

# Le système d'acquisition et de déclenchement d'OPAL (LEP)

P. Farthouat

► **To cite this version:**

P. Farthouat. Le système d'acquisition et de déclenchement d'OPAL (LEP). École thématique. Ecole Joliot Curie "Instrumentation en physique nucléaire et en physique des particules", Maubuisson, (France), du 26-30 septembre 1988 : 7ème session, 1988. <cel-00645642>

**HAL Id: cel-00645642**

**<https://cel.archives-ouvertes.fr/cel-00645642>**

Submitted on 28 Nov 2011

**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.

## LE SYSTEME D'ACQUISITION ET DE DECLENCHEMENT D'OPAL (LEP)

P. Farthouat

DPHPE-SEIPE, CEN Saclay, F-91191 Gif-sur-Yvette Cedex, France

**Résumé :** La description du système de lecture de données de l'expérience OPAL (LEP) est donnée, tant du point de vue matériel que logiciel. Le système de sélection d'évènement est présenté.

**Abstract :** The description of the hardware and software of the OPAL (LEP) experiment data acquisition is given. The hardware trigger system is presented.

## 1. Présentation d'OPAL

OPAL (Omni Purpose Apparatus for Lep) est un détecteur [Ref 1] destiné à l'étude des interactions  $e^+e^-$  jusqu'à des énergies de 100 GeV. Il est en cours de montage auprès du LEP au CERN et doit fournir des résultats dès le mois de juillet 1989.

Les principaux éléments du détecteur [fig 1] sont :

- un détecteur de vertex (chambre à dérive de 50  $\mu\text{m}$  de résolution)
- un détecteur central (chambre de type Jade de résolution azimuthale de 100  $\mu\text{m}$  donnant une mesure de  $dE/dX$  à 3.5%)
- une bobine chaude (0.4 Tesla de champs)
- un compteur de temps de vol (250 ps de résolution)
- un calorimètre électromagnétique (détecteur de pied de gerbe et 12500 blocs de verre au plomb de résolution  $7\%/\sqrt{E}$ )
- un calorimètre hadronique (tubes de larocci de résolution  $100\%/\sqrt{E}$ )
- des chambres à muons

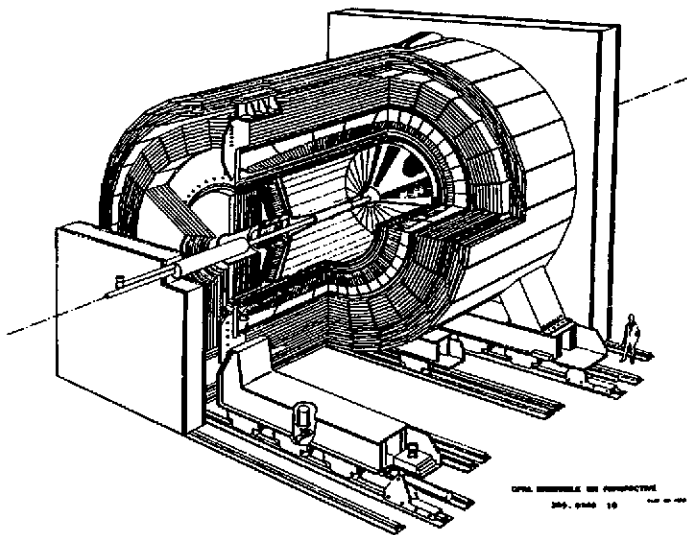


fig 1. Vue d'OPAL

Plus de 150000 signaux analogiques sont traités (préamplification, amplification, codage, suppression de zéro) et un évènement moyen produit 160 koctets de données, qu'il convient de traiter et de stocker; c'est le rôle de l'acquisition de données.

Le LEP fournit 45000 collisions de faisceaux  $e^+e^-$  par seconde. Compte tenu des caractéristiques des faisceaux et des sections efficaces des interactions étudiées, on attend de 0.4 à 1 évènement par seconde. C'est la logique de déclenchement (ou Trigger) qui doit nous permettre d'extraire ces évènements du flot initial.

Ces deux points seront traités dans les chapitres suivant. La séparation entre les différentes parties est souvent arbitraire, car ces fonctions sont très largement imbriquées...

## 2. L'acquisition de données.

Ce chapitre est divisé en trois parties. Dans la première, il sera montré pourquoi il fallait dégager un système standard de lecture et une description détaillée du système de lecture du détecteur de vertex sera présentée. Dans la seconde, une description complète de la collection de données jusqu'à la mise sur bande sera exposée. La troisième partie abordera le logiciel utilisé.

### 2.1 Choix d'un standard

OPAL est une collaboration entre 23 laboratoires européens, américains et japonais. Chaque institut construit tout ou partie d'un détecteur et de l'électronique associée. Pour des raisons de cout (utilisation d'électronique existante,...), de disponibilité industrielle ou de savoir faire, chaque groupe décide du système d'électronique utilisé ainsi que du bus l'abritant. Ainsi dans OPAL, l'électronique de codage proche du détecteur (électronique de front end) utilise différents bus:

- CAMAC
- FASTBUS
- VME
- Spécialisés

Compte tenu du découpage de l'appareillage, on trouve plus de 20 systèmes différents qu'il conviendra de rassembler.

Deux problèmes apparaissent immédiatement :

- avant le montage de l'expérience, puis pour des tests, chaque groupe doit pouvoir acquérir et réduire les données de son appareillage indépendamment des autres et ceci sans interventions matérielles ou logicielles importantes
- pour des raisons de temps mort, il est exclus de lire en série les données brutes des différentes parties de l'expérience.

Ceci impose de définir un système de lecture (matériel et logiciel) identique pour chaque groupe et qui présente les caractéristiques suivantes :

- grande modularité pour permettre une adéquation aisée à chaque type d'électronique par adjonction des interfaces appropriées
- bonne autonomie autorisant des développements confortables
- bonne ouverture sur l'extérieur pour permettre les échanges d'information avec le reste du système d'acquisition, avec le système de controle et avec éventuellement des mini ordinateurs ( $\mu$ Vax, MacIntosh,...)

Nous avons décidé d'utiliser des systèmes VME [Ref 2], à base de processeurs de la famille 68xxx de Motorola. Ce bus répond parfaitement aux critères définis et présente un bon support de la part de l'industrie et des laboratoires de physique:

- on trouve sur le marché de nombreuses cartes processeur, intégrant rapidement les évolutions technologiques (par exemple, nous avons commencé à travailler avec des processeurs 68000 (16 bits) et nous utilisons maintenant des 68020 (32 bits) et des 68030 (32 bits plus rapides avec un cache important))

- il existe de nombreuses interfaces industrielles (Camac, GPIB, Ethernet, SCSI, extension de chassis, VAX,  $\mu$ VAX, Macintosh,...) ou développées dans les laboratoires (Fastbus,...)

- l'industrie supporte de nombreux systèmes d'exploitation (Versados, OS9, UNIX,...)

- il existe un sous bus (VSB [Ref 3]) permettant d'alléger le trafic sur le bus principal

Chaque groupe dispose d'un chassis VME, appelé Local System Crate (LSC) contenant systématiquement [fig 2]:

- une carte processeur 68020 [Ref 4] nommée FIC. A base de 68020 16MHz, adjoint d'un coprocesseur flottant, elle possède jusqu'à 4 MOctets de mémoire double port CPU/VME et peut piloter le bus VME et le bus VSB. D'autre part elle possède un DMA capable de transférer des données entre le VME, le VSB et la mémoire interne. Un puissant mécanisme de «boite aux lettres» permet des échanges simples entre CPUs.

- une mémoire double port [Ref 5] nommée DPM. De capacité maximum de 16 MOctets, elle peut être lue depuis le VME ou le VSB. Elle possède un DMA, capable d'échanger des données avec le VSB à la vitesse de 16 MOctets par seconde.

- une interface Ethernet [Ref 6] nommée LRT. Elle permet au FIC de dialoguer avec n'importe quelle autre machine.

- une carte Trigger [Ref 7] nommée L.TU. Elle interface le LSC avec le système de déclenchement de l'expérience.

- une interface reliant le LSC au niveau supérieur de l'acquisition [Ref 8] nommée VIP. Cette carte permet de faire tous les transferts de données entre deux bus VME distincts ou entre un bus VSB et un bus VME.

A ces modules, chaque groupe ajoute les interfaces dont ils ont besoin :

- interface Camac [ref 9]
- interface FastBus [Ref 10]
- interface  $\mu$ VAX [Ref 11]
- interface MacIntosh II [Ref 12]

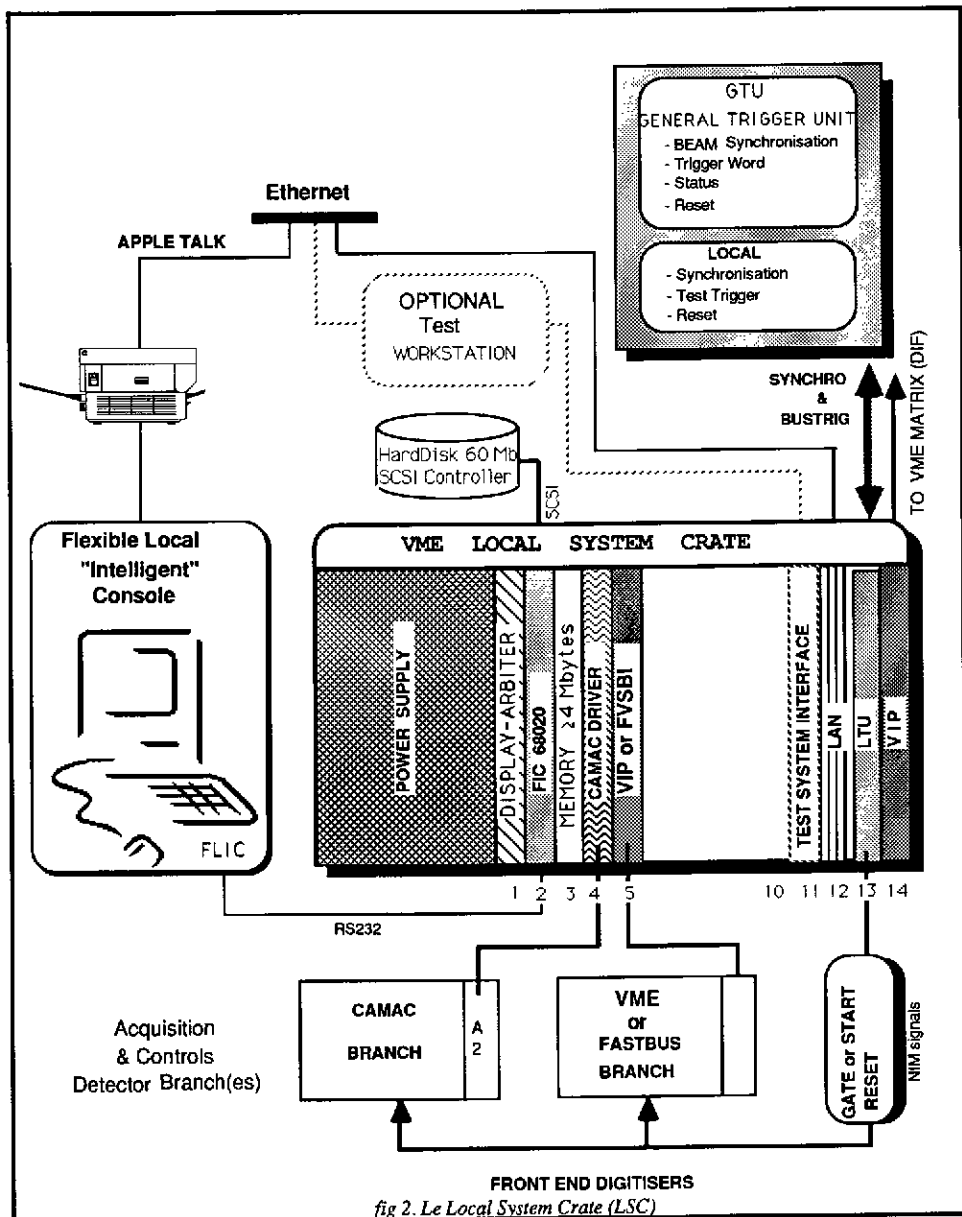


fig 2. Le Local System Crate (LSC)

Enfin, on trouve un disque dur de 60 Moctets [Ref 13] piloté par le FIC [Ref 14]. Ce disque sert à stocker des paramètres et éventuellement de support de développement logiciel. Le système d'exploitation utilisé est OS9.

Quelque soit le mode d'utilisation (test autonome ou acquisition réelle), les opérations effectuées par le processeur d'acquisition (FIC) sont toujours les mêmes du point de vue de l'acquisition et peuvent se résumer à trois tâches:

- 1) une tâche de lecture des données brutes, déclenchée par une interruption en provenance du module de trigger (LTU). Les données sont stockées dans une mémoire tampon (buffer) intermédiaire. Lorsque ceci est fait, le LTU est prévenu; un nouvel évènement peut être accepté.
- 2) une tâche de formattage de l'évènement. L'évènement formatté est stocké dans un autre buffer et sera utilisable par l'étage supérieur de l'acquisition et/ou par la troisième tâche
- 3) une tâche de contrôle/monitoring, qui contrôle l'intégrité des données, fabrique des histogrammes, fait un premier

traitement de cette portion d'évènement.

Le diagramme du flot de données est présenté sur la fig3.

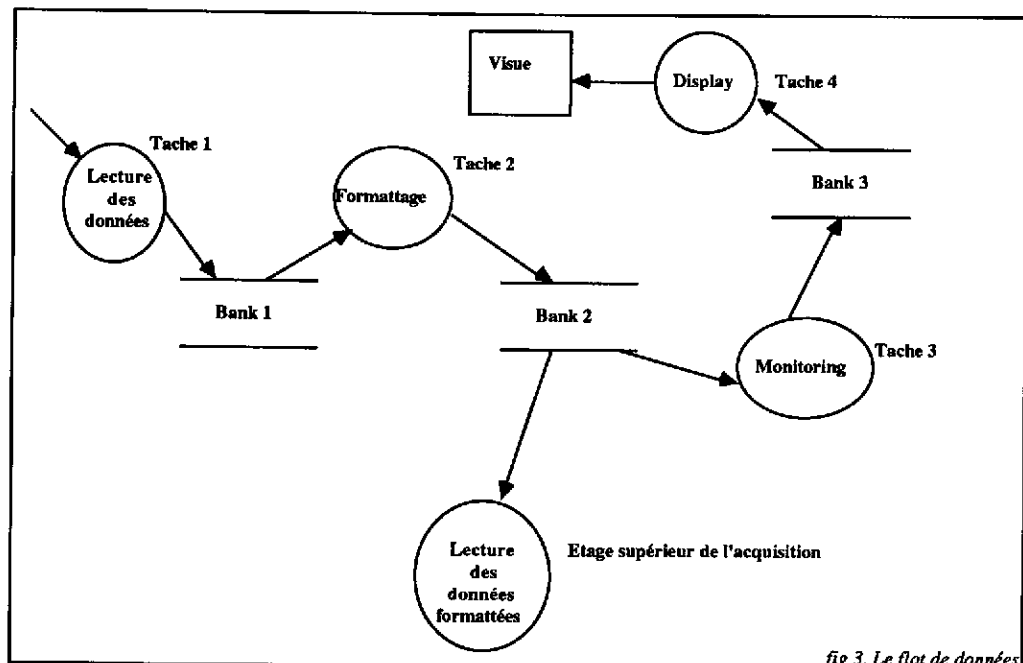


fig 3. Le flot de données

Le stockage des données brutes et des données formatées dans une mémoire tampon permet de réduire le temps mort d'acquisition à la durée de la tache 1 (du moins tant que le buffer n'est pas rempli) et d'utiliser au mieux le temps restant pour effectuer de façon asynchrone par rapport au trigger les taches 2 et 3, puisque l'on n'est plus dépendant du flux irrégulier d'entrée. L'étage supérieur de l'acquisition lira les données formatées dans le deuxième buffer. Cette tache sera elle aussi désynchronisée par rapport au trigger.

Avant de détailler l'acquisition du détecteur de vertex, il faut noter que:

- il y aura une vingtaine de ces sous systèmes d'acquisition
- le temps mort d'acquisition sera faible et égal au temps mort introduit par le plus lent des sous systemes; il sera de l'ordre de 5 ms.
- le trigger devra être distribué à tous les sous systèmes
- il est très facile de déconnecter un LSC du système d'acquisition de l'expérience puisqu'il suffit de ne pas mettre les évènements formatés à la disposition de l'étage supérieur de l'acquisition. Cela ne nécessite aucune modification du matériel.

### 2.1.1 Acquisition du détecteur de vertex

La détecteur de vertex est constitué d'une chambre à dérive stéréo. Le nombre de voies à mesurer est de 648. L'électronique de lecture forme un ensemble de 324 modules abrités dans 20 chassis spéciaux interfacés à une branche Camac. D'autre part, l'électronique de recherche rapide de trace [Track Trigger], située dans 6 chassis FastBus, est incluse dans ce sous ensemble. Les temps de lecture de cet ensemble sont tels qu'ils rendent nécessaires l'adjonction de processeurs spécialisés, tant pour la lecture de l'électronique de la chambre que pour celle du Track Trigger.

L'électronique de la chambre est lue par 10 CBAs [ref15], logés dans 2 chassis Camac. Ces modules sont des processeurs Camac à base de 68000, capables de piloter une branche Camac et possédant 2 Mcoctets de mémoire double port Camac/CPU. Chacun d'eux controle 2 chassis d'électronique de la chambre.

L'électronique du Track Trigger est lue par un processeur (FIC) nommé FEROP (Front End Read Out Processor) auquel une interface FastBus a été adjointe, et situé dans un chassis VME distinct du LSC.

On trouve donc dans le LSC, outre les modules habituels, une interface Camac et une liaison VME-VME qui permettent au FIC de lire et de contrôler les CBAs et le FEROP.

Une vue de l'ensemble est donnée fig4.

Lorsque le trigger arrive, le LTU interrompt le FIC du LSC. Celui-ci prévient le FEROP et les CBAs au moyen d'accès dans des «boites aux lettres». Puis il retourne à d'autres taches. Les CBAs et le FEROP lisent (en parallèle) la partie d'électronique qu'ils contrôlent et stockent les données dans leur mémoire double port. Puis ils signalent au LTU la fin de cette tache (pour autoriser

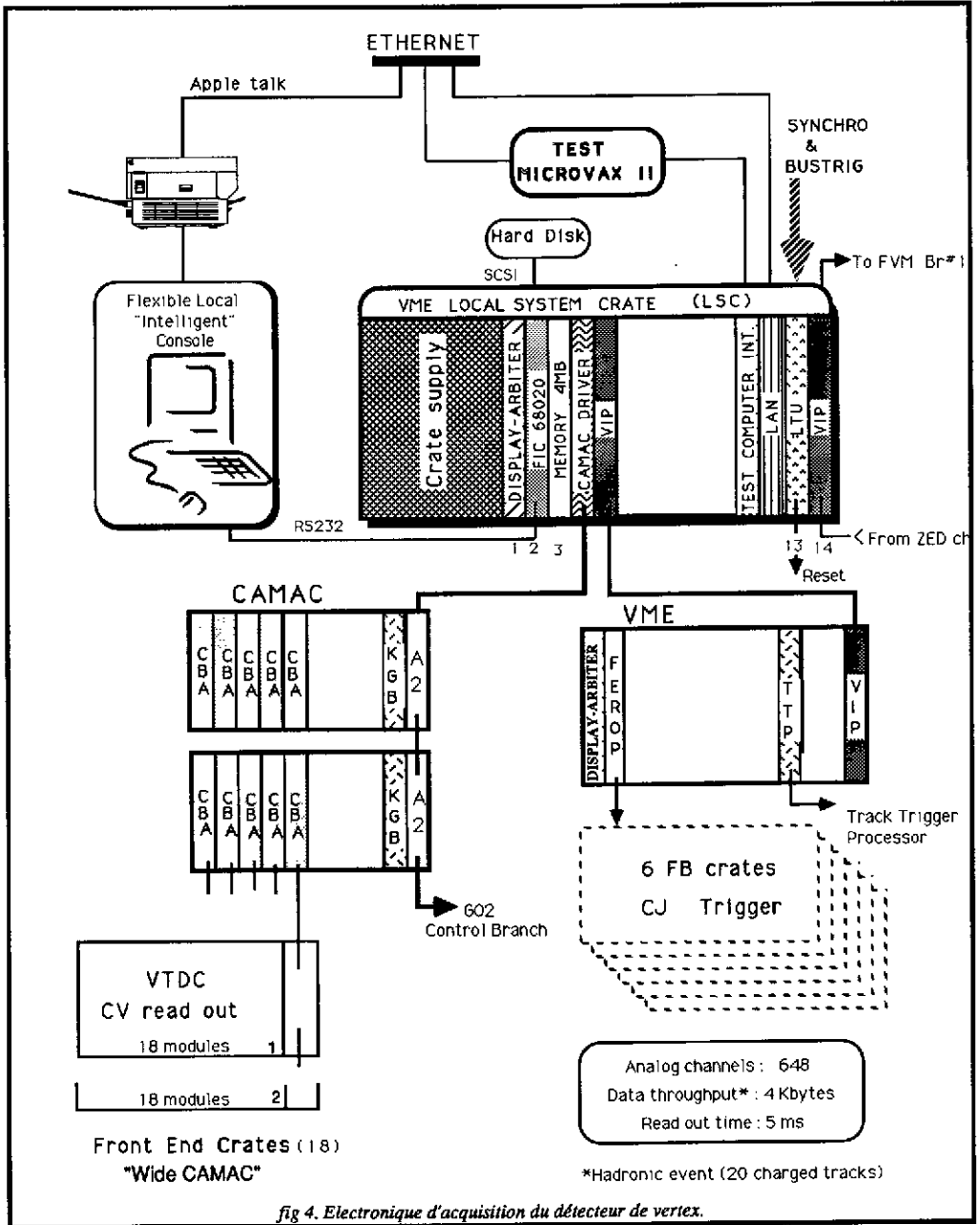


fig 4. Electronique d'acquisition du detecteur de vertex.

la prise d'un nouvel évènement).

Pendant ce temps, de façon désynchronisée par rapport au trigger, le LSC reconstruit les évènements précédant en assemblant les différents morceaux contenus dans les mémoires double port des CBAs et du FEROP, les formate et les stocke dans un buffer qui pourra être utilisé par l'étage supérieur de l'acquisition ou par d'autres tâches du FIC (histogrammation, monitoring, fabrication des tables de calibration, mise à disposition d'évènements au  $\mu$ VAX de test,...).

## 2.2 Reconstruction et traitement des évènements

Nous avons défini le système d'acquisition des différentes parties du détecteur et nous nous trouvons maintenant devant une vingtaine de Local System Crate contenant chacun une partie d'un évènement et disséminés tout autour de l'expérience (les emplacements des différents LSC sont montrés sur la fig5). Il s'agit maintenant de définir un système capable de regrouper ces différents morceaux pour reconstruire un évènement complet et capable de prétraiter ces évènements pour faire éventuellement de la réjection en ligne. En effet, nous avons vu que sur les 45000 croisements de faisceaux par seconde, seul 0.4 évènement seront intéressants. La logique de déclenchement rapide, même si elle est sélective ne nous permettra probablement pas de réduire le flux initial à mieux que 15 à 20 Hz, d'autant plus que certaines études de physique imposent d'avoir un trigger assez «lache». Ce flux n'est pas gênant du tout au niveau des LSC, puisque le temps mort à ce niveau a été réduit à 5 ms. Par contre, stocker sur support magnétique, plus de 4 évènements par secondes (soit 4 blocs de 160 KOctets), présente de grosses difficultés.

Nous devons donc étudier:

- les système de reconstruction d'un évènement complet (Event Builder)
- le système de filtrage des évènements (Filter Matrix)

Une vue de l'ensemble est donnée sur la fig6.

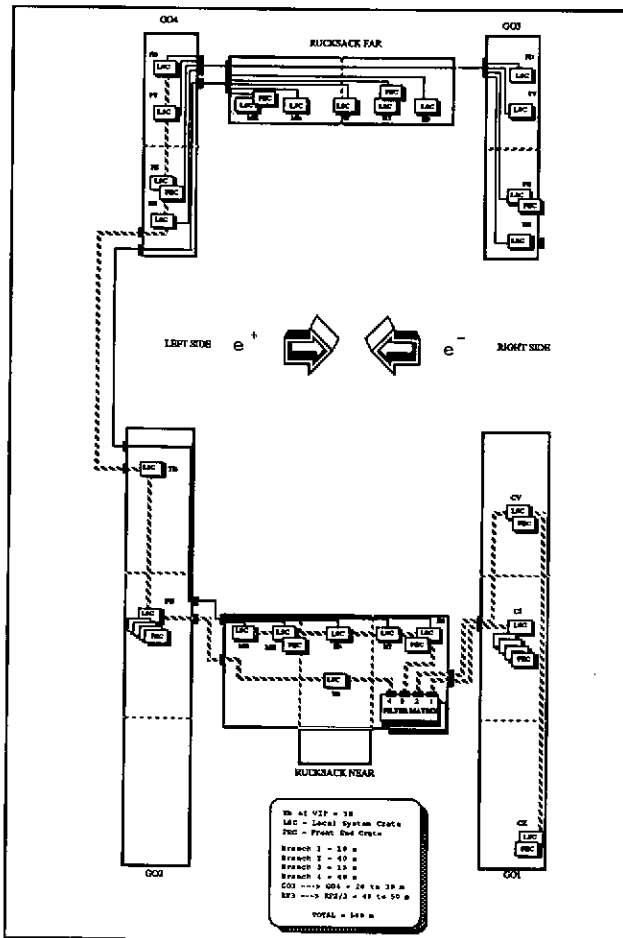


fig 5. Emplacements des différents LSC

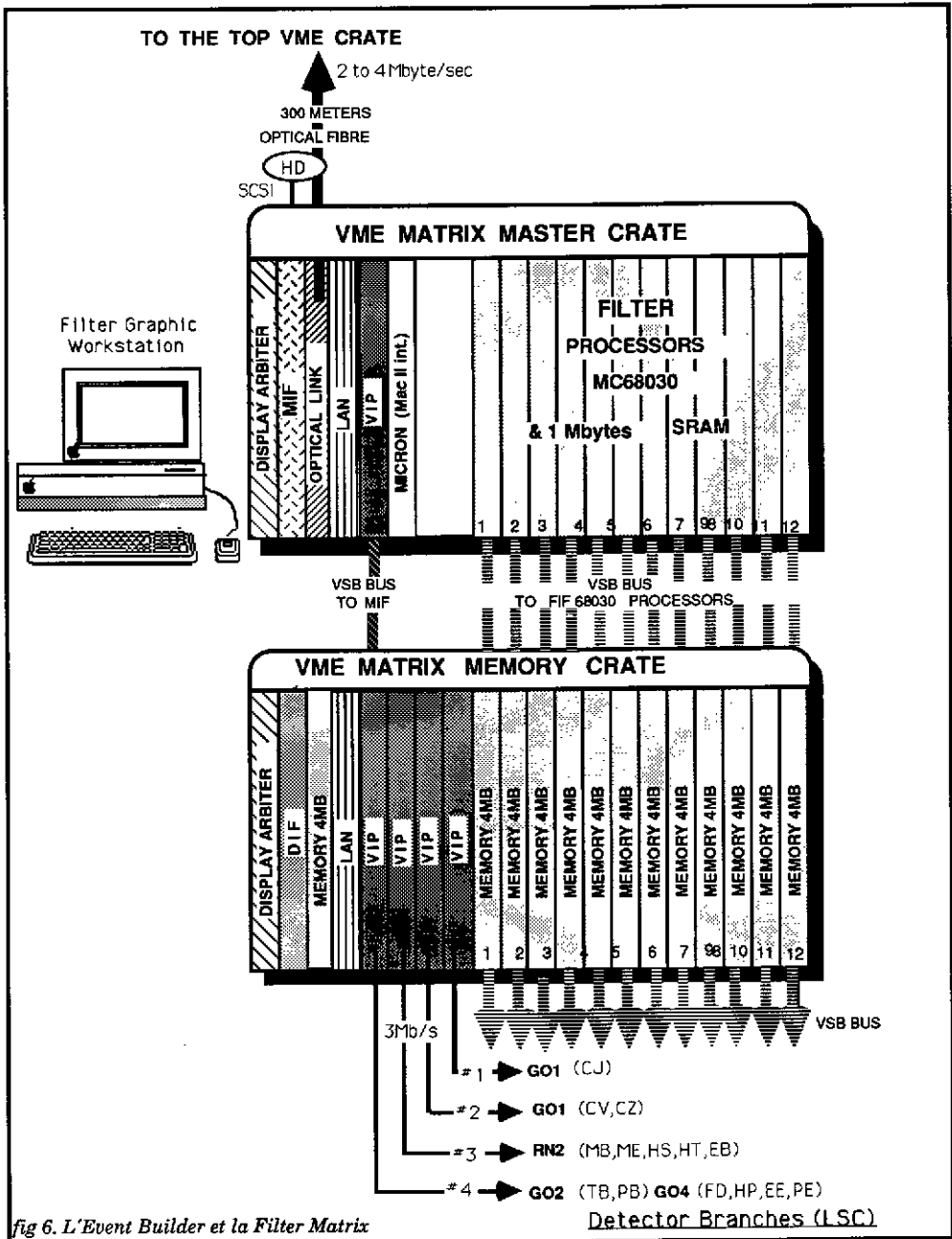


fig 6. L'Event Builder et la Filter Matrix



### 2.2.1 L'Event Builder

L'Event Builder est constitué des 6 modules suivant:

- un CPU (FIC) nommé DIF
- une mémoire DPM de 4 MOctets
- 4 VIPs connectés aux différents LSC

Tous ces éléments sont reliés entre eux par le VME et le VSB. Trois tâches principales sont exécutées par le DIF:

1) vérifier que les différents morceaux d'un évènement sont présents dans les LSC. Lire les paramètres de ces éléments (taille, adresse mémoire où ils sont accessibles). Initialiser le DMA de la mémoire. Ainsi le CPU sera libre pendant la lecture des différents blocs de données. Tous ces transferts utilisent le VSB et se font à la vitesse moyenne de 6 MOctets par seconde.

2) reconstruire un bloc évènement complet à partir des données lues précédemment. Le CPU accède à la DPM en utilisant le VME. L'interaction avec le DMA de la tâche 1 est ainsi minimisée.

3) distribuer les évènements reconstruits à la matrice de filtrage. Ceci est fait en utilisant le DMA du FIC.

Ces trois tâches sont indépendantes les une des autres car elles utilisent des buffers intermédiaires. L'utilisation des DMA fait que seuls le début de la tâche 1 et la tâche 2 sont consommateurs de temps CPU. Ceci permet de tenir un flux d'entrée important.

La tâche 1 peut être considérée comme tâche de fond: le DIF scrute régulièrement les différents LSC, pour «remonter» les morceaux d'évènement. La tâche 2 est démarrée par la tâche 1 dès qu'un évènement est disponible. La tâche 3 est déclenchée par le contrôleur de la matrice de filtrage dès qu'il constate qu'un évènement peut être traité. A ce niveau, il n'y a donc plus aucune relation avec le trigger.

### 2.2.2 La matrice de filtrage

La matrice de filtrage est constituée de 12 processeurs (CPU 68030) només FIF et de 12 mémoires double port VME-VSB, situées dans un châssis différent (cf fig 6). Chaque processeur est relié à une mémoire par une extension du bus VSB. Le principe de fonctionnement est le suivant: dès que l'Event Builder a stocké un évènement dans une mémoire, le processeur qui lui est associé peut le traiter. L'utilisation du VSB rend les 12 processeurs totalement indépendants les uns des autres; la matrice de filtrage est donc capable de traiter jusqu'à 12 évènements simultanément. Si le flux d'entrée est de 20 Hz, cela autorise 600 ms de calcul par évènement.

Le traitement envisagé ne consiste pas à reconstruire l'évènement (reconstruction des traces, identification des particules,.... qui nécessitent 4.4 secondes d'IBM 3090), mais à:

- vérifier l'intégrité de la logique de déclenchement
- marquer les évènements présentant certaines caractéristiques et que l'on désire analyser en priorité
- rejeter les évènements issus du bruit de fond

Le fonctionnement de la matrice est contrôlé par un autre CPU nommé MIF, dont le rôle est le suivant:

- contrôler l'activité des FIFs, en particulier pour connaître ceux qui sont disponibles
- transférer à l'étage supérieur de l'acquisition les évènements sortant du filtre
- donner à l'Event Builder les numéros des mémoires dans lesquelles il peut écrire un nouvel évènement

Les évènements sortant du filtre doivent être envoyés vers des calculateurs rapides (capables d'analyser finement les évènements marqués par le filtre) et le super mini d'acquisition (VAX 8700, chargé du contrôle général de l'expérience, du monitoring, et de la mise sur bande) situés à 300 m de distance. Cette liaison est réalisée au moyen d'une transmission optique [Ref 16] supportant un débit de 100 Mb/s (100 millions de bits par seconde) et permettant de transférer des données entre deux CPUs VME.

### 2.2.3 Le Top VME Crate

Le dernier châssis VME (Top VME Crate), contient un CPU (FIC) qui contrôle les transferts d'évènements à travers la liaison optique et qui les distribue soit directement au VAX pour la mise sur bande, soit à l'un des 5 émulateurs d'IBM 370 pour un traitement immédiat. Le résultat de ce traitement est ensuite transmis au VAX. Une vue de ce châssis est donnée fig 7.

### 2.2.4 Le système de VAX

Le VAX 8700, remplit plusieurs tâches:

- il met les évènements sur bande
- il contrôle des stations graphiques haute résolution
- il maintient une base de données contenant tous les paramètres du détecteur
- il permet aux utilisateurs de communiquer avec les microprocesseurs de l'acquisition

Une vue de la configuration hardware est donnée fig 8.

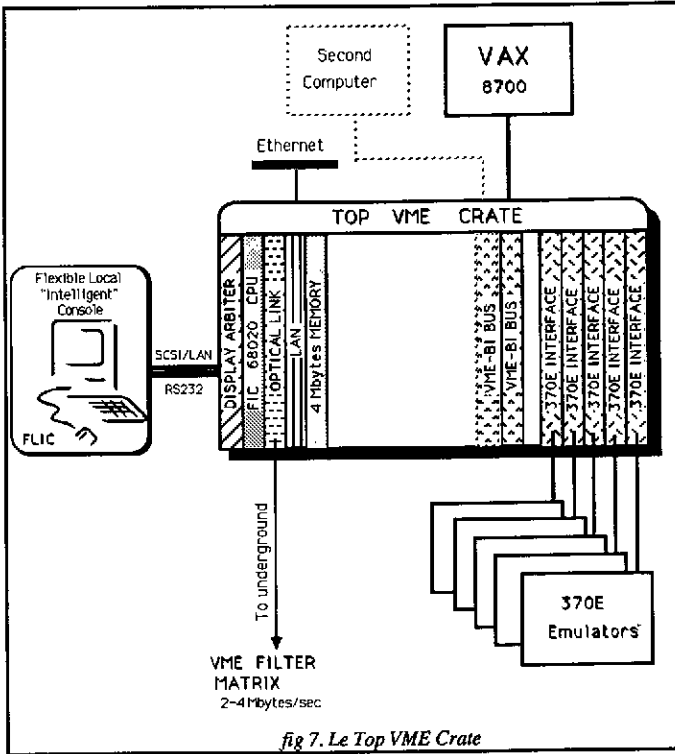


fig 7. Le Top VME Crate

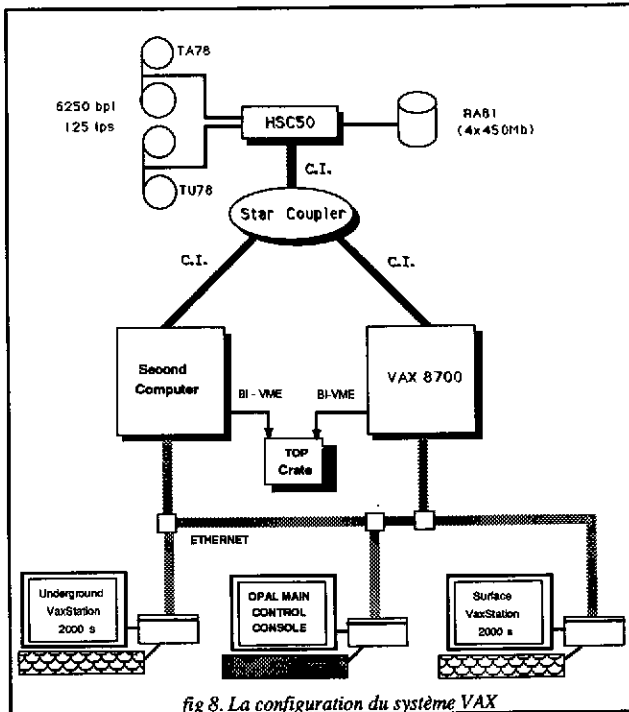


fig 8. La configuration du système VAX

### 2.3 Le logiciel

Il est évident qu'un tel système nécessite de gros développements logiciels et qu'il est nécessaire d'avoir un standard commun à tous les groupes pour:

- le contrôle de la prise de données (Run Control)
- le déclenchement des tâches d'acquisition
- la gestion des buffers événements
- le formatage des événements
- la définition et la fabrication des histogrammes
- la base de donnée
- les procédures de contrôle à distance

Nous ne donnerons que quelques informations générales sur l'ensemble de ces points; le lecteur intéressé se reportera aux références.

#### 2.3.1 Le système d'exploitation

Dans la partie précédente, nous avons beaucoup parlé de «tâches», «déclenchement par interruption», etc..., et nous avons vu que chaque processeur devait exécuté plusieurs tâches. Il est facile d'imaginer l'ajout (traitement supplémentaire, tâche de débogage,...) ou le retrait de tâches dans un processeur particulier. Ces opérations doivent se faire de façon simple sans gros développements logiciels (recompilation, nouvel édition de liens, ...). C'est pourquoi nous avons choisi d'utiliser un système d'exploitation multitâche et temps réel (c'est à dire capable de gérer les interruptions) qui fournit tous les outils dont nous avons besoin. Ce système est OS9 [Ref 17] et ces principales fonctions sont:

- réaliser l'interface entre le processeur et l'utilisateur
- gérer les opérations d'entrée/sortie
- réaliser le chargement et l'exécution des programmes
- créer et gérer un système de répertoire et de fichiers
- gérer le multitâche et le temps partagé
- allouer de la mémoire aux différents objets
- gérer la communication entre tâches
- gérer les ressources communes (par exemples les interfaces)

#### 2.3.2 Le logiciel de contrôle

Le logiciel de contrôle [Ref 18] de la prise de données, permet à un opérateur d'interagir avec les différents programmes qu'il a pu écrire et avec le système en général. Il permet aussi l'interaction avec d'autres tâches exécutées par un processeur quelconque de l'arbre d'acquisition.

Ces principales fonctions sont:

- contrôle de l'interface entre le trigger et le programme d'acquisition de l'utilisateur
- contrôle de l'interface entre le programme utilisateur et le buffer événement
- réalisation de l'interface humaine avec l'opérateur grâce à un programme de menu (Human Interface HI [Ref 19])
- contrôle et état du RUN (arrêt, pause, marche)
- récupération des erreurs et routage vers la tâche qui doit les traiter (Error Message Utility EMU [Ref 20])

L'utilisateur doit donc fournir uniquement les programmes qui lui sont propres (programme d'acquisition, programme de traitement des événements, programme de traitement d'erreur,...). C'est le logiciel de contrôle qui se chargera du bon fonctionnement de l'ensemble.

#### 2.3.3 La manipulation des données

Lorsqu'un événement a été lu, nous avons vu qu'il devait être stocké dans un buffer, puis formater, et mis à la disposition de l'étage supérieur de l'acquisition et/ou d'autres tâches telles que l'histogrammation, l'édition d'événements bruts, etc... Il faut donc un logiciel de gestion du buffer événement, un logiciel de formatage et un logiciel d'histogrammation.

Le logiciel de gestion du buffer (Event Buffer Manager EBM [Ref 21]) permet d'allouer la place mémoire nécessaire à un événement, de gérer l'accès à cet événement par différentes tâches et de restituer la mémoire lorsque l'on n'a plus besoin d'un événement. Toute tâche utilisatrice doit simplement se déclarer pendant la phase d'initialisation, puis utiliser des commandes simples (GET\_EVENT, RELEASE\_EVENT,...) pour disposer des événements. Des routines d'échantillonnage d'événements sont aussi mises à la disposition des utilisateurs; ceci est très utile pour les programmes de monitoring, qui ne traite qu'une partie des événements.

Le logiciel de formatage utilisé est ZEBRA [Ref 22]. Il permet de définir des banques de données et des liens entre ces banques. Dans le cas d'OPAL, cela permet de reconstruire très facilement un événement complet à partir des morceaux fabriqués dans les LSC: chaque groupe fabrique une structure identique à la structure d'un événement complet; les banques correspondant aux autres

sous détecteurs sont bien évidemment vides, mais l'Event Builder n'aura que quelques mises à jour de liens (pointeurs) à faire pour retrouver une structure complète.

Les histogrammes sont fabriqués par un logiciel nommé HBOOK4 [Ref 23] qui permet de définir, de remplir, de transporter, d'effacer, de détruire très facilement des histogrammes.

### 2.3.4 La base de données

Une très grande quantité d'informations sont nécessaires pour traiter les événements tant pendant la phase d'acquisition (suppression des canaux vides) que pendant la phase d'analyse (correction de gain, ...). Ces informations proviennent des résultats des calibrations des détecteurs et de l'électronique (recherche des piédestaux des codeurs par exemple). La gestion efficace de ces informations rend nécessaire l'utilisation d'un logiciel de traitement de base de données. OPAL utilise le logiciel ORACLE [Ref 24].

### 2.3.5 Les appels de procédures à distance

Le système d'acquisition de données est, nous l'avons vu, très largement distribué. Il est certain qu'au démarrage de l'expérience, chaque groupe sera proche de son détecteur et de l'électronique de lecture. La phase de débogage terminée, il sera utile de pouvoir contrôler l'expérience depuis la salle de contrôle principale. Cela impose de pouvoir exécuter toutes les commandes à distance en utilisant les réseaux locaux. C'est pourquoi, un logiciel d'appel de procédure à distance (Remote Procedure Call RPC [Ref 25]) a été développé.

## 2.4 Conclusions

Le choix de cette architecture d'acquisition, ainsi que le choix de matériel et logiciel largement soutenus dans l'industrie, nous a permis de définir un système relativement simple, évolutif (par adjonction de nouveaux modules comme par exemple des processeurs de signaux) et facile à entretenir, pour un coût de développement assez faible.

## 3. Le système de déclenchement

Le but de ce système est de trouver parmi les 45000 croisements de faisceaux par seconde, de 10 à 20 événements susceptibles d'être intéressants. Ce système doit être suffisamment rapide pour ne pas présenter de temps mort, c'est à dire qu'il doit prendre une décision entre deux croisements de faisceaux. Compte tenu des séquences d'initialisation dont ont besoin certaines électroniques de détecteurs, la décision doit être prise 16.5  $\mu$ s après le passage des faisceaux.

Cette logique reçoit des informations en provenance des différentes parties du détecteur. Les éléments utilisés ainsi que les temps d'arrivée des signaux sont donnés fig 9.

Nous allons décrire la partie «Décision Logique» (notée «Fast Decision Logic» sur la figure) ainsi que la logique de synchronisation avec l'acquisition de données.

### 3.1 La logique de décision

On dispose de deux sources d'information en provenance des sous-détecteurs:

1. Chaque sous-détecteur fournit des signaux individuels donnant des informations telles que:
  - multiplicité supérieure à un seuil
  - énergie déposée supérieure à un seuil

Environ 64 signaux de ce type sont reçus par la logique de décision qui les compte et les stockent.

2. Le détecteur est découpé en 24 tranches dans le plan perpendiculaire à l'axe du faisceau (24 phi) et en 6 tranches dans le plan perpendiculaire au premier (6 theta). On définit ainsi 144 cellules élémentaires  $X_{\theta\phi}$  pour chacun des sous détecteurs suivants (fig 10):

- détecteur central
- temps de vol (dans ce cas on n'a qu'un theta)
- calorimètre électromagnétique
- calorimètre hadronique
- chambres à muons

Ces signaux sont traités par un ensemble de modules appelé Theta Phi Matrix (TPM). Cette logique recherche la présence de traces colinéaires, de traces coplanaires avec l'axe du faisceau, de clusters dans le calorimètre électromagnétique, et elle recherche des alignements entre détecteurs.

La TPM fournit ainsi 66 informations supplémentaires que l'on combinera avec les précédentes. En début de prise de données, 32 configurations (parmi les 2<sup>130</sup> possibles) seront choisies pour définir le déclenchement.

Cet ensemble est réalisé dans un châssis Europe (de taille équivalente à deux châssis VME), dans lequel on trouve (fig 11)

- une partie VME contenant les mêmes éléments qu'un Local System Crate

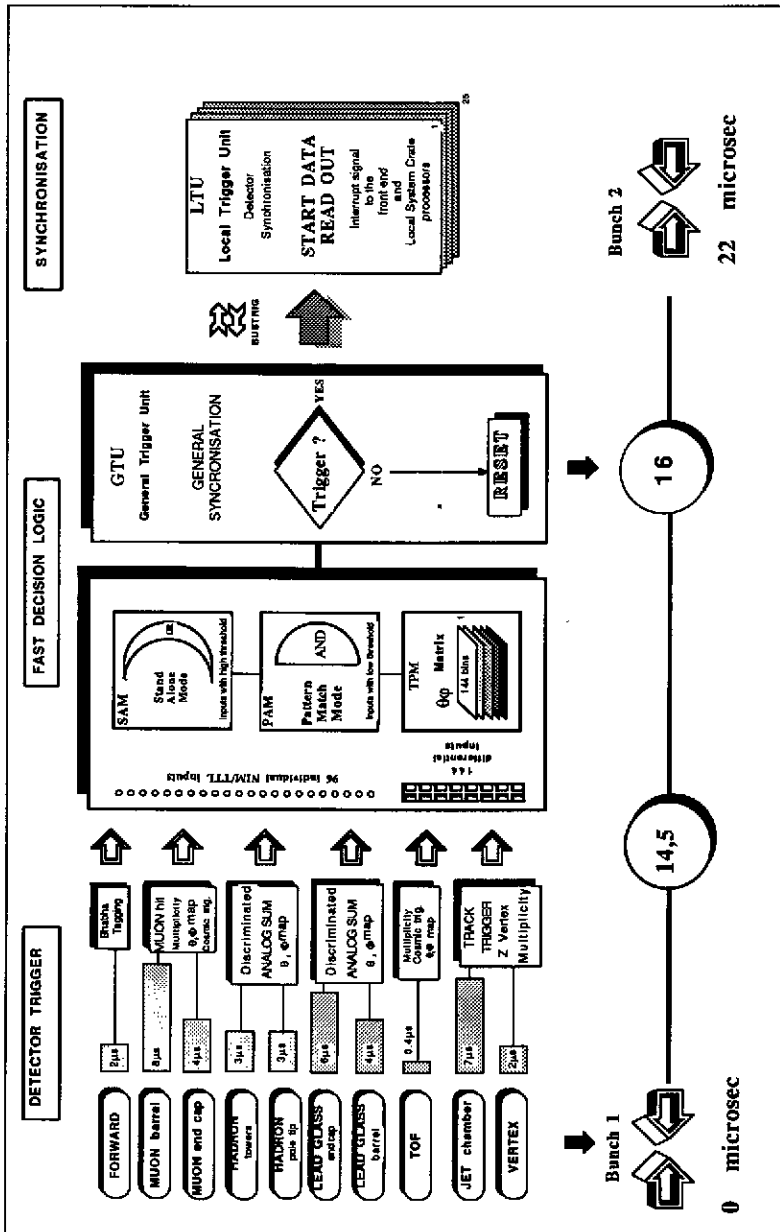


fig.9 Synoptique du Trigger

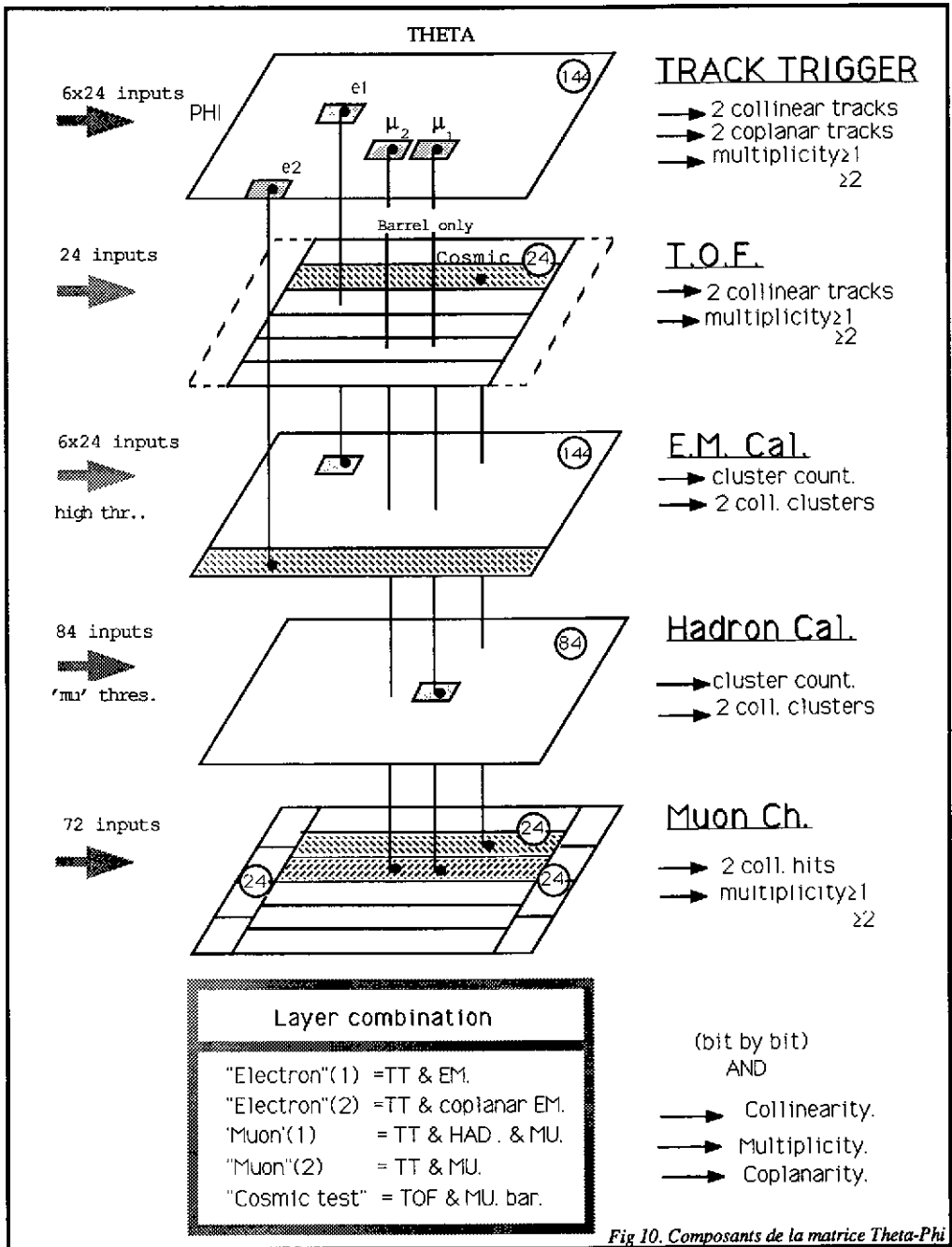


Fig 10. Composants de la matrice Theta-Phi

- la logique de décision constituée de

- 5 cartes pour la TPM
- 1 carte traitant les signaux individuels en provenance des sous détecteurs (SAM)
- 1 carte recherchant les combinaisons admissibles (PAM)
- 1 carte chargée de transmettre le trigger aux Local System Crates (GTU)

Du point de vue de l'acquisition, ce chassis est considéré comme un LSC, et toutes les données entrant dans la fabrication du trigger sont lues, en même temps que celles en provenance des sous détecteurs. Cela permet de vérifier le bon fonctionnement du trigger.

La configuration et le contrôle de la logique de déclenchement sont exécutées à partir d'un MacIntosh II. Un logiciel graphique permet de sélectionner les voies utilisées (il faut pouvoir inhiber certains signaux en cas de panne par exemple), de définir les configurations de trigger et de visualiser les taux de comptage instantanés des différentes entrées.

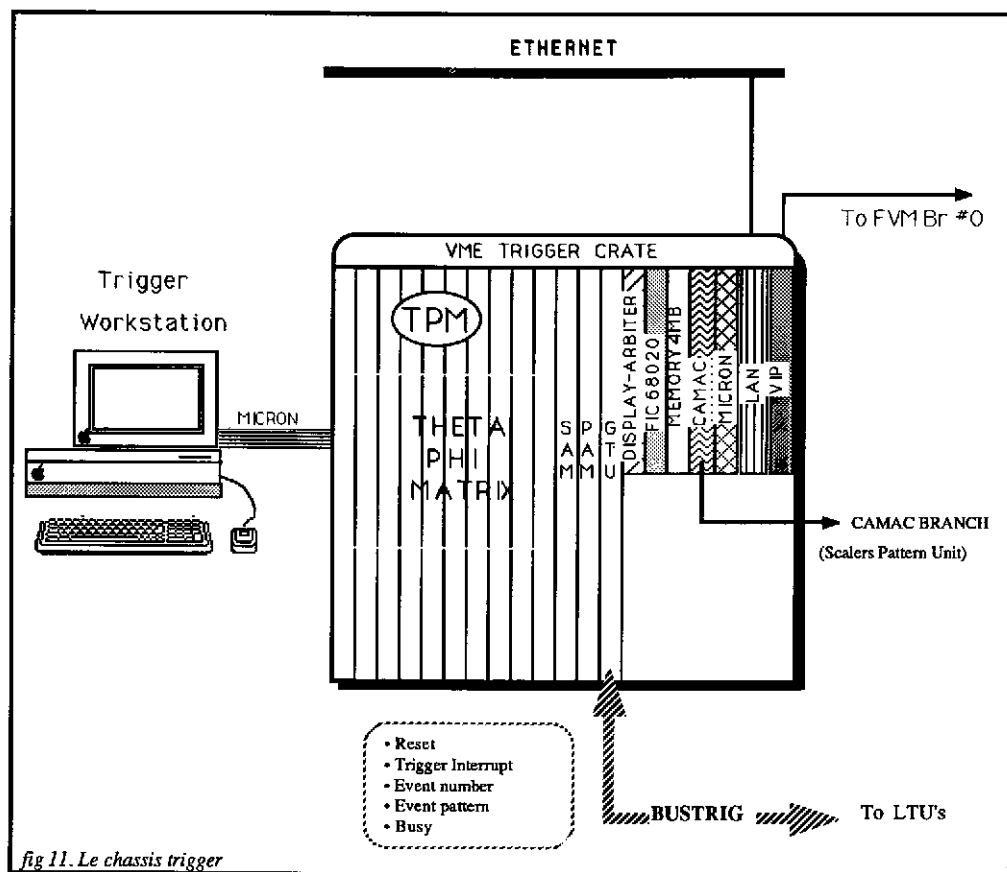


fig 11. Le chassis trigger

### 3.2 La synchronisation avec la prise de données

Un bus rapide, reliant le GTU à tous les LTUs, permet de transmettre le trigger aux processeurs des LSCs et de bloquer la logique de décision tant que l'acquisition de l'évènement n'est pas terminée. La séquence de fonctionnement est la suivante: si un évènement satisfait les critères de décision choisis, le GTU envoie aux LTUs un signal, ainsi que le numéro d'évènement et un mot de trigger (celui-ci peut être utilisé à des fins d'acquisition sélective ou de traitement particulier pour certaines configurations). Les LTUs interrompent les CPUs d'acquisition et renvoient un signal (BUSY) qui restera actif durant la prise de données. Le GTU attend que tous les signaux BUSY redeviennent inactifs pour entamer la séquence de fin d'acquisition; celle-ci consiste à attendre le croisement de faisceaux suivant pour émettre un signal de remise à zéro indispensable au bon fonctionnement de certaines électroniques. Si un évènement ne satisfait pas les critères de décision, le GTU se contente d'émettre le signal de remise à zéro. Dans ce cas, aucun temps mort n'est introduit.

Pour autoriser le fonctionnement autonome des différents sous-détecteurs, il est possible de configurer le LTU en «mode local». Dans ce cas, le LTU ne participe pas aux échanges sur le bus du trigger, et l'utilisateur peut utiliser 4 sources de déclenchement (3 entrées externes, résultats d'une logique de décision propre, ou le trigger général). Il peut aussi sélectionner le signal de synchronisation (1 parmi 8) qu'il désire utiliser. Ce signal sert à fabriquer les portes des codeurs; il peut être le signal de croisement de faisceaux, le signal de détection de particules cosmiques, la synchro d'un laser ou d'un générateur, etc...

#### 4. Conclusions

Un système d'acquisition distribuée a été présenté. Il permet de maintenir une grande souplesse d'utilisation, tout en conservant de bonnes performances de rapidité. Le prix à payer est, d'une part le développement d'un important logiciel de communication entre processeurs, et un système de déclenchement lui aussi distribué.

Ce travail est le fruit d'une collaboration importante entre plusieurs laboratoires et n'a pu être complètement détaillé dans cet article. Le lecteur intéressé pourra se reporter aux références données ci-dessous ... ou rejoindre l'expérience.

#### 5. Références

1. The OPAL detector technical proposal (Rapport CERN 1984)
2. VME Rev.C Document IEC 821
3. VSB IEC 821 Sub Bus
4. FIC 8230 CES 70 route du Pont Butin PO boîte 107 CH 1213 Petit Lancy SUISSE
5. DPM Ph. Farthouat CENS DPhPE-SEIPE 91191 GIF CEDEX
6. LRT - Filtabyte Grande Bretagne
7. LTU Ph. Farthouat
8. VIP Ph. Farthouat
9. CBD 8210 CES (cf 4)
10. Fastbus Interface FVSBI M. Weymann EF CERN
11. IKON USA
12. MAC 7212 CES
13. Seagate 277N Importateur JODE Electronique
14. FIC SCSI DMA Interface CES
15. CBA Ph. Farthouat
16. Optical Link CES
17. OS9 Microware. En France Micro Data Soft 92 Courbevoie
18. Control of OPAL Subdetectors H.Burckhart Cern Document
19. Human Interface R. Hughes-Jones Document Cern
20. EMU F. Meijers Document Cern
21. EBM P. Elcombe Document Cern
22. ZEBRA R.Brun Document Cern
23. HBOOK4 R.Brun Document Cern
24. Data Base access for OPAL DAQ system R. Middleton Document Cern
25. RPC User manual and implementation guide T.Burners-Lee Document Cern
26. Opal hardware trigger Document OPAL
27. OPAL online system design Document OPAL
28. OPAL report on data acquisition and analysis Document Cern