

# LA RÉALISATION D'UN TERMINAL GRAPHIQUE POUR LE CONTRÔLE DU TRAFIC AÉRIEN D'AÉROPORTS MILITAIRES CANADIENS

C. Y. Laporte et J. L. Houle  
Ecole Polytechnique  
Montreal, Quebec  
Canada

## RÉSUMÉ

Les aéroports militaires canadiens ont récemment été dotés de radars utilisant des techniques numériques pour le traitement de l'information. Il fut décidé d'étudier la possibilité d'équiper les tours de contrôle d'un moniteur ayant des capacités limitées par rapport aux terminaux utilisés par les contrôleurs aériens. Un logiciel a été conçu à cet effet. La tâche consiste à analyser ce logiciel, à évaluer ses performances et à proposer une architecture qui permette l'affichage des cibles en temps réel. Les contraintes imposées étaient l'obligation de relier le système d'affichage à l'équipement radar et l'utilisation d'un terminal graphique prédéterminé.

Le présent article décrit des critères de performance du système réalisé. Le premier critère initial imposé par le ministère est que le traitement et l'affichage des cibles ne doivent pas retarder de plus de 22.5 degrés par rapport à la position de l'antenne radar. Le deuxième critère est que le système doit pouvoir traiter jusqu'à 60 cibles à l'intérieur d'un rayon de 32 milles nautiques. Le système d'affichage doit être compatible avec le format et le taux de transmission des mini-ordinateurs. L'affichage des cibles doit être réalisé à l'aide d'un traducteur graphique et d'un écran cathodique moderne. Le système doit être capable de filtrer les différents types de messages pour n'afficher que les messages du type « cible ». L'opérateur doit être capable de changer l'orientation de la plaque à l'aide de commutateurs s'il constate que l'information de la plaque peut cacher une cible. Le système doit réagir à une telle commande à l'intérieur d'une demi-seconde. Toutes les composantes

utilisées doivent répondre aux critères de sélection des normes militaires. Enfin, on a étudié la possibilité d'expansion du système pour le traitement de 90 cibles se trouvant à l'intérieur de 64 milles nautiques.

## 1. INTRODUCTION

Les Forces Armées Canadiennes ont entrepris une série de programmes de modernisation de l'équipement militaire. Un de ces projets consistait à moderniser le système de contrôle de trafic aérien des aéroports militaires. Un budget d'environ \$ 60 millions a été approuvé pour le projet TARP (Terminal Aids Replacement Program).

Le nouvel équipement permet la détection, la corrélation et la poursuite de cibles primaires et secondaires. Aussi, un système intégré de communication permet à l'opérateur de communiquer rapidement avec n'importe quel aéroport du continent. Finalement, l'opérateur a accès à un système d'information météorologique. La figure 1 montre les principales composantes du système TARP. Il fut aussi décidé de doter les tours de contrôle des aéroports militaires d'un moniteur spécial appelé « Radar Situation Display ». Ce moniteur devrait permettre au contrôleur aérien chargé de l'approche finale de visualiser l'état du trafic aux abords de l'aéroport. La tâche consistait à réaliser un système compatible avec les équipements du projet TARP. Ce terminal devait afficher les cibles se trouvant dans un rayon de 32 milles

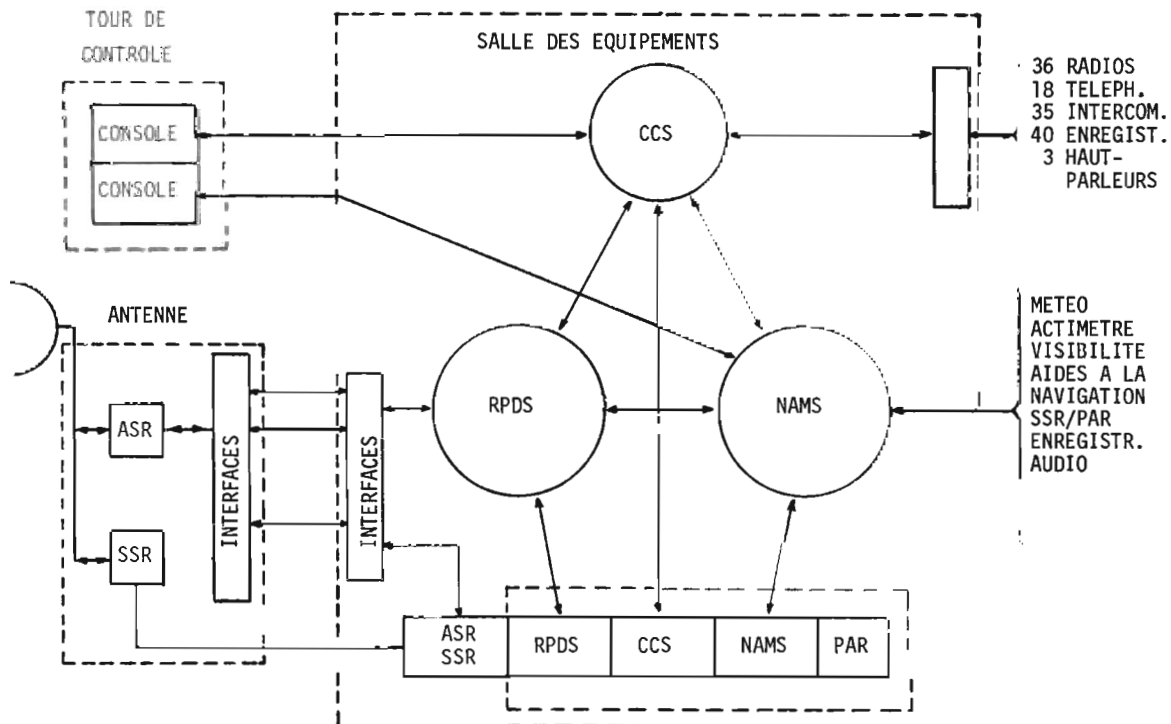


Figure 1: Les composantes du système TARP

nautiques et dont l'altitude était inférieure à 10,000 pieds. Le terminal devait aussi afficher les pistes de l'aéroport ainsi que les « PAR gates » i.e. la surface de terrain convertie par le radar d'approche final. De plus, il fallait afficher les aides à la navigation, les stations VFR et la position des aéroports civils.

La tâche principale du système consiste à lire les données transmises par des mini-ordinateurs NOVA 3, à analyser et traiter les données pour les transmettre selon un format spécifique au terminal graphique. Il y a 12 types de messages transmis par les mini-ordinateurs, la tâche du logiciel consiste à traiter trois types de messages seulement: Target, Sector et End of Sector. La figure 2 donne l'organigramme du logiciel qui est composé de 4 modules principaux: l'exécutif, les routines Sector, End of Sector et Target. La section suivante décrit avec plus de détails ce logiciel.

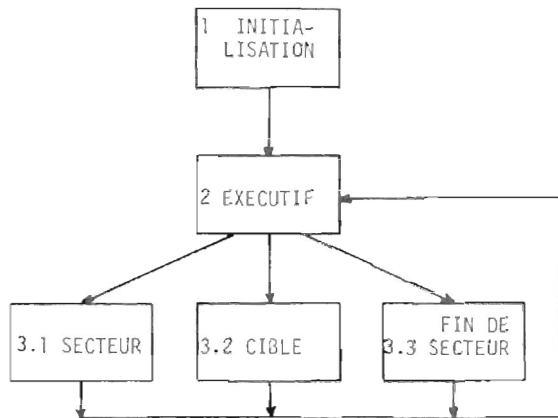


Figure 2: L'organigramme du logiciel du système d'affichage.

## 2. DESCRIPTION FONCTIONNELLE

Cette section est divisée en deux parties: en premier lieu on décrit les messages transmis par les ordinateurs NOVA et en second lieu, on décrit les caractéristiques du logiciel disponible au départ.

### 2.1 DESCRIPTION DES MESSAGES

On a décrit dans l'introduction, les composantes du système TARP. On va maintenant décrire le processus d'acquisition, de traitement et d'affichage des données. L'antenne du radar tourne à une vitesse angulaire de 12 tours par minute (tpm). L'information est transmise à l'ordinateur NOVA qui en effectue le traitement. L'antenne du radar transmet aussi sa position par rapport à un point de référence. Le mini-ordinateur utilise cette information et divise les 360° en 32 secteurs. L'ordinateur effectue le traitement des cibles ainsi que tout objet se trouvant dans un même secteur et transmet ces informations au terminal du contrôleur aérien. Les ordinateurs sont doublés et effectuent les calculs simultanément mais, en tout temps, il n'y a qu'un ordinateur qui transmet les messages aux terminaux. Le deuxième ordinateur ne fait qu'attendre une panne du premier pour prendre la relève à chaud.

L'ordinateur traite l'information de chaque secteur individuellement. Il transmet aux terminaux les informations suivantes: un message de début de secteur, une série de messages indiquant les cibles ou objets volants tels les oiseaux et un message de fin de secteur. Cette transmission s'effectue en mode synchrone à 50 kilobits par seconde. Le format d'un message est le suivant: le premier caractère indique le type de message (secteur, fin de secteur, cible) suivi de l'information pertinente au type de message et, complété par un caractère de fin de message.

Un message cible est composé de 57 octets, le tableau 1 décrit les éléments du message. On y retrouve des informations numériques codées binaires et décimales codées binaires. Il y a aussi des données utilisant le code ASCII. Ils servent à composer la plaque qui sera affichée sur l'écran graphique près du symbole indiquant la position de l'avion. Le système présentement en utilisation affiche les cibles dont la distance est inférieure à 512 milles nautiques du radar.

Les deux autres types de message qu'il faut capter sont le message de début de secteur et de fin de secteur. Le message de début de secteur est composé des éléments suivants: les caractères 02 indiquant le type de

Tableau 1  
Description d'un message cible

TYPE DE MESSAGE
Beacon Code (DCB)
Altitude (binaire)
Distance (binaire)
Azimuth (binaire)
Trait Dot 3 (binaire)
Trait Dot 2 (binaire)
Trait Dot 1 (binaire)
Position actuelle (binaire)
Identification de l'avion (ASCII)
Altitude (ASCII)
Distance (ASCII)
Message (ASCII)

message, deux caractères indiquant le numéro du secteur et les caractères 89 indiquant la fin du message. Le message fin de secteur est semblable: il comporte les caractères 01 indiquant la fin d'un secteur et les caractères 89 indiquant la fin du message. On rappelle que la transmission en mode synchrone transmet en tout temps, c'est-à-dire soit les caractères d'un message, soit des caractères appelés zéro de remplissage (« null fillers »).

### 2.2 DESCRIPTION DU LOGICIEL DISPONIBLE

Le logiciel disponible a été développé pour le micro-processeur MC6809 de la Compagnie Motorola. L'étude des programmes nous a amené à proposer à la section 3, l'architecture d'un système qui permettrait l'exécution du logiciel déjà mis au point. Il y a un programme d'initialisation de circuit de communication synchrone (Advanced Data Link Controller) ADLC. Ce circuit est utilisé pour la communication avec l'ordinateur NOVA. Il y a un programme d'initialisation de circuit de communication asynchrone série ACIA (Asynchronous Communication Interface Adapter). Il sert à la transmission des données au terminal graphique. Enfin, il y a un logiciel de traitement des données. Les sections suivantes décrivent globalement les caractéristiques des blocs modulaires de la figure 2.

#### 2.2.1 L'initialisation

Cette routine est exécutée lors de la mise sous tension du système ou lorsque l'on commande manuellement une nouvelle initialisation à l'aide d'un commutateur. Ce programme masque les interruptions, initialise les pointeurs des files FIFO d'entrée-sortie des données et initialise le circuit de communication série synchrone (ADLC) relié aux ordinateurs NOVA 3. En dernier lieu, il autorise les interruptions des circuits ACIA et ADLC et remet le contrôle au programme principal.

#### 2.2.2 L'exécutif

L'exécutif du programme analyse de manière asynchrone, l'aide de mémoire FIFO, le contenu des messages. L'exécutif joue le rôle de détecteur de messages valides et d'aiguilleur vers l'un ou l'autre des modules de traitement. Il doit aussi s'assurer que l'analyse des messages commence au début d'un secteur.

#### 2.2.3 La routine de service SECTOR

Le module de traitement «SECTOR» a pour tâche de faire la gestion des fichiers à l'intérieur du terminal graphique. Le module «traducteur graphique» nous donne la possibilité de diviser sa mémoire en 32 fichiers et d'autre part le radar divise l'écran en 32 secteurs. Puisqu'une partie de la mémoire du traducteur graphique est réservée à d'autres tâches, il nous est impossible d'utiliser 32 fichiers pour 32 secteurs. Les secteurs sont alors fusionnés en un multiple de 2 par fichier, par le module de traitement «SECTOR» pour former des blocs d'information, pouvant être rafraîchis à chaque balayage du radar.

#### 2.2.4 La routine END-OF-SECTOR

Cette routine effectue la lecture de commutateurs reliés au processeur. Des commutateurs permettent à l'opérateur de modifier la carte géographique qui est affichée par l'écran graphique. D'autres commutateurs permettent d'afficher les secteurs d'observations du radar d'approche pour les différentes pistes de l'aéroport. Cette routine permet ainsi à l'opérateur de choisir le type d'information qu'il désire voir sur l'écran graphique.

#### 2.2.5 La routine de service TARGET

Ce module effectue la lecture de 57 caractères qui composent le message cible. Ce message contient le code «BEACON», l'altitude, la distance, l'azimuth et les trois points de traînée (« trail dots ») qui indiquent la position de la cible pendant les trois tours précédents de l'antenne du radar. Finalement

le message contient l'information de la plaque soit l'identification de l'avion, son altitude, sa distance et un message quelconque. Cette routine détermine, suite à la lecture du message, si la cible doit être affichée i.e. si la cible est à une altitude inférieure à 10,000 pieds et à une distance inférieure à 32 milles nautiques. Si la cible satisfait aux critères de sélection on procède à l'affichage des points de traînée, de la position actuelle de l'avion et de la plaque. Lorsque le traitement et l'affichage de ce message sont terminés on remet le contrôle du processeur à l'exécutif.

### 2.2.6 Analyse des routines de service

Ces routines de service ont déjà été codées en langage assembleur pour un processeur MC6809. On a analysé les 30 routines et sous-routines disponibles pour évaluer leur temps d'exécution. Le tableau 2 donne le temps d'exécution de quelques routines. Une première observation concerne le temps d'exécution (104 usec), de la routine de réception des données ADLC.

Tableau 2  
Temps d'exécution des routines (usec)

ROUTINE DE SERVICE	TEMPS D'EXECUTION
EOS	2987
TARGET	50900
SECTEUR	1285
PLAQUE	35275
ACIA	107
ADLC	104

On verra plus loin que la réception des données monopolise 65% du temps de calcul du processeur. Ceci semble, à première vue, être inacceptable. Une deuxième observation concerne le temps d'exécution de la routine TARGET: 50,900 usec. Ce temps d'exécution est dû aux nombreuses routines de conversions DCB-binaire et aux routines qui effectuent les calculs de transformation pour l'affichage de la cible sur l'écran.

## 3. DESCRIPTION DU MATÉRIEL

Cette section est divisée en trois parties. La première partie décrit les composants imposés qu'il fallait utiliser pour effectuer la réception et l'affichage des données transmises par le mini-ordinateur NOVA: ce sont le circuit de réception des données (ADLC) et le terminal graphique. La deuxième partie décrit les architectures proposées pour la réalisation du système d'affichage. La troisième partie est un calcul de fiabilité pour les architectures proposées.

### 3.1 COMPOSANTES IMPOSÉES

#### 3.1.1 Réception des données

L'ordinateur NOVA 3 transmet les données à 50 kilobits par seconde à l'aide d'un circuit ADLC, le MC6854 de Motorola. Ce circuit peut opérer en mode d'interrogation séquentiel (polling) ou d'interruption. Le logiciel mis au point opère par interruption. Pour un taux de transmission de 50 kilobits par seconde et des caractères de 8 bits, on calcule que l'ADLC interrompt le processeur à chaque 160 micro-secondes.

#### 3.1.2 Le terminal graphique

Une deuxième contrainte imposée était l'utilisation d'un terminal graphique 1351S de la compagnie Hewlett-Packard. Ce système est composé d'un générateur graphique et d'un écran cathodique. Le générateur graphique est muni d'un interface RS-232, qui reçoit les informations d'un ordinateur hôte, les traite et commande l'écran cathodique. Le taux de transmission est de 9600 bauds. Le générateur graphique s'occupe du rafraîchissement de l'écran. La résolution de l'affichage est de 1020 x 1020 points.

### 3.2 ÉTUDE DE DIFFÉRENTES ARCHITECTURES

#### 3.2.1 Analyse d'une première architecture

La section 2.2 présentait le logiciel déjà disponible. On introduit maintenant l'architecture qui permet l'exécution de ce logiciel. La figure 3 illustre cette architecture. Les éléments suivants s'y trouvent: un circuit synchrone de réception des données ADLC MC6854, un bloc de mémoire vive servant comme FIFO pour les données reçues, un micro-processeur de 8 bits le MC6809, un circuit de mémoire morte d'une capacité de 2K octets servant au

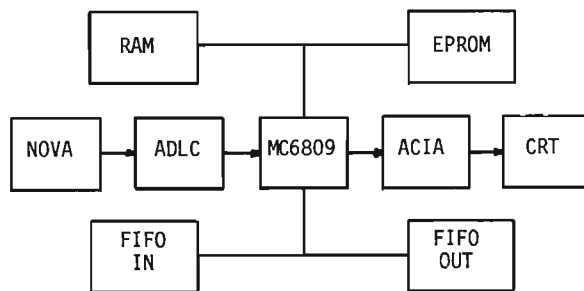


Figure 3 Architecture initiale

stockage du logiciel, un loc de mémoire vive servant comme FIFO pour les données à être transmises, un circuit de transmission des données en mode série asynchrone l'ACIA MC6850, le traducteur graphique ainsi que l'écran cathodique.

La réception des données, par l'ADLC, s'effectue par interruption. La routine de service qui s'occupe de la réception prend 104 cycles machines par caractère reçu. Ce grand nombre de cycles machines s'explique par le fait que le processeur doit répondre à l'interruption, lire la donnée, la stocker dans le FIFO et gérer le pointeur de la pile. Le calcul suivant nous permet d'évaluer le temps consacré par un processeur pour la réception des données:

$$\rho = n s$$

où  $\rho$  = facteur d'utilisation  
 $n$  = nombre moyen d'arrivées par seconde  
 $s$  = temps moyen de service

$$\rho = \frac{50 \text{ kbits/sec} \cdot 104 \text{ cycles/car}}{8 \text{ bits/car} \cdot 10^6 \text{ cycles/sec}} = .65$$

La réception des données monopolise 65% du temps du processeur. Pendant une rotation de l'antenne, le processeur utilise 3.25 sec. pour cette tâche.

Analysons maintenant les routines de traitement des messages. On se rappelle que le système doit être capable de traiter et d'afficher 60 cibles. La routine Target requiert près de 10,600 cycles par cible, c'est-à-dire que le processeur prend .636 sec. pour le traitement des 60 cibles pendant une rotation de l'antenne. Les routines EOS et Sector prennent .035 et .032 sec. respectivement. La transmission des données au traducteur graphique requiert 107 cycles par caractère. Ce temps comprend la gestion de la file FIFO ainsi que la réponse à l'interruption par le circuit ACIA. On calcule que le temps d'exécution pour la routine de transmission des données requiert environ 2.51 sec par rotation de l'antenne.

Le tableau 3 montre les temps d'exécution des différentes routines pour le traitement de 60 messages cibles. On calcule facilement que le temps total d'exécution est de 6.47 secondes. On sait qu'une rotation de l'antenne prend 5 secondes, cette architecture ne rencontre donc pas les critères de performance.

Tableau 3  
Temps d'exécution des routines pour le traitement de 60 cibles

ROUTINE	TEMPS D'EXECUTION
Réception ADLC	3.25
Traitement:	
Target	0.636
EOS	0.035
Sector	0.032
Affichage:	
Target	2.41
EOS	0.06
Sector	0.045
Total	6.47

### 3.2.2 Architecture à deux processeurs

Une autre architecture possible est l'architecture à multi-processeurs. La réalisation d'un tel système est rendue possible grâce à la disponibilité d'une version spéciale du processeur MC6809 c'est la version multi-processeurs MC6809E.

On peut effectuer la répartition des tâches de la façon suivante: un premier processeur s'occupe de la réception et du traitement des données, ceci prend 3.93 secondes; un second processeur s'occupe de la transmission des données, ceci prend 2.51 secondes. Mais, cette architecture est un peu complexe à réaliser puisqu'il faut orchestrer les opérations du premier processeur qui stocke les données dans le FIFO de sortie et du second processeur qui effectue la lecture des données du même FIFO. Le problème consiste à informer le second processeur pour que ce dernier ne transmette pas de mauvaises informations au traducteur graphique. Un second problème consiste à arbitrer les opérations d'accès à la mémoire par les 2 processeurs pour qu'ils ne produisent pas de conflit sur les bus. Ce problème peut être résolu par l'utilisation d'un nouveau circuit de mémoire vive, d'une capacité de 1 kilo-octets à double ports d'accès produit par la compagnie Synertek. Ce circuit SY2130 peut être accédé simultanément par deux processeurs puisqu'il est doté de deux ports d'entrée/sortie.

### 3.2.3 Architectures avec accès direct en mémoire

L'utilisation de l'accès direct en mémoire (DMA) permet de sauver un temps considérable puisque le processeur n'est interrompu par le circuit d'accès direct en mémoire qu'à la fin d'un cycle de stockage d'un bloc de données (voir figure 4). Ce cycle de stockage peut être, par exemple, de 2 kilo-octets. Ceci signifie que le processeur ne serait interrompu qu'une fois plutôt que 2048 fois.

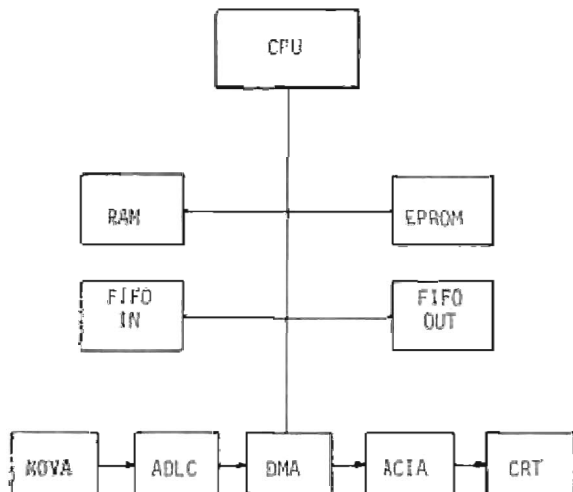


Figure 4: Architecture avec accès direct en mémoire

On sait que le circuit de communication série ADLC délivre une donnée à chaque 160 microsecondes. La méthode d'accès par vol de cycles permet le stockage d'un octet en n'utilisant qu'à peine trois cycles machine par caractère reçu, soit à chaque 160 microsec. Ceci représente une économie considérable par rapport à la méthode précédente ou on utilisait 104 cycles machine par octet reçu. Il faut aussi utiliser quelques cycles machines à la fin d'un cycle de stockage d'un bloc de données pour effectuer la gestion de la file FIFO et amorcer un nouveau cycle DMA. L'équation suivante nous permet de calculer le temps utilisé par le processeur pour effectuer la réception des données par DMA. On utilise la même formule qu'à la section 3.2.1 et on obtient

$$\rho = \frac{50 \text{ kbits/sec} \cdot 3 \text{ cycles/car}}{8 \text{ bits/car} \cdot 10^6 \text{ cycles/sec}} = 0.018$$

Le facteur d'utilisation est donc passé de 65% à moins de 2%. Le temps d'exécution des autres routines n'est à toute fin pratique, pas affectée par cette nouvelle architecture. Le temps total d'exécution est donc de 3.66 secondes soit une réduction de 2.8 secondes par rapport à l'architecture précédente. Le tableau 4 montre les temps d'exécution pour une architecture avec DMA.

Tableau 4  
Temps d'exécution des routines pour une architecture avec accès direct en mémoire pour la réception des données

ROUTINE	TEMPS D'EXÉCUTION (sec)
Réception ADLC	0.093
Traitement:	
Target	0.636
EOS	0.035
Sector	0.032
Affichage:	
Target	2.7
EOS	0.125
Sector	0.045
Total	3.66

### 3.3 CALCUL DE LA FIABILITÉ DU SYSTÈME

On sait que la vie d'une pièce est divisée en trois parties (voir figure 5): la première partie, appelée «Burning-in period», est caractérisée par un haut taux de défaillance. Pour des composantes électroniques il est possible de «sauter» cette première étape en faisant vieillir artificiellement la composante en la chauffant. Cette opération amène la pièce au temps  $T_B$ , c'est-à-dire le début de la vie utile de la pièce. Cette étape est caractérisée par un taux de défaillance faible et constant. Au temps  $T_W$  débute la troisième période appelée «Wearout» ou période d'usure. Elle est caractérisée par un haut taux de défaillance. Pour s'assurer d'un fonctionnement fiable d'un système, il est impératif de connaître le temps  $T_W$  de la composante. On peut alors effectuer les opérations de maintenance requises.

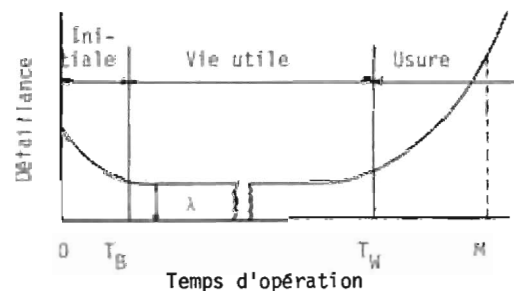


Figure 5 Taux de défaillance en fonction de l'âge.

#### 3.3.1 Calcul de la fiabilité des composantes

Les systèmes militaires doivent répondre à des normes très sévères. La norme militaire «Mil-Hdbk-217C Reliability Prediction of Electronic Equipment», explique la méthode qu'il faut utiliser pour le calcul de la fiabilité de systèmes électroniques. Cette norme montre comment utiliser les modèles de calcul de fiabilité mis au point pour les différentes composantes d'un système électronique telles les transistors, les relais, les commutateurs, les circuits intégrés SSI-MSI-LSI-VLSI. On montre ici la méthode qu'il faut suivre pour calculer la fiabilité des principales composantes du système graphique: soit le processeur MC6809, les circuits de mémoire morte et vive, les circuits de réception ADLC et transmission ACIA ainsi que quelques circuits logiques tels les décodeurs et les étages-tampons de bus.

Les calculs pour obtenir le taux de défaillance du micro-processeur MC6809 sont les suivants:

- A) Le facteur de qualité de la composante est déterminé selon les tests auxquels a été soumise la composante. Ces tests doivent être effectués selon les normes militaires Mil-M-38510 et Mil-STD-883. Le facteur de qualité  $\pi Q$  varie de .5, pour une pièce qui répond aux normes les plus sévères, à 35 pour une pièce de qualité «commerciale». Cette pièce n'a pas été soumise à des tests militaires mais elle répond aux exigences de qualité du fabricant.

- B) Le deuxième facteur,  $\pi E$ , définit l'environnement dans lequel cette composante est utilisée. On en énumère 11 types. Le facteur  $\pi E$  est de 1.0 pour un système utilisé au sol à l'abri des vibrations et des variations de pression et de 10.0 pour un environnement du type «missile launch».
- C) Le troisième facteur,  $\pi T$ , détermine le taux de vieillissement dû à la température de fonctionnement de la composante.
- D) Le facteur  $\pi L$  est déterminé par l'expérience du fabricant pour la production de cette composante. Il est de 10 pour une pièce récemment mise en production et de 1.0 pour une pièce qui est en production depuis au moins 6 mois et qui n'a pas été modifiée depuis le début de sa production.
- E) Deux facteurs, C1 et C2, sont reliés à la complexité de la composante, c'est-à-dire au nombre de transistors, de portes logiques, de bits de mémoire etc.
- F) Un facteur C3 défini le type d'encapsulation (ex.: hermetic with weld seal or glass seal).
- G) Un facteur  $\pi V$  est appelé «voltage derating stress factor».

On fait ici un calcul de fiabilité  $\lambda$  pour le 6809. Selon la norme militaire, la formule est :

$$\lambda = \pi Q [ C1 \pi T \pi V + (C2 + C3) \pi E ] \pi L$$

Les valeurs des facteurs sont obtenues à partir de tables de la norme militaire. Pour notre application, le facteur  $\pi E$  est de 2.5 pour un environnement au sol. Le facteur de qualité,  $\pi Q$ , est de 6.5 pour une composante qui répond aux normes Mil-STD-883, c'est-à-dire pour une pièce qui fait partie d'un système qui ne mettrait pas en péril une mission. Il faut mentionner ici que le coût d'une composante est proportionnelle à la quantité des tests auxquels elle est soumise. Il faut donc se procurer des pièces en fonction de la nature de la tâche à accomplir. Les facteurs de complexité C1 et C2 pour un processeur de 3,000 portes logiques sont de 0.043 et 0.0015 respectivement. Le facteur relié à la température d'opération,  $\pi T$  est de 0.71. Le facteur  $\pi V$  est de 1.0. Le facteur «d'expérience»  $\pi L$  est 1.0 puisque ce processeur est sur le marché depuis quelques années déjà. Enfin le facteur C3 relié au type d'encapsulation est de 0.024. On trouve que le taux de défaillance est :

$$\lambda = 0.612/10^6 \text{ heures.}$$

On peut reprendre les calculs pour une mémoire morte EPROM d'une capacité de 2 kilo-octets, on trouve que  $\lambda$  est :

$$\lambda = 0.7310/10^6 \text{ heures.}$$

Pour le circuit de communication série asynchrone ACIA, les calculs donnent :

$$\lambda = 0.302/10^6 \text{ heures.}$$

Pour le circuit d'accès direct en mémoire MC6844, on trouve :

$$\lambda = 0.538/10^6 \text{ heures.}$$

Pour le circuit de communication série synchrone ADLC 6854, on calcule :

$$\lambda = 0.327/10^6 \text{ heures.}$$

De même pour une mémoire vive statique de 2 kilo-octets on calcule :

$$\lambda = 0.626/10^6 \text{ heures.}$$

Finalement, pour des circuits décodeurs et tampons d'une complexité d'environ 16 portes logiques par boîtier on trouve que le taux de défaillance est :

$$\lambda = 0.1086/10^6 \text{ heures.}$$

### 3.3.2 Analyse de la fiabilité du système

Maintenant que l'on connaît le taux de défaillance des composantes, on peut calculer la fiabilité du système. On tentera aussi d'analyser différentes stratégies soit avec des composantes dont le facteur de qualité est différent soit avec un système redondant. On sait que la fiabilité F pour un système de n composantes en série, se calcule à l'aide de l'équation suivante :

$$F = \exp \left( - \sum_{i=1}^n \lambda_i t \right)$$

Le paramètre  $\lambda_i$  représente le taux de défaillance d'une composante. Le système de traitement et d'affichage est composé des pièces suivantes: 1 CPU, 1 ACIA, 1 DMA, 1 ADLC, 2 mémoires mortes EPROM de 2 kilo-octets, 8 mémoires vives de 2 kilo-octets et 15 circuits divers (tampon, décodeurs). On calcule que la fiabilité du système est :

$$F = \exp (- 8.4 \times 10^{-6} \times t).$$

Pour mieux apprécier l'impact du choix du facteur de qualité sur la fiabilité d'un système, on a tracé des courbes (figure 6) pour des composantes de qualité différente: la courbe du haut représente la fiabilité d'un système dont le facteur de qualité des pièces est 0.5, la seconde courbe pour un facteur de qualité de 6.5 et la troisième courbe pour un facteur de «qualité commerciale» de 35.

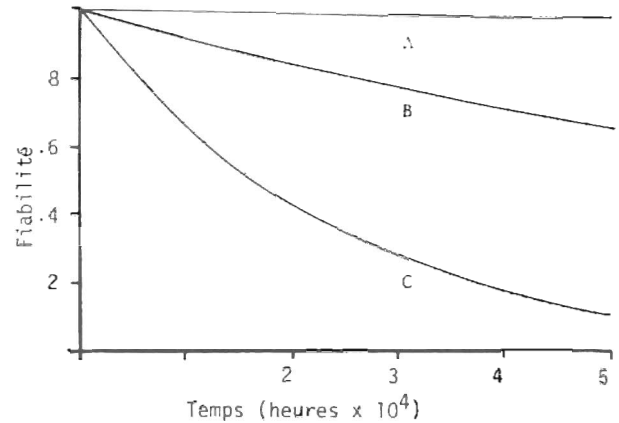


Figure 6: Fiabilité d'un système pour différents facteurs de qualité 0.5(A), 6.5(B) et 35C

On sait que la fiabilité  $F_p$  et le MTBF d'un système en parallèle sont déterminés par les équations suivantes :

$$F_p = 1 - (1 - \exp(-\lambda t))^n$$

$$MTBF = 1/\lambda + 1/2 * \lambda + \dots + 1/n * \lambda.$$

Le paramètre n représente le nombre d'unités en parallèle. La disponibilité, A, d'un système est calculée à partir de l'équation suivante :

$$A = MTBF / (MTBF + MTTR), \text{ où } MTTR = \text{Mean Time to Repair}$$

Il est raisonnable de concevoir un système avec un MTTR d'une heure à l'aide des nouvelles techniques de dépannage telle l'analyse des signatures développée par la compagnie Hewlett Packard. On calcule qu'un taux de défaillance  $\lambda = 8.39E-6$  nous donne un MTBF =  $1.2E + 5$  et une disponibilité A = 99.999%.

Finalement, on peut concevoir un système avec auxiliaire de secours («stand-by»). La fiabilité,  $F_s$ , et le MTBF d'un tel système se calculent à l'aide des équations suivantes :

$$F_s = e^{-\lambda t} (1 + \lambda t)$$

$$MTBF = 2/\lambda$$

Le tableau 5 montre la fiabilité, la disponibilité et le coût relatif en fonction de l'architecture et du facteur de qualité des pièces. Le concepteur peut donc utiliser ce tableau pour faciliter la prise de décision quant au choix de l'architecture la plus performante, la plus fiable et la moins coûteuse.

Tableau 5  
Fiabilité, disponibilité et coût relatif du système selon différentes architectures et facteurs de qualité

ARCHI- TECTURE	QUALITÉ	MTBF	DISPONI- BILITÉ	COÛT
simple	35.0	2.2E + 4	99.99 %	
simple	6.5	1.2E + 5	99.9991%	4
simple	0.5	1.5E + 6	99.9999%	6
parallèle	6.5	1.8E + 5	99.9994%	8
parallèle	35.0	3.3E + 4	99.996 %	2
stand-by	6.5	2.4E + 6	99.9995%	8*
stand-by	35.0	4.4E + 4	99.997 %	2*

## CONCLUSION

Le présent projet a permis d'établir des architectures basées sur micro-processeurs afin de répartir le traitement en temps-réel entre le processeur central et le processeur d'entrées-sorties. De plus des critères de coût et fiabilité ont été étudiés pour évaluer les avantages et inconvénients de structures mono-processeurs versus multi-processeurs.

Les problèmes spécifiques d'une application donnée ont été solutionnés, cependant l'étude peut facilement s'adapter à une vaste classe de cas de même nature.

## BIBLIOGRAPHIE

1. Department of Defense of USA, "Military Standardization Handbook: Reliability Prediction of Electronic Equipment", MIL-HDBK-217C, 1980.
2. Department of Defense of USA, "Military Standard: Definitions of terms for reliability and maintainability", MIL-STD-721C, June 12, 1981.
3. Bazovsky, I., "Reliability Theory and Practice", Prentice Hall Space Technology Series, 1961.
4. Motorola Inc., "MC6809-MC6809E Microprocessor Programming Manual Motorola Publication M6809PM(AD), 1981.
5. Parker, Y., "Multi-Microprocessor Systems", Academic Press, 1983.
6. Motorola Inc., "Hardware consideration for the implementation of direct memory access with the 6809 processor", Application note AN-820.
7. Scales, H., "Multiprocessing with Motorola's MC6809E", Byte Publications Inc., Jul. 1981.
8. Siewiorek, D., Bell, G., Newell, A., "Computer Structures: Principles and Examples", McGraw Hill, 1982.
9. Weitzman, Cay, "Distributed Micro/Minicomputer Systems", Prentice Hall, 1980.