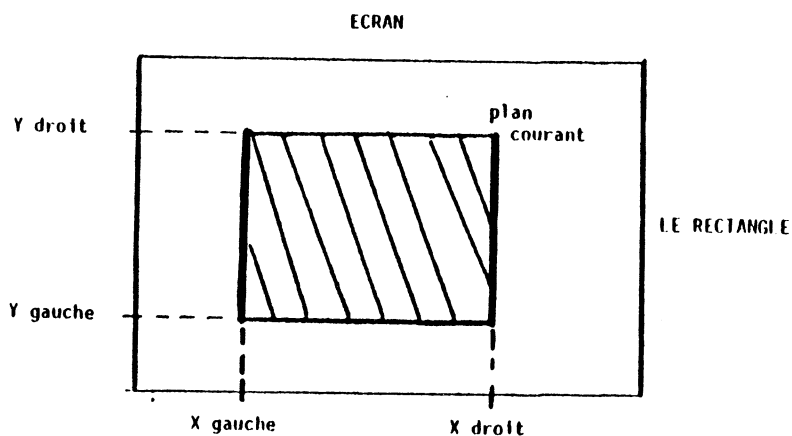
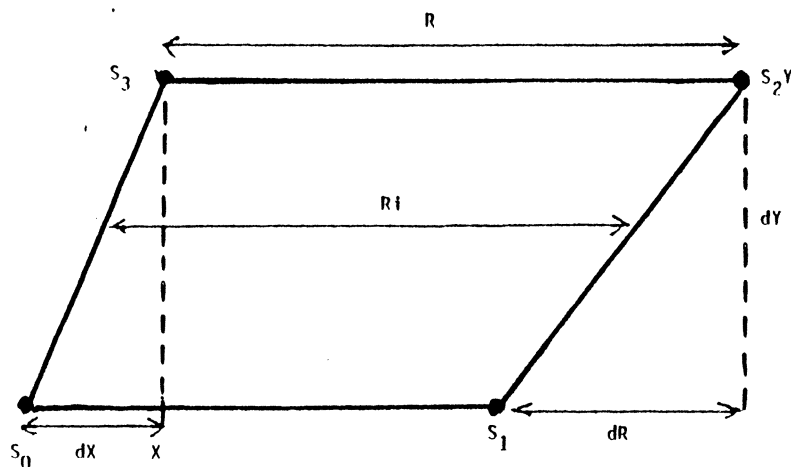


Figure No III.24

Le second cas correspond a la visualisation d'un polygone quelconque, l'ensemble des sommets est enumere. Le remplissage est indique par l'aspect graphique choisi. La syntaxe de la commande :

< esc > < / > < nb-point > ( x1, y1 )... ( xn, yn )

L'algorithme de remplissage mis en oeuvre considere uniquement le cas de trapezes reguliers dont la base est horizontale et definis par quatre sommets :



LA REGION ELEMENTAIRE : TRAPEZE

Figure No III.25

Une partie microprogrammee se charge de determiner la ligne  $i$ , l'abscisse gauche  $x_i$  et le nombre de points  $i$  a visualiser. Une partie cablee se charge du remplissage effectif des memoires de trames, via le module de communication. L'option de compression de donnees est utilisee pour l'ecriture d'une ligne constituee de " $i$ " points.

## ALGORITHME :

```

-----
Debut (*cas ou dx >=0 *)
  ix <- 0
  si dy <> 0 alors
  tant que dx >= dy faire
  debut
    dx <- dx - dy
    ix <- ix + 1
  fin
  registre -adresse <- (y, x) (* premiere ligne *)
  h <- dy div 2
  pour i := y+1 jusqu'a y+dy faire
  (* incrementation automatique *)
  debut
    h <- h + dx
    x <- x + ix
    si h >= dy alors
    debut
      h <- h - dy
      x <- x + 1
    fin
  registre-adresse <- (i,x) (* lignes suivantes *)
fin
fin

```

Dans le cas ou dx est negatif il suffit de changer les bits du registre "incrementation" de maniere a ce qu'il effectue une decrementation. Le declenchement se fait par l'intermediaire du registre de repetition. Pour chaque iteration le registre adresse X est initialise. Les progressions en X,Y sont controlees par le registre-incrementation la texture de remplissage est indiquee au niveau du registre masque, ceci permet un remplissage ajoure du trapeze. Le rectangle est un cas particulier de polygone. Au niveau de l'algorithme, la visualisation du contour ainsi que le remplissage sont plus simples a realiser. L'evaluation des microprogrammes met en evidence trois formules de calcul du temps d'execution :

$$T\text{-polygone} = 82 \text{ cycles} + T\text{-vect} * (\text{nbr-aretes})$$

$$T\text{-rectangle} = 83 \text{ cycles} + T\text{-polygone}$$

$$T\text{-remplissage} = 135 \text{ cycles} + ( 21 \text{ cycles} (\text{nbr-lignes}) )$$

Nbr-aretes : nombre d'aretes constituant le polygone.

Nbr-ligne : nombre de lignes necessaires au remplissage (dy)

T-vect : temps de trace d'un vecteur

(valeur moyenne : 135 microsecondes).

L'etude concerne dans une premiere etape le cas du rectangle soit quatre aretes. Les evaluations sont les suivantes :

T-polygone = 580 microsecondes,

T-rectangle = 622,5 microsecondes (moyenne),

Cette performance permet un debit de 1,6 K-rectangle par seconde. Pour le remplissage les cas extremes sont : le nombre d'aretes maximum soit 512 lignes et le nombre d'arete minimum soit une ligne. L'evaluation obtenue est T-rectangle = 190,5 microsecondes. (moyenne des taches constituees de 1 a 64 lignes), Cette performance permet un debit de 5,25 K-taches par secondes. Il faut aussi tenir compte de deux parametres dans ce cas de remplissage de taches rectangulaires. Il s'agit de la longueur et de la largeur. Si la taille des lignes horizontales est importante (longueur), le processeur d'interpolation sera plus rapide que la logique cablee de visualisation. Dans le cas contraire (taille des lignes horizontale faible par rapport aux lignes verticales, c'est la largeur) le processeur charge de l'interpolation ralentit la logique cablee de remplissage de la memoire de trame qui fonctionne au rythme video. Une rupture dans le temps de reponse a lieu pour des rectangles dont la longueur (ligne horizontale) est superieur a 100 pixels. (ceci correspond au delai d'interpolation necessaire au MC.6809)

La seconde etape de l'evaluation concerne les remplissages de regions elementaires tels que le trapeze ou le triangle. La decomposition en modules des microprogrammes permet d'obtenir les performances suivantes : Le contour du trapeze ou du triangle est visualise par l'intermediaire du microprogramme "polygone", soit un temps moyen de 513 microsecondes. Le remplissage effectif du trapeze se fait en :

T-remplissage = 118 microsecondes + (71 microsecondes \* nbr-aretes)

Soit une moyenne de vitesse de remplissage de 383 microsecondes a 1,5 millisecondes. Cette performance permet un debit environ de 2 K-faces par seconde. (un trapeze avec remplissage : T-execution = 1,9 milliseconde) Ce debit ne tient pas compte des delais de communication. Le gain par rapport a la configuration "calculateur satellite" est de l'ordre de 100 %. (460 millisecondes pour visualiser une tache contre 2 millisecondes pour la configuration "console evoluee").

#### III.4.1.3 GENERATION DE CARACTERES : -

La syntaxe de la commande :

< esc > < ' ' > ( xdeb, ydeb ) < nbcар > < car1 > ... < car-n >

La semantique de la commande :

- ( xdeb, ydeb ) represente le point de l'ecran a partir duquel on effectue l'affichage,
- < nbcар > represente le nombre de caracteres a afficher,
- < car-i > represente le code ASCII d'un caractere.

L'affichage des caracteres se fait en fonction des differents attributs qui leurs sont associes soit la taille, l'orientation et la police. (implicite ou definie par l'utilisateur). L'algorithme de generation de caracteres tient compte de ces trois attributs. Le registre masque permet de definir chaque ligne de la matrice constituant le codage du caractere. Le registre incrementation permet la generation de caracteres dans toutes les directions. Le registre repetition declenche l'operation de remplissage des memoires de trames, la progression est automatique. ( CON 82 ) L'evaluation concerne la generation d'un caractere de taille standard. Le temps total de l'execution de la commande depend de la taille des caracteres et du nombre de caracteres a visualiser. Le temps d'execution total de la commande s'exprime par :

$$T\text{-car} = 84 \text{ cycles} + \text{nbcар} * (432 \text{ cycles} + \text{nbr-colonnes} * 52 \text{ cycles}).$$

"nbcар" represente le nombre de caracteres a afficher, "nbr-colonnes" represente le nombre de colonnes de la matrice d'un caractere. Le temps d'affichage d'un caractere pour une matrice standard est de l'ordre de 388 microsecondes (delais de communications non-evalues) (nbcар = 1, nbr-colonnes = 5) Le gain par rapport a la configuration "calculateur satellite" est peu important. Cependant certaines fonctionalites supplementaires compliquent l'algorithme et sont penalisantes en temps de calcul. (gestion de la taille et de l'orientation des caracteres)

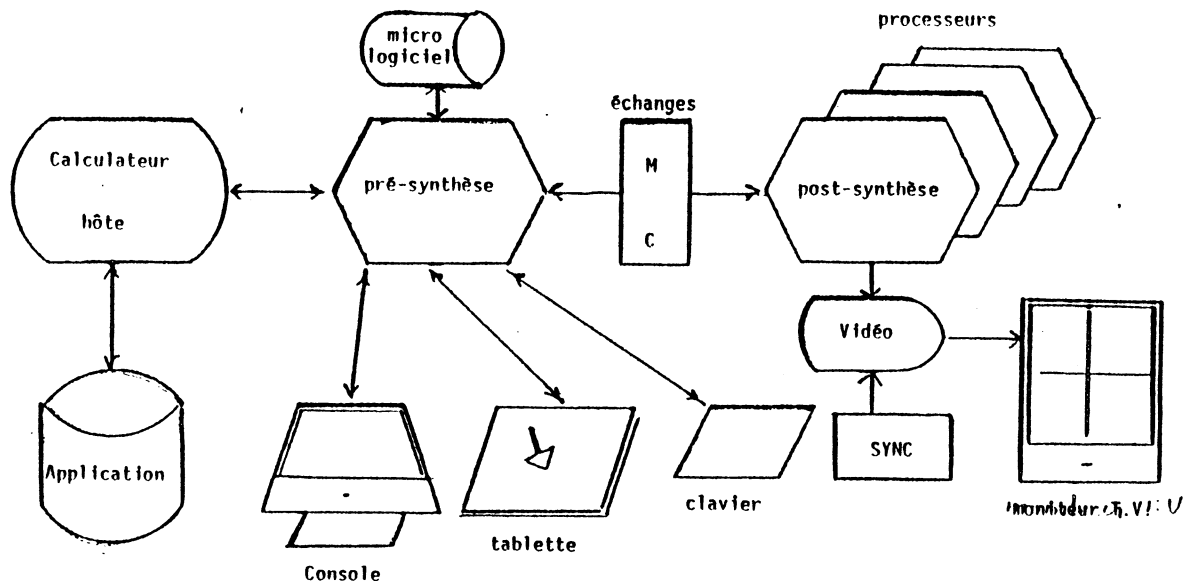


## III.5 TYPES D'INTERACTIVITE DU T.G.I :

Dans la configuration "console evoluee", le processeur MC.6809 gere un ensemble de dispositifs d'interaction :

- Le pupitre a 16 touches,
- La console operateur,
- La tablette a numeriser.

Cette description succincte des fonctions interactives ne fait pas l'objet d'une evaluation. La configuration complete est representee dans la figure suivante.



LA CONSOLE EVOLUEE

Figure No III.26

Deux types d'interaction sont possibles. La consultation ou recuperation des informations introduites par l'application au niveau du terminal, la collecte ou recuperation des informations issues du terminal. L'interaction se fait principalement par le pupitre associe a un reticule visible sur l'ecran.

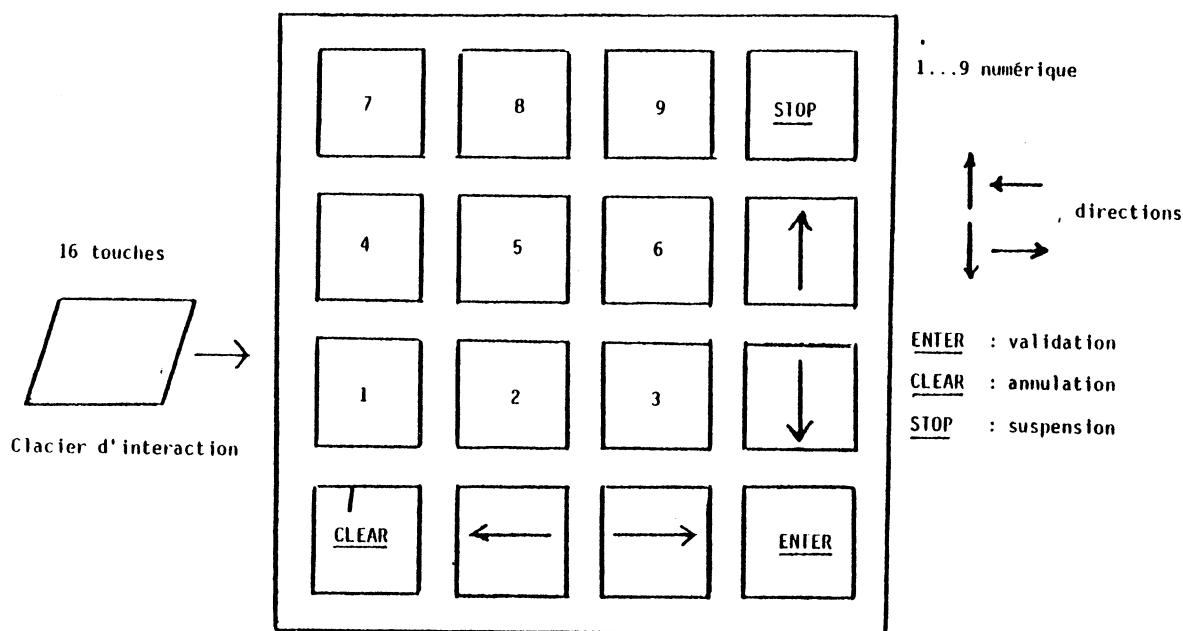


Figure No III.27

III.5.1 DIALOGUE ET CONSULTATION :

Une premiere categorie de ces commandes concernent essentiellement l'operation de collecte de coordonnees : Collecte de coordonnees (X , Y) Cette commande issue du calculateur hote, active la scrutation du pupitre d'interaction implicitement. Une coordonnee peut etre collectee a l'aide d'un des trois dispositifs prevus a cet effet. Dans le cas ou le reticule est associe a l'interaction, les coordonnees sont evaluees :

- Par rapport a l'origine du plan courant,
- Par rapport au repere absolu de l'espace-ecran.

A la fin de la collecte, HELIOS transmet au calculateur hote un acquitement de la commande et le resultat de la collecte. La transmission des coordonnees se fait selon le codage decrit precedemment.

Enfin la commande d'identification de faces permet d'effectuer l'identification d'une face visible sur l'ecran, l'ensemble des dispositifs d'interaction peut etre utilise. Les faces invisibles ne peuvent etre identifiees. A la fin de l'identification, HELIOS transmet vers le calculateur hote une commande d'acquiescement et le resultat de l'identification d'une sequence de faces. Les performances de ces commandes sont entierement determinees par les vitesses de communication des differentes lignes. (seul le pupitre d'interaction est dote d'une interface parallele.)

Une seconde categorie concerne l'ensemble des commandes permettant a l'application de consulter les attributs associes a une face, au plan d'identification, a la source lumineuse et au reticule. Ces commandes constituent des primitives d'accès aux memoires internes des differents processeurs de la post-synthese. La consultation se fait en temps-reel, HELIOS transmet vers le calculateur hote une commande d'acquiescement et l'attribut consulte selon le codage correspondant. Les differentes consultations possibles sont decrites dans la suite du paragraphe.

- 1 Consultation des attributs des faces courantes :
- 2 Consultation des attributs de la source lumineuse :
- 3 Consultation des attributs du plan courant :
- 4 Consultation d'attributs generaux :
- 5 Consultation de l'etat du terminal graphique.
- 6 Consultation d'un point du plan d'identification.

REMARQUE :

Cette seconde categorie de commandes interactives regroupe les fonctions reciproques des commandes d'attribution du terminal. Les performances sont determinees par les delais de communication sur la ligne. Les criteres de symetrie et de coherence de la chaine de synthese sont valides au niveau du repertoire de commandes. Cette caracteristique est inherente a l'architecture de chaque chaine (pre et post-synthese).

### III.5.2 INTERACTIONS SPECIALES :

Cet ensemble de commandes permet d'augmenter les possibilites locales offertes par le terminal graphique et de personnaliser le contexte

d'utilisation afin de répondre aux besoins spécifiques d'une application. Ces commandes constituent une nouvelle fonctionnalité mise en œuvre pour la configuration "console évoluée".

#### 1 Les macro-définitions de fonctions :

Cette commande permet de définir une suite de commandes élémentaires exécutables séquentiellement. Elles peuvent être conditionnées par un événement extérieur. L'événement extérieur peut être l'enfoncement d'une touche particulière du pupitre, l'initialisation du terminal ou encore, un ou plusieurs retours de trame.

#### 2 Contrôle et dialogue implicite :

Cette commande permet de définir une suite de contexte d'utilisation des différents dispositifs d'interaction et de la console-opérateur. De plus, elle permet d'activer un processus d'utilisation locale du terminal à partir des périphériques suivants : la console-opérateur, Le pupitre d'interaction ou la tablette à numériser. L'utilisateur peut inter-agir sur l'image en temps-réel, en effectuant les animations suivantes :

1. Le déplacement et la modification de la source lumineuse,
2. La modification d'une couleur ou d'une texture,
3. La modification de l'attribut visible ou invisible,
4. La modification de la réflexion,
5. La translation ou le "zoom",
6. La génération d'objets graphiques de base,

#### 4 Sauvegarde du contexte HELIOS :

De plus l'utilisateur peut sauvegarder le contexte relatif au terminal. Cette commande permet de sauvegarder l'ensemble des mémoires de la post-synthèse sur un support quelconque :

1. Une mémoire auxiliaire du calculateur hôte,
2. Une unité de disquettes locales gérées par la pré-synthèse,
3. Une imprimante graphique. (copie d'écran ou "hardcopy" d'imprimante)

Les memoires pouvant etre sauvegardees ou restaurees sont celles des differents processeurs de la post-synthese. Il s'agit des plans courants, des tables des couleurs, des normales, de visibilite et des banques des textures et de reflexions. La police de caracteres definie par l'utilisateur peut etre aussi sauvegardee et restauree.

5 Commande de dispositif externe :

Cette commande fournit en sortie une impulsion "TTL" d'une duree programmable multiple du 1/1000 de seconde. La duree maximale de l'impulsion est  $T_{max} = 4,095$  secondes. Aucune commande issue du calculateur hote ne peut etre prise en compte pendant cet intervalle de temps. Le dispositif externe peut etre eventuellement une camera de saisie d'image par image pour la constitution d'un film a l'aide du terminal HELIOS.

## CONCLUSION :

La configuration "console evoluee" a mis en evidence le role primordial de la pre-synthese de base. L'adjonction d'une interface microprogramme a base de MC.6809 a permis :

- La normalisation de l'acces au terminal graphique interactif HELIOS par l'intermediaire d'un jeu de commandes evolue et par la mise en oeuvre d'une interface standard,
- L'adaptation et l'amelioration de la communication entre la pre-synthese et la post-synthese. (mise en oeuvre des registres masque, incrementation et repetition)
- L'implantation d'une "intelligence" residente au niveau du terminal. (le logiciel pilote adapte et evolutif).

Du point de vue des performances, les ameliorations sont sensibles. La communication avec le calculateur hote est optimisee par un codage specifique des informations echangees. Les traitements concernant la generation d'entites graphiques de base, sont performants, le parallelisme entre la pre-synthese et la post-synthese etant le plus souvent utilise. Le bien fonde de l'interface microprogramme est demontre. Les problemes de synchronisation et de repartition des taches entre le calculateur hote et le terminal graphique se posent. La recherche d'un parallelisme optimal implique :

- L'accroissement de la puissance de calcul de la pre-synthese (cas du remplissage de taches et de generation de caracteres),
- Le choix de supports de communication performants, en l'occurrence la mise en oeuvre de mecanismes tels qu'un D.M.A entre le processeur MC.6809 et le module de communication est envisageable.
- la recherche et la mise en oeuvre d'algorithmes mixtes (cables, microprogrammes et programmes).
- D'une evolution de l'architecture du module de communication permettant des acces quasi-directs vers les processeurs de la post-synthese.

Ce dernier point concerne particulièrement "l'équilibrage" des tâches entre les différents processeurs pour les processus de synthèse tels que la génération des polygones quelconques ou encore la gestion d'une scène en trois dimensions. En conclusion, la configuration "console évoluée" offre des performances meilleures que celles de la précédente configuration. De plus la fonctionnalité globale du terminal est nettement améliorée.

On notera que la plupart des évaluations concernent les temps d'exécution des commandes sans tenir compte des délais de communication avec le calculateur hôte. Ceux-ci pénalisent sensiblement les performances globales du système malgré les options retenues.

L'ensemble de ces considérations convergent vers la définition d'une troisième configuration de l'architecture du terminal graphique. Le chapitre IV présente une architecture de pré-synthétiseur tri-dimensionnel réalisant un processus de visualisation d'une scène 3.D. Cette étude constitue une voie d'exploration des architectures de pré-synthèse évoluée et multiprocesseurs. Il s'agit de la version HELIOS III.

## CHAPITRE IV

### REALISATION D'UNE MAQUETTE D'EVALUATION 3.D :

#### REFERENCES BIBLIOGRAPHIQUES :

( BIN 78 ), ( BOU 80 ), ( BOU 83 ), ( BOU 84 ),  
( CLA 76 ), ( DOD 84 ), ( DUG 82 ), ( FON 83 ),  
( FPI 84 ), ( INT 80 ), ( JOL 84 ), ( LUC 77 ),  
( MAR 78 ), ( MAR 82 ), ( MER 79 ), ( MOL 76 ),  
( MOT 79 ), ( SUH 74 ), ( MMC 84 ).



## LE PRE-SYNTHESEUR HELIOS-68000 3.D MONO-PROCESSEUR :

La seconde phase de cette etude consistait a realiser une maquette d'evaluation d'une architecture de pre-synthesiseur tri-dimensionnel et monoprocasseur. Cette evaluation portera principalement sur les etapes de calcul necessaires dans un processus de synthese tri-dimensionnel. Dans le cadre de "la configuration evoluee", on definit une architecture de pre-synthese dont les capacites de stockage, de calcul et d'entree/sortie sont accrues.

L'objectif de cette evolution n'est pas de transposer le micrologiciel decrit dans le chapitre precedant dans un contexte materiel plus sophistique, mais plutot d'elargir le domaine de traitement du processus de pre-synthese. L'univers initial est etendu a la manipulation d'objets graphiques tri-dimensionnels constitues de faces planes. Cette approche permet une evaluation preliminaire concernant :

- Les ressources memoire necessaires,
- Les modes d'echange et de communication,
- La puissance de calcul necessaire.

De plus, on mettra en evidence les operations penalisantes du traitement de scenes 3.D afin de pouvoir proposer des solutions plus performantes sous la forme d'algorithmes mixtes (cables/microprogrammes) ou d'architectures paralleles. On etudiera la possibilite de mise en oeuvre de co-processeurs sous-traitant des fonctions particulieres.

## IV.1 CONCEPTS LIES AU TRAITEMENT D'UNE SCENE 3.D :

D'une maniere generale l'ensemble des transformations geometriques que l'on applique a une scene 3.D peut s'inclure dans un processus de synthese plus complet. Dans ce cas une decomposition en une suite d'etapes coherentes est possible. Ce processus consiste a definir pour une scene donnee, un point de vue et un point de visee choisis par l'utilisateur dans son espace.

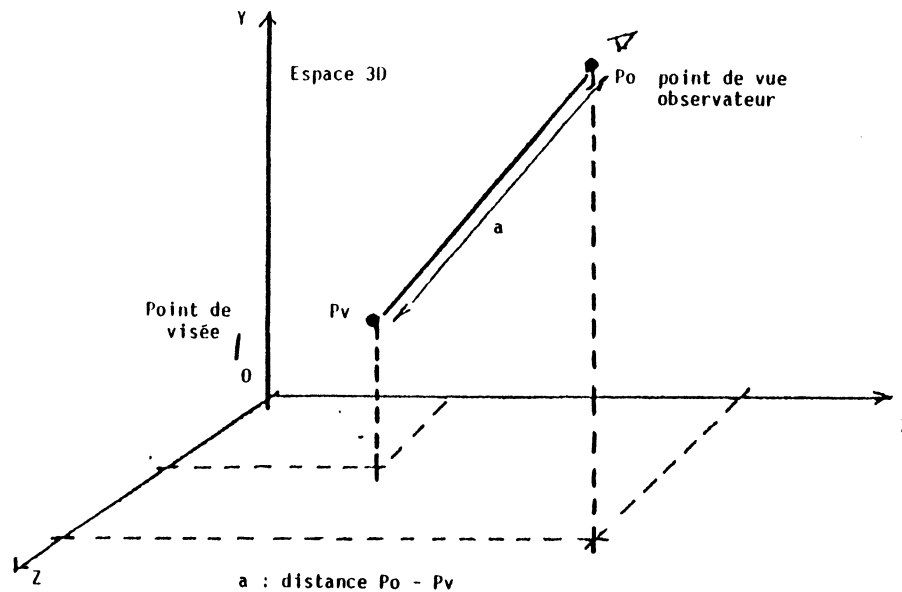


Figure No IV.1

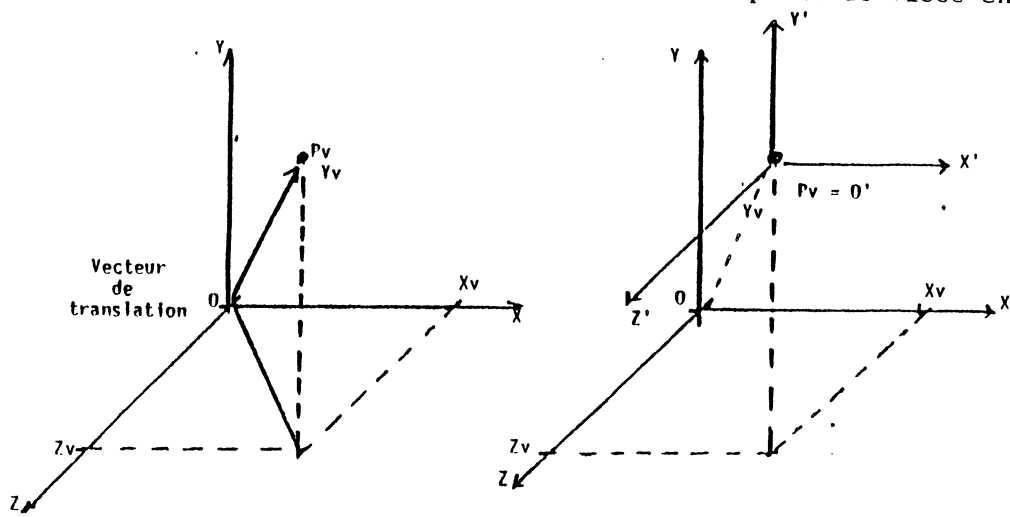
L'utilisateur peut ensuite définir une fenêtre dans son espace, et une cloture dans l'espace de l'écran. Ces informations étant définies, le processus de synthèse visualise une portion de la scène sous un angle donné sur une portion de l'écran. En fonction de ces paramètres on décompose le processus de synthèse en six étapes distinctes :

- Le passage dans le repère du point de vue,
- La projection en perspective,
- Le découpage tri-dimensionnel (fenêtre),
- L'élimination des parties cachées,
- Le passage dans l'espace de l'écran (cloture),
- La visualisation de facettes planes.

L'ordonnement adopté n'est pas obligatoire, l'utilisateur peut sélectionner un sous-ensemble d'étapes cohérentes et nécessaires à son application. L'étape de visualisation est implicite. Il s'agit du remplissage des faces visibles délimitées par leurs contours. En effet certaines étapes peuvent être en partie à la charge de la post-synthèse. (découpage 2.D, Z-Buffer pour l'élimination des faces cachées).

IV.1.1 TRANSFORMATIONS GEOMETRIQUES :

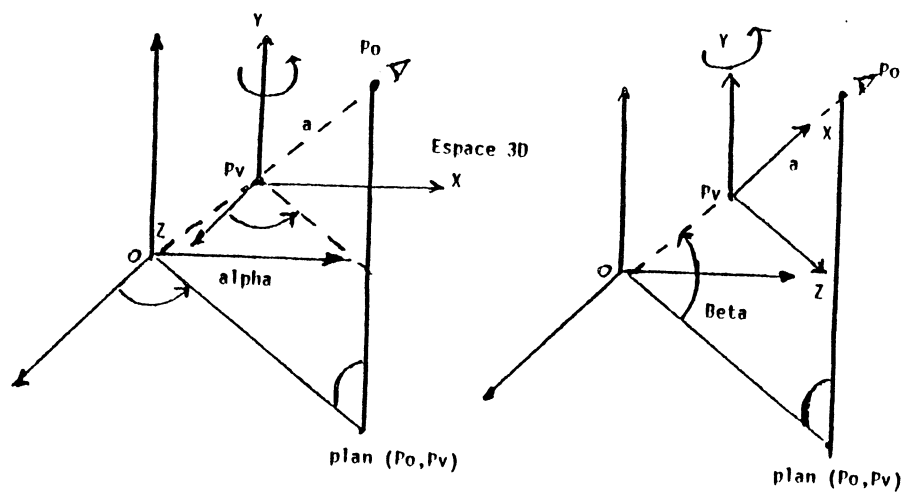
Il s'agit d'effectuer trois transformations geometriques simples. Une translation au point de visee. Cette operation permet de faire confondre l'origine initiale  $O$  du repere-utilisateur, avec le point de visee choisi.



TRANSLATION AU POINT DE VISEE

Figure No IV.2

Une rotation par rapport a l'axe  $Oy$ . Cette operation permet d'amener l'axe  $Oz$  dans le plan de l'espace defini par le point de vue et le point de visee.



ROTATION PAR RAPPORT A O'Y D'UN ANGLE ALPHA

Figure No IV.3

Une rotation par rapport a l'axe Ox. Cette operation permet de rendre co-lineaires l'axe Oz et la demi-droite definie par le point de visee et le point de vue. Cette transformation presente la scene tri-dimensionnelle dans le repere translate au point de visee et face a l'observateur.

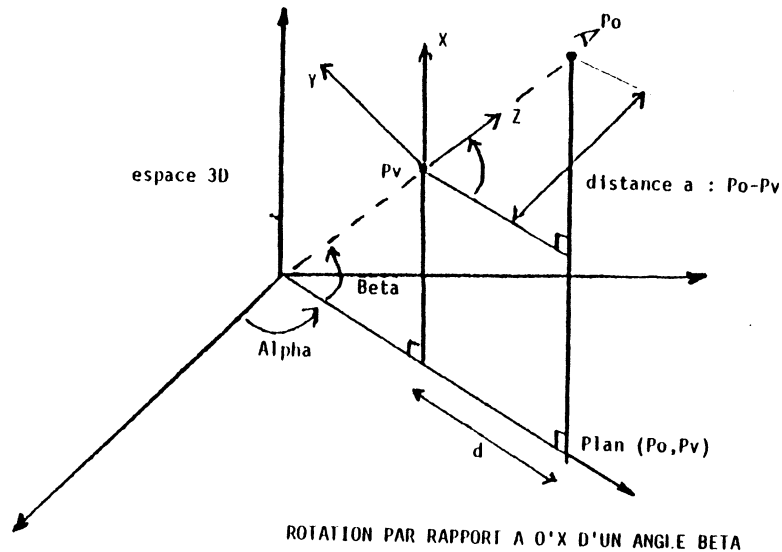


Figure No IV.4

les angles de rotation alpha , beta sont determines par les triangles rectangles delimites par les points Pv , Po et leurs projections respectives selon l'axe Oy. On obtient les relations suivantes :

$$\cos(\alpha) = d / a,$$

$$\sin(\alpha) = ( y_o - y_v ) / a,$$

$$\cos(\beta) = ( x_o - x_v ) / d,$$

$$\sin(\beta) = ( z_o - z_v ) / d.$$

Po (xo, yo, zo) : Point observateur.

Pv (xv, yv, zv) : point de visee.

a : distance entre le point de visee et le point observateur.

Ces trois operations sont effectuees a l'aide des matrices caracteristiques des transformations geometriques de l'espace. On utilise les coordonnees homogenes de maniere a pouvoir composer les transformations aisement. Les matrices de transformation sont :

La translation au point de visee :

```
-----
0    0    0    0
0    0    0    0
0    0    0    0
-xv  -yv  -zv  1
```

La rotation autour de l'axe Oy :

```
-----
cos(beta)  0  sin(beta)  0
0          0    0        0
-sin(beta)  0  cos(beta)  0
0          0    0        1
```

La rotation autour de l'axe Ox :

```
-----
1    0    0    0
0  cos(alpha)  -sin(alpha)  0
0  sin(alpha)   cos(alpha)  0
0    0    0    0
```

La matrice de transformation resultante est  $M.T = T.r * R.y * R.x$  Pour la simplicite des calculs, on considere que le repere de l'observateur est exprime par rapport au repere absolu de la scene. On peut envisager une inclinaison de l'observateur, dans ce cas une rotation supplementaire est necessaire...

#### IV.1.2 PROJECTION EN PERSPECTIVE :

Cette phase consiste a moduler les coordonnees (x, y) par rapport a la profondeur z. Cette modulation tient compte de la distance a qui est la distance entre le point de visee et le point de vue. Pour un point donne de l'espace S (x, y, z), on definit la grandeur omega :

$$\omega = ( a - z ) / a.$$

Les cas extremes sont :

- z = 0, omega = 1, l'effet de perspective est inexistant, le point est dans le plan de projection.
- z = a, omega = 0, l'effet de perspective est indefini, le point est projete a l'infini.

La projection en perspective est définie par :

$$S(x, y, z) \quad \text{---->} \quad Sp(x / \omega, y / \omega, 0)$$

$$\omega = (a - z) / a$$

La distance point de visée - point de vue permet de définir un cône de vision pour l'observateur, la fenêtre définie dans l'espace constitue la base du cône.

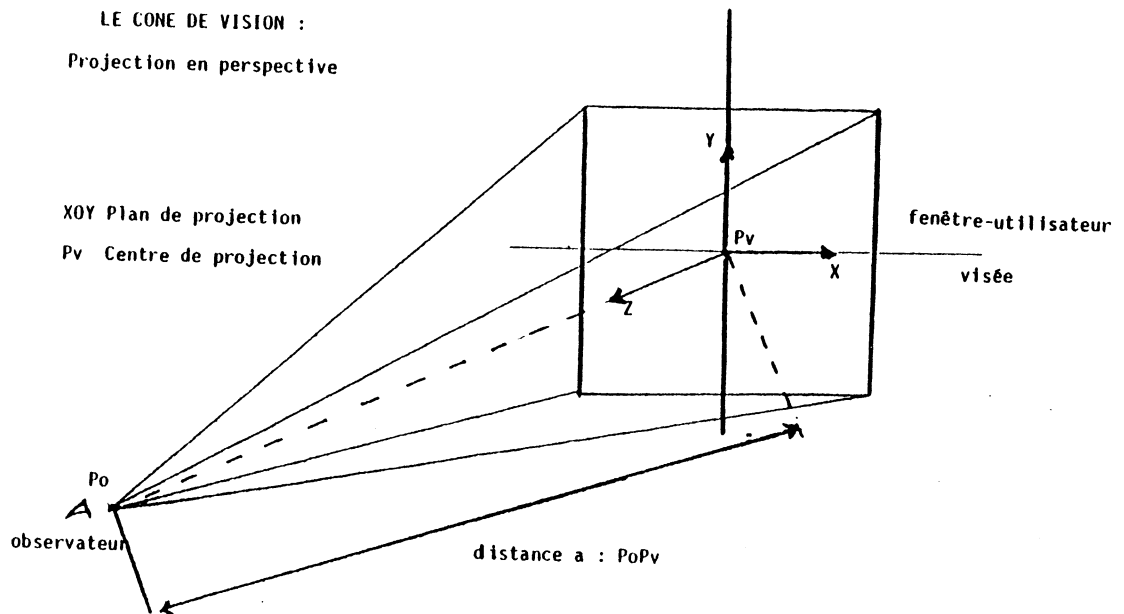


Figure No IV.5

La forme de la fenêtre est carrée, elle est centrée en Pv. Ce choix est restrictif mais assez courant pour les terminaux tri-dimensionnels. La transformation de la projection en perspective fait correspondre à un point de l'espace quelconque, le point du plan correspondant. Le point de l'espace peut être exprimé dans un espace à quatre dimensions par la coordonnée homogène supplémentaire  $\omega$ .

Projection en perspective :

$$\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/a \\ 0 & 0 & 0 & 0 & 1 \end{array}$$

Pour un point quelconque S (x, y, z) de l'espace, si  $\omega < 0$  alors  $z > a$ , le point est derrière le plan limite par  $z = a$ . (Sommet invisible)  
 Dans le cas contraire,  $\omega > 0$  alors  $z < a$ , le point est devant le plan limite par  $z = a$ . (BOU 84) (BIN 78) Pour l'ensemble des sommets appartenant au plan  $z = a$ ,  $\omega = 0$ , le point  $S_p$  est non-défini, on considère que le point est projeté à l'infini, donc invisible. On définit donc deux grandeurs :

- Le plan de devant :

$$z = fg \text{ et } fg = 9 / 10 * ( a ). \text{ ( foreground )}$$

- Le plan de fond :

$$z = bg \text{ et } bg = - 9 / 10 * ( a ). \text{ ( background )}$$

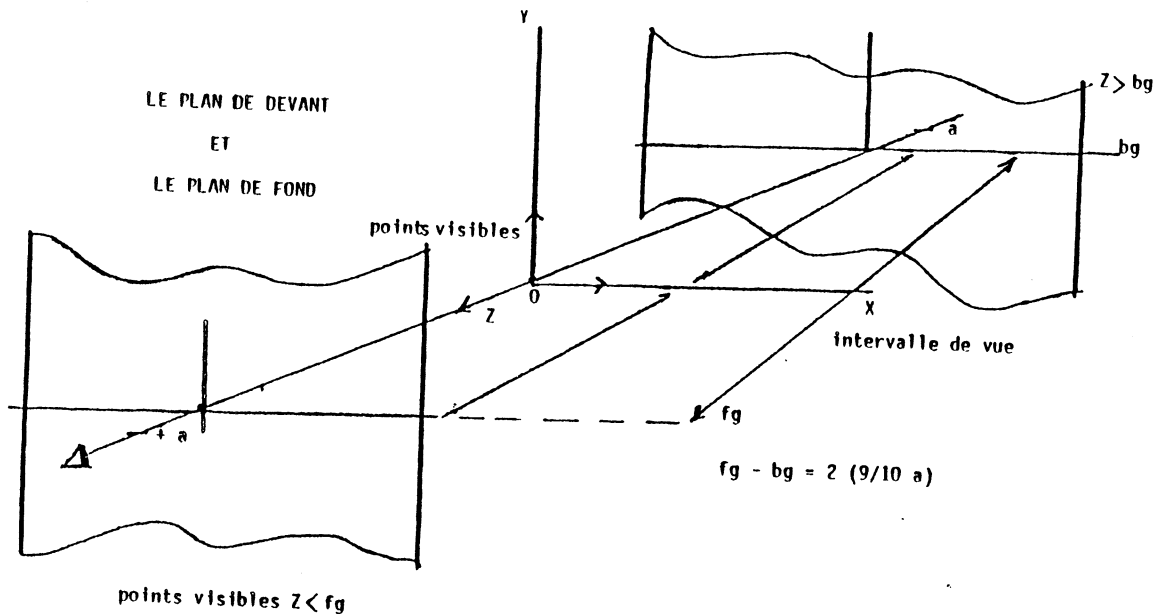


Figure No IV.6

Ces deux plans constituent des limites pour le calcul de la grandeur  $\omega$ . Cette solution permet d'éviter la projection à l'infini et aussi la projection de sommets hors du champ de vision dont la distance maximum est  $2 * a$ , plus exactement :

$$- fg - bg = 9 / 10 * ( 2 * a ).$$

#### IV.1.3 DECOUPAGE TRI-DIMENSIONNEL :

Le découpage consiste à visualiser uniquement les portions de la scène

incluses dans le cone de vision defini par l'observateur. Differentes methodes de decoupage existent. (9 regions, cone canonique) On etudiera le decoupage par plan multiple et la recherche dichotomique pour obtenir un ordre de grandeur des temps de traitement necessaires. Une etude comparative des differente methodes existe. ( SUTH 74 ) ( BOU 80 ) ( HEG 85 )

L'interet de la methode choisie est d'une part la possibilite de parallelisme entre le decoupage 3.D et 2.D, ce dernier pouvant etre realise d'une maniere cablee (cas de la recherche dichotomique). D'autre part nous tenons a signaler que le decoupage 3.D dans une fenetre rectangulaire est le plus souvent propose par les constructeurs de terminaux de haute gamme. Le principal inconvenient de cet algorithme est le decoupage de scenes constituees de polygones convexes. L'algorithme de decoupage considere est caracterise par deux parametres qui sont la distance point de visee-point de vue "a" et la taille de la fenetre "b".

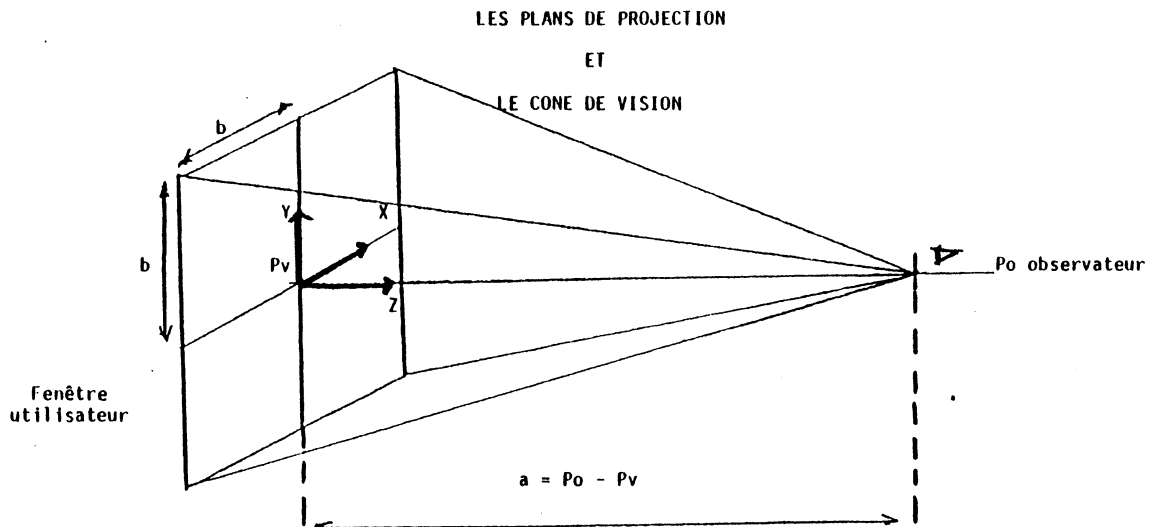


Figure No IV.7

La portion elementaire est une arete delimitée par deux sommets Sd , Sg. L'operation de decoupage determine la portion visible de l'arete donc incluse dans le cone de vision. Le cone de vision est delimité par quatre plans de l'espace definis par les equations suivantes :

- P1 :  $P1 ( S ) = b * z - a * y - a * b$
- P2 :  $P2 ( S ) = b * z + a * y - a * b$
- P3 :  $P3 ( S ) = b * z - a * x - a * b$
- P4 :  $P4 ( S ) = b * z + a * x - a * b$



Un point quelconque est l'interieur du cone de vision si et seulement si pour  $i = 1,2,3,4$  on a  $P_i ( S ) < 0$ ,  $S (x, y, z)$  etant un sommet de l'espace,  $P_i$  represente un plan de l'espace. De plus la projection en perspective implique que tout point de l'espace  $S (x, y, z)$  est transforme en un point projete  $S_p (x/\omega, y/\omega, 0)$ . Ce point  $S_p$  doit etre devant le plan limite  $z = a$  et a l'interieur du cone de vision, soit la condition suivante.

$P ( x , y , z )$  est a l'interieur du cone si et seulement si :

$$\omega < 0$$

$$- b \leq y / \omega \leq + b$$

$$- b \leq x / \omega \leq b$$

Dans ces conditions, le point est devant le plan limite, a l'interieur de la fenetre de projection. Le decoupage d'une arete est dependant de ces deux extremités  $S_d (x_d, y_d, z_d)$  et  $S_g (x_g, y_g, z_g)$ . On determine trois cas typiques :

- $\omega - S_d$  et  $\omega - S_g$  sont positifs, les deux extremités sont entre le plan  $z = a$  et le plan de projection, apres la projection en perspective, il s'agit d'un decoupage dans le plan.
- $\omega - S_d$  et  $\omega - S_g$  sont negatifs, les deux extremités sont derriere le plan  $z = a$  donc hors du cone de vision.
- $\omega - S_d * \omega - S_g < 0$ , dans ce cas l'arete est de part et d'autre du plan  $z = a$ . On recherche l'intersection  $I_s$  avec le plan  $z = a$ , si  $\omega - I_s > 0$  la nouvelle arete est traitee selon le premier cas.

( BIN 78 ) ( BOU 84 ) ( SUH 74 )

la seconde phase de decoupage consiste a determiner la portion de l'arete projetee incluse dans la fenetre. Il s'agit de rechercher les intersections gauche et droite avec les bords de la fenetre definie par l'utilisateur. La recherche de l'intersection avec un bord de la fenetre se fait par divisions successives de l'arete en son milieu. Il s'agit d'une recherche dichotomique. ( BOU 80 )

## ALGORITHME :

```

-----
fonction recherche ( G ; D ) ;
debut
  si D : interieur-fenetre alors
  faire
    tant que ( G = D ) ;
    faire
      M := ( G + D ) / 2 ;
      si M : interieur fenetre alors D := M
      fsi ;
    finfaire ;
  si M : interieur-fenetre alors recherche := M
  sinon recherche := nil
  fsi ;
  finfaire ;
  sinon recherche := D
  fsi ;
fin ;

```

La determination du segment visible a l'interieur de la fenetre se fait par deux recherches :

- Sg := recherche ( G , D ) ;
- Sd := recherche ( D , G ) ;

## IV.1.4 ELIMINATION DES PARTIES CACHEES :

Cette phase du processus de synthese est necessaire pour des images dont le degre de realisme est indispensable a l'application. Le probleme qui se pose est de realiser apres la phase de decoupage, un passage dans le repere de l'ecran sans ambiguïte en ce qui concerne la perception de l'image synthetique. La premiere approche consiste a ne pas visualiser les faces arrieres d'un objet-3.D. (produit scalaire entre le vecteur de visee et la normale de chaque face.) Cependant cette solution n'est pas satisfaisante bien que simple a la mise en oeuvre. En effet, elle permet de lever une partie de l'ambiguïte lors de la visualisation de l'objet-3.D.

Une premiere amelioration consiste a affecter une priorite d'affichage entre les differentes faces des objets-3.D de la scene. Enfin des algorithmes specifiques permettent de realiser de maniere coherente l'elimination des parties cachees d'un objet-3.D par rapport au point de vue et a la scene considerée. Diverses methodes ont fait l'objet d'etudes et d'evaluations. Nous nous interessons aux plus courantes :

- Algorithme de Newell, Newell et Sancha,
- Algorithme de Watkins,
- Algorithme de Warnock et Aterton-Weiler
- Algorithme du Z-Buffer.

Le premier est basé sur un tri de profondeur des faces. Le remplissage des faces se fait dans la mémoire de trames, en tenant compte de la profondeur relative à chaque face. Les faces les plus proches du point de vue sont traitées à la fin du processus. Cette solution implique un tri des faces en fonction de leurs profondeurs respectives. Une ambiguïté demeure, celle des faces qui se chevauchent...

La seconde méthode travaille directement sur la mémoire de trame. On effectue un découpage horizontal correspondant aux lignes de balayage de l'écran. Les segments visibles issus de la phase de découpage sont ensuite traités pour la phase ultime de visualisation en tenant compte, encore une fois, de la profondeur.

L'algorithme de sous-division de l'espace recherche une surface cohérente de polygones visibles sur l'écran en tenant compte du point de vue. La sous-division de l'écran s'arrête au niveau du pixel dans le pire des cas. Cette sous-division s'appuie sur un découpage en carrés de l'écran. Dans chaque carré on effectue comme précédemment une étude de la profondeur, l'affichage se fait de la face la plus éloignée à la face la plus proche. La face la plus proche de l'observateur est considérée comme face de référence.

Enfin l'algorithme du Z-Buffer s'intéresse à la profondeur de chaque point de la face. Il nécessite une mémoire de trame supplémentaire pour l'analyse de la scène. Dans ce cas on considère les faces dans un ordre quelconque. L'élimination des parties cachées a lieu lors du remplissage des mémoires de trame. Pour chaque point on considère la profondeur minimale. On compare la profondeur du point courant et celle définie par l'utilisateur. L'écriture dans la mémoire d'images est conditionnelle:

- La profondeur du point courant est supérieure à celle qui est pré-définie, on invalide l'affichage du point dans la mémoire d'images.
- La profondeur du point courant est inférieure ou égale à celle pré-définie, on valide l'affichage du point dans la mémoire d'images.

Ce traitement est effectué pour toutes les faces constituant la scène 3.D. L'avantage de cette solution est la simplicité de mise en œuvre, surtout si la post-synthèse peut configurer l'une de ces mémoires de trame, en mémoire de profondeur. De plus le problème des faces s'interpénétrant et

se chevauchant ne se pose plus car l'analyse de la scene se fait au niveau du pixel. Dans notre etude on s'interessera a l'agorithme du Z-Buffer qui semble le plus adapte aux terminaux graphiques dont la memoire de trame peut etre reconfigurer en memoire de profondeur. (Cas de la station GETRIS)

( ZAR 85 )

De plus la repartition des taches peut se faire entre la pre-synthese qui effectue le calcul de la profondeur des faces en la comparant a celle qui est pre-definie et la post-synthese qui realise l'écriture conditionnelle dans la memoire de trames en fonction de la profondeur minimale pre-definie. Au niveau des performances, le Z-Buffer semble etre le plus constant car independant de la complexite de la scene. Le seul inconvenient est l'adjonction d'une memoire de profondeur qui doit etre initialisee par la pre-synthese. De plus l'analyse de scenes constituees de faces gauches ne remet pas en cause le processus adopte pour l'eliminations des parties cachees...

#### IV.1.5 PASSAGE DANS LE REPERE-ECRAN :

L'operation de passage dans le repere de l'ecran concerne toute arete projetees puis coupee et incluse dans le cone de vision. Cette portion d'arete est definie par ces deux extremités G ( $x_g, y_g$ ) et D ( $x_d, y_d$ ). La transformation-ecran consiste a faire coincider la fenetre utilisateur et la cloture definie dans l'espace-ecran. La fenetre et la cloture de l'utilisateur sont dans le meme plan.

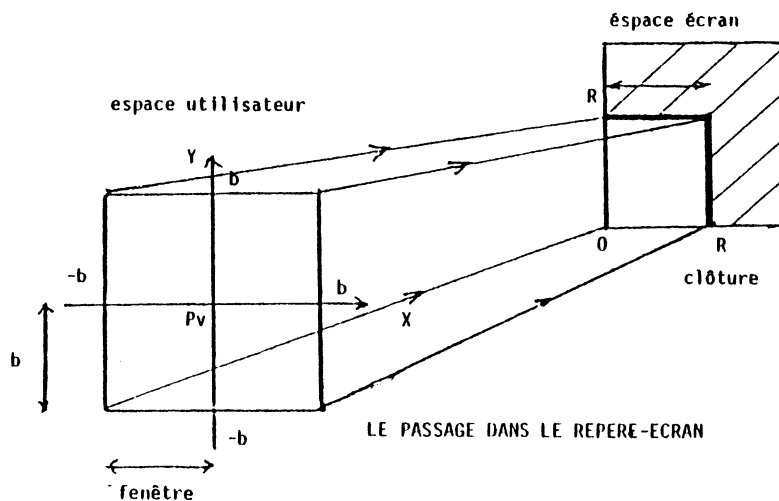


Figure No IV.8

La transformation-ecran est definie par:

$$P ( x_p , y_p ) \text{ ---> } P_e ( x_e , y_e )$$

$$x_e = ( x_p + b ) * r / ( 2 * b )$$

$$y_e = ( y_p + b ) * r / ( 2 * b )$$

Homothetie du plan de rapport  $K = r / ( 2 X b )$ .

Dans ce cas r represente la taille de la cloture carree de l'ecran et b represente la taille de la fenetre de l'utilisateur. L'approche mathematique et la decomposition du processus de synthese tri-dimensionnel a mis en evidence une relative complexite des calculs et des algorithmes au niveau des differentes phases du traitement.

#### IV.2 FONCTIONNALITES DU PRE-SYNTHESEUR :

La fonctionnalite du pre-synthetiseur tri-dimensionnel est essentiellement basee sur la decomposition du processus de synthese decrit precedemment.

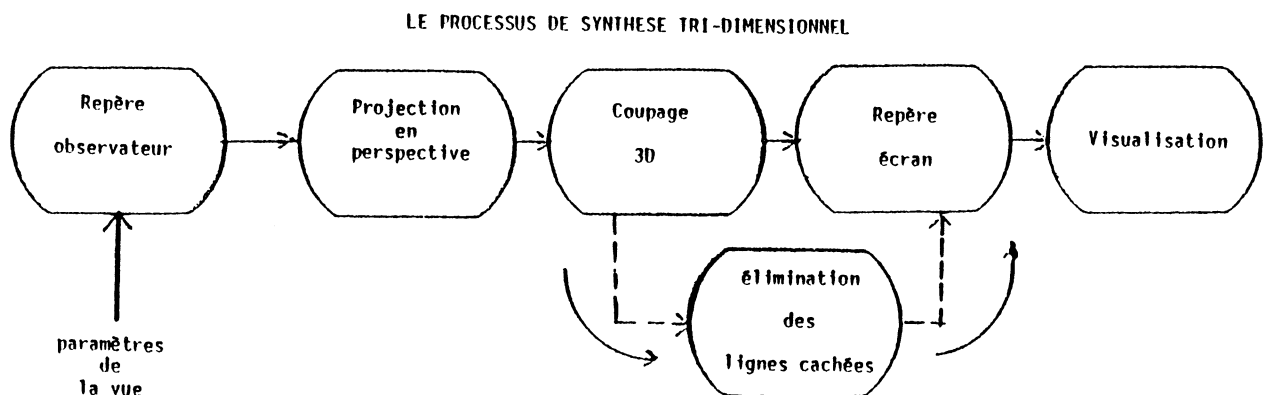
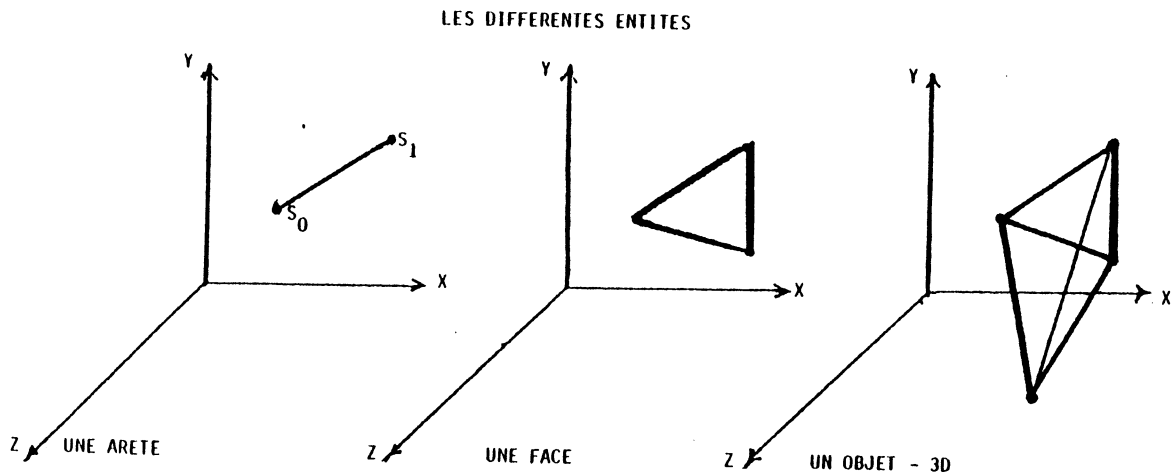


Figure No IV.9

L'application residente au niveau du calculateur hote peut choisir une suite d'etapes coherentes, communiquer les parametres necessaires et activer le processus de synthese. Cette commande suppose que la scene tri-dimensionnelle est deja memorisee au niveau du pre-synthetiseur. Il s'agit de definir l'ensemble des informations echangees entre le calculateur hote et le terminal et les structures de donnees qui leurs sont associees.

Dans une premiere phase, on definira l'ensemble des entites de base. L'entite elementaire est le sommet ou point de l'espace  $S(x, y, z)$ . Un objet de l'espace est une suite de sommets. Une arete est un couple de sommet, une face, une suite finie d'aretes (le contour) et un objet-3.D une suite finie de faces. Cette definition hierarchique est representee par la figure suivante.

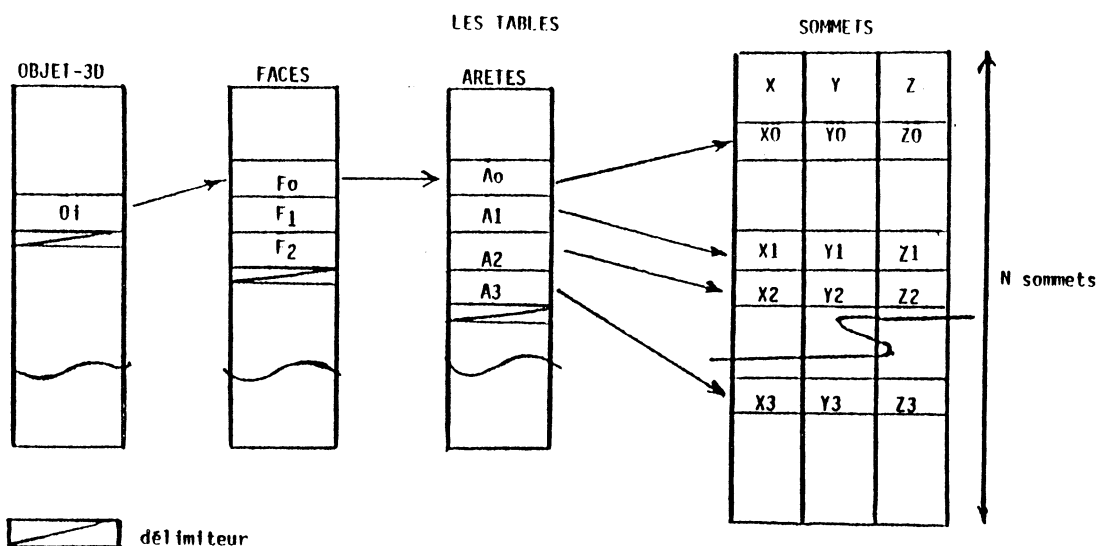


La structure de donnees qui en decoule est un ensemble de tables hierarchisees. Elle permet a partir d'une table des sommets et de tables de descripteurs, de reconstituer la morphologie des differents objets-3.D et la geometrie de la scene. On definit deux tables supplementaires de descripteurs :

- La table des aretes : elle permet d'identifier une suite d'aretes a partir de la table des sommets.
- La table des faces : elle permet d'identifier une suite de faces a partir de la table des aretes.

REMARQUE :

Par extension de cette structure, on peut definir une table des objets-3.D permettant d'identifier une suite d'objets de la scene a partir de la table des faces. Cette structure hierarchisee permet de mettre en oeuvre une table complete et unique de sommets a partir de laquelle toutes les compositions morphologiques et geometriques sont possibles. Le cas des aretes communes est traite dans le paragraphe IV.5



LA STRUCTURE DE DONNEES HIERARCHISEE

Figure No IV.11

On definit un intervalle d'identification < INTERV >. Ce parametre indique un intervalle d'objets-3.D.

0 .. 0 : toute la scene.

I .. J : les objets 3.D de I a J.

I, J sont des entrees dans la table des objets-3.D,  $I < J$

L'intervalle peut designer un ensemble de faces si le pre-synthetiseur ne gere pas la table des objets-3.D. L'ensemble des informations sauvegardees au niveau du pre-synthetiseur respecte les concepts de coherence et de symetrie qui guident la definition de l'architecture. On s'interesse particulierement aux informations d'identite, de morphologie et de geometrie.

#### IV.2.1 GESTION DE L'ESPACE MEMOIRE :

Il est necessaire d'evaluer les capacites de memoire permettant de sauvegarder la structure de donnees. Supposons qu'un objet-3.D soit represente par un cube. Un cube est constitue de huit sommets, douze aretes et six faces. Pour une scene 3.D constituee de 1000 cubes, on obtient 8000 sommets, 12000 aretes et 6000 faces. On considere que le codage d'une valeur entiere se fait sur 16 bits, l'eventail etant dans ce cas assez important. Un sommet est constitue de trois coordonnees codees chacune sur 16 bits. La table des sommets requiert donc 48 K-octets de memoire vive.

Pour que la structure soit homogene du point de vue du codage, on considere qu'un descripteur d'arete est code sur 16 bits et qu'un descripteur de face est code sur 16 bits.

- La table des aretes requiert donc : 24 K-octets de memoire vive,
- La table des faces requiert donc : 12 K-octets de memoire vive.

On suppose que la table des objets-3.D est geree au niveau de l'application, au sein du calculateur hote. De plus une table des sommets transformes est necessaire. L'etape minimale du processus de synthese est la transformation geometrique. (Transformation matricielle) Cette phase de calcul requiert un codage plus precis car les etapes consecutives necessitent des divisions par des valeurs codees sur 16 bits aussi. Un sommet transforme est constitue de trois coordonnees codees chacune sur 32 bits. Ainsi la table des sommets transformes requiert 96 K-octets de memoire vive.

L'interet de cette duplication de la table des sommets est de disposer de la representation initiale de la scene et de la representation courante de la scene apres la derniere transformation geometrique. L'acces a la table des sommets transformes se fait par l'intermediaire des memes tables des aretes et des faces. On remarque aussi la coherence de la structure hierarchisee et la compression des informations de description. La quantite d'espace memoire necessaire pour une scene de 1000 cubes est :

- 180 K-octets de memoire vive.

La recherche de performances optimales implique une communication minimale avec le calculateur hote et une puissance de traitement non-penalisee par des acces a la memoire importants et par des calculs supplementaires inutiles. Ce choix implique de mettre en oeuvre des structures de donnees volumineuses, penalisantes uniquement a l'initialisation du pre-synthetiseur.

#### IV.2.2 GESTION DE LA COMMUNICATION :

Un volume important d'informations est transfere du calculateur hote au pre-synthetiseur. Dans une configuration "console evoluee", la communication se fait via une ligne serie asynchrone a 9600 bauds.



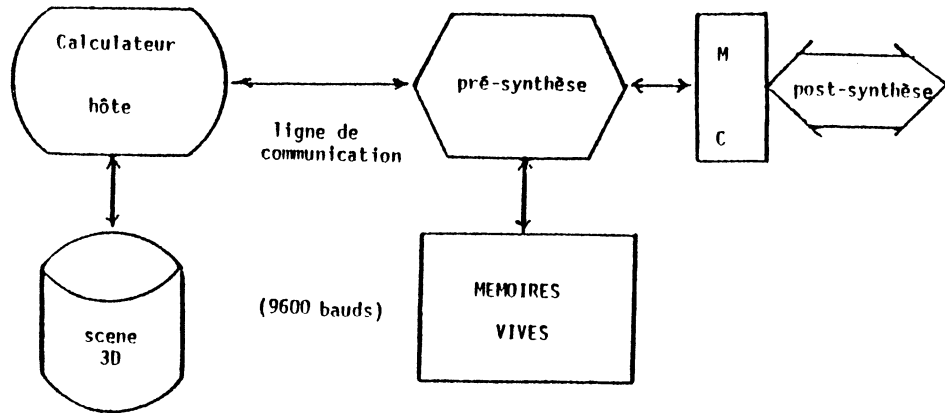


Figure No IV.12

L'évaluation de l'acquisition de la scene de 1000 cubes se fait en ne tenant pas compte des delais de synchronisation lies au protocole d'acquisition, et du codage/decodage de l'information. On obtient dans ce cas les evaluations suivantes :

L'acquisition de la table des sommets se fait en 50 secondes.

L'acquisition de la table des aretes se fait 25 secondes.

L'acquisition de la table des faces se fait 12,5 secondes.

Soit environ une minute et trente secondes de delai de communication pour une vitesse de transmission relativement elevee. Cette contrainte d'acquisition de la scene 3.D implique la mise en oeuvre d'interfaces de communication plus performantes ou la possibilite d'un archivage local au niveau de la pre-synthese. Par exemple, la realisation d'une interface de communication parallele avec un protocole d'echange de type "demande et acquittement", sur une largeur de 16 bits permet de reduire ces delais :

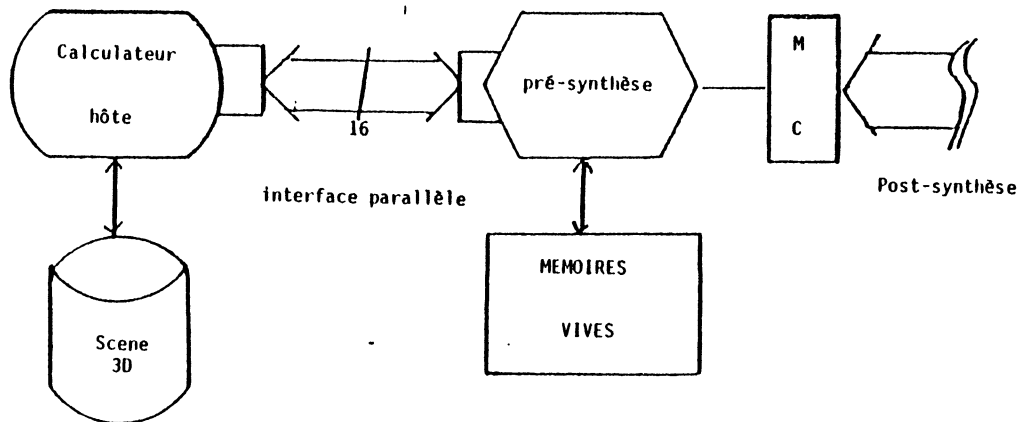


Figure No IV.13

La procédures d'échange d'une information peut être programmée de la façon suivante :

```

procédure reception-donnee
debut
  lire-port-parallele (etat) ; (* attente : etat-pret *)
  lire-port-parallele (donnee) ;
fin
    
```

```

procédure emission-donnee ;
debut
  lire-port-parallele (etat) ; (* attente : etat-pret *)
  écrire-port-parallele (donnee) ;
fin
    
```

Un délai d'échantillonnage équivalent à 3 accès-mémoires est nécessaire. Le test représente un accès mémoire de 6 cycles d'horloge à 500 nanosecondes. On détermine ainsi un délai de 9 microsecondes. Les temps de transfert obtenus sont :

- La table des sommets : 216 millisecondes,
- La table des arêtes : 108 millisecondes,
- La table des faces : 54 millisecondes.

Soit un total de 378 millisecondes, ce délai minimum du point de vue de l'évaluation est acceptable. (limite  $\leq 1$  seconde) Le choix de l'interface optimale doit tenir compte des normes en vigueur, (facilite de connexion du terminal) et de la simplicité de mise en oeuvre (coût de l'interface). Il

est a noter que d'autres types d'interfaces existent (D.M.A), ceci depend fortement du calculateur note...

#### IV.2.3 FONCTIONS DU PRE-SYNTHESEUR :

Les fonctions disponibles au niveau de la pre-synthese permettent d'accéder globalement ou selectivement a la structure de donnees representant la scene 3.D. L'utilisateur peut commander :

- La visualisation complete ou partielle de la scene,
- La suppression, la duplication, l'insertion d'un objet 3.D,
- L'identification d'un objet 3.D a l'aide d'un dispositif d'interaction, etc.

Ces fonctions concernent les processus d'identification, d'attribution, de consultation et la visualisation (implicite). Un objet-3.D peut porter un ensemble d'attributs, les sous-ensembles d'attributs etant associes aux differentes faces le constituant. Dans cette etude nous nous interesserons plus particulierement aux processus affectant la geometrie de la scene 3.D dans son ensemble. Il s'agit des transformations geometriques. L'utilisateur peut specifier un ensemble d'objets-3.D sur lequel porte la transformation :

- La translation,
- La rotation par rapport a  $Ox$ ,  $Oy$ ,  $Oz$ ,
- Les homotheties,
- La projection en perspective.

Deux propositions sont envisageables. La matrice de transformation est evaluee et communiquee par le calculateur note, ou bien les parametres de la transformation geometrique sont communiquees au pre-synthesiseur qui calcule a son tour la matrice de transformation. Dans le premier cas, la matrice est evaluee en virgule flottante et transmise en virgule fixe au pre-synthesiseur par un ordre de transformation. Seuls les coefficients significatifs sont transmis. Il s'agit des treize coefficients codes sur 16 bits chacun. Les trois derniers sont implicites. ( BOU 84 )

Dans le second cas, les parametres caracterisant la transformation sont transmis au pre-synthesiseur. Dans ce cas le calcul de la matrice est a la charge du pre-synthesiseur 3.D. Cette phase de transformation produit en sortie la table des sommets transformes pour les etapes suivantes du processus de synthese. Son evaluation sera proposee dans le paragraphe IV.5

## LA TRANSFORMATION MATRICIELLE

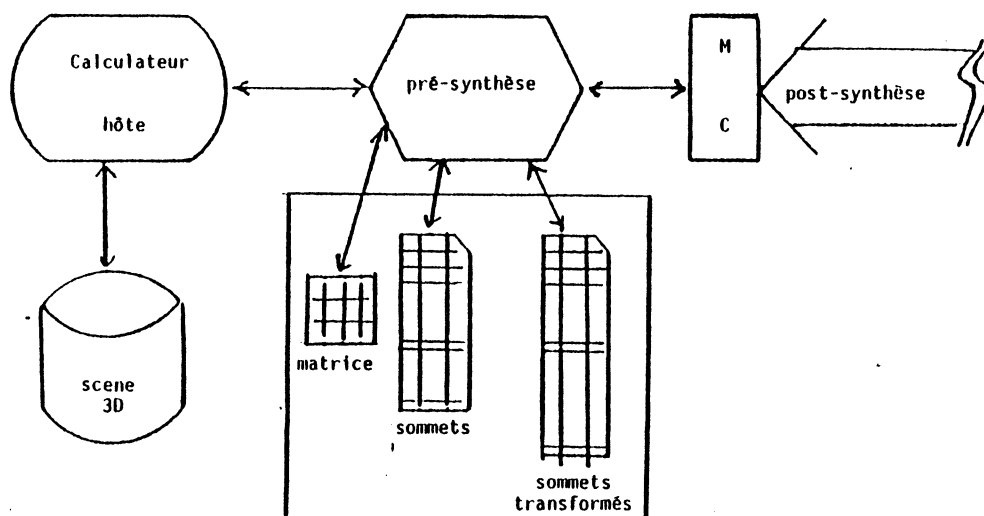


Figure No IV.14

Pour les différentes transformations le problème de la précision des calculs se pose. Cependant une étude a montré que le processus de visualisation produit une image quasi-identique selon deux cheminement de calculs.

- Le calcul en virgule flottante jusqu'au passage dans le repère de l'écran. Le passage en virgule fixe a lieu pour la phase de visualisation.
- Le calcul en virgule flottante de la matrice de transformation uniquement, L'évaluation d'une matrice entière et le calcul en virgule fixe jusqu'à la phase de visualisation.

## REMARQUE :

Pour des coefficients matriciels qui peuvent être inférieurs à un, des facteurs de compensation sont mis en œuvre. Il s'agit du cas des rotations et des homothéties essentiellement. La compensation qui consiste à multiplier ces coefficients par une puissance de deux, est ensuite annulée après la transformation matricielle par une opération de décalage arithmétique à droite.

## IV.3 ARCHITECTURE DU PRE-SYNTHETISEUR 3.D :

Les différentes étapes du processus de synthèse requièrent des opérations de multiplication et de division. Les operands manipulées sont

representees sur 16 et 32 bits. Cette taille constitue la precision minimum pour ce type de calcul. De plus d'autres types de calcul interviennent dans le processus de synthese 3.D (produit scalaire, produit vectoriel ...). Notre choix s'est tourne vers les nouveaux microprocesseurs 16 bits disponibles et integrant dans leur jeu d'instructions :

- La multiplication : 16 X 16 bits, le resultat etant sur 32 bits signes,
- La division : 32 / 16 bits, le resultat etant sur 16 bits signes.

Ces processeurs adressent des espaces memoires de l'ordre de 8 a 16 M-octets, cette possibilite resoud definitivement le probleme de memorisation locale des structures de donnees de la scene, au niveau du pre-synthetiseur. Pour une meilleure precision des calculs matriciels, la mise en oeuvre d'un co-processeur en arithmetique flottante est possible. Du point de vue des performances, l'amelioration est enregistree au niveau du parallelisme possible entre les deux processeurs. Ce point est detaille dans le chapitre V.

#### IV.3.1 CHOIX D'UN PROCESSEUR 16/32 BITS :

On considere trois criteres de comparaison qui sont l'horloge de fonctionnement du processeur, le temps d'acces a la memoire et la capacite d'adressage. La plupart des processeurs 16/32 bits utilisent des bus de communication asynchrones et meme multiplexes. Il faut choisir le composant disposant d'une horloge de base la plus rapide possible afin d'obtenir des performances en temps de cycle les meilleures, et ceci pour une famille de composants donnee. Ce tableau regroupe les caracteristiques principales d'un ensemble de ces processeurs.

TABLEAU RECAPITULATIF :

PROCESSEUR	HORLOGE	ACCES-MEMOIRE	MUL	DIV	DONNEES	ADRESSAGE
MC.68020	16 MHz	250 nano-s	5	10	32 bits	16 M-octets
NS.16000	10 MHz	400 nano-s	8	17	16 bits	16 M-octets
IT.8086	8 MHz	500 nano-s	17	21	16 bits	8 M-octets
Z.8000	4 MHz	750 nano-s	18	25	16 bits	64 K-octets

Le temps d'accès memoire correspond a la valeur minimale. Les temps d'execution des operations de multiplication et de division sont les valeurs maximales exprimees en microsecondes. La taille des donnees correspond a la largeur du bus externe. (16 bits) Le Z.8000 est capable de faire une multiplication 32\*32 bits, le resultat etant sur 64 bits, en 72 microsecondes et une division 64/32, le resultat etant sur 32 bits, en 182 microsecondes. Cette remarque est aussi valable pour le NS.16000. Ces deux processeurs acceptent le format "entier long". La figure suivante recapitule les temps d'accès memoire des differents processeurs.

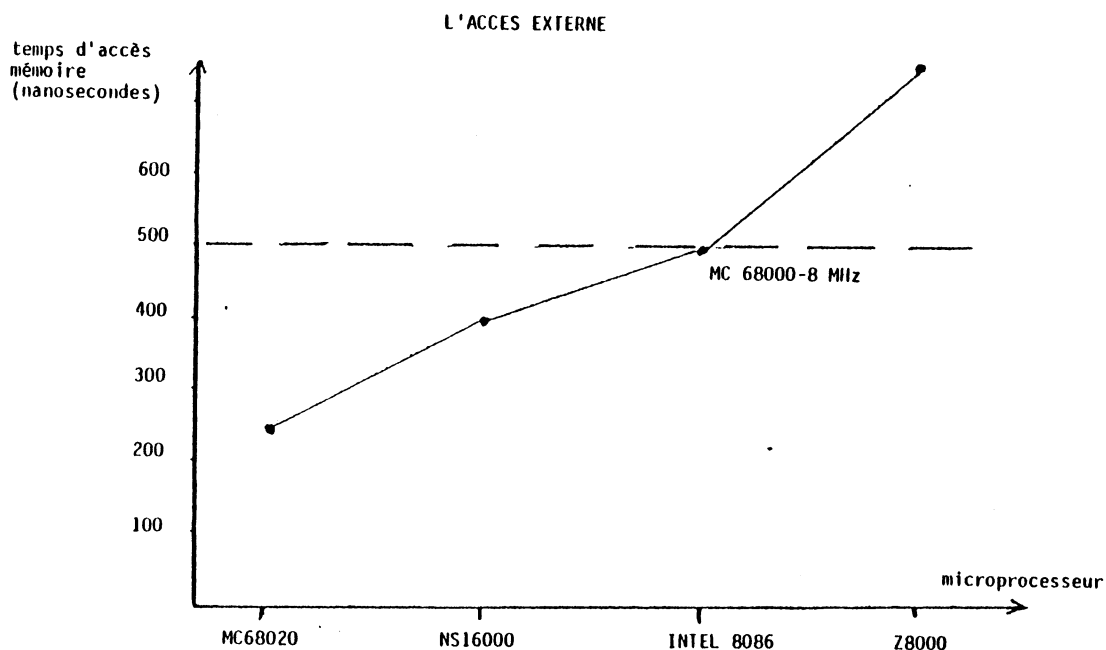


Figure No IV.15

L'utilisation d'un MC.68000 a 8 MHz pour le pre-synthetiseur 3.D a permis d'avoir des ordres de grandeur des delais de calcul dans le processus de visualisation 3.D. L'etude comparative des performances montre que seuls les processeurs MC.68020 et NS.16000 peuvent apporter une amelioration. Le processeur Intel.8086 possede des performances relativement identiques a celles du MC.68000 a 8 MHz, a l'exception de la difference de capacite d'adressage.

Cette etude ne tient pas compte des nouveaux processeurs de la famille Intel fonctionnant a 10 MHz. (Intel 80186, 80286 et 80386) L'interet des modes d'adressage et du jeu d'intructions du MC.68000 au niveau d'un pre-synthetiseur microprogramme, est etudie dans le paragraphe IV.3.3. La figure suivante recapitule les temps de calcul des deux operations MUL et DIV pour les differents processeurs.

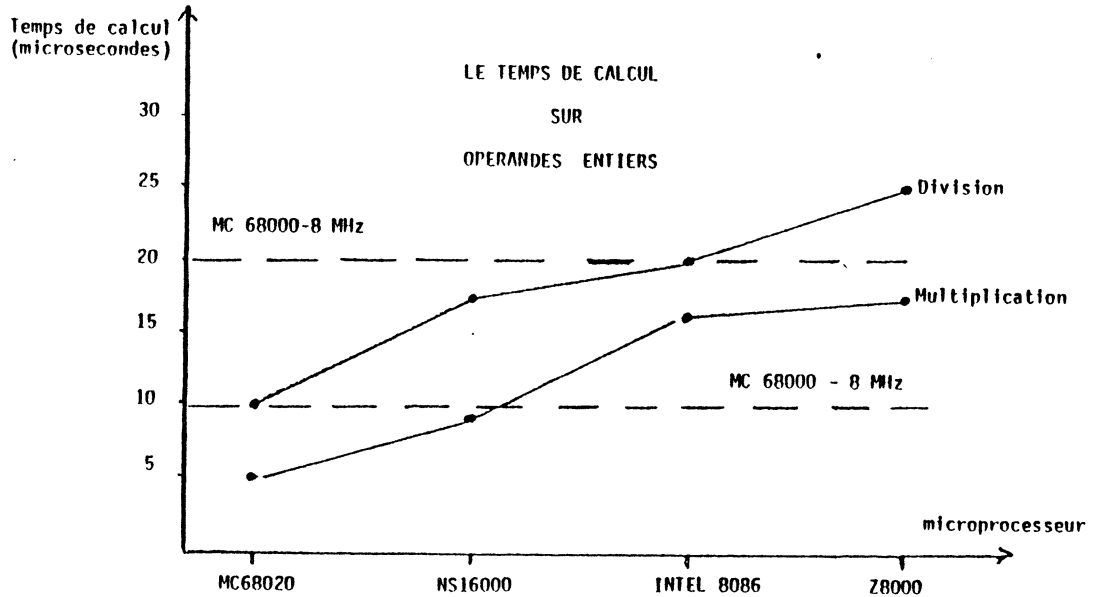


Figure No IV.16

#### IV.3.2 CARACTERISTIQUES TECHNIQUES :

La première étape de définition d'une architecture consiste à choisir le processeur central. L'étude comparative nous a permis de démontrer que le processeur le plus adapté serait le MC.68000 dans une version initiale à 8 MHz. L'évolution de l'architecture est possible par la disponibilité de processeurs identiques fonctionnant à 10 MHz et même 16 MHz. Il s'agit d'un micro-processeur 16 bits externe (32 bits interne), disposant d'un bus asynchrone fonctionnant par acquittement. Les principales caractéristiques sont :

- L'espace d'adressage de 16 M-octets,
- Les entrées/sorties en configuration mémoire,
- 5 types de données acceptées :
  - . Le bit, le digit BCD (4 bits), l'octet (8 bits), le mot (16 bits), le mot long (32 bits).

Il gere 8 niveaux d'interruptions externes avec priorite ainsi que des deroutements internes. La notion d'instructions privilegiees existe, deux modes d'executions sont possibles le mode maitre et le mode esclave.

Ce processeur 16/32 bits dont l'orthogonalite du jeu d'instructions est le principal atout se repand de plus en plus sur le marche. La compatibilite des coupleurs auxiliaires d'interface 8 bits est conservee. De plus MOTOROLA a annonce et echantillone une nouvelle famille de composants O.E.M en 16 bits, associee au processeur MC.68000.

Le choix de ce processeur permet de doter la pre-synthese d'une puissance de calcul appreciable. Dans un environnement minimum, le MC.68000 disposera d'une capacite de memoire vive importante. Cette memoire sera realisee a partir de RAMs dynamiques, un processus de rafraichissement sera prevu. Des interfaces d'entrees/sorties standards seront integres a l'architecture ainsi qu'une interface specialisee d'accès au module de communication.

#### IV.3.2.1 UNITES D'ENTREES / SORTIES : -

Le processeur MC.68000 gere trois lignes asynchrones de type V.24-RS.232. Les coupleurs d'entrees/sorties serie sont implantes dans une page memoire specifique appelee : "VPA". La mise en oeuvre d'un decodage particulier pour la page d'entree/sortie permet au processeur MC.68000 de reconnaitre un acces a destination d'un coupleur 8 bits et d'engendrer les differents signaux compatibles necessaires.

De plus une interface parallele est disponible. Il permet un echange bi-directionnel d'information sur 16 bits. Le protocole d'echange est de type "Demande - acquittement". Enfin le processeur MC.68000 gere un pupitre d'interaction 16 touches a travers un double circuit d'encodage et d'amplification. L'affectation des differentes interfaces est identique a celle de la configuration "console evoluee".



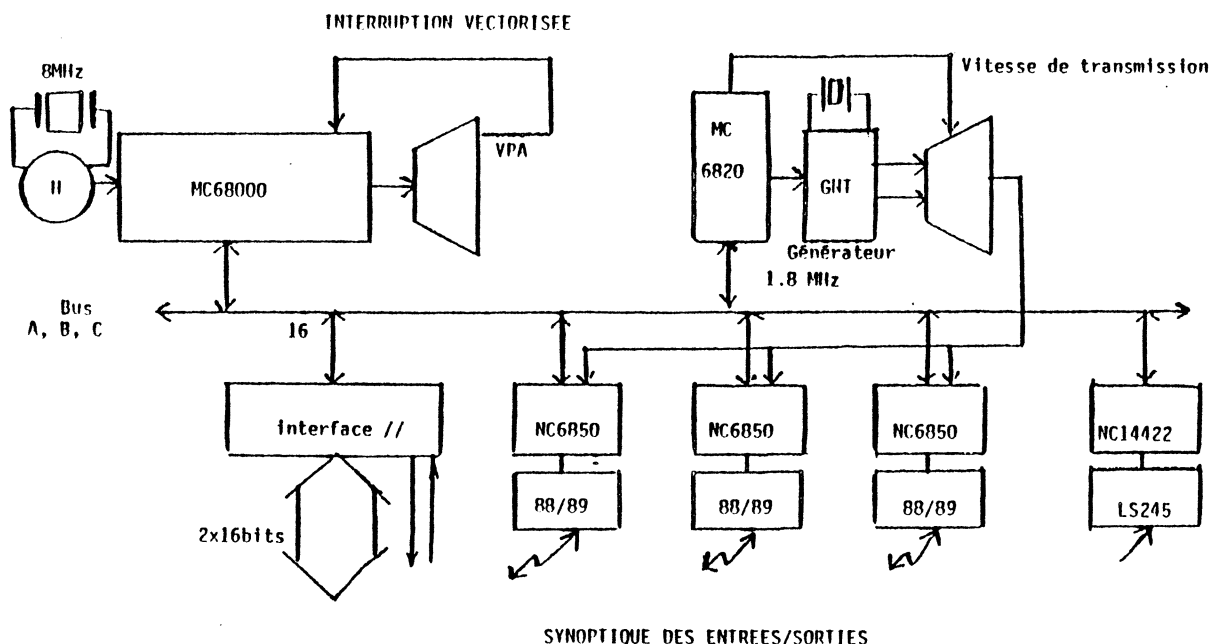


Figure No IV.17

D'autre part, le decodage "VPA" permet la gestion par interruption des entrees/sorties; le mode de fonctionnement est appele "mode auto-vectorise". En effet le MC.68000 accepte jusqu'a 8 niveaux d'interruptions avec priorite, auxquels sont associes automatiquement des adresses de sequences d'exception. Le protocole de la configuration precedente peut etre reconduit.

#### IV.3.2.2 MEMOIRES LOCALES : -

L'ensemble des memoires locales adressables par le processeur MC.68000 est constitue de deux blocs principaux, le bloc des octets de forts poids et celui des octets de faibles poids. Les bus d'echange du MC.68000 sont de type asynchrone. Chaque type de memoire dispose d'un acquittement specifique du cycle en fonction de son temps d'accès.

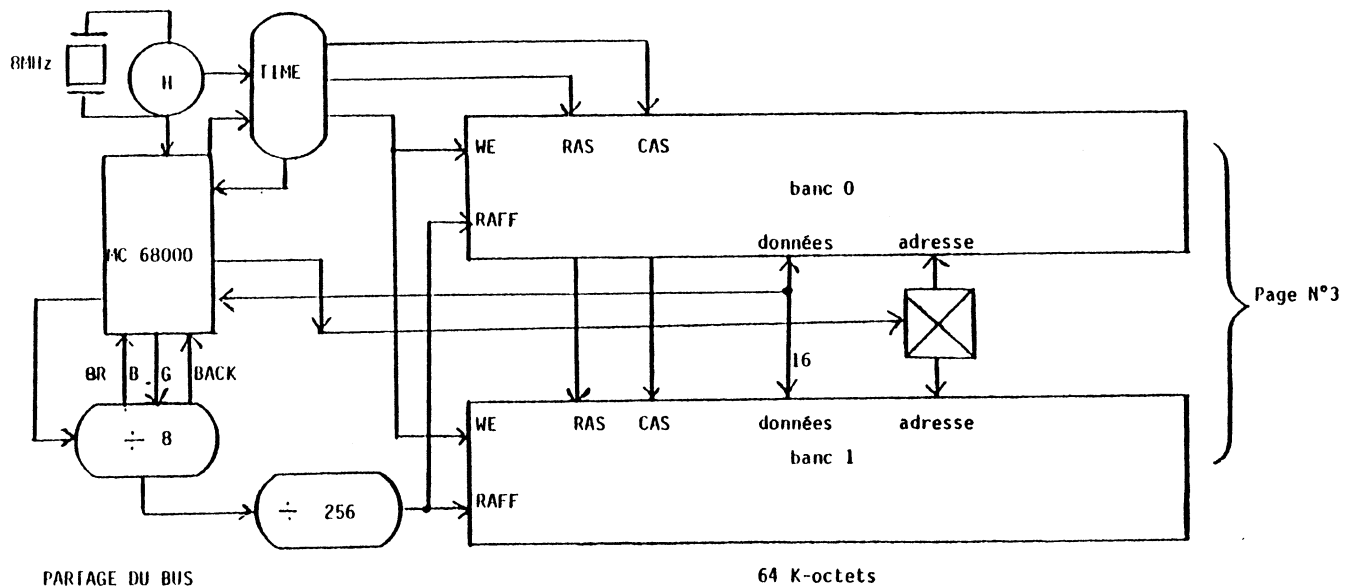
- Les memoires mortes PROM : 450 nanosecondes,
- Les memoires vives RAM statiques : 400 nanosecondes,
- Les memoires vives RAM dynamiques : 250 nanosecondes.

Cette repartition permet aux programmes utilisateurs de s'executer avec un cycle-memoire inferieur, a l'exception des appels au moniteur resident en memoire morte. Cette option permet d'optimiser les acces aux donnees ou aux

programmes residents en memoire vive, en particulier aux tables hirarchisees. La premiere page memoire est implantee en adresse basse : "000.000" Les 32 premiers mots sont stockes en ROM et affectes aux differents vecteurs d'initialisation du processeur : Reset, SSP, TRAP d'urgence. Les 256 mots suivants sont stockes en memoire vive de type RAM statique. Ils constituent la table des interruptions et des deroutements du processeur, ainsi que la pile systeme. La gestion des interruptions retenue est de type "auto-vectorise".

Les deux pages suivantes de memoire morte ont une capacite totale de 16 K-mots (32 K-octets) organisees en blocs de 8 K-octets. Cette memoire contient d'une part un moniteur d'exploitation de 40 commandes avec un assembleur/desassembleur ligne a ligne, resident. D'autre part le micro-logiciel associe au pre-synthetiseur y est sauvegarde.

La derniere page memoire est affectee a l'espace memoire vive de l'utilisateur, dont la capacite est de 64 K-mots (128 K-octets). Elle est organisee en deux blocs de 32 K-mots chacun. Il s'agit de memoires dynamiques, un processus de rafraichissement transparent est disponible. Ce processus est base sur une requisition des bus toutes les 8 microsecondes afin de rafraichir les 256 lignes des deux bancs de memoire en moins de 4 millisecondes.



LA RAM DYNAMIQUE : 128 K-octets

Figure No IV.18

Le dispositif de rafraichissement a ete concu de facon a etre commun aux deux bancs memoire de 64 K-octets. Une extension de la capacite memoire ne pose aucun probleme, le processus de rafraichissement peut etre partage indefiniment, jusqu'a concurrence de la "sortance" des amplificateurs d'adresse et de donnee.

IV.3.2.3 INTERFACE SPECIALISEE : -

L'accès au module de communication est simplifié par rapport au pré-synthétiseur de la configuration "console évoluée". L'accès aux différents registres se fait en une seule phase sur une largeur de données de 16 bits. Le module de communication se présente de la même façon, comme une mémoire locale de 8 registres de 16 bits. Le processeur MC.68000 accède à cette mémoire par adresse paire de 0 à 16. Une page spécifique est réservée au module de communication. Cette page est décodée comme un bloc d'entrée/sortie de type "VPA". Ainsi la gestion par interruption du module de communication est possible.

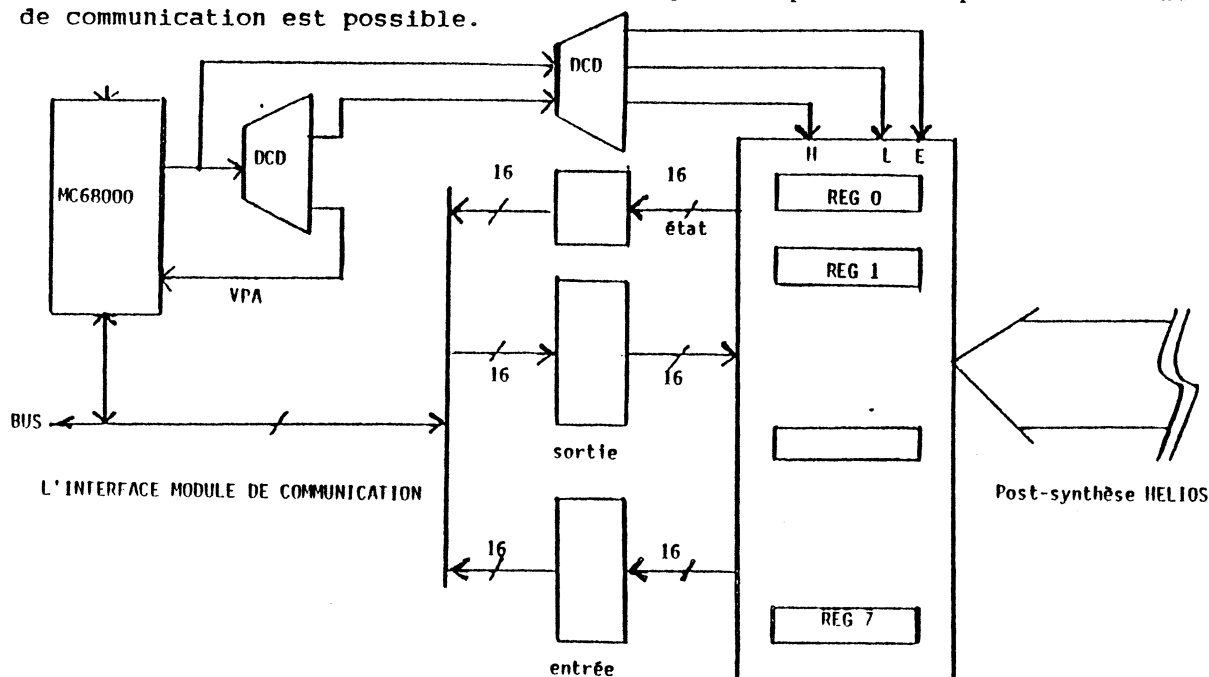


Figure No IV.19

Une information à destination du module de communication est conservée jusqu'à une nouvelle définition explicite. L'acquiescement de fin de cycle-bus concernant les accès au module de communication peut se faire de deux façons.

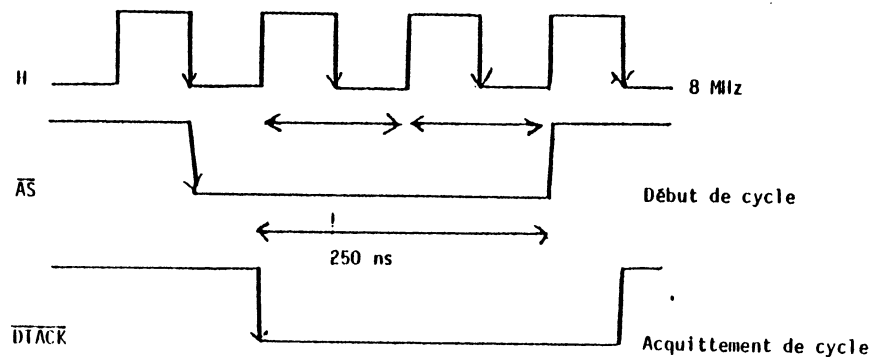
**L'ACCES "VPA" :** Dans ce cas, il s'agit d'une émulation d'un bus 8 bits. L'accès peut se faire sur 16 bits, cependant l'acquiescement est pris en compte après dix périodes de l'horloge de base à 125 nanosecondes, soit un temps d'accès de 1,25 microseconde.

**L'ACCES RAPIDE :** Dans ce cas il s'agit d'un accès spécifique ou l'acquiescement est produit en début de cycle bus soit un temps d'accès inférieur à 300 nanosecondes. Dans cette configuration, le mode d'interruption auto-vectorisé n'est pas utilisable. Ce type d'accès permet une gestion optimale du module de communication.

En conclusion, l'architecture du pre-synthetiseur 3.D presente des performances intrinseques meilleures. Il s'agit maintenant d'affiner les evaluations concernant d'une part les traitements associes a la pre-synthese et d'autre part les acces a la post-synthese.

IV.3.2.4 EVALUATION DES ECHANGES : -

Le processeur MC.68000 fonctionne a partir d'une horloge de base de 8 MHz, soit un temps de cycle de 125 nanosecondes. L'accès mémoire minimum est obtenu pour un acquittement de bus en debut de cycle :



CYCLE - COURT - MC 68000

Figure No IV.20

Le module de communication est en acces rapide. Du point de vue du temps acces a la post-synthese, on considere tout d'abord le delai d'initialisation d'un registre sur 16 bits, soit 1,125 microseconde. Ce delai doit etre double car l'information est prealablement extraite de la memoire vive du pre-synthetiseur. Il s'agit du temps moyen en fonction des modes d'adressage possibles. L'accès a une memoire d'un processeur de la post-synthese necessite :

- Une initialisation du registre module,
- Une initialisation du registre d'adresse : X et Y,
- Une initialisation ou une consultation du registre de donnee.

Cet acces est possible si le registre d'etat indique une disponibilite du module de communication. Cette fonction est detectee par scrutation du registre d'etat.

$$T\text{-acces a la post-synthese} = T\text{-scrutation} + T\text{-acces-module,}$$

L'evaluation donne le resultat suivant, avec une scrutation minimale. Le temps d'acces a la post-synthese est de l'ordre de 5,125 microsecondes. Le gain par rapport a l'architecture a base de MC.6809 est de 60 % pour les processus de lecture ou d'ecriture des memoires de la post-synthese. Ce gain decoule du temps de cycle ameliore et du transfert via un bus de donnees externes de 16 bits. Dans cette evaluation on ne decompte pas le delai supplementaire necessaire a l'initialisation d'un registre interne du processeur et ceci pour que la comparaison avec le MC.6809 soit coherente.

Pour un processus d'ecriture des memoires de trame du processeur de visibilite, une initialisation supplementaire est necessaire, celle du registre repetition. Il s'agit en particulier du remplissage des plans d'identification pour les operations de generation de vecteurs, du remplissage des taches par exemple. Dans ce cas le delai d'echange avec la post-synthese est de l'ordre de 6,25 microsecondes. Une compression des transferts peut etre mis en oeuvre a l'aide des instructions "move" multiple du MC.68000, une seule instruction permettrait d'initialiser un ensemble de registres du module de communication. Les debits moyens obtenus sont 2,7 M-mots de 16 bits ou 1,6 M-mots de 32 bits.

Le debit via le module de communication a partir d'un pre-synthetiseur 3.D architecture autour d'un MC.68000 a 8 MHz correspond a une vitesse de 15 M-bits par seconde. Ce debit ne tient pas compte de l'option de compression de donnees disponible au niveau du registre repetition.

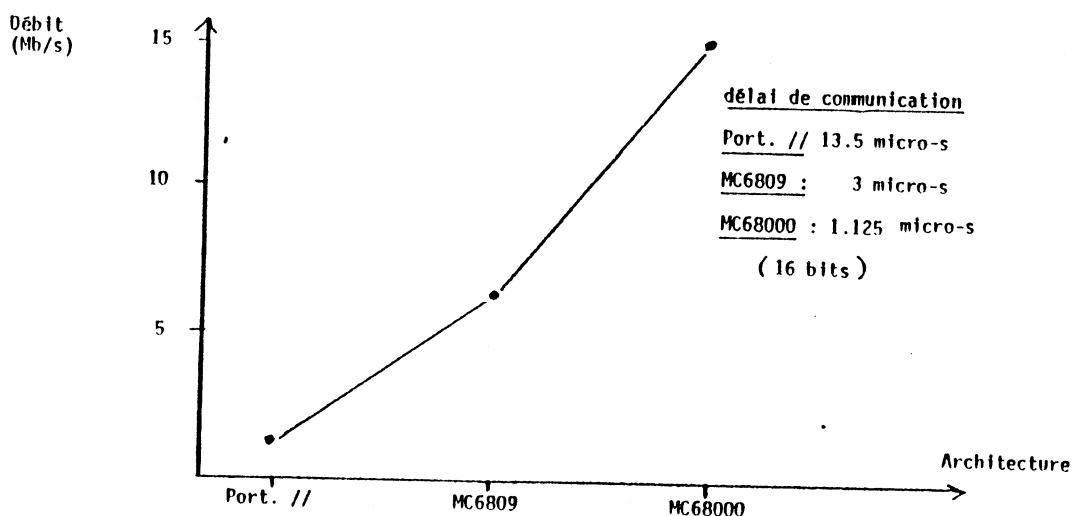


Figure No IV.21

#### IV.4 SOLUTION MICRO-PROGRAMMEE :

L'architecture du pre-synthetiseur tri-dimensionnel a base de MC.68000 constitue dans une premiere etape une maquette d'evaluation du processus de synthese 3.D.

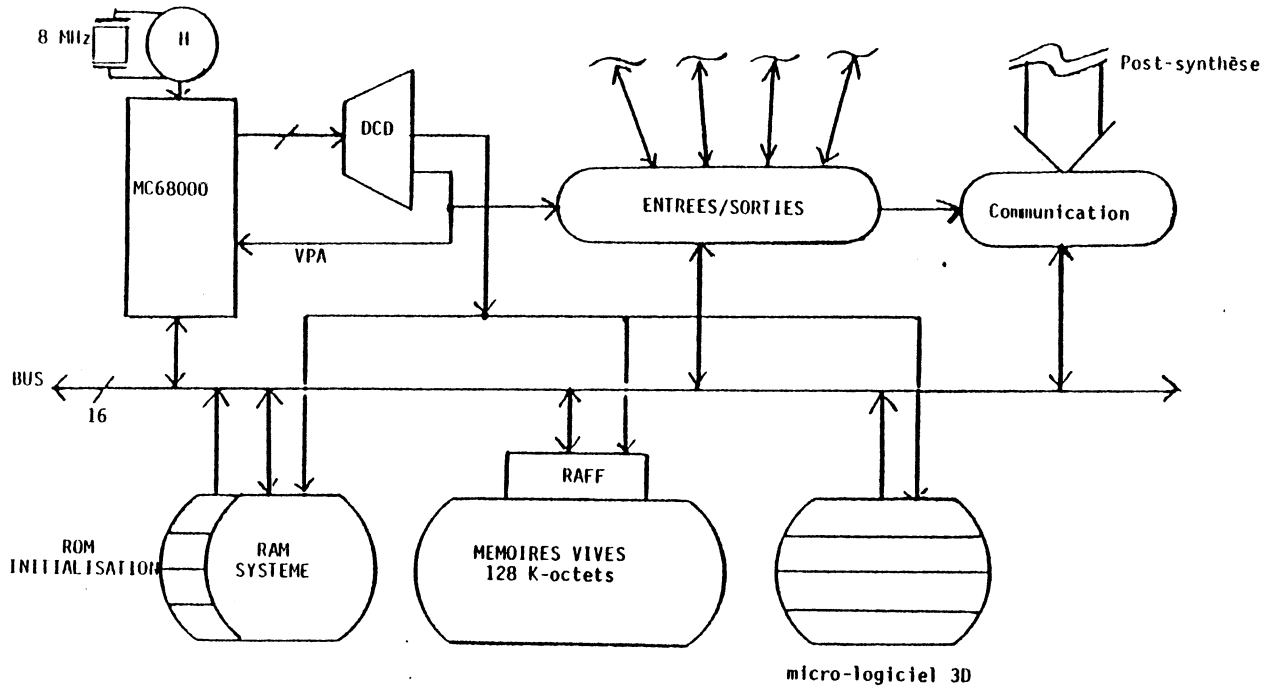


Figure No IV.22

La configuration de developpement et de test du micrologiciel est la suivante :

- Le developpement en assembleur MC.68000 sous le systeme VMS-GAGE, le calculateur hote est un VAX-780.
- Le transfert des fichiers intermediaires vers la maquette de test dotee d'un moniteur de base d'aide a la mise au point.

La manipulation de la structure de donnees decrivant la scene 3.D se deroule selon deux phases. La phase d'initialisation qui effectue la construction et l'initialisation de la structure au niveau du VAX-780. La phase d'acquisition qui effectue la recuperation et la memorisation de la structure de donnees au niveau du pre-synthetiseur 3.D. Le traitement associe a cette structure de donnees permet de definir plusieurs cheminements possibles d'une entite graphique :

- Le cheminement d'une arete,
- Le cheminement d'une face,
- Le cheminement d'un objet-3.D,
- Le traitement global de la scene.

L'ensemble de ces cheminements a ete evalue pour un processus de visualisation partiel ou complet, la composition du processus etant parametree par l'utilisateur. La conception du logiciel a tenu compte des contraintes temps-reel liees au traitement d'une scene tri-dimensionnelle. Il ne s'agit pas de faire une animation 3.D en moins de 40 millisecondes, mais plutot d'identifier et de localiser les operations penalisantes associees aux differentes phases du processus. L'utilisateur construit et initialise la scene 3.D au niveau du calculateur hote puis la transmet au pre-synthetiseur 3.D. Un ensemble de parametres caracteristiques de la vue sont recuperes :

- La cloture d'ecran,
- La fenetre de l'utilisateur,
- La distance point de visee - point de vue,

La matrice de transformation est pre-calculee puis transmise au pre-synthetiseur 3.D. L'evaluation des coefficients est faite au niveau du calculateur hote. Le processus de base consiste a traiter la scene globalement. La projection en perspective est traitee au cours du decoupage 3.D.

ALGORITHME :

-----

```

debut
  transformation matricielle (table-sommets, table-sommets transformes) ;
  calcul coordonnees homogenes (table des sommets, a, table omega) ;
  tant que non (fin)
  faire
    Pour la face-courante
    faire
      extraire (arete) ;
      decoupage 3.D (arete) ;
      si visible
        passage repere ecran (arete) ;
      fsi;
    finfaire;
  visualiser(face-courante)
  (* contour et remplissage eventuel *)
  finfaire;
fin

```

Un processus specifique consiste a manipuler la composante de base de l'objet-3.D courant, il s'agit de l'arete. Dans ce cas le processus fait cheminer un couple de sommets a travers les differentes phases de traitement. L'acces a la structure de donnees est plus selectif. La progression n'est plus conditionnee par le traitement global de la table des sommets mais par l'acces a deux sommets de l'arete courante. Cette arete

appartient a la face courante de l'objet-3.D en traitement.

ALGORITHME :

```

-----
debut
  tant que non (fin)
  faire
    Pour la face-courante
    faire
      Pour chaque-arete
      faire
        extraire (sommet gauche, sommet droit) ;
        transformation matricielle (sommet gauche, somet droit) ;
        calcul coordonnees homogenes (sommet gauche, sommet droit, a) ;
        decoupage 3.D (sommet gauche, sommet droit) ;
        si visible alors
          passage repere ecran (sommet gauche, sommet droit) ;
        fsi ;
      finfaire
    visualiser(face-courante);
  finfaire;
fin

```

REMARQUE :

L'utilisateur peut effectuer un ensemble d'operations de base sur la structure de donnees par l'intermediaire d'un ensemble de commandes de base emises depuis le calculateur hote. Ces operations sont :

- La creation, la destruction, la duplication d'objets 3.D.

Les operations de base peuvent porter sur la table des faces si la table des objets-3.D n'est pas disponible au niveau du pre-synthetiseur. La structure de donnees choisie evite la duplication de l'information. La table des sommets est unique. Cependant elle est mal adaptee aux operations de mise a jour. Le gestionnaire de la structure comptabilise les emplacements libres contigus et non-contigus consecutifs aux operations de destruction. Un utilitaire de type "RAMASSE-MIETTES" serait necessaire. L'activation du "RAMASSE MIETTES" peut etre declenchee si le nombre maximum de destruction d'objets-3.D est atteint ou si une demande de creation d'un d'objet 3.D est non-satisfait.

IV.4.1 STRUCTURE DU MICROLOGICIEL 3.D :

Le micrologiciel 3.D est organise en sept modules successifs. Chaque module integre un ensemble coherent d'actions cooperant dans le cadre d'une phase donnee du processus. Le module d'aquisition de la structure de



donnees se charge de recuperer l'ensemble des parametres necessaires au processus de visualisation 3.D, au niveau du pre-synthetiseur.

## ALGORITHME :

```

-----
debut
  acquerir (table des faces) ;
  acquerir (table des aretes) ;
  acquerir (table des sommets) ;
  acquerir (matrice - transformation) ;
  lire (distance - a) ;
  lire (fenetre - b) ;
  lire (cloture - r) ;
  lire (facteur - compensation) ;
  lire (code - processus) ;
fin

```

Deux modules realisent la transformation matricielle et le calcul des coordonnees homogenes pour la scene 3.D complete. Les quatre derniers modules realisent la projection en perspective et le decoupage tri-dimensionnel de la scene. Un module supplementaire effectue le passage dans le repere de l'ecran en tenant compte de la cloture choisie pour la portion d'arete visible determinee precedemment.

Le module principal realise l'enchainement des differents modules. Il se charge du sequencement des differentes phases du processus compose par l'utilisateur. A l'initialisation, il effectue une remise a zero des registres du pre-synthetiseur 3.D et active le module d'aquisition pour recuperer l'ensemble des parametres de la vue ainsi que la commande utilisateur issue du calculateur hote.

## IV.4.2 METHODE D'EVALUATION :

La phase finale de cette etude consiste a evaluer les temps de traitement necessaires aux differentes phases du processus de synthese. L'objectif est de localiser les operations penalisantes. Dans une premiere etape on effectue une evaluation sequentielle des traitements au niveau de la pre-synthese, sans tenir compte du parallelisme possible avec la post-synthese. La seconde etape mettra en evidence le parallelisme entre la pre-synthese et la post-synthese d'une part, et entre les differents modules de la pre-synthese d'autre part. Ainsi de nouvelles architectures de pre-synthetiseurs 3.D seront proposees.

La modularite du micro-logiciel permet de realiser une evaluation globale ou selective. Au sein d'un module on evaluera une action, un sous-ensemble d'actions cooperant, toutes les actions soit un module. Cette premiere approche permet d'avoir des ordres de grandeur des temps de calcul pour la solution mise en oeuvre. Au sein du micrologiciel on evaluera un

module, un sous-ensemble de modules cooperant entre eux, tous les modules soit le micrologiciel 3.D. Les resultats proposes sont des moyennes obtenues a partir de differentes scenes enregistrees et traitees. (ensemble de cubes) La methode d'evaluation du temps de traitement est simple. Il s'agit de comptabiliser les nombres de cycles MC.68000 necessaire a l'execution d'une sequence d'instructions.

REMARQUE :

Le temps d'evaluation du calcul integre des delais supplementaires. Il s'agit de l'acces a la structure des donnees hierarchisee et du rafraichissement de la memoire dynamique par vol de cycles. (un cycle toutes les 8 microsecondes)

La mise en oeuvre de cette evaluation se fait sous deux aspects. L'aspect materiel, un circuit de comptabilisation du temps de type PIT-MC.68230 a ete integre a l'architecture du pre-synthetiseur. L'aspect logiciel, un module supplementaire constitue de cinq actions est dedie a l'evaluation du micro-logiciel 3.D. Une sequence cible peut etre evaluee de la facon suivante :

ALGORITHME :

```

debut ;
  initialiser;
  valider-decomptage;
  appel-module-cible ;
  devalider-decomptage;
  evaluation-temps-ecoule ;
fin
    
```

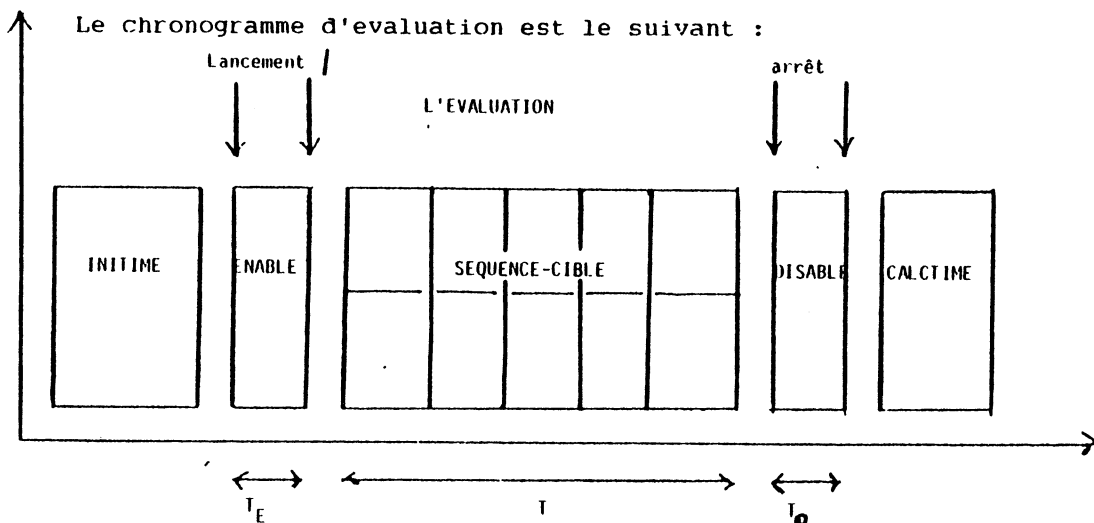


Figure No IV.23

La precision de cette methode depend des rapports  $TE / T$  et  $TO / T$ . L'appel de la sequence cible est decouvert dans l'evaluation. Afin d'ameliorer la precision on peut iterer "N" fois sur l'appel de la sequence cible pour reduire artificiellement les proportions  $TE / T$  et  $TO / T$ .

Le temps de calcul sera dans ce cas  $Te = T / N$  (en microsecondes). Une evaluation preliminaire a permis de determiner  $TE$  et  $TO$  :

- $TE = 8,375$  microsecondes,
- $TO = 7$  microsecondes.

#### IV.4.3 PERFORMANCES DU MICROLOGICIEL :

La modelisation d'un objet-3.D est faite a partir d'un cube. La scene est constituee de  $N$  cubes,  $N$  etant fixe. Dans une premiere phase l'evaluation portera sur le traitement d'un seul cube. On s'interessera particulierement au probleme des transformations matricielles. Les resultats bruts obtenus sont resumes dans les tableaux suivants. Il s'agit du temps de reponse moyen obtenu pour une arete et pour les differentes phases du processus 3.D. (dimension variable de l'arete)

TRANSFORMATION MATRICIELLE : 155 microsecondes.

PROJECTION EN PERSPECTIVE : 54,75 microsecondes.

DECOUPAGE 3.D : 155 microsecondes.

TRANSFORMATION-ECRAN : 69,65 microsecondes.

Les proportions de temps de calcul concernant les differentes entites sont resumees dans le tableau suivant. Il sagit maintenant de tenir compte des delais d'affichage dans le cadre de faces planes.

TABLEAU RECAPITULATIF DES PROPORTIONS :

PHASE DU PROCESSUS	OBJET-3.D	FACE	ARETE
TRANSFORMATION MATRICIELLE	16,4 %	22,2 %	35,7 %
PROJECTION PERSPECTIVE	5,6 %	7,8 %	12,5 %
DECOUPAGE 3.D	56,4 %	50,1 %	35,7 %
TRANSFORMATION-ECRAN	21,6 %	19,9 %	16,0 %

L'evaluation des performances pour une contrainte de temps-reel de 40 millisecondes soit 25 images par seconde, faite pour chaque etape du processus indique les resultats suivants pour une arete, une face etant delimitée par trois ou quatre aretes.

TRANSFORMATION MATRICIELLE : 258 aretes

PROJECTION EN PERSPECTIVE : 730 aretes

DECOUPAGE 3.D : 258 aretes

TRANSFORMATION ECRAN : 574 aretes

Le cheminement a travers le processus complet d'une arete est de 434 microsecondes, soit 92 aretes traitees en temps reel. Les differentes compositions a travers le processus 3.D correspondant au traitement d'une arete indiquent les resultats suivants :

TYPE DE COMPOSITION  
-----

TEMPS -----	PERFORMANCES TEMPS-REEL -----
1-Transformation matricielle, projection en perspective, decoupage 3.D 202 microsecondes	198 aretes
2-Transformation matricielle, projection en perspective 132 microsecondes	303
3-Transformation matricielle, transformation-ecran 142,2 microsecondes	272
4-Projection en perspective, transformation-ecran 124,4 microsecondes	322

Une premiere moyenne des evaluations dans differentes compositions du processus indique que 70 faces sont traitees en temps-reel. Dans cette configuration, les performances obtenues pour des scenes tri-dimensionnelles en dessin au trait sont pour une face :

- Temps de traitement du processus : 697 microsecondes,
- Temps de visualisation : 40 microsecondes pour le contour uniquement. (La generation de vecteurs se fait en 2 a 3 millions points par seconde)

Une face est constituee de 3 ou 4 aretes de dimension 10 points \* 10 points. Le temps total de traitement est : 737 microsecondes. Soit 55 faces traitees en temps-reel, ou 1357 faces traitees en une seconde. Par contre s'il fallait tenir compte des delais de remplissage d'une face, on obtiendrait les resultats suivants en s'appuyant sur les performances de HELIOS-II. Pour une vitesse de 5,25 K-taches par seconde, on traite completement une face en 877 micro-secondes soit 1140 faces traitees en une seconde. (une face de dimension 10X10 points est visualisee en 180 microsecondes) Cette evaluation a permis de mesurer les proportions de calcul inherentes a chaque etape en tenant compte des delais d'affichage. (vitesse de remplissage de taches du terminal HELIOS dans la configuration "console evoluee").

Le parallelisme est possible durant l'écriture des mémoires de trames séquentielles par le module de communication. Cependant cette amélioration est peu sensible étant donné la disproportion des temps de traitement entre la pré-synthèse et la post-synthèse. Cette disproportion croît avec la complexité des calculs liés au processus 3.D. De plus la phase d'élimination des parties cachées n'est pas évaluée dans notre cas. La solution la plus adaptée concernant cette phase du traitement, serait un Z-BUFFER traité en partie par la post-synthèse. En effet, la mise en place d'une mémoire de profondeur pour chaque pixel, le traitement étant ensuite confié à un automate à écriture conditionnelle dans la mémoire de trame.

Les phases de transformation matricielle, de projection en perspective et de transformation-écran peuvent être améliorées au niveau algorithmique. Ainsi la transformation matricielle d'un sommet peut être réduite de 30 % en utilisant le maximum de registres internes du processeur et les instructions de "move" multiple. Dans ce cas, un sommet et une colonne de la matrice seraient pré-chargés, le calcul étant effectué de façon interne.

## ALGORITHME :

-----

```

debut
  extraction du sommet courant;
  chargement-multiple des trois coordonnées;
  pour i:=1 jusqu'a 3
    faire
      chargement-multiple de trois coefficients significatifs;
      (* colonne courante de la matrice *)
      calcul des produits intermediaire;
      cumul des produits et de la composante de translation;
    finfaire
  rangement-multiple du sommet transforme;
fin

```

## REMARQUE :

Les temps de calcul obtenus sont de 50 à 55 microsecondes environ. Les instructions "move" multiples du MC.68000 permettent d'obtenir des débits importants d'échange avec la mémoire, leur application peut être étendue aux échanges avec le module de communication. Les échanges multiples peuvent se faire sur une largeur de 16 ou 32 bits. Au niveau opératoire un co-processeur peut apporter une légère amélioration.

Il serait aussi possible de conserver les résultats intermédiaires des arêtes à travers les différentes étapes et appartenant à plusieurs objets-3.D. Cela constituerait une économie en temps de calcul appréciable. Ainsi, si deux faces possèdent une arête commune le temps de réponse concernant ces deux faces est divisé par deux. Deux sommets sur quatre seront traités seulement pour la face F1. L'arête (S0, S1) sera traitée qu'une seule fois.

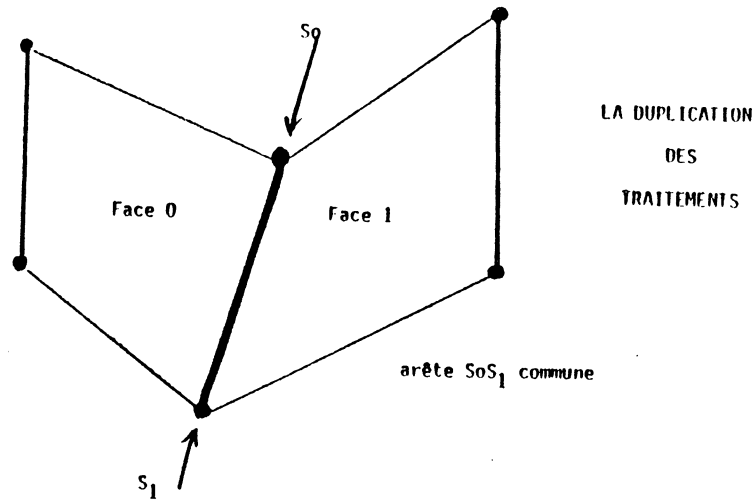


Figure No IV.24

Si l'on considère le processus complet et séquentiel de synthèse la représentation temporelle des proportions est la suivante :

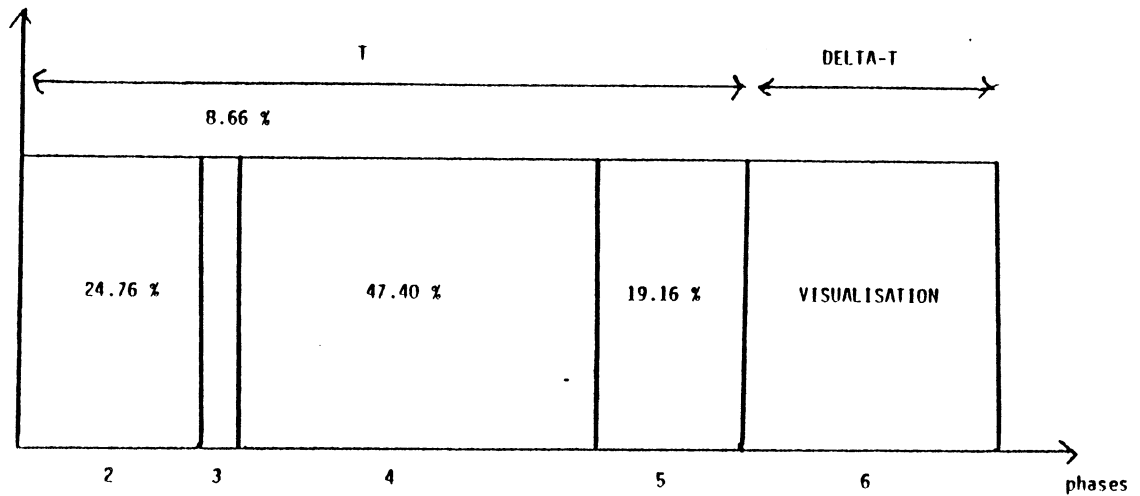


Figure No IV.25

Le temps de traitement lié à la phase de visualisation est variable dépend directement de la complexité des objets-3.D constituant la scène. La configuration "console évoluée" visualise 5,25 K-taches par seconde, soit une tâche visualisée en 620 microsecondes. Dans ce délai de traitement, une partie du temps peut être récupérée par la pré-synthèse, il s'agit des temps de remplissage des mémoires de trames et de profondeur (cas du Z-Buffer). Un recouvrement optimal permettrait le traitement de 60 faces environ en temps réel. Ainsi, des axes de perfectionnement et d'amélioration de

l'algorithmique du processus de synthese, se degagent :

- L'amelioration de la puissance de calcul,
- Le parallelisme au niveau operatoire par l'adjonction de co-processeurs,
- L'amelioration de la repartition des taches associees au processus de synthese

L'objectif est de reduire l'intervalle de temps necessaire au processus en profitant du parallelisme pre-existant entre la pre- et la post-synthese et en mettant en oeuvre un parallelisme intrinseque au niveau de la pre-synthese.



CONCLUSION :

L'architecture du pre-synthetiseur 3.D a base de processeur 16/32 bits a montre ses limites, les performances obtenues sont insuffisantes. Cette etude confirme donc que le concept de pre-synthese evoluee implique la mise en oeuvre d'architectures plus completes que celle proposee dans ce chapitre. Le parallelisme semble etre la seule solution garantissant une reduction des temps de calcul. Ce parallelisme peut intervenir a deux niveaux :

- Au niveau operatoire : le processeur central est dote de co-processeurs ou d'operateurs de calcul,
- Au niveau processus de synthese : le processeur central est banalise au sein d'une architecture multiprocesseurs. ( CIG 84 )

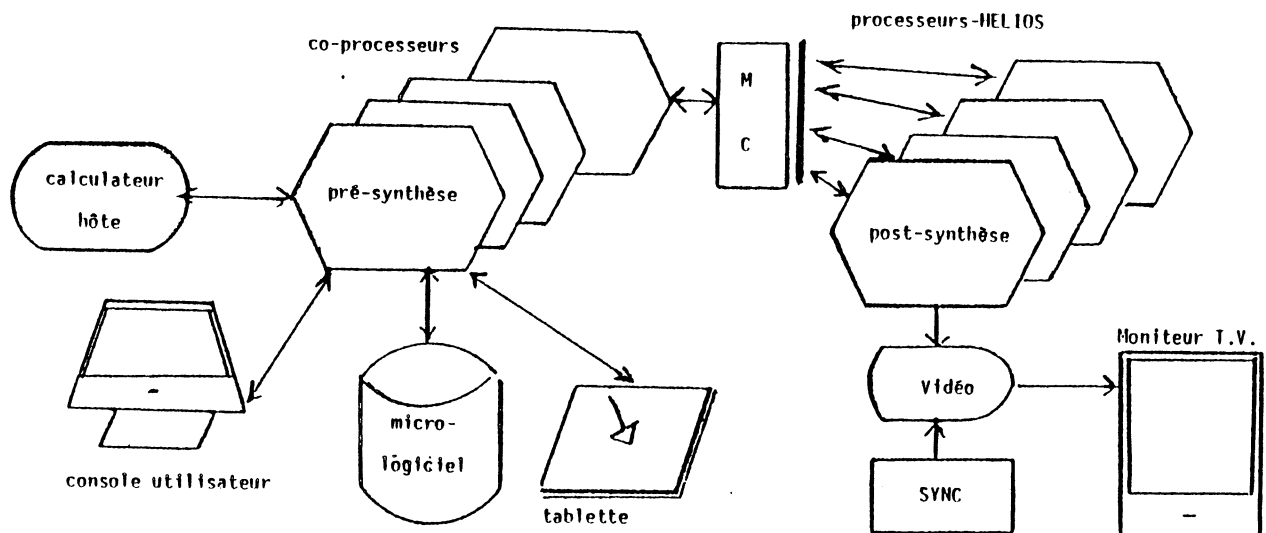


Figure No IV.26

La pre-synthese evoluee est donc basee sur des architectures plus complexes dont le degre de parallelisme est a determiner. le chapitre V propose des solutions dans les deux cas cites precedemment.

CHAPITRE V

EVALUATIONS D'ARCHITECTURES PARALLELES :

REFERENCES BIBLIOGRAPHIQUES :

( AIF 84 ), ( AMD 84 ), ( BGR 79 ), ( CHM 84 ),  
( CHU 84 ), ( ELE 84 ), ( GRA 80 ), ( GRO 85 ),  
( HUI 84 ), ( IKE 84 ), ( INT 80 ), ( LUC 77 ),  
( MGE 84 ), ( MOK 84 ), ( MYE 84 ), ( NEB 77 ),  
( NES 79 ), ( RAG 83 ), ( SMR 85 ), ( WES 81 ).

## V.1 MULTI-PROCESSEUR 3.D CENTRALISE :

Il s'agit de proposer des architectures de pre-synthese susceptibles d'ameliorer les performances de calcul concernant un processus de visualisation donne. La premiere etape consiste a doter le processeur principal d'un parallelisme au niveau operatoire. Pour le processus de synthese tri-dimensionnel, le chapitre precedent a mis en evidence l'importance des calculs arithmetiques. La premiere solution consiste a realiser des co-processeurs de calcul dans des technologies differentes, l'objectif etant d'obtenir des performances meilleures.

Les operations de multiplication et de division manipulant des operandes entieres signes ou non-signes, sont rarement disponibles sur tous les microprocesseurs 8 et 16 bits. Les performances obtenues sur les microprocesseurs 16 bits disposant de ces deux operations au niveau du jeu d'instruction, sont mediocres. (cf chapitre IV) A titre indicatif, une tranche-AMD.2901 dont l'horloge de base est a 10 MHz, effectuee a l'aide de 4 tranches, une multiplication microprogrammee en 5 a 7 microsecondes. La multiplication cablee a l'aide de circuits TTL (LS,S) se fait en 2 a 3 microsecondes pour une horloge de base a 5-8 MHz, en 1 a 1,5 microsecondes pour une horloge de base a 10-15 MHz. (Les algorithmes utilises sont bases sur des operations d'addition et de decalage des operandes codes sur 16 bits)

Pour une division il faut quasiment doubler les temps de calcul. L'insuffisance de ces performances apparait pour des applications ou les contraintes de temps-reel sont importantes et ou le volume de calcul est non negligeable, par exemple 1000 multiplications 16 X 16 bits effectuees par un MC.68000 a 8 MHz necessitent un delai de calcul de 8 a 10 millisecondes. Une realisation en logique cablee reduit de 75 % le temps de calcul d'une multiplication (idem pour la division).

Dans cette etude, on decrira rapidement l'impact des co-processeurs, puis on s'interessera d'une part aux multiplieurs/diviseurs rapides 16 X 16 bits paralleles avec accumulateurs dotes des possibilites d'accumulation de resultats ou de produits intermediaires. Cette caracteristique est essentielle dans la mise en oeuvre d'operateur de calcul matriciel. D'autre part on etudiera les multiplieurs/diviseurs rapides sans accumulateurs dont les temps de calcul sont legerement inferieurs aux precedents.

## V.1.1 CO-PROCESSEURS ARITHMETIQUES :

L'interet du co-processeur est de pouvoir travailler sur des operandes en virgule flottante jusqu'a la phase de transformation-ecran et ceci pour obtenir une meilleure precision de calcul. De plus le co-processeur permettrait de calculer la matrice de transformation localement a l'aide des fonctions trigonometriques dans le cas de rotations.

- L'AMD.9511 requiert un minimum de trois acces pour effectuer une operation flottante.

La tendance est de mettre en oeuvre deux composants de la meme famille qui sont le processeur central et le co-processeur flottant. L'architecture ainsi obtenue permettrait de disposer au niveau du processeur central d'un jeu d'instructions en arithmetique flottante. Cette amelioration permet de simplifier les echanges entre les deux unites. Les exemples les plus courants sont :

- MC.68000 et MC.68881,
- Intel.80286 et Intel.80287,
- NS.16000 et NS.16081.

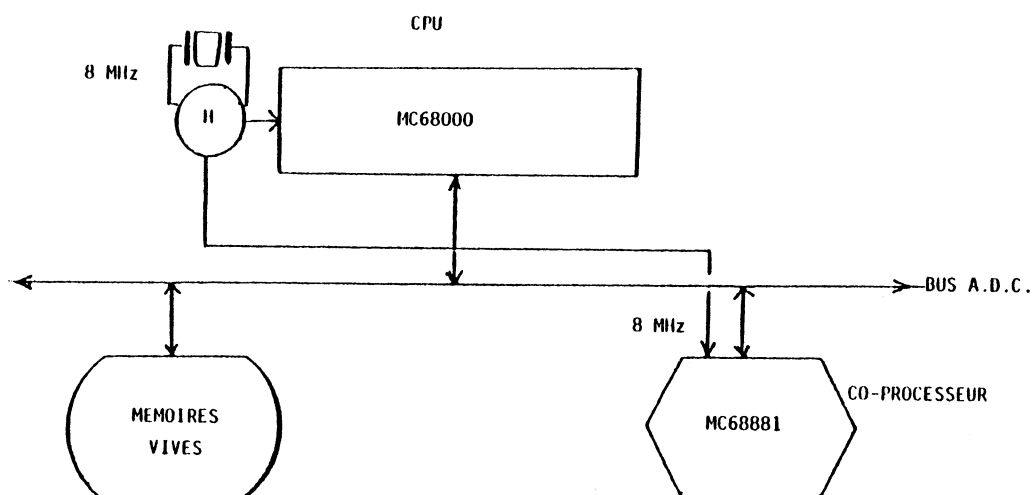


Figure No V.1

Cette solution ne permet pas de resorber le temps de calcul necessaire a la phase de transformation matricielle, excepte pour certains co-processeurs qui effectuent une multiplication ou une division flottante plus rapidement que le CPU travaillant sur des operandes de type entier. De plus cette solution est assez couteuse, le prix d'un co-processeur se situant aux alentours de 2000 a 3000 francs hors taxes. Dans une configuration evoluee de type "station de travail", l'option du co-processeur flottant est systematiquement proposee au niveau de la partie calculeur. (cf Chapitre VI) Le co-processeur offre des operations arithmetiques diverses sur des reels en virgule flottante. Les criteres de comparaison sont :

- Le temps de calcul de la multiplication et de la division flottantes,
- L'horloge de fonctionnement du co-processeur.

Ce tableau recapitule les principales performances d'un ensemble de co-processeurs flottants.

TABLEAU RECAPITULATIF :

CO-PROCESSEUR	HORLOGE	MUL	DIV	DONNEES
AMD.9511	4 MHz	45	45	16 / 32 bits
Intel.8087	8 MHz	14	26	32 / 64 bits
Intel.86287	8 MHz	20	40	32 / 64 bits
NS.16000	10 MHz	7	13	32 / 64 bits
Z.8070	10 MHz	5	6	32 / 64 / 80 bits

Les donnees correspondent a la taille des operandes acceptees par le co-processeur. Certains co-processeurs presentent des performances comparables a celles des processeurs de base pour les operations de multiplication et de division mais en virgule flottante. Seuls les co-processeurs Z.8070 et NS.16081 ont des temps de traitement reduits de moitie par rapport au processeur central, sans compter le parallelisme possible. Cependant il faut tenir compte des delais supplementaires lies a la gestion d'un co-processeur en particulier l'initialisation et les primitives d'utilisation des fonctions de calcul.

#### V.1.2 MULTIPLIEURS / DIVISEURS RAPIDES :

##### SANS ACCUMULATEURS PARALLELES :

Ce tableau recapitulatif permet de presenter un ensemble de multiplieurs rapides monolythiques ainsi que leurs performances pour les operations de multiplication :

FAMILLE -----	TEMPS DE REPONSE -----	TYPE D'OPERANDE -----
AMD.29516/517	45 a 65 nanosecondes	16 X 16 => 32 bits
ADSP.1016	75 a 90 nanosecondes	16 X 16 => 32 bits
MMI 745556	75 nanosecondes	16 X 16 => 32 bits
TI.74.ALS.1616	55 nanosecondes	16 X 16 => 32 bits
TRW-MPY.016	45 nanosecondes	16 X 16 => 32 bits
WEITEK.2516.D	45 nanosecondes	16 X 16 => 32 bits

Ce sont les multiplieurs/diviseurs les plus rapides. Pour chaque famille le temps de calcul varie entre 45 et 220 nanosecondes maximum. Le temps moyen de calcul d'une multiplication 16 X 16 => 32 bits est de l'ordre de 100 nanosecondes.

#### AVEC ACCUMULATEURS PARALLELES :

Ce tableau recapitulatif permet de presenter un ensemble de multiplieurs rapides avec possibilite d'accumulation, ainsi que leurs performances respectives.

FAMILLE -----	TEMPS DE REPONSE -----	TYPE D'OPERANDE -----
AMD.29510	80-100 nanosecondes	16 * 16 => 32 bits
ADSP.1010	95 nanosecondes	16 X 16 => 32 bits
EFCIS.29747	200 nanosecondes	16 X 16 => 32 bits
TRW-TDC.1043	100 nanosecondes	16 X 16 => 32 bits
WEITEK.2010	65-150 nanosecondes	16 X 16 => 32 bits

De meme pour chaque famille le temps de calcul varie de 65 nanosecondes a 240 nanosecondes maximum. Le temps moyen de calcul d'une multiplication 16 X 16 => 32 bits est de l'ordre de 140 nanosecondes.

L'operation de multiplication consiste a realiser deux acces a la memoire pour fournir les operandes et un acces a la memoire pour recuperer le resultat, ce dernier etant immediatement disponible car pendant la recherche de l'instruction de lecture du port "R", la multiplication s'effectue entre les operandes A et B.

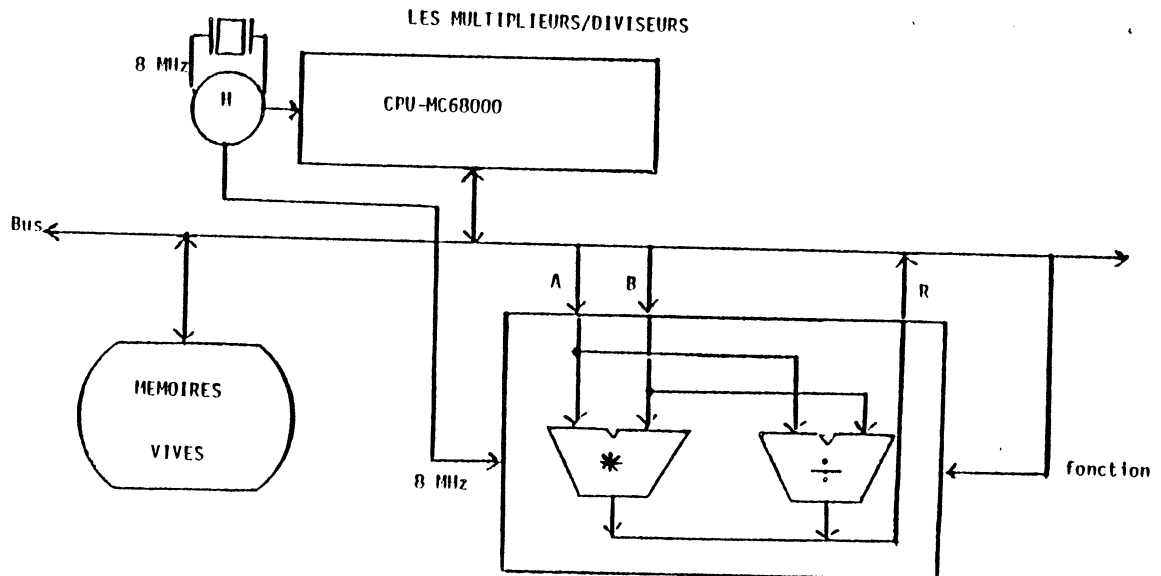


Figure No V.2

Une multiplication se fera donc a l'aide de trois instructions d'accès au peripherique de calcul, soit environ 1,5 a 3 microsecondes. Dans ce cas les registres du processeur sont prealablement charges. (Cette plage de temps de reponse est fonction du delai d'acquiescement du cycle du processeur MC.68000, DTACK). Dans cette configuration de calcul, la transformation matricielle se fera en :

19,75 microsecondes a 33,25 microsecondes pour un sommet de l'espace.

Le gain est de 56 % a 75 % par rapport a la transformation microprogramme.

Ce gain est assez rentable mais est-il interessant d'utiliser ces multiplieurs rapides dont la mise en oeuvre est complexe pour simplement se substituer aux instructions de multiplication et de division du processeur MC.68000 ? Il apparait que la mise en oeuvre de ce type de multiplieurs rapides necessite l'adjonction d'une dizaine de boitiers T.T.L. supplementaires. Dans ce cas il serait plus avantageux de developper des operateurs plus complets. ( WES 81 )

## V.2 OPERATEURS A BASE DE MULTIPLIEURS :

Une premiere amelioration est de realiser un operateur cable qui effectue la transformation matricielle d'un sommet. Il faut definir les operations que l'on veut sous-traiter.

Deux solutions seront proposees. La premiere est intermediaire, la seconde est totale.

## V.2.1 OPERATEUR MATRICIEL PARTIEL : O.M.P

Cet operateur calcule le produit d'une ligne "i" de la matrice par les coordonnees (X, Y, Z) du sommet courant a transformer. Il s'agit d'un operateur cable intermediaire qui necessite :

- Trois multiplieurs rapides 16 X 16 => 32 bits,
- Un additionneur a quatre operandes, le resultat etant sur 32 bits,
- Trois registres d'entrees : X, Y, Z,
- Un registre de sortie : R-XYZ,
- Un registre de commande : CMD de lancement de l'O.M.P (optionnel).

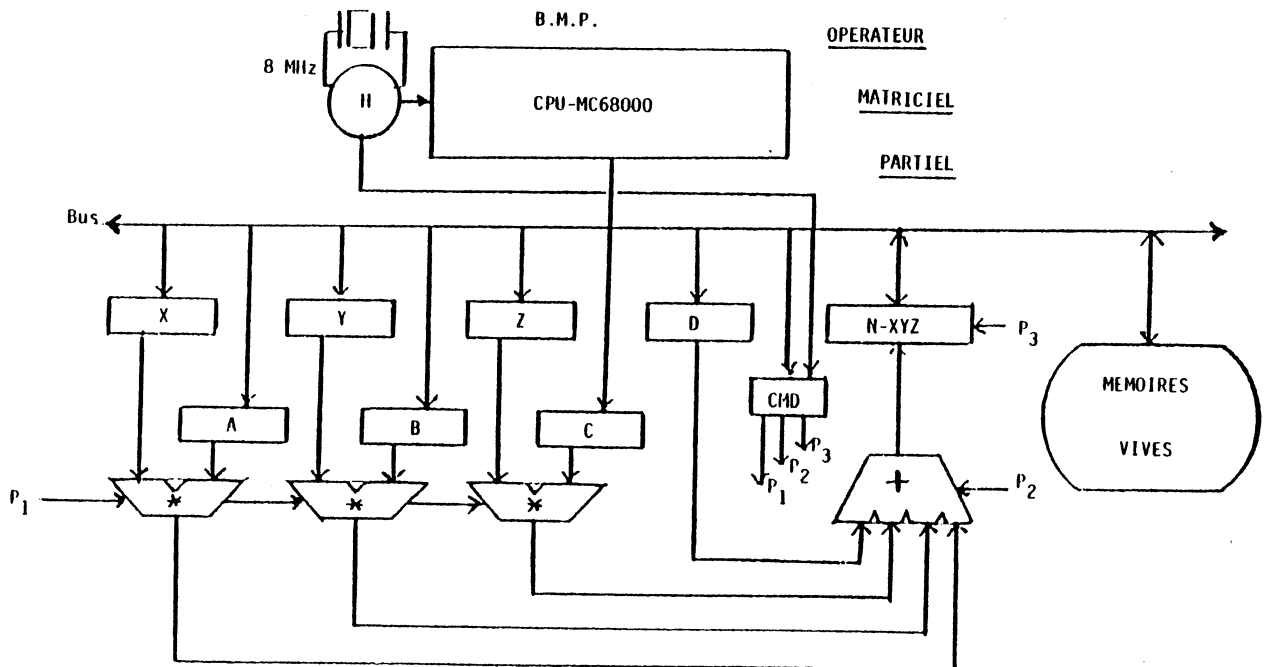


Figure No V.3



Les registres X, Y, Z sont sur 16 bits. Le registre R-XYZ est sur 32 bits.

La transformation matricielle necessitera :

- Trois instructions d'accès pour stocker le sommet (X, Y, Z),
- Quatre instructions d'accès pour stocker la ligne courante de la matrice,
- Un accès pour lancer l'opération : CMD (optionnel),
- Un accès pour récupérer le résultat partiel : R-XYZ.

Au total l'O.M.P requiert neuf instructions d'accès à la mémoire pour calculer la coordonnée transformée d'un sommet, soit X, ou Y, ou Z. Dans cette évaluation, on suppose que les registres internes du processeur central, sont préalablement initialisés. Dans le cas contraire il faut pratiquement doubler le nombre d'accès. La transformation matricielle d'un sommet requiert donc  $9 \times 3 = 27$  accès. Le temps de réponse est de 13,5 microsecondes à 27 microsecondes.

On obtient une amélioration de 18 à 31 % par rapport à la solution à l'aide d'un seul multiplieur rapide. La mise en œuvre de l'O.M.P nécessite l'utilisation de trois multiplieurs et d'un additionneur. Une amélioration du coût d'un tel opérateur consiste en l'utilisation d'un seul multiplieur doté d'un séquencement interne. Dans ce cas, les performances se dégradent légèrement, car une mémoire locale serait nécessaire ainsi que trois cycles internes pour les multiplications. Une autre solution consiste à réaliser l'addition finale au sein du processeur central afin de simplifier l'architecture de l'opérateur. ( WES 81 )

#### V.2.2 OPERATEUR MATRICIEL COMPLET : O.M.C

Cet opérateur calcule la transformation matricielle d'un sommet. Il s'agit d'un opérateur cable spécifique. Il est constitué d'une mémoire locale de 16 mots ou est pré-stockée la matrice de transformation de l'espace. (13 mots de 16 bits et 3 mots de 32 bits pour la translation de l'espace) Un séquenceur interne effectue la fonction d'un opérateur matriciel partiel en fournissant consécutivement la nouvelle ligne de la matrice. Il nécessite :

- Trois multiplieurs rapides  $16 \times 16 \Rightarrow 32$  bits,
- Un additionneur à quatre opérands, le résultat étant sur 32 bits,
- Trois registres d'entrée : X, Y, Z,

- Trois registres de sortie : NX, NY, NZ,
- Un registre de commande : CMD de lancement de l'O.M.C.
- Des systemes de multiplexage.

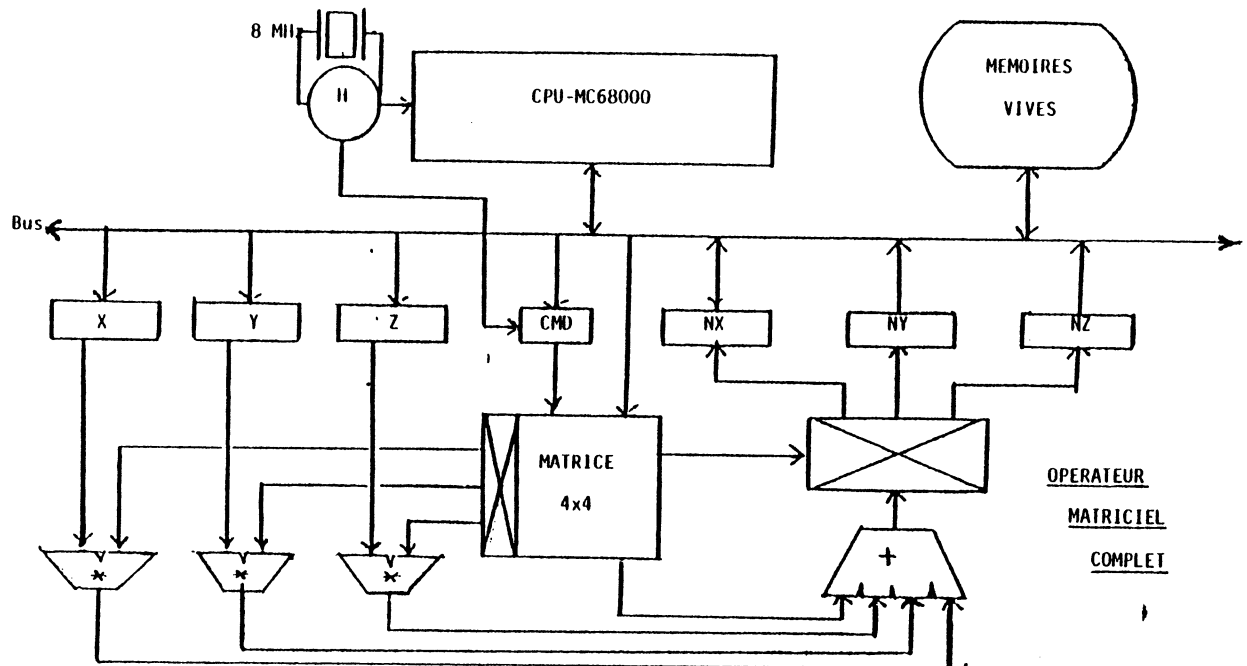


Figure No V.4

La transformation matricielle necessitera :

- Trois instructions d'accès pour stocker le sommet : X, Y, Z
- Un délai de calcul des trois lignes significatives de la matrice,
- Trois instructions d'accès pour récupérer le sommet transformé : NX, NY, NZ.

REMARQUE :

On suppose que la matrice est pré-stockée dans la mémoire locale de l'O.M.C. Les 16 accès à la mémoire d'initialisation de l'O.M.C, ne seront donc pas comptabilisés dans l'évaluation.

L'O.M.C requiert trois cycles supplémentaires pour le séquençage du calcul des trois lignes significatives de la matrice. Un système de multiplexage permettrait à chaque phase de choisir la ligne courante. Un cycle suffit au calcul matriciel correspondant à une ligne et à

l'échantillonnage du résultat. Une multiplication rapide sur 16 bits signée se fait en 100 nanosecondes. La mémoire locale où est stockée la matrice, doit être à accès rapide soit près de 100 nanosecondes.

Son organisation doit permettre un accès à quatre mots mémoires simultanément, soit une ligne de la matrice.

Au total l'O.M.C requiert trois instructions d'accès pour déposer les coordonnées du sommet à transformer et trois instructions d'accès pour récupérer les coordonnées du sommet transformé, plus trois cycles internes environ pour le séquencement du passage à la ligne suivante. La transformation matricielle se fait en 12 accès plus un accès de lancement de l'opération par l'intermédiaire du registre CMD, soit au total 13 accès. Le temps de réponse est de l'ordre de 6,5 à 13 microsecondes. On obtient une amélioration de 52 % par rapport à la solution précédente. Comparativement l'O.M.C est quasiment deux fois plus rapide que l'O.M.P. Cependant la solution de l'O.M.C est certainement plus coûteuse et plus complexe à mettre en œuvre. (les registres du processeur central sont préalablement initialisés) Cet opérateur matriciel peut être utilisé pour toute sorte de transformation du plan ou de l'espace.

### V.2.3 ETUDE DES PERFORMANCES :

La courbe d'évaluation de la transformation matricielle d'un sommet visualise les différentes performances obtenues :

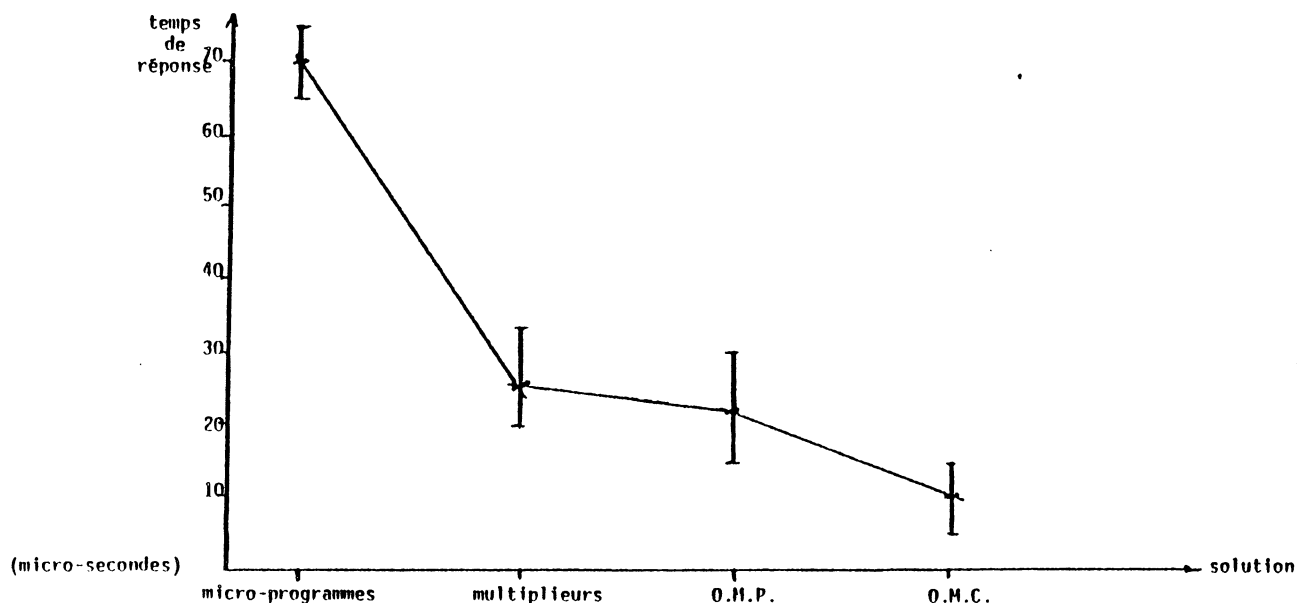


Figure No V.5

Il est evident que la solution microprogramme est inadaptee, l'introduction des multiplieurs rapides introduit une amelioration sensible de 56 a 75 %. La progression est une suite plus lineaire bien que la complexite de l'operateur augmente nettement. L'O.M.P semble la solution la plus appropriee, elle est simple a mettre en oeuvre, peu couteuse, elle apporte une amelioration de l'ordre de 80% par rapport a la solution microprogramme. Son architecture moyennant quelques aménagements permettrait d'effectuer d'autres types de calculs. Ce point sera developpe dans la suite de l'etude.

### V.3 OPERATEURS SPECIFIQUES DE TRANSFORMATION :

#### V.3.1 TRANSFORMATION-ECRAN :

Le traitement realisant le passage du repere-utilisateur dans le repere-ecran constitue une phase de calcul finale. Pour chaque sommet on effectue un calcul sur les coordonnees (X, Y) pour obtenir les coordonnees (Xe, Ye) dans la cloture de l'ecran. Cette phase du processus de visualisation 3.D occupe une proportion relativement importante. On peut dire que 1/5 du temps de calcul est necessaire pour effectuer la tranformation-ecran. Il serait donc judicieux d'analyser le cheminement des calculs inherent a cette phase et d'evaluer les solutions susceptibles de reduire ces temps de reponse. La transformation-ecran est definie par :

$$X_e = ( X_i + b ) * r / ( 2 * b )$$

$$Y_e = ( Y_i + b ) * r / ( 2 * b )$$

(Xi, Yi) sont les coordonnees d'un sommet visible dans le repere-utilisateur.

Il apparait donc que deux multiplications et deux divisions sont necessaires pour realiser la transformation-ecran d'un sommet, cette transformation s'appliquant a une arete visible c'est-a-dire une paire de sommets gauche et droit (Sg, Sd). Dans ce cas, la division par deux se fait par un decalage avec conservation du signe.

#### V.3.1.1 DIFFERENTES PROPOSITIONS : -

##### MULTIPLIEURS / DIVISEURS :

La premiere solution consiste a doter le C.P.U de deux operateurs externes pour realiser ces operations. Cette demarche est semblable a celle de la phase de transformation matricielle. (utilisation du multiplieur/diviseur SN-74516 de Texas Instruments.) Une multiplication necessite trois instructions d'accès, deux ecritures des operandes A et B, une lecture du resultat R avec recouvrement. Une division necessite de la meme facon trois instructions d'accès plus un delai supplementaire de dix cycles par rapport a la multiplication.

Dans ce cas, la transformation-ecran se fera en 21,75 a 27,75 microsecondes. L'amelioration est de l'ordre de 60 a 68 % pour la transformation-ecran d'une arete, par rapport a la solution microprogramme. Ce composant est en fait un co-processeur evolue capable de realiser les deux operations. La mise en oeuvre est donc plus rentable que dans le cadre des transformations matricielles.

#### UTILISATION DE O.T.E :

Cette ultime optimisation permet de sous-traiter completement la transformation-ecran de sorte a n'avoir aucun calcul microprogramme a realiser au niveau du processeur central. Cet operateur de transformation-ecran O.T.E sera parametre par des valeurs caracteristiques fournies par l'utilisateur :

- La taille fenetre :  $b$ ,
- La taille de la cloture d'ecran :  $r$ ,

une mise a jour automatique devra etre mise en oeuvre si ces parametres peuvent evoluer d'une prise de vue a une autre. L'O.T.E est un operateur specifique. Il requiert dans son architecture :

- Deux additionneurs 16 bits,
- Deux multiplieurs  $16 \times 16 \Rightarrow 32$  bits,
- Deux diviseurs  $32 / 16 \Rightarrow 16$  bits,
- Deux registres d'entree  $X_i, Y_i$
- Un registre de commande CMD, (optionnel)
- Deux registres de sortie  $Y_e, Z_e$ ,
- Deux registres d'entree parametres :  $B, R$

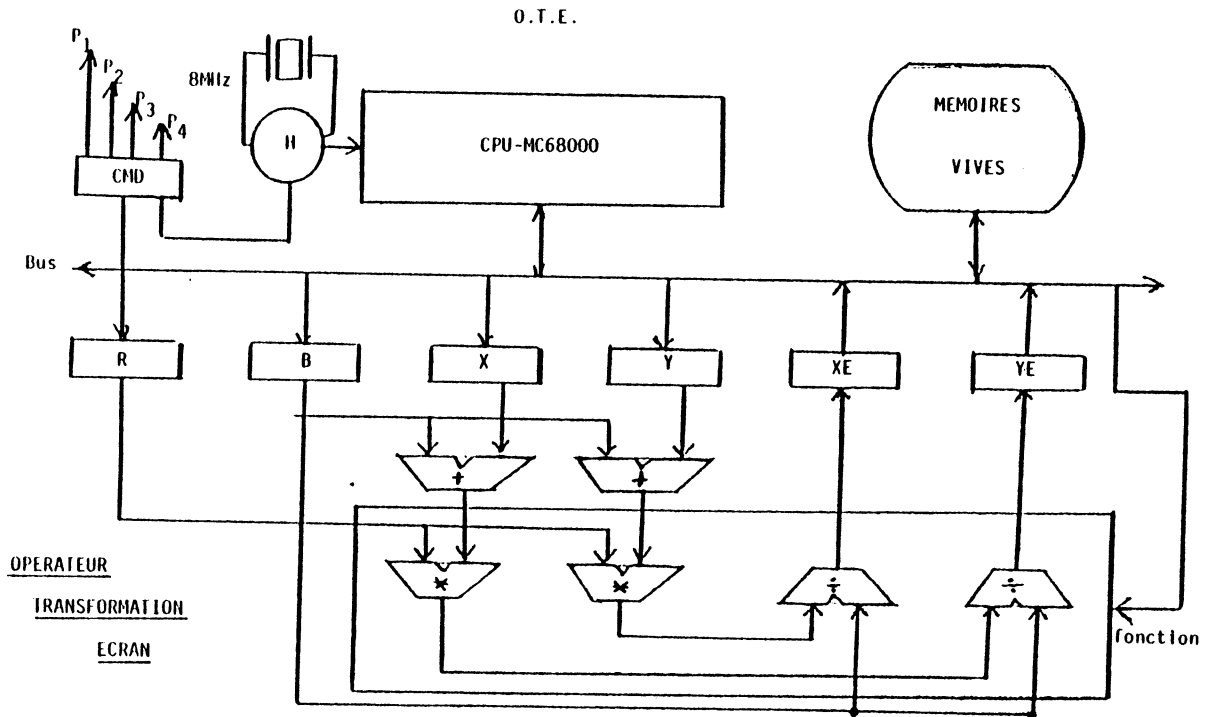


Figure No V.6

Les registres sont sur 16 bits chacun. Il suffit de deux instructions d'accès pour déposer les coordonnées  $X_i, Y_i$  d'un sommet puis d'un accès au registre CMD pour lancer la transformation-ecran d'un sommet puis de deux instructions d'accès pour récupérer les coordonnées-résultat  $X_e$  et  $Y_e$ . (les registres du processeur central sont préalablement initialisés.) Pour une arête visible cette opération doit être effectuée deux fois de suite. Il faut traiter une paire de sommets. On suppose que les registres paramètres sont initialisés avant le lancement du processus. Les accès à B, R ne sont donc pas pris en compte dans cette évaluation

Au total on obtient 5 accès pour un sommet soit 10 accès pour la transformation-ecran d'une arête, le temps de calcul est de 8 à 13 microsecondes. Ce temps réponse introduit une amélioration de :

- 64 à 77 % par rapport à la solution des multiplieurs/diviseurs rapides,
- 85 à 92 % par rapport à la solution microprogrammée.

L'architecture proposée est conçue de telle sorte que les opérations d'addition et de multiplication soient réalisées avec un recouvrement correspondant à la recherche de l'instruction de stockage de la seconde coordonnée ( $Y_i$ ), de même les opérations (+, \*) sont effectuées avec un recouvrement correspondant à la recherche de l'instruction de lecture du premier résultat de la première coordonnée  $X_e$ . Les opérations de division introduisent un délai supplémentaire de 3 microsecondes inhérent aux

composants SN.74516, soit 3 "NOP" au niveau du processeur MC.68000. Cet operateur permet de realiser la transformation-ecran en 8 a 13 microsecondes plus l'operation de decalage a droite avec conservation du signe pour realiser la division par deux. Soit au total 11 a 16 microsecondes pour une arete. L'amelioration obtenue par rapport a la solution microprogrammee est de l'ordre de 77 a 84 %.

## REMARQUE :

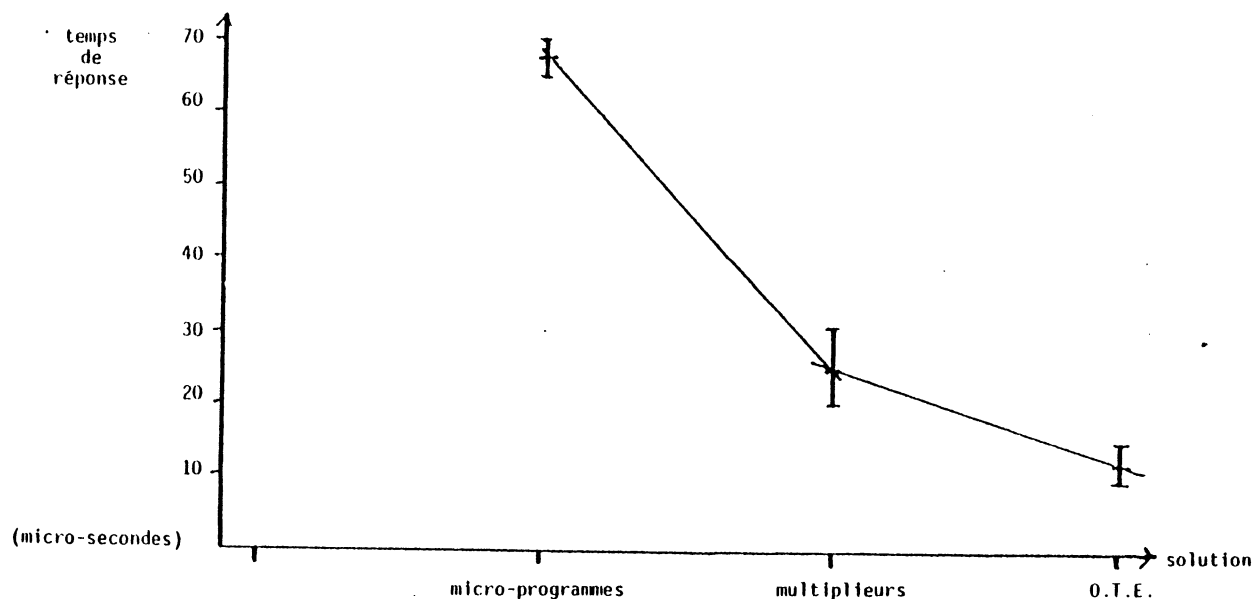
L'architecture de l'O.T.E peut etre simplifiee. En effet l'utilisation des multiplieurs/diviseurs SN.74516 permet de conserver des resultats partiels dans des accumulateurs et de commander une operation differente : La division avec un seul operande et l'accumulateur. Dans ce cas les diviseurs disparaissent de l'architecture. De plus cet operateur peut etre re-utilise pour le calcul des coordonnees homogenes. Ce point est explicite dans la suite de l'etude.

## V.3.1.2 ETUDE DES PERFORMANCES : -

Une recapitulation des performances des differentes solutions pour la phase de transformation-ecran traitant une arete fournit les resultats suivants :

TYPE DE SOLUTION -----	TEMPS DE REPONSE -----	PROPORTION -----
solution microprogrammee	69,65 microsecondes	15,31 %
solution avec multiplieurs	21,75 a 27,75 microsecondes	6,71 %
solution avec l'O.T.E :	11 a 16 microsecondes	4 %

La solution des multiplieurs/diviseurs rapides et de l'O.T.E reduisent les proportions de calcul de cette phase avec une difference de 2,71 % en faveur de l'O.T.E.



les criteres de choix d'une solution sont la complexite de la mise en oeuvre et Le cout de realisation, Pour une contrainte de temps-reel de 40 millisecondes, l'O.T.E traite les 1000 faces en 16 millisecondes, par contre les multiplieurs/diviseurs rapides traitent les 1000 faces en 27 millisecondes. Le choix doit aussi tenir compte de l'architecture globale du pre-synthetiseur. La solution des multiplieurs/diviseurs rapides est commune a trois phases du processus de visualisation 3.D, sa mise en oeuvre est plus rentable. Ces trois phases sont la transformation-ecran, la transformation matricielle et la projection en perspective.

### V.3.2 PROJECTION EN PERSPECTIVE :

Cette phase de calcul est decomposee en deux etapes. Pour un sommet Si (Xi, Yi, Zi) on calcule d'abord :

$$\omega_{i} = a - Z_{i}$$

a : distance point de visee - point de vue (observateur),

Au niveau de l'algorithme de decoupage, "a" etant positif, on ne calcule au depart que  $\omega_{i}$ , puis si l'arete est visible on effectue la projection en perspective :



$$\omega_{i0} = \omega_i / a$$

$$S(X_i, Y_i) \rightarrow S_p(X_{pi}, Y_{pi}),$$

$$X_{pi} = X_i / \omega_{i0}$$

$$Y_{pi} = Y_i / \omega_{i0}$$

La multiplication est effectuee en priorite pour minimiser la perte de precision liee a la division entiere. L'evaluation du chapitre precedent indique que la proportion moyenne de calcul de la phase de projection est 7 %. La projection est une phase qui s'applique a une arete visible, le calcul se fait sur une paire de sommets.

#### V.3.2.1 DIFFERENTES PROPOSITIONS : -

##### MULTIPLIEURS / DIVISEURS RAPIDES :

L'architecture proposee a pour but de sous-traiter les operations penalisantes de la projection a l'aide de composants plus performants. Il s'agit bien sur des operations de multiplication et de division. Le multiplieur/diviseur SN.74516 peut effectuer la soustraction  $\omega_i = a - Z_i$ . Cette operation n'entre pas en jeu dans l'evaluation mais sa mise en oeuvre permettrait de reduire le temps de calcul global de la phase de projection. Le decoupage des acces indique une amelioration de l'ordre de 50 %. Le temps de calcul de la projection est divisee par deux grace a la solution des multiplieurs /diviseurs rapides.

##### UTILISATION DE L'O.T.E :

L'O.T.E effectue le calcul intermediaire  $X_e = (X_i + b) * r / b$  et  $Y_e = (Y_i + b) * r / b$ , la division par deux est realisee par le processeur central par decalage. Le calcul de la projection est defini par :

$$X_{pi} = (a * X_i) / (a - Z_i)$$

$$Y_{pi} = (a * Y_i) / (a - Z_i)$$

a : distance point de vue - point de visee

Posons :  $B = a - Z_i$  et  $R = a$

Les registres de l'O.T.E B,R seront correctement initialises avant de lancer l'operation de projection. La valeur  $a - Z_i$  est calculee precedemment par le C.P.U lors d'une phase intermediaire. Au niveau du fonctionnement de l'O.T.E, il suffit de substituer a l'addition  $X_i + B$  et  $Y_i + B$  une non-operation. Cette option peut etre indiquee dans le registre de commande de l'O.T.E. On obtient alors :

$$X_{pi} = X_e = ( X_i * a ) / ( a - Z_i )$$

$$Y_{pi} = Y_e = ( Y_i * a ) / ( a - Z_i )$$

Dans ce cas la phase de projection s'effectue avec un temps de reponse identique a celui de la phase de transformation-ecran. Les evaluations sont pour une arete 11 a 16 microsecondes et pour une face 38 a 88 microsecondes. Un temps supplementaire de calcul de l'expression  $a - Z_i$  est a prendre en compte car cette operation est realisee par le processeur central. Soit d'apres l'evaluation un delai supplementaire de 7,23 microsecondes pour un sommet. L'evaluation de la phase de projection pour un sommet de 12,73 a 15,21 microsecondes.

Ce temps de reponse est du meme ordre de grandeur que celui obtenu avec la solution des multiplieurs/diviseurs rapides pour la valeur maximale. Pour la valeur minimale on introduit une amelioration de l'ordre de 19,55 %. Pres d'un cinquieme du temps de reponse est donc epargne, ceci est non-negligeable car pour le traitement de 1000 faces, on evite un temps de calcul supplementaire de 7,6 millisecondes.

#### REMARQUE :

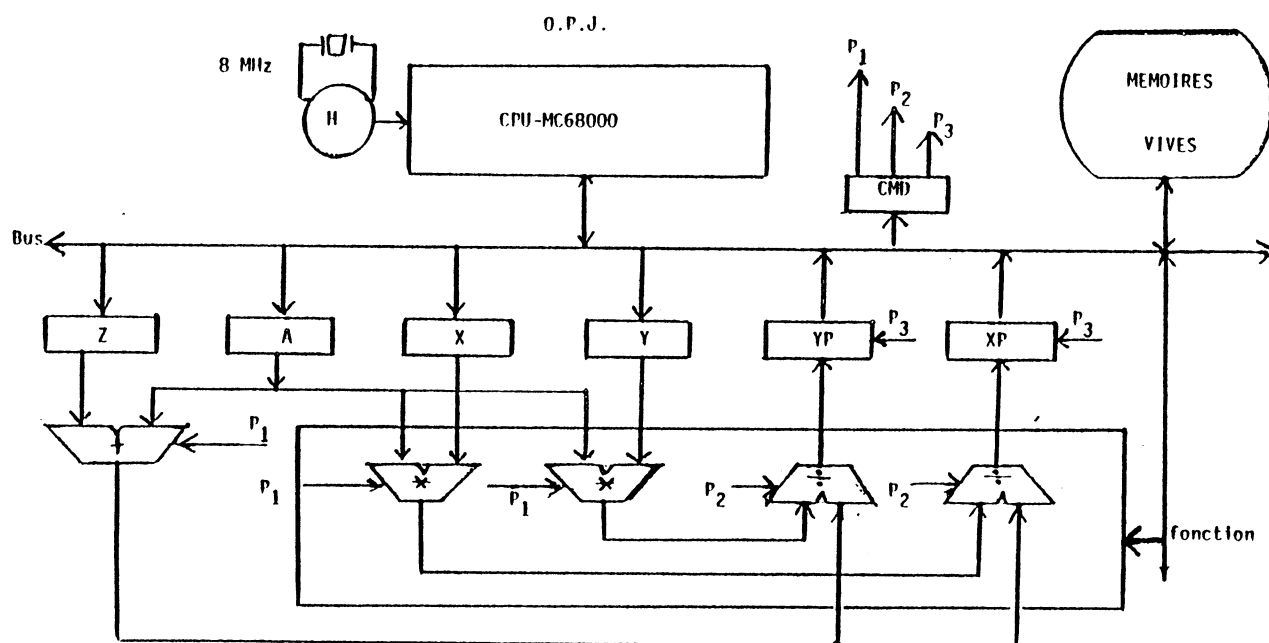
Une fois la phase de projection terminee, les registres B,R sont correctement initialises pour que la phase de transformation-ecran puisse s'effectuer correctement. L'operateur des additions est valide a partir du registre CMD.

#### OPERATEUR CABLE DE PROJECTION : O.P.J

Il s'agit de realiser l'operation de projection a l'aide d'un operateur specifique, aucun calcul supplementaire n'etant necessaire au niveau du processeur central. L'operateur de projection O.P.J sera parametre par la valeur de  $a$  qui est la distance entre le point de visee et le point de vue. La proposition d'une architecture de cet operateur requiert :

- Trois diviseurs rapides 32 / 16 bits,

- Un additionneur a deux entrees, le resultat etant sur 32 bits,
- Trois registres d'entree : X, Y, Z,
- Un registre parametre : A
- Un registre de commande : CMD (optionnel)
- Deux registres de sortie : XP, YP car ZP = 0



OPERATEUR DE PROJECTION

Figure No V.7

Il suffit de stocker en trois acces  $X_i$ ,  $Y_i$ ,  $Z_i$  pour un sommet  $S_i$ . On suppose que le registre parametre "A" est pre-initialise. Puis d'un acces est necessaire pour lancer l'operation de projection. Enfin deux acces finaux sont sequences par le processeur central pour recuperer les valeurs XP, YP du sommet projete. Au total, on realise une projection en cinq instructions d'accès pour un sommet. Si on ajoute la complementation a deux du registre  $X_i$  necessaire a la coherence des calculs, on obtient le temps de reponse moyen de 8 a 11 microsecondes pour un sommet. Soit une amelioration de :

- 82 % par rapport a la solution microprogrammee,
- 64 % par rapport aux multiplieurs/diviseurs rapides,
- 60 % par rapport a l'operateur O.T.E

Dans cette architecture le recouvrement est total, la lecture des registres XP et YP a lieu une fois le calcul effectue, pendant la recherche des intructions d'accès aux registres resultats de l'operateur.

#### V.3.2.2 ETUDE DES PERFORMANCES : -

On obtient pour la projection en perspective, trois types de solutions, resumees dans le tableau recapitulatif suivant :

#### PROJECTION EN PERSPECTIVE DE L'ARETE :

TYPE DE SOLUTION	TEMPS DE REPONSE
La solution microprogramme	37,23 microsecondes
La solution multiplieurs / diviseurs rapides	22,37 microsecondes,
La solution de l'operateur cable O.T.E	20,73 microsecondes,
La solution de l'operateur cable O.P.J	10 microsecondes.

Le gain est donc appreciable, les nouvelles solutions permettent de reduire quasiment de moitie la proportion de temps de calcul liee a la phase de projection. L'amelioration apportee par l'utilisation de l'O.T.E n'est pas tres importante par rapport aux multiplieurs/diviseurs rapides, mais l'interet est de pouvoir "rentabiliser" l'architecture mise en oeuvre en la sollicitant lors de deux phases de calcul.

#### V.3.3 PHASE DE DECOUPAGE TRI-DIMENSIONNEL :

Cette phase se decompose en deux etapes. L'etape de recherche d'une intersection de l'arete a couper avec le plan  $z = a$  Dans ce cas seule une portion de l'arete est a l'interieur du cone de visualisation 3.D. L'etape de recherche des intersections de l'arete a couper, avec les bords de la fenetre de visualisation de taille b. Dans ce cas une fois la projection effectuee, on recherche la portion de l'arete visible par rapport a la fenetre de visualisation. Un parallelisme peut etre mis en oeuvre entre ces deux etapes, le decoupage aux clotures de l'ecran etant a la charge de la post-synthese.

Pour la premiere etape on a  $\omega_i = a - Z_i$ , pour un sommet  $S_i(X_i, Y_i, Z_i)$   $\omega_i$  permet de situer la position d'un sommet par rapport au plan  $z = a$ . Pour une arete on peut detecter l'intersection avec le plan  $z = a$  en fonction de  $\omega_i$  et  $\omega_j$ ,  $S_i, S_j$  etant les extremités de l'arete. Pour la seconde etape, une fois la projection des deux sommets  $S_i, S_j$  calculee, on determine a l'aide de tests simples si les sommets sont :

- A l'interieur du cone de visualisation,
- A l'interieur de la fenetre de visualisation,
- Si une intersection avec un bord est a calculer.

L'evaluation a permis d'identifier les sequences de l'algorithme les plus sollicitees en temps de calcul. L'evaluation des differentes actions associees au module de decoupage 3:D est donnee a titre indicatif pour une methode donnee. (cf chapitre IV) La proportion moyenne de temps de calcul de cette phase est de l'ordre de 47,4 %. Une decomposition de l'algorithme en actions a permis d'affiner les evaluations d'un ensemble d'utilitaires traitant une arete. Ces resultats ont ete obtenus pour differentes tailles de l'arete (multiple de la largeur de la fenetre) et pour differentes positions (horizontal, verticale et diagonale).

FONCTION -----	TEMPS DE REPONSE -----
Tests de l'arete "hors de la fenetre"	50 a 60 microsecondes
Tests du sommet "interieur de la fenetre"	35 microsecondes
Calcul du milieu de l'arete	16 microsecondes
Calcul de la convergence	23 microsecondes
Recherche dichotomique	10 a 20 microsecondes

Si la post-synthese assure un decoupage dans le plan, le temps de reponse peut etre reduit de pres de 38 %. C'est le cas du terminal HELIOS qui assure un decoupage aux frontieres de la cloture d'ecran. Certaines solutions microprogrammees consistent a repartir les operations de decoupage entre plusieurs processeurs, chaque processeur effectue le traitement suivant deux plans de l'espace. C'est le cas du "geometry engine" et de la cellule "CSI". (cf chapitre VI)

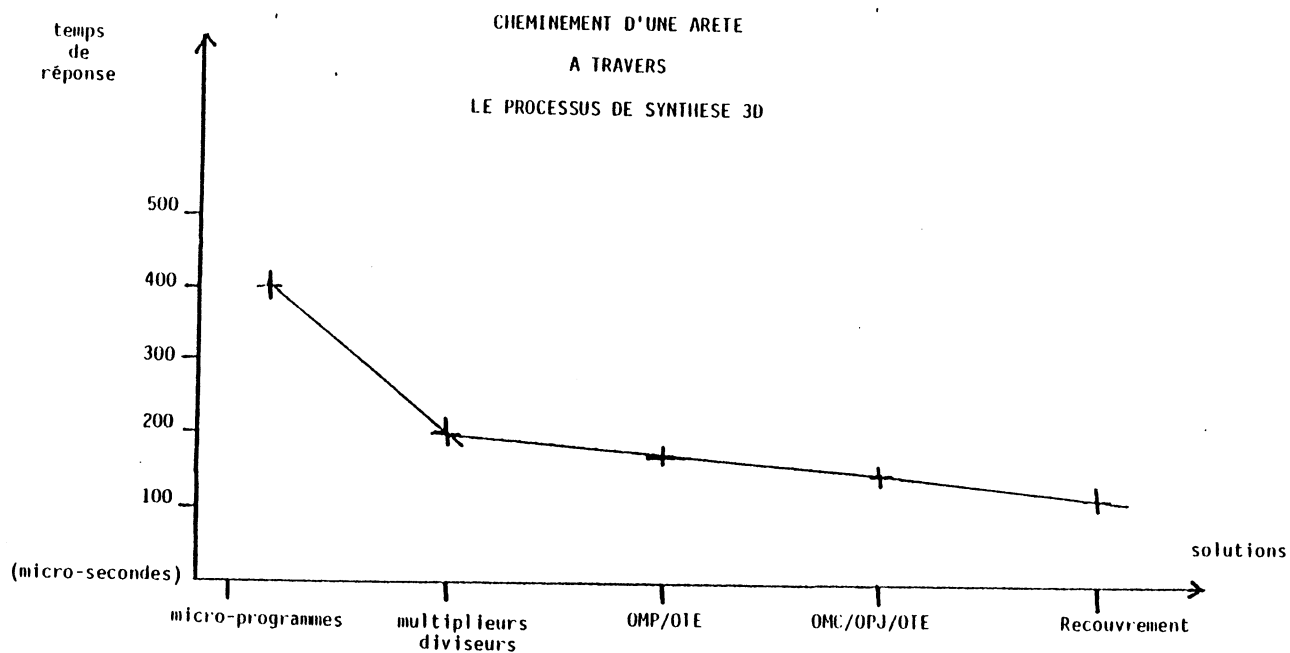
## V.4 PROCESSUS DE VISUALISATION 3.D AMELIORE :

Les differentes propositions d'architectures ont permis d'amoindrir les proportions de calcul des differentes phases du processus 3.D. La solution microprogrammee realise le cheminement d'une arete a travers le processus 3.D en 435 microsecondes, la visualisation etant non-evaluee. Pour la solution des multiplieurs/diviseurs rapides le cheminement se fait en 200 a 255,37 microsecondes. Pour la solution des operateurs cables intermediaires, l'utilisation de l'O.M.P et de l'O.T.E uniquement indique que le cheminement se fait en 175 a 229,2 microsecondes. Pour la solution des operateurs cables specifiques, l'utilisation de l'O.M.C, de l'O.P.J et de l'O.T.E, le cheminement se fait en 143 a 198 microsecondes.

## REMARQUE :

Ces evaluations sont faites sans tenir compte du parallelisme possible entre les differents operateurs cables qui presentent cet avantage par rapport aux multiplieurs/diviseurs rapides. Cependant le processeur central peut etre dote de plusieurs unites de multiplieurs/diviseurs rapides.

Ainsi la transformation-ecran d'une arete visible peut s'effectuer en meme temps que la transformation matricielle de l'arete suivante. Ce recouvrement est parfait car l'O.M.C et l'O.T.E ont les memes temps de reponse environ 13,5 microsecondes. Dans ce cas on peut re-evaluer la troisieme solution, globalement le cheminement se fait en 129 a 184,5 microsecondes. De la meme facon on peut projeter l'arete a couper et en meme temps realiser la transformation matricielle de l'arete suivante. Le recouvrement est possible, le temps de reponse de l'O.P.J est superieur a celui de l'O.M.C (idem pour l'O.T.E). On peut re-evaluer la troisieme solution, le cheminement se fait de 116 a 171 microsecondes.



Il faut préciser que ces méthodes de recouvrement nécessitent des mécanismes d'accès parallèles. Le processeur central alimente par le biais de mémoires privées les différents opérateurs. La phase d'élimination des parties cachées reste en partie à la charge du processeur central. (tri des faces) La post-synthèse se charge de l'exécution d'un Z-Buffer en interprétant les informations contenues dans la mémoire de profondeur. L'opérateur de visualisation peut être un processeur spécifique séquencant les accès à la post-synthèse via le module de communication. Il serait éventuellement alimenté directement par l'opérateur O.T.E. Cette technique de recouvrement est illustrée par l'architecture proposée dans la figure suivante :

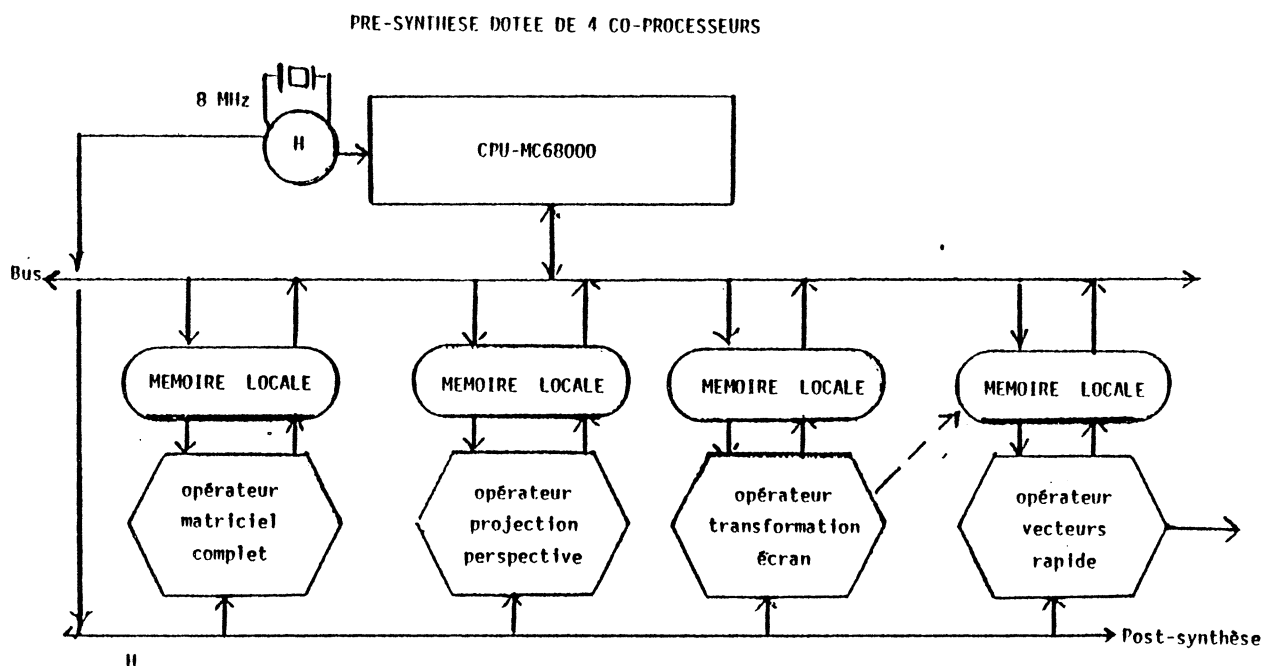


Figure No V.8

Les files de communication sont des zones mémoire partagées uniquement par le processeur central et l'opérateur concerné. La synchronisation de ce "pipeline" est à la charge du processeur central.

Il serait assez délicat d'évaluer les temps de réponse d'une telle architecture, il faut déjà préciser que les files de communication doivent être réalisées à l'aide de mémoires à accès rapides et de petites capacités. Cependant le recouvrement calculé dans cette évaluation considère que le processeur central accède à un opérateur par l'intermédiaire d'une page mémoire privée, donc indépendamment des autres opérateurs. La fin de l'exécution d'un opérateur peut être signalée au processeur central par un mécanisme d'interruption. (cf chapitre IV auto-vectorisation)

La solution microprogrammée ne pourrait traiter que 28 faces en temps réel, La solution utilisant les opérateurs spécifiques avec recouvrement peut traiter près d'une centaine de faces. Il semble que malgré les

ameliorations apportees pour resorber le temps de calcul necessaire au processus de visualisation 3.D, les performances obtenues restent mediocres.

#### V.4.1 PERFORMANCES RELATIVE A CHAQUE PHASE :

Il s'agit d'evaluer les phases de calcul proprement dit du processus et pour chaque type d'operateur, le nombre d'entites de base traitees en 40 millisecondes. L'entite de base consideree est la face plane.

#### LA TRANSFORMATION MATRICIELLE :

TYPE D'OPERATEUR	NOMBRE DE FACES
Multiplieur / diviseurs rapides	214
Operateur intermediaire O.P.C	282
Operateur specifique O.P.C	846

#### LA PROJECTION EN PERSPECTIVE :

TYPE D'OPERATEUR	NOMBRE DE FACES
Multiplieur / diviseurs rapides	510
Operateur intermediaire O.T.E	552
Operateur specifique O.P.J	714



## LA TRANSFORMATION-ECRAN :

TYPE D'OPERATEUR	NOMBRE DE FACES
Multiplieur / diviseurs rapides	461
Operateur specifique O.T.E	846

Le cheminement sequentiel initialement propose n'est pas obligatoire, certaines phases peuvent etre court-cicutees. Differentes compositions vont etre proposees dans la suite de l'evaluation.

## TRANSFORMATION MATRICIELLE, PROJECTION ET VISUALISATION :

Dans ce cas l'utilisateur definit un espace de travail inclus dans l'espace-ecran. Une fois la transformation matricielle et la projection effectuees, le terminal graphique realise la visualisation. Si des aretes se situent apres calcul hors de l'espace-ecran, elles ne sont pas visualisees. (decoupage cable primaire a partir des coordonnees des sommets). L'evaluation finale inclut la phase de visualisation du contour d'une face, soit environ 40 microsecondes. ("modele" de 10 points X 10 points) Les performances sont les suivantes : (phase de visualisation comprise)

TYPE D'OPERATEURS	CONTRAINTE TEMPS-REEL
multiplieurs / diviseurs rapides	131 faces
Les operateurs O.M.P et O.T.E	157 faces
Les operateurs O.M.C et O.P.J	280 faces

La solution des operateurs specifiques semble etre plus adaptee pour des contraintes de temps-reel. Cependant, l'option de remplissage a une vitesse de 5,25 K-taches par seconde, degrade ces performances de maniere constante, d'environ 60 %. Cette remarque est valable tout au long de cette evaluation.

TRANSFORMATION MATRICIELLE / ECRAN ET VISUALISATION :

-----

Dans ce cas, l'utilisateur se definit un espace de travail inclus dans l'espace-ecran, cependant il dispose de la possibilite de definition d'une cloture d'affichage dans l'espace-ecran. Il faut donc effectuer un calcul de correspondance pour faire coincider l'espace utilisateur observe dans le sous-espace-ecran defini. Le decoupage n'est pas realise. La transformation matricielle inclut la phase de projection en perspective. Ce processus propose les performances suivantes :

TYPE D'OPERATEURS	CONTRAINTE TEMPS-REEL
-----	-----
multiplieurs / diviseurs rapides	100 faces
Les operateurs O.M.P et O.T.E	132 faces
Les operateurs O.M.C, O.P.J et O.T.E	210 faces

D'autres compositions du processus de synthese sont envisageables ... En conclusion, il apparait que la solution des operateurs cables qu'ils soient intermediaires ou specifiques, est une option interessante. De plus les performances presentees sont intrinseques et ne tiennent pas compte d'un parallelisme possible entre les differents operateurs. Dans une architecture "pipeline" ou le recouvrement serait tel que le debit de la chaine serait fonction de celui de l'etage le moins performant, on obtiendrait la performance suivante :

- Operateurs specifiques avec visualisation, la phase de decoupage 2.D etant a la charge de la post-synthese, 416 faces traitees en temps reel.

#### V.4.2 MULTI-PROCESSEUR 3.D DECENTRALISE :

Cette premiere etude a decrit des nouvelles architectures ou le processeur central est dote de co-processeurs de sous-traitance du calcul avec des evaluations qui permettent de valider l'interet du multi-processeur au niveau operatoire. La deuxieme possibilite consiste a proposer des architectures multi-processeurs ou chaque processeur assure un processus ou une phase donnee au sein du processus general de visualisation 3.D. Dans ce cas chaque processeur assure les traitements qui lui sont affectes eventuellement a l'aide de co-processeurs de sous-traitance.

Une ebauche d'evaluations est proposee dans ce chapitre. En effet, il est difficile d'evaluer avec precision et de facon theorique les performances de ce type d'architecture. Du point de vue fonctionnel on peut imaginer un "pipeline" pour la chaine de pre-synthese avec une communication inter-processeur par files ou par memoires communes a un couple de processeurs. Dans ce cas l'architecture de pre-synthese evoluee decrite precedemment se transforme en une chaine de processeurs gerant separement des operateurs cables de calcul.

## LA PROPOSITION : LE "PIPELINE" SYNCHRONE

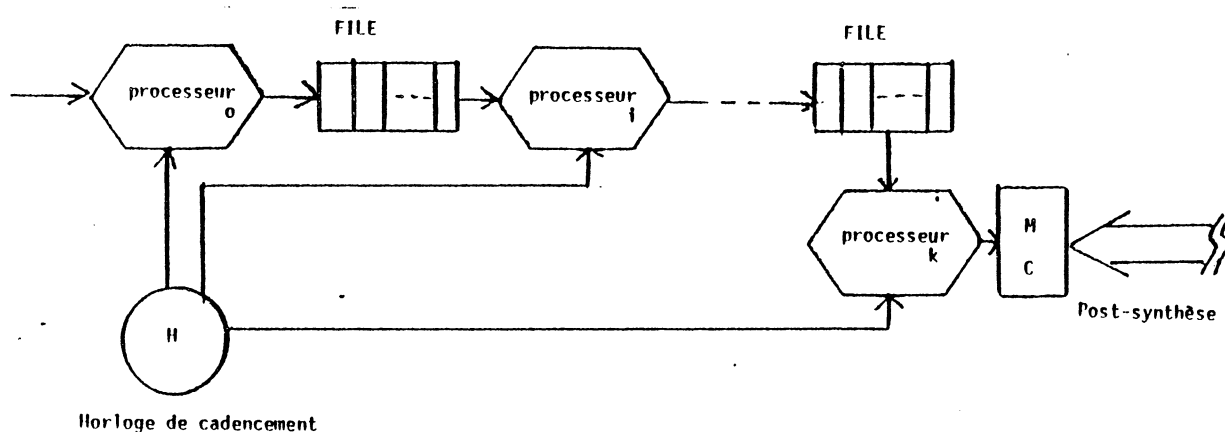
LA PRE-SYNTHESE MULTI-PROCESSEUR  
PROCESSUS DE SYNTHESE - 3D

Figure No V.9

La mise en oeuvre d'une architecture parallele au niveau du processus de synthese implique le dimensionnement des files de communication, le choix du processeur adapte a chaque phase du processus 3.D et enfin un choix de synchronisation inter-processeur. (synchrone ou asynchrone.)

Dans ce cas, la performance globale de l'architecture sera conditionnee par le temps de calcul necessaire a la phase la plus complexe du processus 3.D, le decoupage 3.D ou l'elimination des lignes cachees. Cette premiere proposition d'architecture consiste a choisir un processeur performant manipulant des informations sur 32 bits et dont l'horloge de base est superieure a 8 MHz. Ensuite il s'agit de dedier un processeur a chaque phase du processus 3.D de visualisation. Cette etude preliminaire concerne en particulier les phases de calcul. De nombreux constructeurs de stations de synthese d'images optent pour ce type d'architectures. (cf chapitre VI)

## V.4.2.1 PROCESSEURS SYSTOLIQUES : -

Il s'agit d'etudier l'impact de processeurs specialises dans les architectures de terminaux graphiques dotes de performances en temps-reel. L'approche consiste a effectuer une repartition des differentes phases d'un processus de synthese d'images entre les differents processeurs d'un "pipeline" classique. ( MGE 84 ) ( AIF 84 )

Le processeur IMST.424 est en fait appelle "transputer". Il s'agit de derives de microprocesseurs en tranches. La caracteristique principale de ce type de composant est la mise en oeuvre d'une unite centrale a jeu d'instructions reduit mais performant. Il s'agit de machines a "R.I.S.C" signifiant :

-----  
 Reduced Instruction Set for Compact program.  
 -----

De plus ces processeurs sont dotes de liens d'entrees/sorties dont les vitesses de transmission sont de l'ordre de 15 M-bits par seconde. La capacite memoire adressable est de 25 M-octets. Chaque processeur est dote d'une memoire locale de 4 K-octets dont le temps d'accès est de 40 nanosecondes. Une interface d'extension memoire permet d'adresser jusqu'a 4 G-octets.

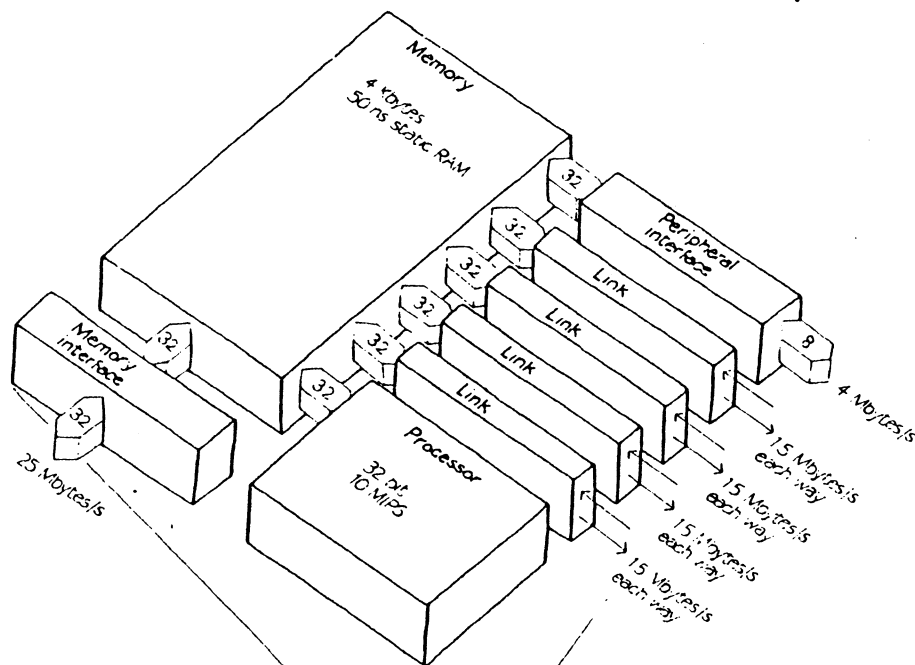


Figure No V.10

L'horloge de base est de 5 MHz. Ces processeurs peuvent etre agences selon differentes formes. Les communications inter-processeurs sont assurees par les liens d'entrees/sorties disponibles. L' unite arithmetique effectue les quatre operations sur des operandes de 32 bits. Les temps de calcul sont :

- L'addition : 50 nanosecondes,
- La multiplication 16 X 16 => 32 bits : 950 nanosecondes,
- La division 32 / 16 => 16 bits : 1,950 microsecondes.

Une evaluation primaire indique qu'une transformation matricielle se fait en 8,85 microsecondes, ce temps de reponse est meilleur que celui obtenu avec un operateur matriciel complet. (O.M.C) Cette performance permet de transformer pres de 4520 sommets en temps-reel.

L'approche de conception d'une architecture "pipeline" se prete bien a ce type de processeur. Les echanges d'informations entre les differents etages se font par l'intermediaire d'instructions d'entrees/sorties a travers les liens disponibles. Ces intructions de base gerent la synchronisation entre les differents processeurs. La syntaxe est IN ou OUT "adresse du lien concerne". ( AIF 84 ) La premiere proposition consiste a affecter un "transputer" a chaque etape du processus de visualisation 3.D. Si on considere l'amelioration obtenue pour la phase de transformation matricielle comme ordre de grandeur de progression pour les autres phases de calcul, on obtient les evaluations suivantes :

- La transformation matricielle d'une face : 31,5 microsecondes,
- La projection en perspective d'une face : 20,6 microsecondes,
- Le decoupage tri-dimensionnel d'une face : 479,5 microsecondes,
- La transformation-ecran d'une face : 33 microsecondes.

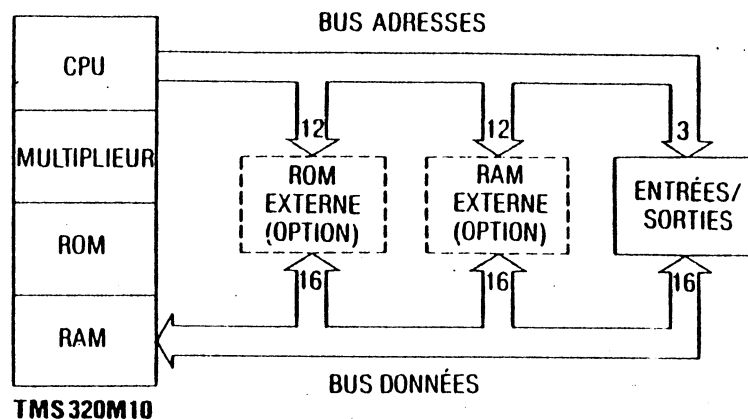
Si la phase de decoupage tri-dimensionnel est en partie traitee par la chaine de post-synthese, on synthetisera pres de 318 faces en temps-reel. Dans une seconde evaluation, on considere que le recouvrement existe entre les differents etages. L'etage le moins performant determine le temps de reponse global du "pipeline". Les delais de communication ne sont pas decompes dans cette evaluation. On obtient dans cette configuration 84 faces traitees et visualisees en temps-reel avec le decoupage et 1000 faces sans decoupage.

En conclusion il apparait que ce type de processeur permet, grace a son architecture adaptee, de repondre aux contraintes de temps-reel de maniere satisfaisante. Il est certain qu'une etude plus approfondie serait necessaire, ce n'est pas l'objet de notre analyse. L'inconvenient majeur de ce composant est son cout actuel qui est de l'ordre de 5000 F H.T.

#### V.4.2.2 ETUDE DE PROCESSEURS SPECIFIQUES : -

Il s'agit du processeur de traitement de signal TMS.320 dont les temps de reponse en calcul arithmetique sont extremement performants. ( ELE 84 ) Son implantation au sein de la pre-synthese du terminal HELIOS a ete envisagee pour la configuration intermediaire.

L'interet primordial de ce processeur est de pouvoir effectuer une operation arithmetique quelconque en un cycle interne unique de 200 nanosecondes et sur des operandes de 16 et 32 bits. Cette vitesse de calcul ne permet une comparaison qu'avec les multiplieurs rapides decrits precedemment. La mise en oeuvre n'est pas prohibitive du point de vue du cout. Ce processeur coute actuellement pres de 800 F H.T. Son point faible est la capacite d'adressage reduite. Il dispose d'une memoire vive interne de 288 octets et un memoire morte interne de 3 K-octets. Un port d'entree/sortie permet de realiser une extension memoire avec un debit de 40 M-bits par seconde. (soit un temps d'accès de 2,5 microsecondes)



#### TMS 320M10

Extension mémoire jusqu'à 8 octets sans affecter la vitesse avec le TMS 32010. Le TMS 320M10 (mode micro-calculateur) exécute les instructions contenues d'une part dans la mémoire ROM intégrée (adresses 0 à 1536) d'autre part celles contenues en niveau externe (adresses 1536 à 4096).

Figure No V.11

Ce processeur fonctionne en mode de recouvrement de cycles. Il effectue la recherche de l'instruction "i+1" en meme temps que l'execution de l'instruction "i". L'affectation d'un processeur TMS.320 a chaque phase du processus de visualisation tri-dimensionnel dans le cadre d'une architecture "pipeline" permet obtenir les performances suivantes en tenant compte des temps d'accès necessaire a la recherche des operandes.

Si l'on suppose que les variables necessaires au calcul sont dans la memoire vive interne du processeur, deux cycles memoire sont utilises pour la recherche de l'information. Ces delais supplementaires de 400 nanosecondes par variables attenuent legerement les performances.

## EVALUATION DES ECHANGES AVEC LA MEMOIRE :

PHASE DU PROCESSUS	TEMPS DE REPONSE
Transformation matricielle	12,4 microsecondes
Projection en perspective	7,2 microsecondes
Transformation-ecran	4 microsecondes

La phase de transformation matricielle necessite 22 acces. La phase de projection en perspective necessite 6 acces. La phase de transformation-ecran necessite 6 acces. Le TMS.320 est moins rapide que l'operateur matriciel complet car les acces a la memoire interne le penalisent. Par contre il est plus rapide que les operateurs de projection et de transformation-ecran. Une architecture "pipeline" permettrait d'obtenir des performances comparables a celles des IMST.424. Cependant la manipulation de structures de donnees a travers le port d'extension memoire penaliserait le temps de reponse global du systeme. ( AIF 84 )

## V.4.2.3 CAS DES PROCESSEURS EN TRANCHES : -

Il s'agit de processeurs "bit slice" qui sont de plus en plus utilises dans les architectures de terminaux graphiques evolues. Le processeur en tranches est constitue de deux unites qui sont l'unite de controle et de sequencement et l'unite de calcul arithmetique et logique. On s'interessera dans cette etude a la famille de processeurs en tranches de la famille A.M.D. (Advanced Micro Device.) Les deux unites sont appelees respectivement :

- La partie-sequenceur AMD.2909,
- La partie operative AMD.2901 sur 4 bits.

La gestion de la partie operative et des unites peripheriques associees se fait par l'intermediaire d'un jeu de micro-instructions interpretees et executees par la partie-sequenceur.

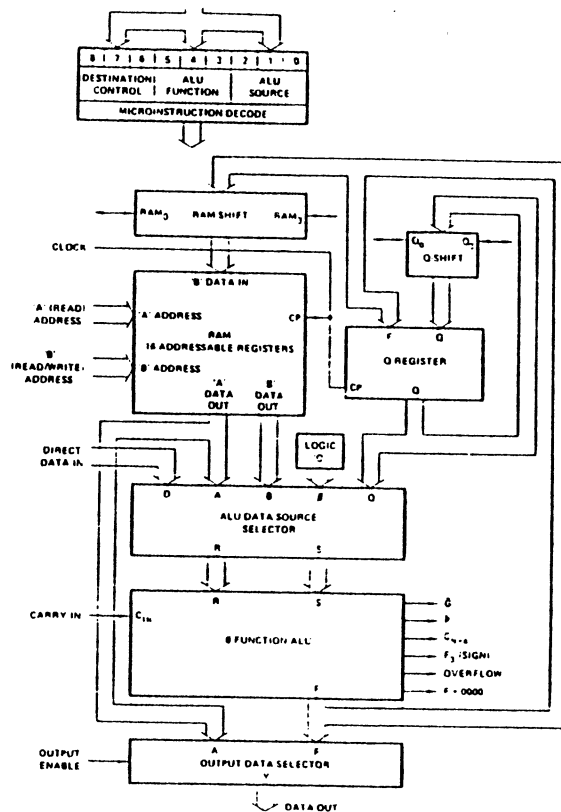


Figure No V.12

AMD propose des parties sequenceurs ameliores. Il s'agit de l'AMD.2903 qui integre des bus supplementaires d'accès a la memoire permettant l'extension de la memoire locale de 16 registres de 4 bits appartenant a l'AMD.2909. L'AMD.2903 permet de plus la gestion simplifiée des operations de decalage au niveau de la partie operative. Il est capable d'adresser jusqu'a 4 K-mots de micro-instructions grace a un compteur interne de 12 bits. Cependant il n'est pas cascadable. Enfin AMD propose la partie-sequenceur 29112 de huit bits cascadables qui est actuellement l'unité la plus performante de la famille AMD. L'horloge de base du systeme est celle de l'organe le plus lent en l'occurrence la partie operative, soit environ 10 MHz. La propagation de la retenue se fait sans anticipation en 100 nanosecondes, avec anticipation en 15 nanosecondes. Le temps d'execution d'une micro-instruction est donc compris entre 100 et 115 nanosecondes.

( ELI 84 ) ( AMD 84 ) ( CHU 84 )

Dans une premiere etape la mise en oeuvre d'un multi-processeur au niveau operatoire est possible. Une premiere evaluation permet de determiner les temps de calculs des operations de base la multiplication et la division. AMD propose des algorithmes simples pour mettre en oeuvre les microprogrammes associes. Trois micro-instructions sont necessaires :



- L'addition et le decalage gauche sur condition booleenne,
- La soustraction et le decalage droit sur condition booleenne,
- Le branchement sur condition.

Initialement le microprogramme recupere les deux operandes sur 16 bits chacun. Les evaluations obtenues sont pour la multiplication 2 microsecondes et pour la division 2,5 microsecondes. (performances de l'AMD.2901-2909) Ainsi on obtient pour les differentes phases de calcul du processus de visualisation 3.D, les performances suivantes concernant le traitement d'une arete.

LES PERFORMANCES DE L'AMD.2901-2909 :

PHASE -----	TEMPS DE REPONSE -----
Transformation matricielle	21 microsecondes
Projection en perspective	9,7 microsecondes
Transformation-ecran	10,4 microsecondes

Ce co-processeur fonctionne pour ce type de calcul 2 a 3 fois plus vite que le processeur MC.68000 a 8 MHz. Une architecture a processeurs en tranches peut constituer une solution rentable et plus flexible que celle des operateurs cables specifiques ou intermediaires. La flexibilite est une consequence de la possibilite de modifier ou de faire evoluer les microprogrammes associes a l'architecture qui peut realiser les trois phases de calcul eventuellement. Des propositions d'architecture existent et sont souvent equipees de multiplieurs rapides associes au processeur en tranches. Des memoires double-acces sont mises en oeuvre pour ameliorer la performance. Les temps des traitements obtenus sont de l'ordre de 10 microsecondes. Pour une architecture evolutive a base de microprocesseurs en tranches on obtient des performances pour la transformation matricielle de l'ordre de 12 a 6.75 microsecondes. La cellule "CSI" concue a l'universite de Rennes constituee de deux processeurs flottants (100 nanosecondes pour une operation quelconque) presente les performances suivantes :

- Transformation matricielle : 3.5 microsecondes
- Produit entre deux matrices 4X4 : 10.2 microsecondes

- Produit scalaire ou vectoriel : 1.2 a 1.8 microsecondes

L'architecture "pipeline" peut etre envisagee, l'affectation d'une unite centrale a chaque phase du processus est possible, et ceci meme pour la phase de visualisation. Les algorithmes de generation de vecteurs peuvent etre microprogrammes, le processeur pilotera de plus la post-synthese. Ce type d'architecture est celui des terminaux graphiques du constructeur americain APOLLO dont une presentation sera faite dans le chapitre suivant. ( SMR 85 ) ( ROU 86 )

## CONCLUSION :

En conclusion generale les differentes solutions proposees semblent offrir des performances interessantes. Le choix du concepteur doit s'appuyer sur les performances visees d'une part, et sur la gamme de configuration materielle consideree d'autre part. Il s'agit aussi de tenir compte du type d'applications considere.

Il est possible de proposer des choix de realisation des operateurs des differentes phases du processus de synthese tri-dimensionnel en fonction d'applications graphiques caracteristiques. Ainsi pour des applications de C.A.O ou de production d'images synthetiques des operateurs partiels ou des multiplieurs/diviseurs peuvent etre envisages. Cependant, pour des applications d'animation en temps-reel, seul le multiprocesseur decentralise dote eventuellement d'operateurs specifiques et complets; peut repondre aux performances attendues.

L'utilisation de co-processeurs classiques est pratiquement inevitable bien que l'apport en performance soit mediocre dans ce cadre. Cependant l'adjonction d'unites de calcul specifiques semble etre de plus en plus courant sur les terminaux graphiques de haut de gamme. Ainsi le concepteur peut retenir pour des configurations orientees vers la production d'images synthetiques en 3.D, des operateurs tels que l'O.M.C, l'O.T.E et l'O.P.J; la phase de decoupage 2.D devant etre realisee par la post-synthese.

Des configurations plus modestes de milieu de gamme seraient equipees d'un operateur matriciel unique qui serait exploite aussi bien en 2.D qu'en 3.D. On notera que les terminaux de milieu de gamme sont au moins bi-processeurs au niveau de la pre-synthese. Il semble que le multiprocesseur decentralise soit l'architecture retenue pour les stations de synthese d'images. Les criteres de choix du processeur de base pourraient etre le cout et la complexite de la mise en oeuvre. A titre indicatif on peut rappeler les prix suivants :

Le TMS.320	: 800 F H.T
Le O.M.C	: 3000 F H.T
L'IMST.424	: 5000 F H.T
L'AMD.2901-2909	: 250 a 1000 F H.T

## CHAPITRE VI

### LE CONCEPT DE LA STATION DE TRAVAIL :

#### REFERENCES BIBLIOGRAPHIQUES :

( ATO 85 ), ( CHR 84 ), ( CIG 84 ), ( CID 84 ),  
( DOD 84 ), ( DSS 85 ), ( GAU 85 ), ( GEG 85 ),  
( GRO 85 ), ( HUB 84 ), ( HOF 84 ), ( ITM 85 ),  
( JCK 85 ), ( LAG 85 ), ( MAR 84 ), ( MCC 84 ),  
( MET 84 ), ( MGE 84 ), ( NIC 84 ), ( VIA 85 ),  
( WIL 85 ), ( ZAR 84 ), ( ZAR 85 ).

## VI.1 LA NOUVELLE TENDANCE :

Initialement, l'architecture d'un terminal graphique interactif se resumait a un post-synthetiseur connecte a un moniteur TV de visualisation et un calculateur principal dote d'une application programme. Cette configuration de "console bas de gamme" a mis en evidence les points faibles d'une pareille architecture. Les inconvenients majeurs sont la dependance de l'application vis a vis du materiel, l'inexistence de flexibilite et enfin la pauvreté des possibilites graphiques du terminal. ( MAR 82 )

## VI.1.1 EVOLUTION DES ARCHITECTURES :

Le point de depart de l'amelioration des fonctionnalites du terminal est l'evolution du post-synthetiseur. Il s'agit maintenant d'une chaine de post-synthese assurant de nombreux processus associes a la visualisation. la pre-synthese s'occupe par exemple du trace de vecteurs, de l'affichage de caracteres, de l'effacement de l'ecran, de la designation d'un point a l'ecran par le biais d'un reticule. Le programme d'application a la possibilite d'utiliser ces primitives de base au niveau du calculateur hôte.

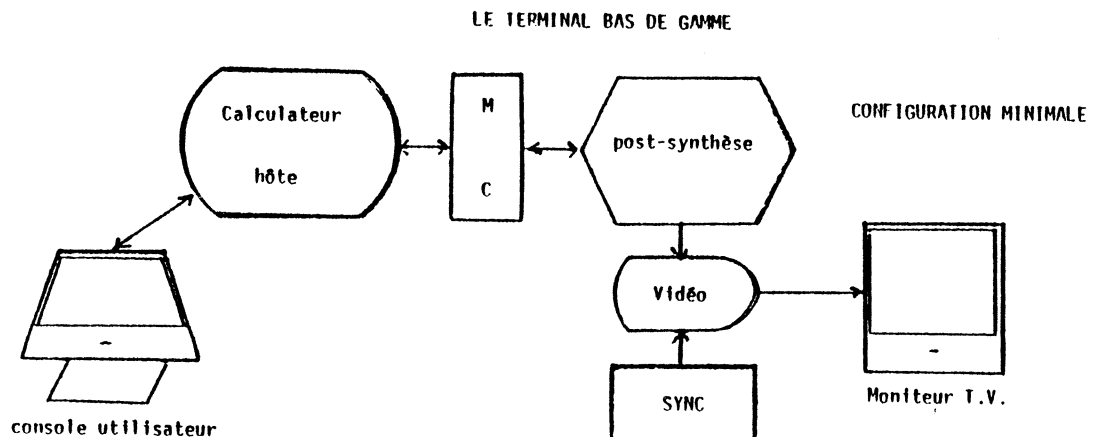


Figure No VI.1

Cette configuration se materialise par l'apparition des co-processeurs graphiques ou de logique cablee specialisee a base de composants V.L.S.I EF.9365, NEC.7220 et HITACHI 63464, ou encore de microprocesseurs en tranches AMD 2909-2901. Il s'agit de la configuration : "console evoluee"

On assiste ainsi a une migration d'un ensemble de processus associes a la synthese d'images du calculateur hôte vers le terminal graphique. Ainsi la conception d'un systeme graphique consiste a decomposer le processus

general de synthese d'images et a affecter chaque etape a un processeur donne. L'ensemble des processeurs constitue l'architecture "pipeline". La performance globale depend de l'organisation du "pipeline". Cette demarche de conception qui aboutit aujourd'hui aux stations de travail proposees par de nombreux constructeurs, se base sur trois approches fondamentales :

- Le systeme "pipeline" general,
- Le systeme a partitionnement de l'ecran,
- Le systeme a partitionnement de l'objet de l'ecran.

( CIG 84 )

#### LE SYSTEME "PIPELINE" GENERAL :

Il s'agit generalement de systemes performants privilegiant la contrainte de temps-reel. Chaque etage est capable de traiter une trame d'image. A un instant donne, plusieurs trames successives d'image circulent dans la chaine des processeurs. L'etape finale est la visualisation sur le moniteur T.V.

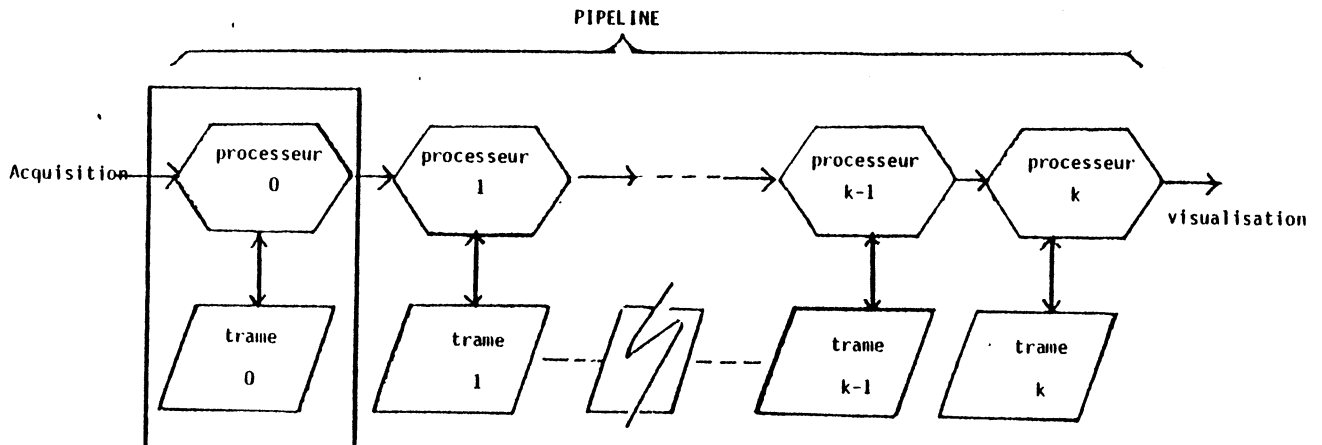


Figure No VI.2

Chaque processeur est capable d'effectuer une transformation du processus de synthese d'une trame d'image complete en 40 milisecondes. La transformation associee a un processeur peut etre :

- Une transformation matricielle geometrique,
- Un decoupage de la scene,
- L'elimination des parties cachees,
- Le calcul des ombres portees.

L'entite de base manipulee est pre-definie, il s'agit un polygone quelconque. Dans ce cas, l'architecture est entierement determinee par le type d'objet manipule. Le domaine d'application de ce type de systeme est specifique. Il s'agit des simulateurs de vols et de la production cinematographique. L'exemple le plus interessant illustrant ce type d'architecture est La machine CT-3 concue et realisee par EVANS et SUTHERLAND. Cette station de synthese d'images en temps-reel est constituee de 800 cartes memoires et processeurs. 100 types de carte differentes existent. Cette complexite "quasi-dementielle" demontre la dependance technologique actuelle des constructeurs de stations de travail graphiques.

#### LE SYSTEME DE PARTITIONNEMENT DE L'ECRAN :

Cette architecture est caracterisee par le decoupage de l'espace-ecran en portions d'images. Chaque portion est affectee a un processeur donne appartenant a un "pipeline". Le processeur est responsable du traitement concernant une region de l'ecran. La complexite d'un etage est moindre car le calcul affecte a une portion d'image est plus faible. Les temps de reponse sont meilleurs. Les problemes que souleve ce type d'architecture parallele sont :

- Le choix d'un type de partitionnement,
- La repartition des taches entre les differents processeurs,
- la reconstitution de l'image a partir des portions issue des differents processus,

La complexite de l'architecture depend directement de la resolution de l'espace-ecran et de la finesse du decoupage de l'espace-ecran. Ce decoupage consiste en l'assignation d'un groupe de points de l'ecran. Le nombre de points peut etre fixe ou variable. Le premier cas correspond au modele propose par J.H CLARK. Il s'agit du "Geomery Engine" dont la description sera faite dans la suite du chapitre. La structure est reguliere :

- Un processeur par point du groupe  $N \times M$  points,

- Un processeur pour les lignes paires,
- Un processeur pour les lignes impaires.

L'écran est généralement découpé en rectangles. Deux concepts fondamentaux caractérisent ce découpage de l'écran. Il s'agit du découpage en régions contigües et du découpage en régions non-contigües. Le second cas correspond au modèle de Kaplan, initialement les différents processeurs sont libres. Un ordonnanceur se charge d'activer les étages du "pipeline" et leur affecte des traitements spécifiques selon le découpage de l'écran retenu. Les autres cas de découpage sont :

- L'assignation d'une ligne de la trame d'image à un processeur,
- L'assignation d'une portion de lignes à un processeur, la taille de la portion est déterminée dynamiquement,

#### LE PARTITIONNEMENT DE L'OBJET DE L'ECRAN :

Il s'agit de mettre en oeuvre une architecture parallèle ou un groupe de processeurs opère sur un objet de l'écran. Le processus de synthèse d'un type d'objet est reparté sous forme de tâches attribuées à un groupe de processeurs. Cette architecture est capable de gérer de nombreux types d'objets graphiques pré-définis. La difficulté de réalisation de ce type d'architecture est l'allocation d'un objet de l'écran à un groupe de processeurs et le partage des tâches.

Il paraît important de noter la complexité croissante des systèmes graphiques actuels. Cette complexité est déterminante du point de vue des fonctionnalités du terminal considéré, pour une configuration d'utilisation donnée. Dans une configuration de type "station de travail", le contexte d'utilisation du T.G.I permet le développement local d'une application sans nécessité d'un ordinateur hôte. De plus les performances proposées sont meilleures.

#### VI.1.2 SYSTEMES MULTI-PROCESSEUR :

Il apparaît alors deux entités de base appelées la partie calculateur et la partie générateur images. Le multiprocesseur est mis en oeuvre au niveau de chaque partie de la station de travail. La partie calculateur est constituée d'un microprocesseur 16/32 bits et de son environnement, un co-processeur de calcul accompagne souvent cette architecture. (le co-processeur se charge généralement des transformations géométriques associées au processus de synthèse 2.D ou 3.D) La partie générateur d'images est constituée de deux unités, la première est un ensemble de co-processeurs en tranches chargés des tâches graphiques élémentaires. (génération de



vecteurs, cercles, trapeze ..., remplissage de taches, etc...) la seconde unite est la logique cablee de visualisation fonctionnant au rythme du signal video. ( memoire de trame, table de correspondance, interface video digital-analogique ...)

Le developpement d'une version commercialisee du terminal graphique interactif HELIOS s'inscrit dans ce cadre. Il s'agit la de l'integration de chaine complete de synthese a l'architecture d'un micro ou mini-ordinateur classique.

## VI.2 STATION GRAPHIQUE GETRIS-3.D :

Il s'agit de la version commercialisee du terminal HELIOS par la societe grenobloise GETRIS.IMAGES. Le processeur principal de la station de travail est en fait le processeur de la chaine de pre-synthese. Il s'agit d'un micro-processeur 16 bits, Intel.8086 dont l'horloge de base est a 8 MHz. Ce choix est dicte par des raisons commerciales bien que les evaluations des chapitres precedents demontrent que le micro-processeur de la famille Motorola MC.68020 est plus performant. Le multi-processeur est de type operatoire car le processeur principal est dote d'un co-processeur arithmetique flottante de la meme famille (Intel.8087) Ce co-processeur intervient dans le calcul matriciel des transformations geometriques en 2.D ou 3.D.

### VI.2.1 LA PARTIE CALCULATEUR :

Cette partie calculateur assure un ensemble de fonctions :

- Les operations de base necessaires a la synthese d'images,
- Le support du logiciel graphique 2.D/3.D,
- La gestion de la memoire de masse de la station (unites de disques durs et souples)
- La gestion des coupleurs d'entree/sortie pour les divers peripheriques, (dispositif d'interactions compris)
- L'interface avec le module de communication de la chaine de post-synthese,
- La gestion des memoires vives necessaires (memorisation des structures de donnees).

## REMARQUE :

La partie calculateur de la station de travail est constituée d'un ensemble de cartes communicant à travers un bus normalisé de type MULTI-BUS. La post-synthèse correspond dans cette configuration à la partie générateur d'images.

## VI.2.2 PARTIE GÉNÉRATEUR D'IMAGES :

La post-synthèse constituée de la partie générateur d'images a une structure modulaire. Cette architecture diffère très peu de celle du terminal HELIOS. Certaines modifications consistent à intégrer deux modules différents sur une même carte. Il s'agit en particulier du module de communication et de l'interface vidéo. Cependant l'ordonnement du processus à travers la chaîne de synthèse est identique à celui décrit dans les chapitres précédents. La modularité de l'architecture de ce système graphique permet de définir deux synthétiseurs optionnels qui sont le processeur d'éclairage et le processeur des textures. Les mémoires de la post-synthèse sont extensibles :

- Les plans de mémoire ont une profondeur maximale de 8 plans,
- La table de correspondances mémorise les couleurs, le nombre de faces possibles.

L'interface directe avec la partie calculateur de la station de travail et la partie générateur d'images est appelée :

- "COMMUNICATION ET GÉNÉRATION VIDÉO".

Ce module pilote les différents bus de la chaîne de post-synthèse et synchronise les échanges d'informations. De plus ce module intègre les horloges de génération des différents signaux de synchronisation vidéo qui sont le retour-trame, le retour-ligne et l'horloge de base. Il réalise la conversion digitale analogique pour engendrer les différentes composantes de la couleur d'un signal vidéo en synthèse additive.

Un module intègre l'ensemble de la mémoire constituant :

- LES PLANS DE MÉMOIRE :

Il s'agit des plans d'identification et la mémoire de trame de la logique câblée de visualisation. Cette mémoire est accessible par l'intermédiaire de la partie communication du module précédent en temps-réel soit à une vitesse de 13,5 millions de points par seconde. Ce module permet en temps-réel :

- L'enregistrement simultane de 4096 informations differentes,
- La translation du plan d'image,
- La gestion de la profondeur et des priorites des differents plans,
- La gestion de la visibilite ou de l'invisibilite,
- La creation de fenetres dans le plan d'image,
- La synchronisation des ecritures et des lectures via le module de communication.

La capacite d'un plan de memoire est 1024 X 1024 points avec 12 bits de profondeur d'informations. L'information de base codee sur 12 bits est une couleur parmi 4096 en synthese additive. Ce module est extensible jusqu'a 8 plans memoires de meme caracteristiques et ceci selon le besoin de l'application.

Le module "TABLE DE CORRESPONDANCE" est constitue d'une memoire mono-dimensionnelle permettant de stocker les numeros logiques des faces constituant les objets d'une scene quelconque. Il offre la possibilite d'identifier en temps-reel un sous-ensemble de faces appartenant a un objet donne de la scene. Cette facilite permet en temps-reel :

- La modification de la couleur associee a la face,
- La designation et l'identification de la face par l'intermediaire d'un dispositif d'interaction,
- L'affichage de 4096 couleurs parmi 16 millions de couleurs possibles (mise en oeuvre de deux tables de correspondances de 4096 couleurs :  $4 K \times 4 K = 16 M$  couleurs au total).

Il est necessaire de faire une difference entre les modules extensibles et les modules optionnels. Les modules PLAN MEMOIRE et TABLE DE CORRESPONDANCE sont des modules dits extensibles. Dans une configuration multi-plan, le T.G.I permet une gestion par juxtaposition ou superposition des differents plans. Ce choix est programmable par l'utilisateur. Un interet du mode multi-plan est d'integrer une partie cablee de l'algorithme du Z-buffer pour l'elimination des parties cachees. Le module des TABLES DE CORRESPONDANCE peut gerer jusqu'a 2 tables.

Cette extension permet d'afficher simultanement 16 millions de couleurs. Elle permet d'identifier jusqu'a 16000 faces d'objets d'une scene quelconque. (soit 4 tables de correspondance et 4 plans memoire). Cette option est valide en mode identification. On note ici la faculte de reconfiguration des differentes memoires de la station GETRIS. Il s'agit la d'une architecture banalisee. ( ZAR 83 et 85 )

Les modules "ECLAIRAGE et TEXTURE" sont typiquement optionnels et n'intéressent que les configurations de T.G.I destinées à de la synthèse d'images à haut degré de réalisme et d'interactivité. Ces modules sont dotés de processeurs câblés de calcul de l'éclairage et de la projection des textures associées aux faces d'un objet de la scène. Le module d'éclairage calcule en temps-reel et en fonction de la normale de chaque face :

- Le déplacement de la source lumineuse dans l'espace,
- Le changement de la couleur ou de l'intensité de :
  - . La source lumineuse,
  - . La lumière ambiante.
- Le calcul de la couleur et de l'intensité en chaque point.
- Le calcul de la réflexion (brillance) des matériaux pour chaque face. (14 niveaux de réflexion sont possibles, du mat au brillant)

La configuration de la partie ordinateur et générateur d'images est représentée par la figure suivante :

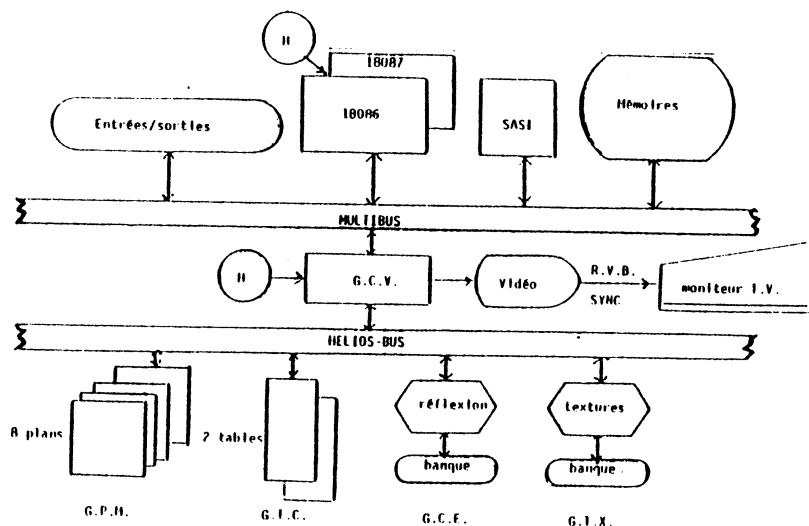


Figure No VI.3

En conclusion on peut dire que cette description technique du terminal graphique interactif HELIOS dans une configuration "STATION de TRAVAIL", est caractérisée essentiellement par sa modularité et son extensibilité. Deux configurations d'utilisation du T.G.I sont disponibles la "console évoluée" et la "station de travail". Dans la configuration "console évoluée", la

communication se fait a travers une interface de type "emulation MULTIBUS", cette option recente assure une connexion avec tous les micro-ordinateurs de type IBM-PC.AT ou compatible.

### VI.2.3 ETUDE DES PERFORMANCES :

La logique cablee de visualisation est constituee de l'ensemble des operateurs de la chaine de post-synthese d'HELIOS III. L'architecture modulaire associee a la partie generateur d'images permet de definir une configuration de base pour un utilisateur et un type d'application donne. La station de travail GETRIS associee au moniteur de visualisation permet de synthetiser des images a haut degre de realisme. Elle permet de creer et de visualiser des objets tri-dimensionnels composes de faces et de surfaces gauches, la modification et l'animation de ces objets etant possible en temps-reel.

Des fonctionnalites de niveau superieur sont realisees par le micro-logiciel execute par le processeur central de la station. Elles concernent la gestion des structures de donnees des scenes 2.D et 3.D et les operations associees a ces structures. Les performances obtenues dans la configuration "station de travail" sont les suivantes :

- Vitesse de traces de vecteurs 6 Millions de points par seconde, (soit 75 K-vecteurs de 2 cm par seconde ou 3000 vecteurs en temps-reel)
- Remplissage de taches a 14 Millions de points par seconde, (soit 8000 carres de 2 cm de cote)
- Vitesse d'affichage des caracteres 20 K-caracteres par seconde, (programmable en 16 tailles et 8 directions)

Les transformations geometriques realisees par la partie calculateur dotee d'un co-processeur Intel.8087 ont les performances suivantes, 20000 points par seconde en 2.D et 10000 points par seconde en 3.D. Pour une evaluation des operations de transformation matricielle et de visualisation de quadrilateres de l'espace, on obtient les performances suivantes, 2000 a 6000 figures par seconde. Globalement les performances sont encore accrues d'une part par l'adjonction du processeur Intel.8086-8087 et d'autre part, par l'amelioration de la communication et des debits vers la post-synthese. (acces direct aux differents processeurs, mise en oeuvre d'une pile de memorisation accessible par le processeur de la partie calculateur, amelioration du cycle d'accès HELIOS 70 nanoseconde par pixel.)

### VI.3 STATION GRAPHIQUE METHEUS :

## VI.3.1 PARTIE CALCULATEUR :

L'architecture de base de la partie calculateur est de type bi-processeur. Elle est constituée de deux MC.68000 pilotes par une horloge de base de 12,5 MHz. Un processeur se charge de la gestion de la mémoire centrale (allocation de pages virtuelles) d'une capacité de 1 M-octets extensible à 4 M-octets, de plus il s'occupe de la gestion de l'ensemble des interfaces d'entrées/sorties. Le second processeur se charge de l'exploitation globale des ressources de la station. Il supporte un système multi-tâche et multi-utilisateur de type UNIX dans la configuration de base.

La partie calculateur est dotée d'une mémoire de masse de 33 M-octets sur disque dur et d'un M-octets sur disque souple. Cette mémoire de masse est extensible par l'adjonction d'une seconde unité de disques durs de 33 M-octets. Un ensemble varié d'interfaces d'entrée/sortie est disponible. (réseau ETHERNET, lignes série ...)

## VI.3.2 PARTIE GENERATEUR D'IMAGES :

La mémoire centrale est partagée par les deux processeurs. Cette mémoire est banalisée. Elle intègre la mémoire de trames. En fait la gestion de la mémoire de trame avec une profondeur de 24 bits est particulière. Un bit parmi les 24 possibles indique un niveau de priorité dans la composition des couleurs.

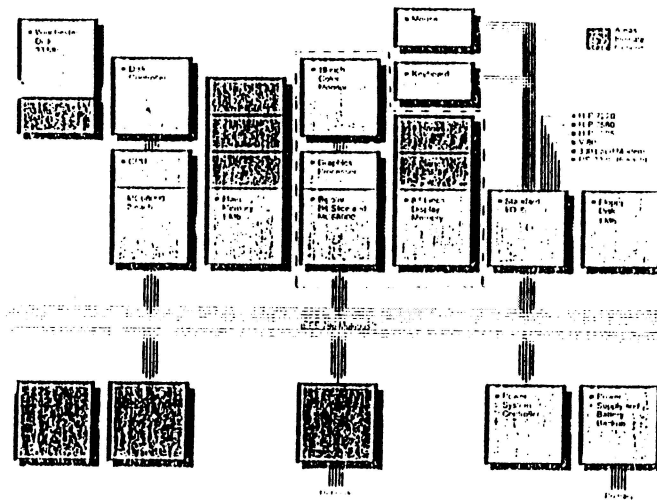


Figure No VI.4

Deux cas de compositions sont possibles, le groupe "haute priorité" masque le groupe "basse priorité" ou bien les deux groupes sont mélangés selon une synthèse additive des couleurs. La couleur est codée sur 24 bits. Cette option permet en particulier de gérer des attributs d'aspect comme la transparence ou l'opacité. La configuration de la mémoire de trames est

versatile : Une premiere configuration permet d'affecter a chaque groupe de huit plans memoire une priorite d'affichage :

- Le groupe 0 : 8 plans (256 couleurs) "haute priorite",
- Le groupe 1 : 8 plans (256 couleurs) "moyenne priorite",
- Le groupe 2 : 8 plans (256 couleurs) "basse priorite",

Ce systeme de priorites en cascades est valable pour une configuration avec deux groupes de huit plans memoires. Une deuxieme configuration permet d'affecter une composante des trois couleurs de base a chaque groupe avec 256 nuances par couleur.

### VI.3.3 ETUDE DES PERFORMANCES :

Cette configuration permet d'obtenir une palette de 16 millions de couleurs possibles simultanement. Le generateur d'images propose les performances suivantes :

- La generation de vecteurs : un million de points par seconde,
- Le transfert compresse de points permet des vitesses de remplissage de la memoire de trame de pres de quatre millions de points par seconde, (PIXBLT : Pixel Block Transfert)
- Le remplissage de taches polygonales : 16 millions de points par seconde (FLASHFILL-POLYGON)

Ces vitesses de generation d'images correspondent a des vitesses de remplissage maximales de la memoire de trame, de l'ordre de 284 millions de bits ecrits par seconde. Il est possible que l'architecture "BIT-SLICE" soit basee sur le dernier microprocesseur en tranches de la famille AMD.29116 oriente "bit-map" capable de traiter 16 points (bits) au rythme de 100 nanosecondes. (soit un point toutes les 6,25 nanosecondes, soit 160 millions de bits par seconde). Le moniteur de visualisation utilise est a haute resolution soit 1024 \* 768 points en mode entrelace. ( CHU 84 ) ( AMD 84 )

En conclusion on remarque que cette station de travail dispose d'une architecture bi-processeur au niveau de la pre-synthese, et d'une architecture a base de microprocesseurs en tranches au niveau de la post-synthese. On remarque l'analogie existant entre le "PIXBLT" et l'option de compression de donnees disponible au niveau du module de communication de la station GETRIS.

( MET 84 )

#### VI.4 STATION GRAPHIQUE CHROMATICS :

##### VI.4.1 PARTIE CALCULATEUR :

L'architecture de la station de travail "CHROMATICS" de la serie 7900 est basee sur un processeur 16 bits, le MC.68010 et son environnement. La memoire vive est extensible a 4 M-octets. La memoire morte du calculateur necessaire au moniteur resident de la station de travail a une capacite de base de 64 K-octets, extensible a 512 K-octets. La memoire de masse est constituee de deux unites de disques souples ayant une capacite de 1 M-octets chacune. En option, il est possible de disposer de disques durs d'une capacite de 8 M-octets a 64 M-octets. Des unites de derouleurs de bandes magnetiques peuvent etre connectees a la station de travail comme unites de sauvegarde supplementaires.

Les interfaces disponibles sont classiques. (serie, parallele, DMA, Hardcopy...) Un quadruple port d'entree/sortie serie est gere par un processeur dedie. (Zilog.80) Une memoire partagee de 1 K-octets est utilisee comme port de communication entre les processeurs MC.68000 et Zilog.80. Les dispositifs d'interactions sont clavier de 151 touches dont 24 touches de fonctions programmables, la tablette a numeriser, un photostyle et une manette d'interaction a trois axes X,Y,Z. ( MET 84 )

##### VI.4.2 PARTIE GENERATEUR D'IMAGES :

L'architecture de la partie generateur d'images est assez simple. La memoire d'images est constituee de plans de memoire de trame de 128 K-octets. Une memoire d'images peut etre constituee de huit plans maximum. Le generateur d'images peut gerer au maximum deux memoires d'images soit au total 2 M-octets de memoire. Cette memoire correspond a une resolution d'ecran classique de 1024 \* 768 points visibles de 8 bits de profondeur en version de base. (1024 X 1024 X 8 bits de memoire d'images) La palette des couleurs est de 256 couleurs visualisables parmi 16 millions possibles. On dispose des trois composantes fondamentales le Rouge, le Vert et le Bleu et de 256 nuances par composante. Une table de correspondance de 1 K-octets est dotee de 256 entrees pour les 256 couleurs disponibles.

##### VI.4.3 ETUDES DES PERFORMANCES :

Le processeur graphique du generateur d'images est constitue d'un generateur de vecteurs rapide de 50000 vecteurs par seconde. Un vecteur faisant en moyenne 25 points, la vitesse de trace est de l'ordre de 1,250 million de points par secondes. Un zoom cable en X et Y est mis en oeuvre avec un facteur de grossissement de 0 a 16. Il permet la visualisation en



mode de recouvrement de texte alphanumerique. la gestion de 8 plans de devant (forground) et de 8 plans de derriere (background) permet la superposition et la transparence sur des images. Les plans dans ce mode d'utilisation ont une profondeur de 1 bit.

Certaines extensions au niveau de l'architecture de la partie generateur d'images permettent de mettre en oeuvre des primitives graphiques performantes tel que l'accès direct a la memoire d'images, les rotations cablees, la gestion de la symetrie 2.D/3.D , l'inversion de l'image (video).

En conclusion, cette station de travail est plus modeste du point de vue des ressources du systeme aussi bien au niveau materiel que logiciel. Cependant la simplicité de sa conception est interessante :

- La pre-synthese tri-processeur : MC.68000 / Zilog.80 et bit-slice.
- La post-synthese dotée d'une memoire d'images et de son environnement. (tables de correspondance et interface video)

#### REMARQUE :

CHROMATICS propose la serie 1500 qui est une station de travail plus evoluee inspiree de l'architecture decrite precedemment. Les ameliorations mises en oeuvre sont :

- L'utilisation d'un processeur 32 bit pour la gestion d'une liste de visualisation de 8 M-octets, (32 K-octets de memoires vives a acces rapide),
- La mise en oeuvre d'un "ARRAY PROCESSEUR" effectuant 10 millions d'operations par seconde en virgule flottante sur 32 bits avec memorisation de 64 matrices de transformation de 16 coefficients,
  - . La transformation matricielle 2.D : 2 microsecondes
  - . La transformation matricielle 3.D : 6 microsecondes
- L'utilisation d'un generateur de vecteurs cable : La vitesse de visualisation est de 1 point toutes les 50 nanosecondes.
- La possibilite d'operations logiques ET, OU, NON, au niveau de la memoire d'images,
- L'accès a la memoire d'images par l'intermediaire d'un masque (gestion cablee des textures de traits)
- Le decoupage-2.D cable.

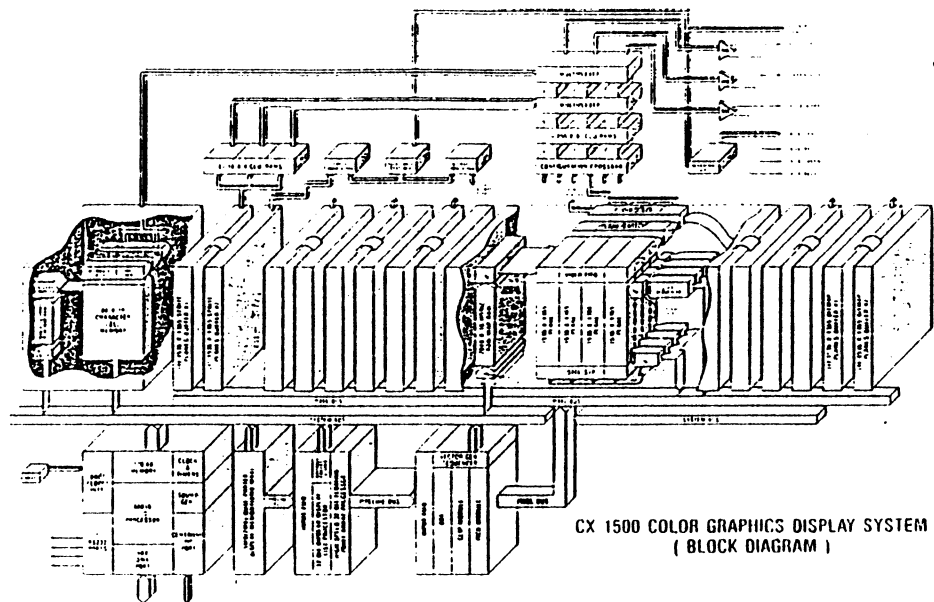


Figure No VI.5

#### VI.5 STATION GRAPHIQUE : SILICON-GRAPHICS

(Serie 2000, 2400 et 2500)

##### VI.5.1 LE COMPOSANT "GEOMETRY-ENGINE" :

L'architecture de cette station de travail est essentiellement basee sur un composant particulier et specifique, le "GEOMETRY-ENGINE" Il s'agit d'un composant "V.L.S.I" developpe a l'universite de STANFORD par James H.Clark ( MGE 84 ) de la compagnie M.V.C. La particularite de ce composant est d'etre configurable du point de vue de sa fonctionnalite. Il est constitue de 7 unites principales :

- 4 unites fonctionnelles,
- Un generateur d'horloge de base,
- Un compteur de microprogrammes ,
- Une pile systeme (gestion des appels de sous-programmes),
- Une memoire de micro-programmes de 40 K-bits.

( ATO 85 ) ( CID 84 ) ( MGE 84 ) ( WIL 85 )

Chaque unite fonctionnelle est constituee d'une U.A.L a virgule flottante a deux operandes. La memoire de micro-programmes contient un micro-logiciel capable d'effectuer une transformation geometrique quelconque en 2.D ou 3.D de l'espace utilisateur vers l'espace-ecran. Les systeme de coordonnees sont definis par l'utilisateur pour des objets graphiques tels que les points, les vecteurs, les polygones, les courbes et les caracteres.

Ces composants V.L.S.I sont mis en oeuvre dans une architecture "PIPELINE". Il s'agit de la station de travail IRIS-1400. La fonctionnalite de chaque "GEOMETRY-ENGINE" est configurable par l'intermediaire d'un registre de fonction. On peut a l'initialisation de la station affecter a chaque groupe de composants "V.L.S.I" une etape d'un processus de visualisation 2.D ou 3.D. Les quatre premiers "G.E" sont affectes a la transformation matricielle flottante, les 6 suivants "G.E" sont affectes au decoupage bi et tri-dimensionnel, les deux derniers "G.E" sont affectes a la transformation-ecran. Le groupe decoupage 3.D associe chaque composant V.L.S.I a une action de decoupage par rapport a la fenetre utilisateur. Le decoupage s'effectue selon les tests de situations suivants:

- Les bord gauche ou droit de la fenetre, .
- Les bord superieur ou inferieur de la fenetre,
- Les plan de devant ou de derriere,

(cf Chapitre IV)

Le groupe transformation-ecran effectue le passage en virgule fixe des differentes coordonnees pour la visualisation dans la cloture d'ecran definie par l'utilisateur. Le resultat est un entier sur 24 bits, une portion etant significative selon la resolution de l'ecran du moniteur T.V de visualisation considere. Des registres de profondeur peuvent etre associes aux points en traitement pour simuler une intensite (nuance) de perspective au niveau de la visualisation.

#### VI.5.2 PARTIE CALCULATEUR :

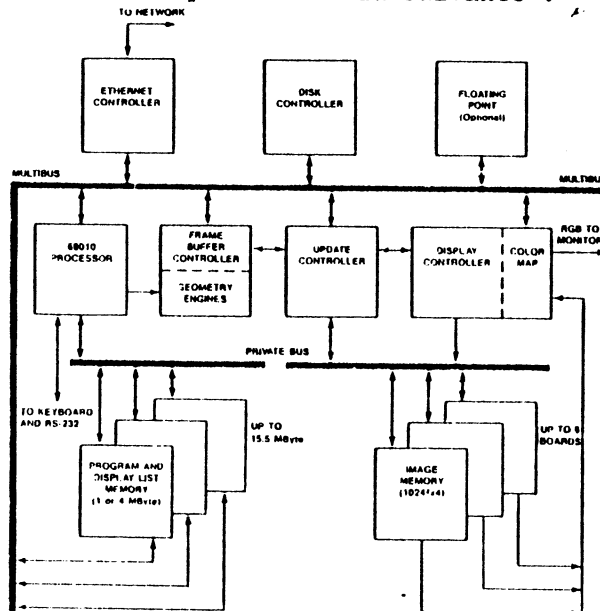
L'architecture globale de la station est tres simplifiee par la presence du "Geometry Engine Pipeline". le processeur principal est un MC.68010 pilote par une horloge a 10 MHz. Le processeur MC.68010 est dote d'un co-processeur flottant optionnel, probablement le MC.68881. Il se charge de la gestion de la liste de visualisation et de la synchronisation avec la partie generateur d'images et le "G.E.P". Il intercepte, decode et execute les commandes graphiques. La memoire vive de la partie calculateur a une capacite de 1 M-octet en version de base, extensible a 4 M-octets. Les entrees/sorties sont disponibles un interface de communication de type

serie, un interface clavier (pupitre d'interaction) et un interface ETHERNET (reseau). Une unite de disques constituant une memoire de masse est proposee en option. De plus l'architecture de la partie calculateur peut etre configuree en bi-processeur par l'ajonction d'un second microprocesseur MC.68010. Le bus general de communication de la partie calculateur est un bus normalise de type "MULTIBUS"

VI.5.3 PARTIE GENERATEUR D'IMAGES :

Elle est constituee de deux sous-systemes le sous-systeme "G.E.P" decrit precedemment et le sous-systeme processeur graphique. Le processeur graphique est mis en oeuvre a partir de processeurs en tranches du type AMD.2903. Il accede a la memoire d'images en effectuant la generation de vecteurs, le remplissage de taches, la generation de caracteres et la gestion des couleurs et des textures. ( AMD 84 )

La memoire d'images est constituee de plans memoire de 1024 X 1024 X 4 bits, une extension est possible, jusqu'a six plans memoires. Soit une memoire d'images de 1024 \* 1024 \* 24 bits, donc une capacite totale de 3 M-octets. Des tables de correspondance gerees par un controleur de mise a jour de la memoire d'images, permettent l'acces aux 16 millions de couleurs possibles de la palette de cette station de travail. Le synoptique de l'architecture generale du systeme est la suivante :



IRIS 2400 System Block Diagram  
Figure No VI.6

## VI.5.4 ETUDE DES PERFORMANCES :

D'après J.H. Clarck le "G.E" effectue une transformation geometrique plus rapidement qu'un VAX-780 et 200 a 400 fois plus rapidement qu'un processeur MC.68000. Le "GEOMETRY.ENGINE" effectue plus de 6 millions d'operations en virgule flottante par seconde. ( CID 83 ) ( ATO 85 )

Cette architecture "PIPELINE" de 12 "GEOMETRY ENGINES" traite en moyenne une scene de 65000 coordonnees par seconde en virgule flottante soit 32500 sommets par seconde en 2.D et 21670 sommets par seconde en 3.D. Cette performance constitue une moyenne de 250 faces 3.D traitees en temps-reel. Le groupe de transformation matricielle effectue des transformation 2.D ou 3.D flottante en 15 microsecondes environ. Cette performance est comparable a celles evaluees dans le chapitre V mais les operateurs consideres n'acceptent que des operandes a virgule fixe. Cette unite est capable d'engendrer des objets graphiques 3.D en calculant un point toutes les 10 microsecondes. Les differentes transformations geometriques du plan et de l'espace sont traitees en temps-reel. De plus, l'architecture specifique du "G.E.P" permet d'utiliser la station comme un processeur "geometrique" ou matriciel general pour des applications variees.

( ATO 85 )

En conclusion, on peut dire que l'IRIS-1400 grace a son architecture originale constituee de trois blocs fonctionnels, est une station de travail aux performances temps-reel exceptionnelles. Le parallelisme mis en oeuvre est inter et intra synthetiseur. Il s'agit d'une architecture de type tri-processeurs dont l'un des processeurs est le "Geometry Engine Pipeline". Les deux autres sont la partie calculateur a base de MC.68010 et la partie generateur d'images a base de microprocesseur en tranches A.M.D. De plus la partie calculateur peut etre dotee d'un co-processeur en virgule flottante.

## VI.6 STATION GRAPHIQUE : APOLLO-DN.550-660

## VI.6.1 PARTIE CALCULATEUR :

La partie calculateur est de type bi-processeur. Un processeur central assurant la gestion des differentes composantes de l'architecture. Un processeur specialise dedie au calcul en arithmetique flottante, realise a partir de microprocesseurs en tranches de la famille A.M.D. Elle est architecturee autour d'un microprocesseur MC.68010 dont l'horloge de base est a 10 MHz. Les versions prochaines seront equipees d'un veritable processeur 32 bits, il s'agit du MC.68020 dont l'horloge de base est a 16 MHz accompagne d'un co-processeur flotant le MC.68881. La memoire vive de la partie calculateur a une capacite minimale de 1 M-octets, extensible a 3 M-octets. La memoire de masse est constituee d'un disque dur de 50 M-octets et d'une unite de cartouches amovibles de 45 M-octets. Ces deux unites sont a la charge du processeur central. Les differents interfaces disponibles sont deux lignes de communication serie a 19.2 K-bauds, un interface reseau de type ETHERNET etc ...

## REMARQUE :

Les stations APOLLO peuvent être organisées dans une configuration de réseau local ou le partage des ressources est entièrement banalisée au niveau de chacun des nœuds. Cette possibilité de décentralisation est offerte par le réseau spécifique DOMAIN. Elle constitue l'atout principal des stations graphiques APOLLO. La communication interne se fait à travers un bus normalisé de type IEEE.796 MULTIBUS. Quatre connecteurs d'extension sont disponibles en fond de panier pour des applications spécifiques de l'utilisateur. ( JCK 85 )

## VI.6.2 PARTIE GENERATEUR D'IMAGES :

Elle est constituée d'une architecture à base de microprocesseurs en tranches (série AMD.29116) gérant une mémoire d'images de capacité de 2 M-octets. ( AMD 84 ) ( CHU 84 ) Cette mémoire d'images est organisée en 1024 X 2048 points X 24 bits. La configuration de base est un plan mémoire extensible à huit plans. La palette des couleurs est de 256 parmi 16 millions de couleurs possibles. Le remplissage de la mémoire d'images se fait par compression de données. Il existe l'option de transfert appelée "Bit Block Transfert" qui permet d'atteindre des vitesses de transmission moyennes de l'ordre de 160 M-bits par seconde. Cette figure récapitulative présente l'architecture générale de la station APOLLO :

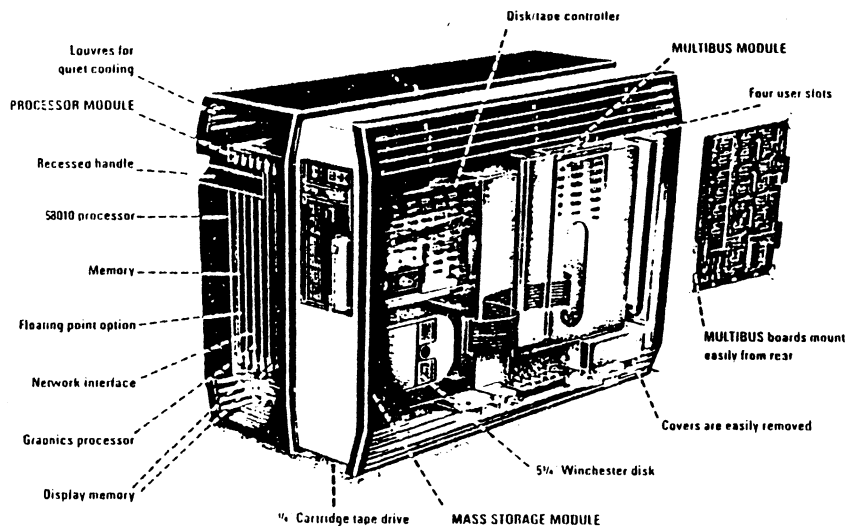


Figure No VI.7

VI.6.3 ETUDE DES PERFORMANCES :

Les performances graphiques indiquees par le constructeur sont la generation de vecteurs a 1 million de points par seconde, le remplissage de taches a 320 millions de points par seconde et les operations logiques sur la memoire d'images a 100 M-bits par seconde. Pour les differents logiciels graphiques manipulent des images 2.D et 3.D, Les phases de transformation matricielle et de decoupage 2.D se font a la vitesse de 10000 vecteurs par seconde.

En conclusion, il faut signaler la simplicite de conception de cette station dont l'atout principal est le reseau DOMAIN. L'architecture est de type Tri-processeurs. La pre-synthese est dotee d'une puissance de calcul toujours accrue. (cas de la station DN.660, 1 a 5 Mips). Au niveau de la post-synthese, l'operateur de remplissage des memoires d'images possede l'option de compression de donnees. Cette caracteristique couplee aux performances du processeur en tranches AMD.29116 permet d'obtenir des vitesses de generation tres appreciables.

## CONCLUSION :

En conclusion cette etude succincte de quatre stations de travail "concurrentes" de la station GETRIS, fait apparaitre quelques points de reflexion. La partie calculateur est souvent architecturee autour d'un bus normalise de type Q-BUS, MULTIBUS, VME BUS, afin de disposer d'une modularite et d'une extensibilite pre-existante. Les cartes sont developpees par le constructeur ayant adopte une norme. De plus cette option permet de doter la station de systemes d'exploitation standard tels que UNIX et UNIX-like, CPM, MSDOS, VMS...

Les logiciels graphiques proposes sont en partie developpes par le constructeur de la station, il s'agit du niveau 1. le niveau 2 peut etre un logiciel normalise du type GKS, GRAF, GSPC, IGL, PHIGS ou CLOVIS, ou encore d'une bibliotheque specifique a la station donc tres performante, (cas de GETRIS) accessible a partir d'un langage evolue qui peut etre le FORTRAN 77, Le PASCAL, et tres souvent Le langage C.

La partie calculateur est caracterisee par l'extensibilite de sa memoire vive (1 a quelques Mega-octets), de ses entrees/sorties serie et parallele, de ses memoires de masse (1 a 4 unites de disques durs et souples) et la possibilite de rajouter des co-processeurs classiques de sous-traitance. (arithmetique flottante) ou des co-processeurs specialises (multiplieur , processeur matriciel ...)

La partie generateur d'images est constituee d'une memoire de trame extensible en resolution et en profondeur. La palette des couleurs geree par des tables de correspondance fait 24 bits en moyenne. Cette memoire et ces tables de correspondance sont accedees et gerees par des processeurs graphiques de la partie generateur d'images :

- Co-processeurs monolithiques : EF.936X / NEC.7220 / HITACHI 63484 etc.
- Microprocesseurs en tranches, (famille A.M.D)
- Logique cablee discrete,

Le choix d'une technologie de realisation est un compromis entre trois criteres qui sont le cout, la flexibilite et la performance. Cette evolution montre a quel point le T.G.I devient un veritable calculateur dedie a la synthese d'images. Pour une configuration "station de travail", le calculateur hote constitue un processeur supplementaire que l'utilisateur peut integrer librement a l'architecture parallele.

( CIG 84 )





CONCLUSION

REFERENCES BIBLIOGRAPHIQUES :

( BED 81 ), ( CIG 84 ), ( DOD 84 ), ( DSS 85 ),  
( GAU 85 ), ( HOF 84 ), ( INP 85 ), ( MCC 84 ),  
( MOK 84 ), ( WIL 85 ).

## CONCLUSION

Globalement l'architecture d'un systeme graphique interactif est de plus en plus complexe. Deux phenomenes justifient cette complexitee croissante. Le premier est l'integration de processus de synthese d'image evolues au niveau des fonctions locales proposees par le terminal. il s'agit en particulier des transformations geometriques, des operations de coupage et d'elimination des partie cachees, et de la gestion des attributs d'aspect. Le second est la vulgarisation des composants V.L.S.I tels que les processeurs standards, les memoires vives, les interfaces et les co-processeurs specialises.

Un systeme graphique de haut de gamme est constitue en moyenne de 150 a 200 circuits integres dont la majorite sont des memoires vives. Le controle de cette "proliferation materielle" pose un certain nombre de problemes relatifs au cout de l'architecture, au volume de la machine, et aux performances.

Le probleme crucial actuel du developpement de cette tendance est que le marche du V.L.S.I graphique n'est pas suffisamment porteur pour pousser les constructeurs de semi-conducteurs a developper des composants specifiques. ( CIG 84 )

Seules quelques exceptions existent et demontrent la pre-occupation de quelques constructeurs mondiaux aussi bien au niveau de la pre-synthese qu'au niveau de la post-synthese, en voici des exemples :

- L'Iris 1400 dote des "Geometry Engine",
- Le Selliac-7 dote d'un multiplieur matriciel 4 X 4,
- La cellule C.S.I configurable pour la synthese d'image,
- La famille AM.815-X : Advanced Micro-Device.
  - . Le gestionnaire de la memoire d'image : AM.8150 (Display Refresh Controler).
  - . Le registre de serialisation : AM.8157 (Video Shift Register).
  - . Le module de synchronisation video : AM.8158 (Video Timing Controler).
  - . Le circuit "palette" : AM.8158

- Le EF.9367, circuit "palette" de Thomson.

La premiere tendance V.L.S.I concerne l'utilisation des techniques de "Gate Array". La terminologie francaise est "reseau pre-diffuse" Cette technique permet la conception de circuits integres de type V.L.S.I a la demande et specifique appellees "semi-custom components" par opposition aux circuits classiques appellees "full-custom components" Ces deux dernieres annees l'utilisation des reseaux pre-diffuses est devenue assez courante. Le concepteur dispose de bibliotheques de reseaux appropriees permettant l'integration d'une logique cablee discrete. L'avenement du circuit integres a la demande peut etre une solution pour le developpement d'architectures de systeme graphiques modulaires, performants, fiables et economiques.

Ainsi un reseau pre-diffuse de 1000 cellules peut remplacer une logique cablee discrete de 50 circuits integres S.S.I ou M.S.I. Cette technologie se repend lentement car les couts de developpement sont assez importants. Soit environ 500 K.F pour une "puce" de moyenne integration. Il est certain que les architectures futures des stations de travail, peuvent etre simplifiees et seraient plus performantes, en particulier au niveau de la post-synthese ou la logique cablee discrete est importante. Ainsi la partie generateur d'images peut etre rammenee a quelques V.L.S.I.

Au niveau de la partie calculateur, la disponibilite des ressources n'est pas remise en cause, par contre des operateurs de calcul peuvent etre integres. Le cas de la transformation matricielle en general constitue a notre sens une operation de calcul realisable en reseaux pre-diffuses. Il existe deja des processeurs flottants realises en reseau pre-diffuse effectuant 1 million d'operations par seconde sur des operandes de 16 et 32 bits.

La resolution de l'image et la qualite des couleurs ne peuvent que progresser, etant donne la regression des couts des composants memoire. Bien sur la technologie des dispositifs d'affichage doit suivre cette evolution. Le probleme crucial est le choix d'une architecture ayant des performances en temps-reel acceptables pour des images dont le degre de realisme est suffisant. Ces qualites sont obtenues si une repartition judicieuse des taches est effectuee.

Les stations de travail constituent une option satisfaisante actuellement. De plus, la combinaison des V.L.S.I standards dont les performances sont en constante progression et des V.L.S.I a la demande en reseaux pre-diffuses constitue une voie d'exploration a ne pas negliger afin de negocier au mieux le compromis TEMPS-REEL ET REALISME.

En conclusion finale de cette etude, il semble que les evaluations presentees dans ce document ont permis de caracteriser les operations penalisantes lors de la phase de pre-synthese d'images. Ainsi une option est prise pour la conception d'un co-processeur de calcul matriciel pour les versions futures de la station GETRIS II. Ce co-processeur sera realise a l'aide de composants en tranche de la famille A.M.D. Au niveau de l'optimisation des echanges geres par le module de communication, un mecanisme base sur une memoire a acces rapide de type pile, a permis une amelioration des performances graphiques de base. (generation de vecteurs, de cercle, de caracteres, remplissage de polygones...)

Ces choix judicieux ont pu etre retenus apres avoir "radiographier" les processus de synthese d'images aussi bien au niveau materiel qu'au niveau logiciel. De plus l'architecture de la post-synthese a ete sensiblement amelioree. (capacite des plans memoires, acces directs aux processeurs, recadrage des performances pour le calcul de l'eclairage, possibilites de seuillage et de zonage). Un module supplementaire et optionnel d'acquisition de video analogique en couleur pour la station GETRIS a ete realise. Ainsi le melange d'images synthetiques et d'images reelles est possible.

( ZAR 85 )

Des outils logiciels pour le pilotage de support de memorisation tel que le magnetoscope, sont en cours d'elaboration, l'objectif etant de constituer un poste complet de production d'images pour des domaines d'applications audio-visuels. Il semble que l'orientation de la station GETRIS se precise au sein du laboratoire ARTEMIS, l'aspect utilisateur reste a developper...

## BIBLIOGRAPHIE

- [ AIF 84 ] : Advanced Information INMOS  
IMST.424 Transputer.  
INMOS Corporation Colorado Springs USA 1984.
- [ AMD 84 ] : AMD Data sheet  
AMD.29116 architecture speed pixel manipulation in  
interactive bit-mapped graphics. A high performance  
8 bits slice microprogram sequencer with its related  
support circuits.
- [ ATO 85 ] : A Technical Overview  
Silicon Graphics sales and Services offices.  
Iris Series 2000 Dallas Septembre 1895.
- [ BED 81 ] : A Bellon G Dupenloup  
Les reseaux prediffuses.  
Rapport de recherche No-359 Mars 1983.
- [ BGR 79 ] : C Brie R Gerber  
Les microprocesseurs en tranches.  
Techniques et Documentations Paris 1979.
- [ BIN 78 ] : J.F Blinn M.E Newell  
Clipping using homogeneous coordinates.  
Computer Graphics SIGGRAPH-ACM Vol 12 No 3 78.
- [ BLZ 84 ] : A Boulze  
Le logiciel pilote D'HELIOS integre a CLOVIS.  
Rapport interne ARTEMIS Fevrier 1984.
- [ BLZ 84 ] : A Boulze  
Dans le cadre de CLOVIS, un logiciel pilote du terminal  
de synthese d'images realistes HELIOS.  
These d'ingenieur CNAM Octobre 1984 Grnoble.
- [ BOU 80 ] : Ph Boulle  
Etude et realisation d'algorithmes pour la visualisation de  
scenes tridimensionnelles.  
These de Docteur-ingenieur septembre 1980 Grenoble.
- [ BOU 83 ] : Ph Boulle  
Une nouvelle approche des problemes lies a la description de  
scenes tridimensionnelles.  
Rapport de recherches No-359 IMAG Mars 1983 Grenoble.
- [ BOU 84 ] : Ph Boulle  
Etude de fesabilite d'un processeur graphique d'animation de  
scenes tridimensionnelles.  
Rapport interne ARTEMIS Janvier 1984.

- [ BRE 65 ] : J.E Bresenham  
Algorithm for Computer control of a digital plotter  
IBM system journal Vol 4 No 1 65
- [ CHR 84 ] : Chromatics systems.  
Color graphic display systems of the CX-1500 series.  
Documentation technique.
- [ CHM 83 ] : P. Chu & W. Miller  
Microprocesseur architecture suits bi-mapped graphics.  
Electronic Design January 20 - 1983
- [ CIG 84 ] : B.J Shachter  
Computer image generation.  
Willey-Interscience N Y 1984
- [ CID 83 ] : J.H Clark T Davis  
Work station unites real-time graphics with Unix and  
Ethernet.
- [ CLA 76 ] : J.H. Clark  
Hierarchical model for visible surface algorithms.  
ACM Communications Vol 19, No 10, Octobre 1976
- [ CON 82 ] : Convention de recherche No-680.712  
Convention de recherche No-82.225  
Developement d'un systeme interactif pour la synthese  
d'images.  
Premier et Deuxieme rapport Decembre/Juillet 1982 Grenoble.
- [ CHU 84 ] : P Chu W Miller  
AMD.29116 Architecture speed pixel manipulation in  
interactive bit-mapped graphics.  
Technical data sheet AMD 1984.
- [ DOD 84 ] : D.J Doornik J.C Dalrymple  
The architectural evolution of a high performance  
graphics terminal. Tektronix Inc  
IEEE and Computer Graphics Applications April 1984.
- [ DSS 85 ] : A Daniel A Strupat A Sfarti  
Un jeu de circuits integres pour les systemes graphiques  
a haute performance.  
Electronique industrielle No 83 Fevrier 1985.
- [ DUG 82 ] : R Dubois D Girod  
Les microprocesseurs 16 bit a la loupe.  
Col Eyrolles 1982.

- [ ELE 84 ] : Electron  
Le processeur de signal TMS.320, la famille de processeurs  
du signal 16 / 32 bits.  
Revue technique No 546 Fevrier 1984.
- [ FER 81 ] : F.N Ferreira  
Conception et realisation d'un systeme interactif pour  
la synthese d'image realiste : HELIOS.  
These de Docteur-ingenieur septembre 1981 Grenoble
- [ FON 83 ] : A.B Fontaine  
Le microprocesseur 16 bits Intel-8086 Materiel Logiciel  
et systeme d'exploitation.  
Ed Masson 1983
- [ FPI 84 ] : A Fujimoto C.G Perott K.Iwata  
A 3.D Graphics Display System with depth buffer and  
pipeline processor.  
IEEE and Computer Graphics applications June 1984
- [ GEG 85 ] : Ph Genoud J.F Grabowieky  
A general purpose graphic environment CLOVIS.  
IASTED IMAG 1985.
- [ GEN 84 ] : Ph genoud  
Modelisation tri-dimensionnelle et modelisation geometrique.  
Rapport interne ARTEMIS 1984 Grenoble.
- [ GRA 80 ] : M Grave  
Etude d'un noyau de systeme de synthese d'images.  
Application a la visualisation de scenes tridimensionnelles  
These de Docteur-ingenieur Decembre 1980 Lille.
- [ GAU 85 ] : A Gaulene  
Terminaux graphiques couleur. Performances et tendances.  
Le Nouvel Automatisme Juin 85.
- [ GRA 80 ] : R. Gray  
Bit-map architecture realizes raster display potential.  
Computer Design July 1980.
- [ GRO 85 ] : C. Gross  
Des processeurs graphiques pour des images plus vivantes.  
Electronique industrielle No 83 Janvier 1985.
- [ HUI 84 ] : J. Huisman  
Pipe-line modele Promotes Extensible Display System.  
Computer Design May 1984



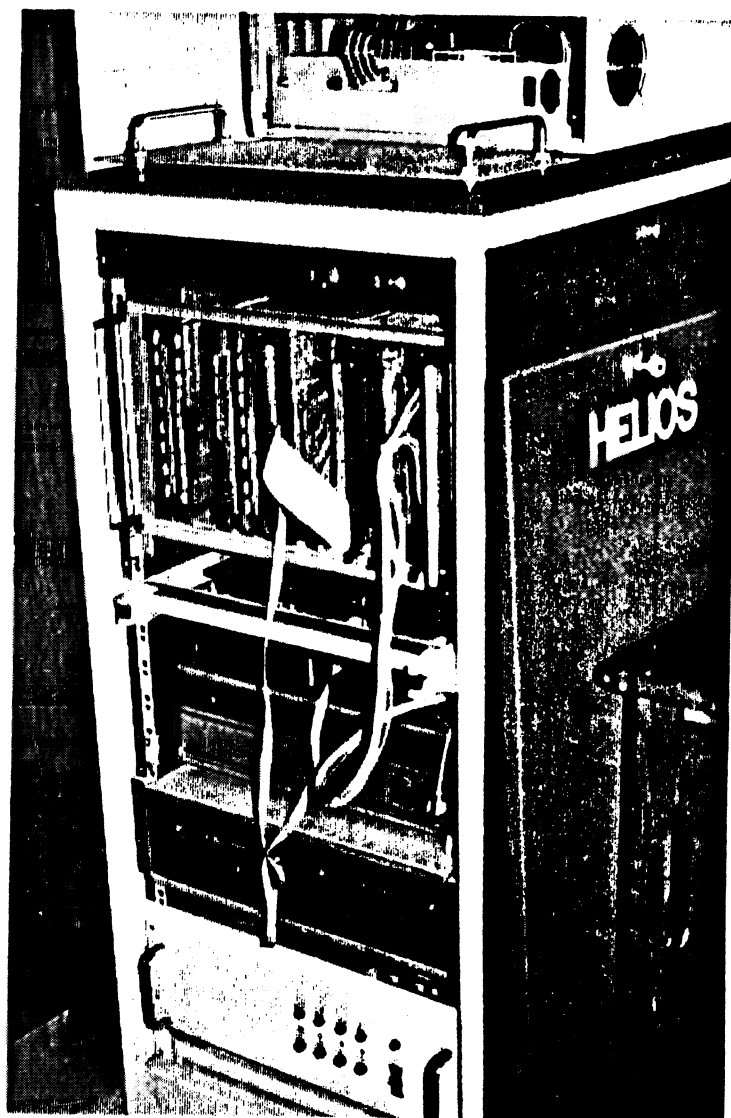
- [ HOF 84 ] : T.R Hoffman  
The role of LSI and VLSI in computer image generation  
Willey-interscience N Y 1984.
- [ IKE 84 ] : T Ikedo  
High speed techniques for 3.D color graphics terminal.  
Seillac Corporation.  
IEEE and Computer graphics applications June 1984.
- [ INP 85 ] : Support de cours  
La conception logique et topologique de circuits  
integres complexes.  
Fascicule IV ENSIMAG Module A-1.
- [ ITM 85 ] : Dossiers techniques  
La station de creation d'image synthetique en temps-reel  
GETRIS. ( ITMI et EFISYSTEME )
- [ JCK 85 ] : J.C Jackelin  
Dossiers techniques de description des architectures des  
stations de travail DN.550 et DN.660.  
APOLLO COMPUTERS Documentation Novembre 1984.
- [ JOL 84 ] : P Jolland  
Le microprocesseur 16 bits MC.68000 et sa programmation.  
Ed Eyrolles Paris 1984.
- [ LAG 85 ] : G Lagrange  
Comment choisir un terminal graphique.  
Electronique industrielle No 83 Fevrier 1985.
- [ LUC 77 ] : M Lucas  
Contribution a l'etude des techniques de communication  
graphique avec un ordinateur. Elements de base des  
logiciels graphiques interactifs.  
These de Doctorat d'etat Decembre 1977 Grenoble.
- [ MAT 78 ] : P Matherat  
A chip for low-cost raster scan graphic display.  
Computer Graphics SIGGRAPH-ACM Vol 12 No 3 1978.
- [ MAR 78 ] : F Martinez  
Les concepts lies a la description de l'animation.  
Exemples de realisation.  
Seminaire d'informatique Mars 1978.
- [ MAR 82 ] : F Martinez  
Vers une approche systematique de la synthese d'image.  
Aspect materiel et logiciel.  
These de Doctorat d'etat Novembre 1982 Grenoble.

- [ MAR 84 ] : F Martinez  
La synthese d'image Concepts materiel et logiciel.  
Col Editest 1984.
- [ MAR 84 ] : F Martinez  
Etude comparative des systemes actuels de synthese d'image.  
Rapport interne ARTEMIS Grenoble 1984.
- [ MCC 84 ] : J.J McC Ormick  
Present futur color display technologies for graphics.  
Computer and Graphics Vol 8 No 3 1984
- [ MER 79 ] : M Meriaux  
Etude et realisation d'un terminal graphique interactif  
couleur tridimensionnel fonctionnant par tache.  
These de Doctorat ingenieur Janvier 1979 Lille.
- [ MET 84 ] : Metheus  
Graphic work station Metheus  
Metheus Corporation Oregon.
- [ MGE 84 ] : W Myers  
Staking aut the graphics display pipeline.  
Special feature of pipelined architectures.  
IEEE and Computer Graphics Applications June 1984.
- [ MMC 84 ] : Revue Mini et Micro  
Panorama des co-processeurs pour le calcul arithmetique.  
Les multiplieurs numeriques. Principes et Panorama.  
Un processeur en arithmetique flottante : Le NS.16081.  
( Numeros 128, 136, 216 )
- [ MOK 84 ] : N. Mokhoff  
Thirty-two bit micros power workstations.  
Computer Design June 1984.
- [ MOL 76 ] : P Morvan M lucas  
Image et ordinateurs. Introduction a l'infographie  
interactive.  
Lib Larousse Ed 1976.
- [ MOT 79 ] : Motorola data sheet  
16 bits microprocessor user'manual.  
Motorola previous product data sheet, the floting  
point processor MC.68881.
- [ MYE 84 ] : W. Myers  
Staking out Graphics Display Pipe-line.  
IEEE Computer Graphics and Applications July 1984.

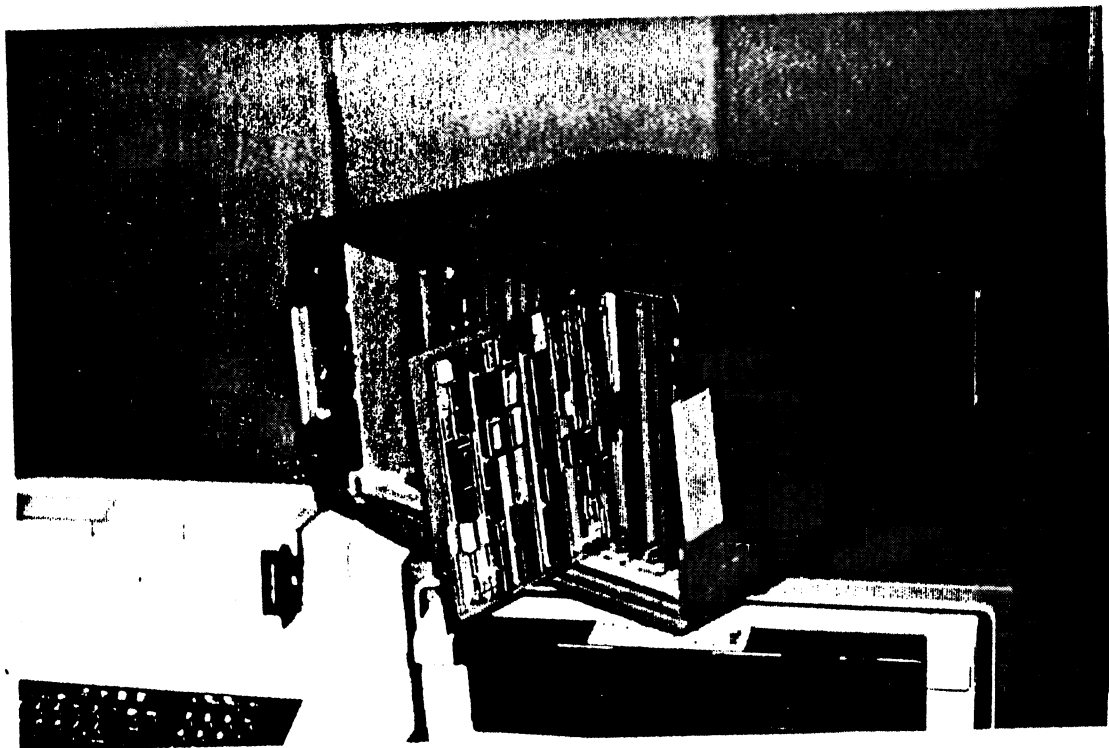
- [ NEB 77 ] : M. Newell & J. Blinn  
The progression of realism in computer generated images.  
Proc Annal conference A.C.M October 1977 Seattle.
- [ NES 79 ] : W Newman R.F Sproul  
Principles of interactive computer graphics  
2nd ed Mc G.Hill N.Y 1979.
- [ NIK 84 ] : R. Nickel  
The Iris Workstation.  
IEEE Computer Grphics and Applications August 1984
- [ RAG 83 ] : S Ragab  
Etude et realisation d'un systeme multiprocesseur  
pour le traitement des algorithmes paralleles.  
These de Doctorat-ingenieur Octobre 1983 Grenoble.
- [ ROU 86 ] : F ROUSEE  
Un processeur modulaire pour la synthese d'image.  
Seminaire de l'image electronique  
Deuxieme colloque images. Nice Avril 1986
- [ RTS 83 ] : Raster technology system  
Model one graphics systems and display controller.  
Technical revue Decembre 1983.
- [ SAR 82 ] : M.T SARRASIN  
Analyse et implantation du logiciel pilote d'un  
synthetiseur d'image : HELIOS.  
These d'ingenieur CNAM Octobre 1982.
- [ SMR 85 ] : M.K Stojcev Ph.D Milonanovic Z.C Radonjic  
Some shifting methods for matrix multiplication.  
IEEE Procedings Vol 132 January 1985.
- [ SUH 74 ] : I.E Sutherland G.W Hodgman  
Reentrant polygon clipping.  
CACM Vol 17 No 1 January 1974
- [ VIA 85 ] : J Viard  
Etude des specifications d'une station de travail  
pour la CAO de VLSI.  
Rapport de DEA Juin 1985.
- [ WES 81 ] : R.W Blasco V Coli C Hastings S Rajpal  
Supercharges MC.68000 arithmetics.  
Monolithic Memories revue.

- [ WIL 85 ] : T Williams  
Graphics processing migrate from host to work station.  
Computer Design July 1985.
- [ ZAR 84 ] : V.H Zarate Silva  
Architectures modulaires d'HELIOS.  
Rapport interne ARTEMIS Grenoble 1984.
- [ ZAR 85 ] : V.H Zarate Silva  
Etude et réalisation d'un synthetiseur d'images  
base sur une architecture banalisée.  
These de Docteur-ingenieur Octobre 1985 Grenoble.

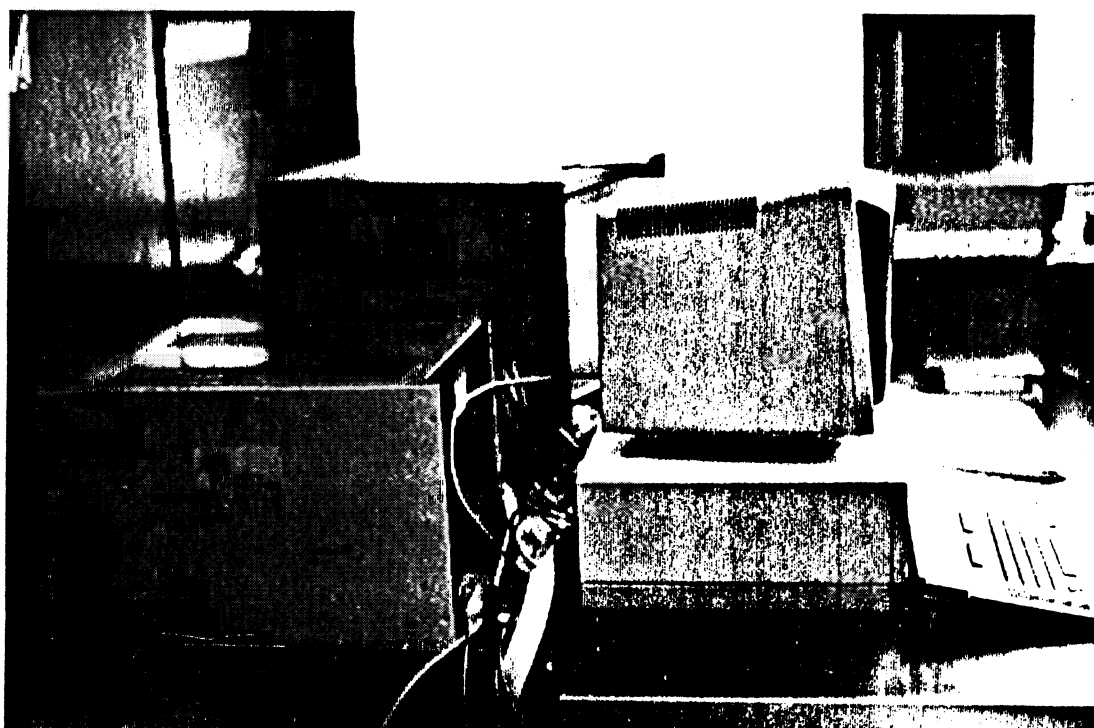
LA CONFIGURATION : CALCULATEUR SATELLITE



LA CONFIGURATION : CONSOLE EVOLUEE



LA CONFIGURATION : STATION DE TRAVAIL



```

1 :
2 : PROGRAMME CALCULANT LA TRANSFORMATION MATRICIELLE
3 :
4 : D' ENSEMBLE DE SOMMETS D'UNE SCENE 3.D
5 :
6 : S[I].X.Y.Z X MATRANS -> NEWS[I].X.Y.Z
7 :
8 :
9 :
10 : XREF DEBFAC,JDTFAC,IDSOMT,NEWSOM
11 : XREF OMEGA,VISIB,LONGS,NFACE
12 : XREF AVAL,BVAL,RVAL,MATRIC
13 : CALMAT ASECT 2
14 :
15 : ORG 0600000H
16 :
17 : PRODMAT EQU *
18 : MOVEQ #02.DO
19 : CLR D5
20 : CLR D6
21 : LEA.L IDSOMT,A0 ; TABLE DES SOMMETS
22 : LEA.L NEWSOM,A1 ; TABLE DES SOMMETS
23 : ADD.L D6,A1 ; TRANSFORMES
24 : MOVE.W LONGS,D1 ; NOMBRE DE SOMMETS
25 : TRCORD LEA.L MATRIC,A2 ; ADRESSE DE LA MATRICE
26 : ADD.L D5,A2
27 : CLR D4
28 : MOVEQ #2,D2 ;
29 : MOVE.W (A0+),D3 ; PRODUIT D'UNE LIGNE
30 : MULS.W (A2+),D3 ; DE LA MATICE PAR UNE
31 : ADD.L D3,D4 ; ORDONNEE D'UN SOMMET
32 : DBRA D2,TRLIG ; X , Y , Z
33 : CLR.L D3
34 : MOVE.W (A2),D3 ; CUMUL DE LA COMPOSANTE
35 : ADD.L D3,D4 ; HOMOGENE =1
36 : MOVE.L D4,(A1+) ; SAVE DE LA NOUVELLE
37 : ADDO #8,A1 ; ORDONNEE DU SOMMET
38 : DBRA D1,TRCORD ;
39 : ADDO #8,D5
40 : ADDO #4,D6 ; PASSAGE A L'ORDONNEE
41 : DBRA DO,TRSDMT ; SUIVANTE DU SOMMET
42 : TRAP #14
43 :
44 : END
    
```

PAS D ERREUR.



```

00608000
00608000
608000 3 4282
608002 3 4280
608004 3 R 41F900000000
60800A 3 R 43F900000000
608010 3 R 323900000000
608016 3 R 343900000000
60801C 3 2618
60801E 3 4483
608020 3 0682
608022 3 87C2
608024 3 32C3
608026 3 5048
608028 3 51C9FFF2
60802C 3 4E4E

1 : CE PROGRAMME CALCULE LES VALEURS DU
2 :
3 :
4 : FACTEUR W POUR CHAQUE SOMMET S(X,Y,Z)
5 :
6 : APRES LA TRANSFORMATION MATRICIELLE
7 :
8 : W = (A-X) / A
9 :
10 :
11 CALONG ASECT 3
12 :
13 XREF DEBFAC.IDTFAC.IDSOMT.NEWSOM
14 XREF OMEGA.VISIB.LONGS
15 XREF NFACE.AVAL.BVAL.RVAL
16 :
17 ORG 0608000H
18 :
19 :
20 COMEGA EQU #
21 CLR.L D2
22 CLR.L DO
23 LEA.L NEWSOM.AO
24 LEA.L OMEGA.A1
25 MOVE.W LONGS.D1
26 MOVE.W AVAL.D2
27 NEXTSO MOVE.L (AO+).D3
28 NEG.L D3
29 ADD.L D2.D3
30 DIVS.W D2.D3
31 MOVE.W D3.(A1+)
32 ADDQ #8.AO
33 DBRA D1.NEXTSO
34 TRAP #14
35 :
36 END

: TABLE DES SOMMETS TRANSFORMES
: TABLE DES OMEGAS CLCULES
: NOMBRE DE SOMMETS
: AVAL : DISTANCE POINT DE VISEE-OBSERVATEUR
: D3=-X
: D3=A-X
: D3=A-X/A
: SAVE W
: NEXT X (LONG=32 BITS +8)

```

PAS D ERREUR.



```

56 : CE SOUS-PROGRAMME REALISE LA VISUALISATION SUR
57 :
58 : LE SYNTHETISEUR HELIOS DE LA PORTION D'ARETE VISIBLE
59 :
60 VISUH EOU *
61 :
62 MOVEM.L AO,A1,DO,(-A7) : SAVE
63 MOVEQ #3,DO : TRANSFERT DE 4 COORDONNEES
64 LEA.L VISDY,AO : 16 BITS SIGNIFICATIFS POUR VISUH
65 LEA.L YCOLEF,A1
66 :
67 RETRAN ADDQ.L #2,AO : POSITIONNEMENT SUR F.W
68 MOVE.W (AO+),A1+ : VECTEUR ( VISDY,VISOZ,VIZEY,VIZEZ )
69 DBRA DO,RETRAN : NEXT
70 :
71 : VISUALISATION SIMPLE POUR L'EVALUATION
72 :
73 JSR VECTEUR : TRACE DE L'ARETE VISIBLE
74 MOVEM.L (A7+),AO,A1,DO : RESTAURE
75 :
76 RTS
77 :
78 :
79 :
80 :
81 : CE SOUS-PROGRAMME ANALYSE UNE ARETE DETERMINE LA
82 : PARTIE VISIBLE DANS LA FENETRE (EVENTUELLEMENT)
83 :
84 : TSTCUT= 0: ARETE REJETEE 1: ARETE COUPEE A VISUALISER
85 :
86 : LES VARIABLES UTILISEES:
87 : TSTCUT BOOLEEN INDICANT SI LE COUPAGE DE L'ARETE EST REUSSIE
88 : VISDY ORDONNEE EN Y DE L'ORIGINE DE L'ARETE A VISUALISER
89 : VISOZ ORDONNEE EN Z DE L'ORIGINE DE L'ARETE A VISUALISER
90 : VISEY ORDONNEE EN Y DE L'EXTREMITEE DE L'ARETE A VISUALISER
91 : VIZEZ ORDONNEE EN Z DE L'EXTREMITEE DE L'ARETE A VISUALISER
92 : FORGD PLAN DE DEVANT POUR LE COUPAGE // Y.O.Z EN X
93 : BACGD PLAN DE DERRIERE IDEM
94 : AROUND PARAMETRE INTERVENANT DANS L'INTERPOLATION LINEAIRE
95 :
96 :
97 CUTART EOU *
98 :
99 : A4= ADRESSE DU SOMMET I (EXTREMITTE GAUCHE) ORIGINE LEFT
100 : A5= ADRESSE DU SOMMET J (EXTREMITTE DROITE) EXTREMITTE RIGHT
101 :
102 : VISDY,VISOZ: COORDONNEES DU SOMMET ORIGINE DE L'ARETE VISIBLE
103 : VISEY,VIZEZ: COORDONNEES DU SOMMET EXTREMITTE DE L'ARETE VISIBLE
104 :
105 MOVEM.L ALL,(-A7) : SAUVEGARDE DU CONTEXTE
106 CLR.B TSTCUT : ARETE SUPPOSEE NON-VISIBLE
107 MOVE.L (A4),DO : DO=XI
108 MOVE.L 4(A4),D1 : D1=YJ SOMMET I
109 MOVE.L 8(A4),D2 : D2=ZI
110 MOVE.L (A5),D3 : D3=XJ
    
```

0060FC00

60FC00 8 48E780C0  
 60FC04 8 7003  
 60FC06 8 R 41F900000000  
 60FC0C 8 R 43F900000000

60FC12 8 5488  
 60FC14 8 32D8  
 60FC16 8 51C8FFFA

60FC1A 8 R 4EB900000000  
 60FC20 8 4CDF0301

60FC24 8 4E75

0060FC26

60FC26 8 48E7FFFF  
 60FC2A 8 R 423900000000  
 60FC30 8 2014  
 60FC32 8 222C0004  
 60FC36 8 242C0008  
 60FC3A 8 2615

60FC3C	8	282D0004	MOVE.L	4(A5).D4	: D4=YJ	SOMMET J
60FC40	8	2A2D0008	MOVE.L	8(A5).D5	: D5=ZJ	
60FC44	8	3C12	MOVE.W	(A2).D6	: D6=WI	
60FC46	8	3E13	MOVE.W	(A3).D7	: D7=WJ	
111	:		SEQUENCE COUPAGE PAR RAPPORT A X (OMEGA)			
112	:					
113	:					
114	:					
115	:					
116	:					
117	:					
118	:					
119	:					
120	:					
121	:					
122	:					
123	:					
124	:					
125	:					
126	:					
127	:					
128	:					
129	:					
130	:					
131	:					
132	:					
133	:					
134	:					
135	:					
136	:					
137	:					
138	:					
139	:					
140	:					
141	:					
142	:					
143	:					
144	:					
145	:					
146	:					
147	:					
148	:					
149	:					
150	:					
151	:					
152	:					
153	:					
154	:					
155	:					
156	:					
157	:					
158	:					
159	:					
160	:					
161	:					
162	:					
163	:					
164	:					
165	:					





```

00600000
600000 1 7002
601770 1 4245
601D4C 1 4246
60348C 1 41F900601D4C
60639C 1 43F90060348C
60686C 1 D3C6
60733C 1 32390060733C
60733E 1 45F900607346
607340 1 D5C5
607342 1 4244
607344 1 7402
607346 1 3618
00607366
607366 1 7002
607368 1 4245
60736A 1 4246
60736C 1 41F900601D4C
607372 1 43F90060348C
607378 1 D3C6
60737A 1 32390060733C
607380 1 45F900607346
607386 1 D5C5
607388 1 4244
60738A 1 7402
60738C 1 3618
60738E 1 C7DA
607390 1 D883
607392 1 51CAFFF8
607396 1 4283
607398 1 3612
60739A 1 D883
60739C 1 22C4
60739E 1 5049
6073A0 1 51C9FFDE
6073A4 1 5045
6073A6 1 5846
6073A8 1 51C8FFC2
6073AC 1 4E4E

1 : PROGRAMME CALCULANT LA TRANSFORMATION MATRICIELLE
2 :
3 :
4 : D' ENSEMBLE DE SOMMETS D'UNE SCENE 3.0
5 :
6 : S[I].X.Y.Z X MATRANS -> NEWS[I].X.Y.Z
7 :
8 :
9 START ASECT 1
10 :
11 : ORG 0600000H
12 DEBFAC DS.B 6000
13 IDTFAC DS.B 1500
14 IDSOMT DS.B 6000
15 NEWSOM DS.B 12000
16 OMEGA DS.B 2000
17 VISIBL DS.B 2000
18 LONGS DS.B 2
19 NFACE DS.B 2
20 AVAL DS.B 2
21 BVAL DS.B 2
22 RVAL DS.B 2
23 MATRIC DS.B 32
24 :
25 PRODMAT EQU *
26 MOVEQ #02.D0
27 CLR D5
28 CLR D6
29 TRSOMT LEA.L IDSOMT.A0 ; TABLE DES SOMMETS
30 LEA.L NEWSOM.A1 ; TABLE DES SOMMETS
31 ADD.L D6.A1 ; TRANSFORMES
32 MOVE.W LONGS.D1 ; NOMBRE DE SOMMETS
33 TRCORD LEA.L MATRIC.A2 ; ADRESSE DE LA MATRICE
34 ADD.L D5.A2
35 CLR D4
36 MOVEQ #2.D2 ;
37 TRLIG MOVE.W (A0+).D3 ; PRODUIT D'UNE LIGNE
38 MULS.W (A2+).D3 ; DE LA MATICE PAR UNE
39 ADD.L D3.D4 ; ORDONNEE D'UN SOMMET
40 DBRA D2.TRLIG ; X . Y . Z
41 CLR.L D3
42 MOVE.W (A2).D3 ; CUMUL DE LA COMPOSANTE
43 ADD.L D3.D4 ; HOMOGENE =1
44 MOVE.L D4.(A1+) ; SAVE DE LA NOUVELLE
45 ADDO #8.A1 ; ORDONNEE DU SOMMET
46 DBRA D1.TRCORD ;
47 ADDO #8.D5
48 ADDO #4.D6 ; PASSAGE A L'ORDONNEE
49 DBRA D0.TRSOMT ; SUIVANTE DU SOMMET
50 TRAP #14
51 :
52 :
53 : CE PROGRAMME CALCULE LES VALEURS DU
54 :
55 : FACTEUR W POUR CHAQUE SOMMET S(X.Y.Z)

```

AUTORISATION de SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974

VU les rapports de présentation de Messieurs

- . F. MARTINEZ, Professeur
- . R. CAUBET, Maître de conférences

**Monsieur CHIBANE Karim**

est autorisé à présenter une thèse en soutenance en vue de l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Informatique".

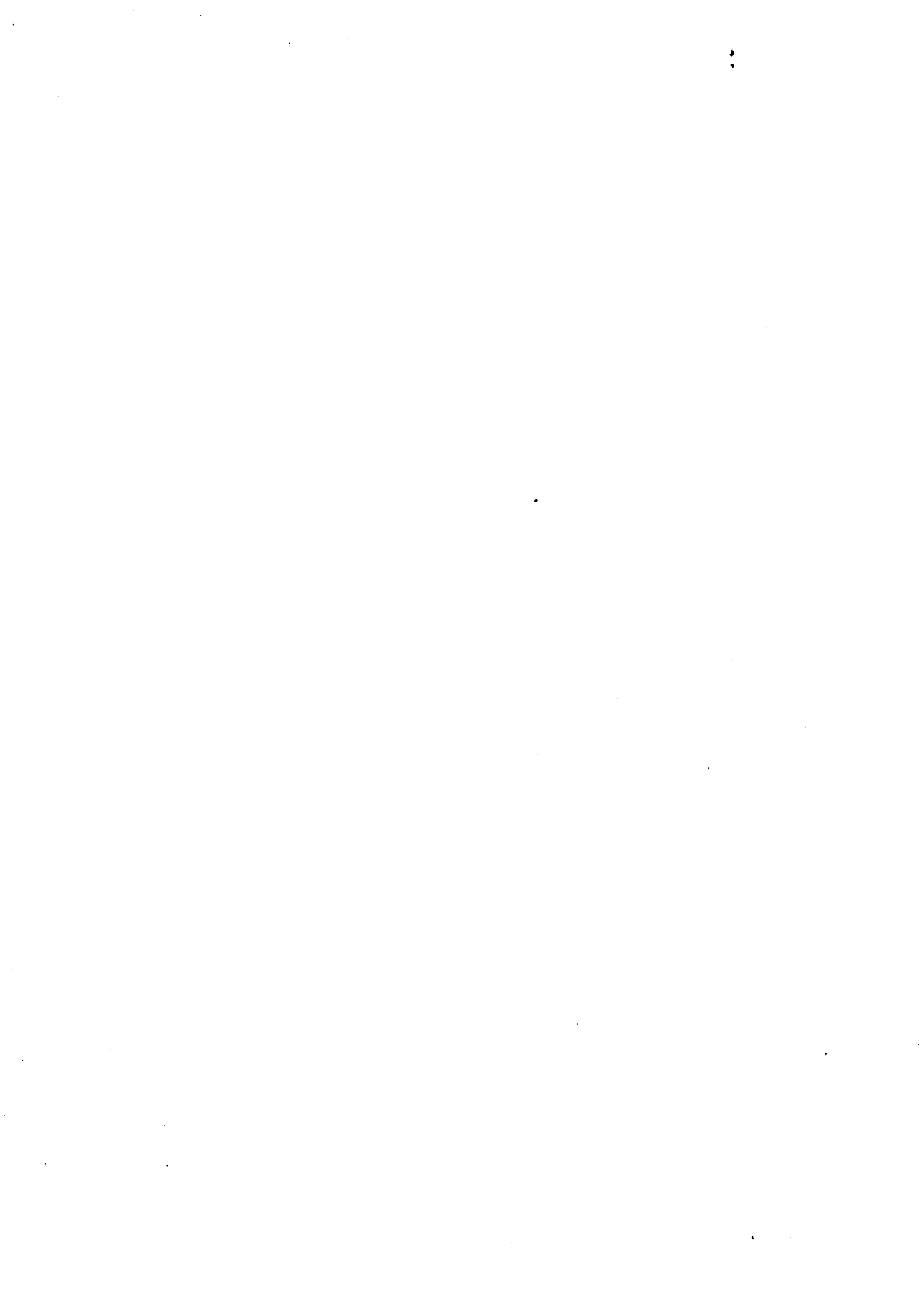
Fait à Grenoble, le 20 octobre 1986

**D. BLOCH**  
Président  
de l'Institut National Polytechnique  
de Grenoble

P.O. le Vice-Président.







## RESUME :

Du point de vue technologique, pour les terminaux graphiques, 1983 fut l'annee de la couleur; 1984 celle des ecrans video a haute resolution; 1985 est pour sa part dediee au realisme et a l'animation en temps reel.

Cette etude retrace en partie cette evolution en mettant l'accent sur la fonctionnalite et les performances toujours ameliorees des architectures proposees. A travers l'etude du systeme de synthese d'images realistes HELIOS, le lecteur pourra au fil des chapitres mesurer la progression technique d'une version du terminal a une autre.

- HELIOS-I : configuration "calculateur satellite",
- HELIOS-II et III : configuration "console evoluee",
- GETRIS : configuration "station de travail" et version commercialisee par la societe grenobloise GETRIS-IMAGES.

Cette mutation technologique est en cours, cette these indique certaines tendances adoptees par de nombreux constructeurs pour les annees a venir. Une attention particuliere est portee sur l'impact du choix de conception d'une architecture par rapport aux performances obtenues en temps reel pour des images realistes en trois dimensions (processeurs, interface de communication, type de memoires etc).

## MOTS CLES

Terminal Graphique Interactif - Pre-synthese - Post-synthese -  
Processus de synthese Tri-dimensionnel - Realisme -  
Co-processeur - Operateurs Cables et Microprogrammes - Temps-reel -  
Architecture Parallele - Pipeline - Station de Travail.

