94 CHAM S023

LAPP-T-94/06

Université de Savoie
Laboratoirc d'Annecy-Le-Vieux de Physique des Particules
BP 110, 74941 Annecy-Le-Vieux Cedex, France

THÈSE

présentée pour obtenir le titre de

Docteur en Sciences
Spécialité : Physique Experimentale & Instrumentation

par

Xudong CAI

Contribution à l'élaboration et à la mise en oeuvre du système de déclenchement et d'acquisition de l'expérience L3 au LEP

Soutenue le 6 Octobre 1994 devant la Commission d'Examen

L. Massonnet	(Président)
J. J. Blaising	(Directeur de thèse)
O. Callot	(Rapporteur)
P. Charpentier	(Rapporteur)
A. Degré	
C. Dionisi	

Contribution to the trigger and data acquisition system of the L3 experiment at LEP

Xudong CAI

6 October 1994

Annecy, France

For my parents

Résume

Cette thèse est consacrée au système de déclenchement et d'acquisition de l'expérience L3 au LEP. Il est distribué à plusieurs niveaux, afin de déclencher, sélectionner, collecter et enregistrer les événements détectés par L3. Le système de déclenchement à trois niveaux permet de réduire un éventuel taux de déclenchement de premier niveau de 100 Hz à quelques Hz. Les systèmes de lecture et d'assemblage disposent de mémoires intermédiaires afin que seul la numérisation des données des détecteurs introduise du temps mort.

Dans cette thèse l'accent est mis sur le système de lecture du calorimètre hadronique, le système central d'assemblage et le filtre de troisième niveau.

Mots clefs : *L3–Déclenchement–Acquisition–FASTBUS–Transputer*

Abstract

This thesis is devoted to the L3 trigger and data acquisition system of the L3 experiment at LEP. It is a large distributed system with multiple levels to trigger, collect and record events detected by the L3 detector. The three trigger levels are designed to reduce a possible first level trigger rate of 100 Hz to a few Hz. The readout and event building systems have intermediate buffering so that the only dead time is introduced during digitization of the detector signals.

The emphasis of this thesis is on the Hadron Calorimeter (one of the subdetectors) readout system, the L3 event building system and the upgraded level-3 trigger system. Both hardware and software are described.

Keywords : *L3–Trigger–Data Acquisition–FASTBUS–Transputer*

Contents

List of Figures

List of Tables

Introduction (French)

La découverte en 1983 des bosons vecteurs Z^0 et $W^{\pm}$, par les expériences UA1 at UA2 [1, 2, 3, 4] au CERN, a été une confirmation remarquable des prédictions du modèle electrofaible de Glashow, Salam et Weinberg [5, 6, 7, 8, 9].

Depuis 1989 le LEP (Large Electron Positron collider) produit des collisions e^+e^- qui permettent détudier les propriétés et des couplages du boson Z^0, gŕace à sa production directe, ainsi que la production de nouvelles particules. En attendant une augmentation de l'énergie des faisceaux pour atteindre la production de paires de W, plusieurs millions de Z^0 ont déja été produits.

L'étude de la variation des sections efficaces en fonction de l'énergie dans le centre de masse pour chacun des canaux $e^+e^- \rightarrow f^+f^-$ avec $f = e\ , \mu,\ \tau,\ q$ a permis de déterminer la masse m_Z, la largeur totale Γ_Z, les sections efficaces leptoniques, la section efficace hadronique au pic σ^0_{had} ainsi que les différentes largeurs partielles Γ_{had}, Γ_{ee}, $\Gamma_{\mu\mu}$, $\Gamma_{\tau\tau}$ et par suite la largeur invisible Γ_{inv}. Γ_{inv} est la largeur partielle de désintégration en paires $\nu\bar{\nu}$. Les valeurs obtenues sont en très bon accord avec le modèle standard et montrent que le nombre de canaux de désintégration du Z^0 est tel que l'Univers ne peut compter que trois types de neutrinos légers ν_e, ν_μ, ν_τ.

Les mesures des assymétries angulaires avant-arrière de la charge des fermions produits dans les canaux $e^+e^- \rightarrow f^+f^-$ (avec $f = e\ , \mu,\ \tau$) ainsi que la mesure de la polarisation des tau conduisent à la détermination de l'angle de mélange de Weinberg $\sin^2 \Theta_w$ qui est un des paramètres essentiels du modèle electrofaible.

Le Z^0 se désintègre à 70% en une paire quark-antiquark. Il offre donc la possibilité de tester la chromodynamique quantique (CDQ), théorie de l'interaction forte incluse dans le modèle standard, et de mesurer à la masse du Z^0 l'intensité du couplage quark-gluon $\alpha_s(m_Z)$.

Jusqu'à présent aucune particule supersymetrique ou composée n'a été mise en évidence dans la limite de masse cinématique permise. Le boson scalaire de Higgs, qui dans le modèle standard génère la masse des particules, n'a pas été observé, mais les résultats obtenus indiquent que sa masse devrait être supérieure à 65 Gev. Les désintégrations rares de Z, n'ont pas permis de mettre en évidence de nouvelles interactions.

Avec les statistiques obtenus, et la mise en oeuvre de détecteurs de traces au silicium, la physique du quark b au LEP devient compétitive avec celle faite sur les autres machines. 20 désintégrations hadroniques ont lieu en paires $b\bar{b}$ permettant d'étudier la spectroscopie des hadrons "beaux" et de mesurer le paramètre de mélange B^0-$\bar{B}^0$. La mesure de la largeur partielle $\Gamma_{b\bar{b}}$ du Z^0 en paires $b\bar{b}$ permet de déterminer le couplage du Z^0 au quark b et de tester l'universalité du couplage des quarks. L'angle de mélange $\sin^2\Theta_w$ du modèle électrofaible peut également être obtenu grace à la mesure des asymétries angulaires avant-arrière de la charge ($A_{b\bar{b}}$) des paires $b\bar{b}$.

Le LEP est également un excellent collisioneur de photons, à l'énergie des paires de W les interactions photon-photon produiront la majeure partie des événements.

L'expérience L3 [10] est l'une des quatre expériences qui sont en fonctionnement depuis juillet 1989 auprès du collisioneur électron, positron du CERN. Elle a été optimisée pour la détection des électrons des photons et de muons. Le dispositif expérimental est constitué de plusieurs détecteurs disposés en couches cylindriques successives autour de l'axe du faisceau, ainsi que de deux bouchons situés aux extrémités du cylindre. En allant du point d'interaction vers l'extérieur on trouve:

- deux compteurs de luminosité en Germanate de Bismuth (BGO) mesurant la diffusion $e^+e^- \rightarrow e^+e^-$ à petit angle $24.7\ mrad \leq \theta \leq 69.3\ mrad$.

- un détecteur de trace au silicium (SMD, Silicon Microvertex Detector) installé en 1993.

- un détecteur de trace et de vertex (TEC, Time Expansion Chamber) qui mesure la trajectoire, la charge et l'impulsion des traces chargées issues du point d'interaction.

- un calorimètre électromagnétique (ECAL) constitué de 10734 cristaux en BGO qui mesure l'énergie des électrons et des photons avec une résolution en énergie de 1.3% à $E = 45$ GeV et 5% à $E = 100$ MeV.

- un hodoscope de 30 compteurs à scintillation dont la bonne résolution en temps ($< 1ns$) permet de rejeter les muons cosmiques.

- un calorimètre hadronique (HCAL) constitué de quatre longueurs d'absorption d'uranium, instrumentées par des chambres proportionelles à fils. Il mesure l'énergie des hadrons avec une résolution de $(55/\sqrt{E} + 5)\%$

- un spectromètre à muons (MUCH) qui mesure l'impulsion des muons avec une précision de 2.5% à 45 GeV.

Tout cet appareillage est contenu dans un solénoide de 7800 t produisant un champ magnétique uniforme de 0.5 Tesla.

De 1989 à 1992 le LEP a fonctionné avec 4 paquets d'électrons et 4 paquets de positrons au rythme de 44982 croisements par seconde. Depuis septembre 1992 pour atteindre la luminosité nominale de 1.5 $10^{31} cm^{-2} s^{-1}$ dans le LEP circulent 8 paquets d'électrons et 8 paquets de positrons (89964 croisements par seconde), qui entrent ainsi en collision toutes les $11.116 \mu sec$. A chaque croisement des faisceaux le système de déclenchement et d'acquisition analyse les données produites, reconnait si une interaction e^+e^- s'est produite et décide de l'enregistrement de l'événement.

A lénergie du Z^0, lorsque la luminosité est de 1.5 $10^{31} cm^{-2} s^{-1}$ le LEP produit 0.45 Z^0 par seconde, le système de déclenchement et d'acquisition doit sélectionner ces évènements parmi le bruit de fond produit par les interactions des faisceaux avec le gaz du tube à vide, les rayons cosmiques, l'électronique des détecteurs, l'activité de l'uranium du calorimètre hadronique.

Afin d'éffectuer cette sélection le système de déclenchement et d'acquisition est organisé en plusieurs niveaux:

- Le déclenchement de premier niveau utilise une logique cablée capable de prendre une décision en 11 μsec.
- Le déclenchement de deuxième niveau comporte 3 microprocesseurs.
- Le filtre de troisième niveau est constitué de 4 stations VAX.
- Le système d'acquisition central utilise le standard Fastbus.

Les deux premiers niveaux fonctionnent sur des données de faible résolution (10^4 canaux) obtenues par regroupement des capteurs. Le troisième niveau à accès à l'information complète (10^5 canaux de haute résolution).

A chaque croisement des faisceaux le système déclenche la numérisation des données résumées. Celles ci sont mémorisées dans les mémoires du niveau-2 pendant que le niveau-1 analyse l'événement. Si la décision est négative, l'électronique est remise à zéro. Si la décision est positive, les données principales sont numérisées, l'événement est construit au niveau de chaque détecteur et le niveau-2 est activé (temps de calcul $\sim$ 2 ms). Si la décision du niveau-2 est positive, les données principales, des détecteurs et les données résumées sont assemblées, l'événement est structuré, puis envoyé à l'une des stations VAX du niveau-3 (temps de calcul $\sim$ 100 ms).

Si la réponse du niveau-3 est positive l'événement est transféré dans une mémoire de la VAX7610. Un programme lit ces événements et les écrit sur bande ou les transfère vers les disques du centre de calcul de L3.

De 1989 à 1993 le système d'acquisition de L3 a permis de recueillir un total de plus de 1.5 millions de Z^0.

Le travail presenté dans ce mémoire est destiné à décrire l'architecture ainsi que les performances actuelles du système d'acquisition. En effet depuis 1989 le système a déja subi différentes améliorations.

Le premier chapitre est consacré à la description du dispositif expérimental de L3.

Le second chapitre décrit le système de déclenchement et d'acquistion.

Entre 1988 et 1990 j'ai participé a la conception et à la réalisation du système d'acquisition du calorimètre hadronique. Le troisième chapitre, présente donc l'aspect conceptuel et fonctionnel de l'acquisition de ce détecteur, la synchronisation et l'assemblage d'un événement lorsque le système est utilisé pour la calibration du calorimètre.

Le quatrième chapitre est consacré à la synchronisation et à l'assemblage des données provenants des différents détecteurs. ainsi qu'a la connection au filtre de troisième niveau.

Le dernier chapitre est consacré au filtre de troisième niveau qui a été modifié en 1993. Il décrit également les performances du système, et introduit les dévelopements en cours de réalisation.

Reference

[1] UA1 Collaboration, G.Arnison, *et al.*. **Experimental Observation of Isolated Large Transverse Energy Electrons with Associated Missing Energy at $\sqrt{s}$ = 540 GeV**. *Physics Letters* ***B 122***, pages 103–116, 1983.

[2] UA1 Collaboration, G.Arnison, *et al.*. **Experimental Observation of Lepton Pairs of Invariant Mass around 95 GeV/c^2 at the CERN SPS Collider**. *Physics Letters* ***B 126***, pages 398–410, 1983.

[3] UA2 Collaboration, P.Bagnaia, *et al.*. **Evidence for $Z^o \rightarrow e^+e^-$ at the CERN $p\bar{p}$ Collider**. *Physics Letters* ***B 129***, pages 130–140, 1983.

[4] UA2 Collaboration, R.Ansari, *et al.*. **Measurement of the Standard Model Parameters from a Antiquark pairs**. *Physics Letters* ***B 186***, pages 440–451, 1987.

[5] S.L.Glashow. **Partial-Symmetries of Weak Interactions**. *Nuclear Physics* ***22***, pages 579–588, 1961.

[6] S.L.Glashow, J.Ilipoulos, and L.Maiani. **Weak Interactions with Lepton-Hadron Symmetry**. *Physical Review* ***D 2***, No. 7, pages 1285–1292, 1970.

[7] S.Weinberg. **A Model of Leptons**. *Physical Review Letters* ***19***, No. 21, pages 1264–1266, 1967.

[8] A.Salam and J.C.Ward. **Electromagnetic and Weak Interactions**. *Physics Letters* ***13***, No. 2, pages 168–171, 1964.

[9] A.Salam. **Elementary Particle Theory**. Almquist and Wiksell, Sockolm, 1968.

[10] L3 Collaboration, B.Adeva, *et al.*. **The Construction of the L3 Experiment**. *Nuclear Instruments and Methods in Physics Research* ***A 289***, pages 35–102, 1990.

Chapitre 1

Le LEP et l'expérience L3

1.1 Le LEP

Le collisioneur Electron Positron (LEP) [1, 2] est installé dans un tunnel circulaire ayant une circonférence de 26.66 km. Le tunnel est creusé dans la molasse à une profondeur moyenne de 100m, entre les montagnes du Jura et le lac Leman, à la frontière Franco-Suisse (Figure 1.1).

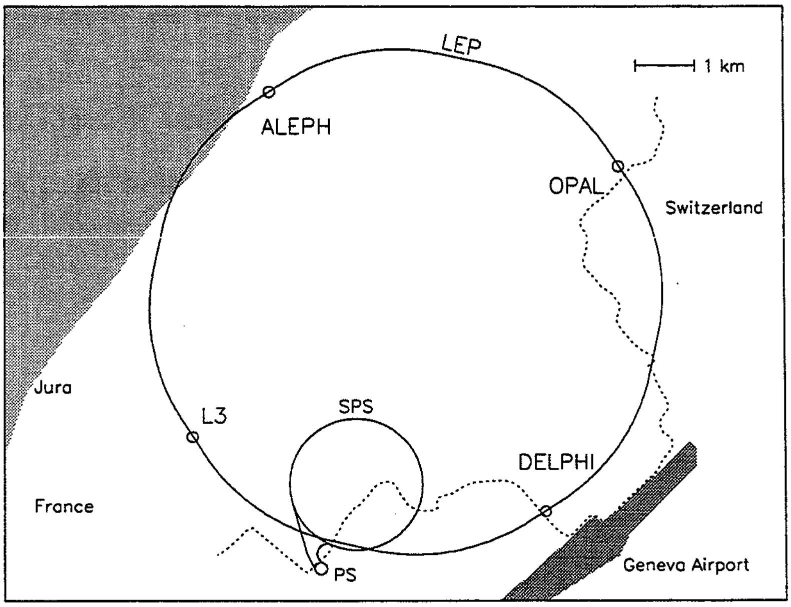


Figure 1.1: Le LEP

Les électrons et les positons circulent en sens opposé dans un tube a vide. Plus de 3000 aimants dipolaires courbent les particules le long de leurs trajectoires circulaires et environ 2000 quadrupoles focalisent le faisceau. Aux points de collision la dimension du faisceau est de 200 μm suivant la direction horizontale et de 10 μm dans la direction verticale. Le construction du LEP prévoie plusieurs phases correspondant à un accroissement de l'énergie des faisceaux. La première phase d'exploitation de LEP a commencé en 1989. La table 1.1 présente quelques paramètres de LEP pour cette première phase.

Table 1.1: Principaux paramètres de LEP (Phase 1)

Circonférence	26658.883 m
Rayon de courbure des dipoles	3096.175 m
Section horizontale du faisceau	200 μm
Section verticale du faisceau	10 μm
Fréquence de révolution	11245 Hz
Nombre de paquets par faisceau	4/8
Nombre de points de collision	4
Nombre de cavités RF	128
Frequence RF	352.20904 MHz
Energy à l'injection	20 Gev
Energie maximum	$\approx$ 60 Gev
Luminosity (3 mA par faisceau)	1.6 $\times 10^{31} cm^{-2} s^{-1}$

Les deux faisceaux peuvent entrer en collision en 8 points disposés symétriquement le long de l'anneau du LEP. Quatre d'entre eux sont utilisés par des expériences appelées L3, ALEPH, OPAL et DELPHI. Celles-ci se trouvent respectivements situées aux points No. 2, 4, 6 et 8.

Les électrons et les positons sont groupés en ***paquets*** et accélerés de 20 GeV (énergie d'injection) à 45 GeV en 15 minutes environ. De 1989 à 1992 le LEP a fonctionné avec 4 paquets d'électrons et 4 paquets de positons, chaque paquet ayant une longueur d'environ 15 mm suivant la direction des faisceux. Depuis Octobre 1992 le nombre des paquets délectrons et de positons a été augmenté à 8 afin d'accroitre la luminosité de LEP.

1.2 Le Detecteur L3

1.2.1 Introduction

Le détecteur L3 [3, 4] est le plus grand des quatre détecteurs en fonctionnement aupres du LEP. Il permet de mesurer l'impulsion et l'angle des muons, des

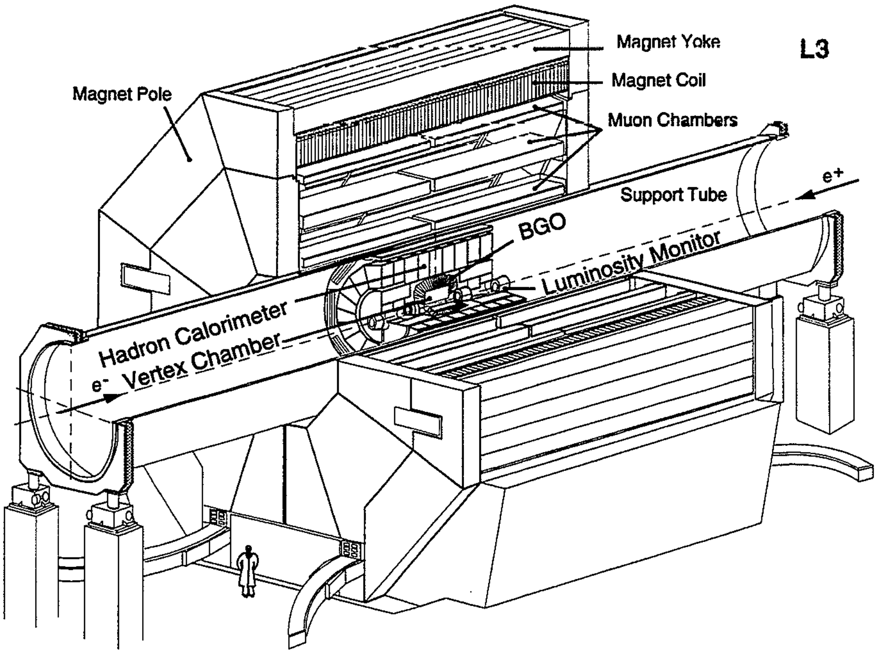


Figure 1.2: Shema du detecteur L3

électrons, et des photons, la position du vertex de l'interaction ainsi que l'energie des jets de hadrons. La Figure 1.2 représente le détecteur L3.

Le détecteur est installé à l'interieur d'un aimant, ses divers éléments sont positionnés à l'aide d' un tube d'acier de 4.45 m de diamètre et de 32 m de long. L'aimant solenoidal, de forme octoganale (16 m de diamètre et 12 m de long) est constitué de 6700 tonnes d'acier et d'une bobine en aluminium d'un poids de 1100 tonnes. Lorsque le courant est de 30338 A le champ magnétique atteint 0.5 T. A partir de la ligne de faisceau les principaux éléments du détecteur sont:

- Le détecteur central de trace.
- Le calorimètre électromagnetique.
- Les compteurs à scintillation.
- Le calorimètre hadronique.
- Le spectromètre à muon.
- Les moniteurs de luminosité.

La Figure 1.3 indique le système de coordonnées utilisé dans L3.

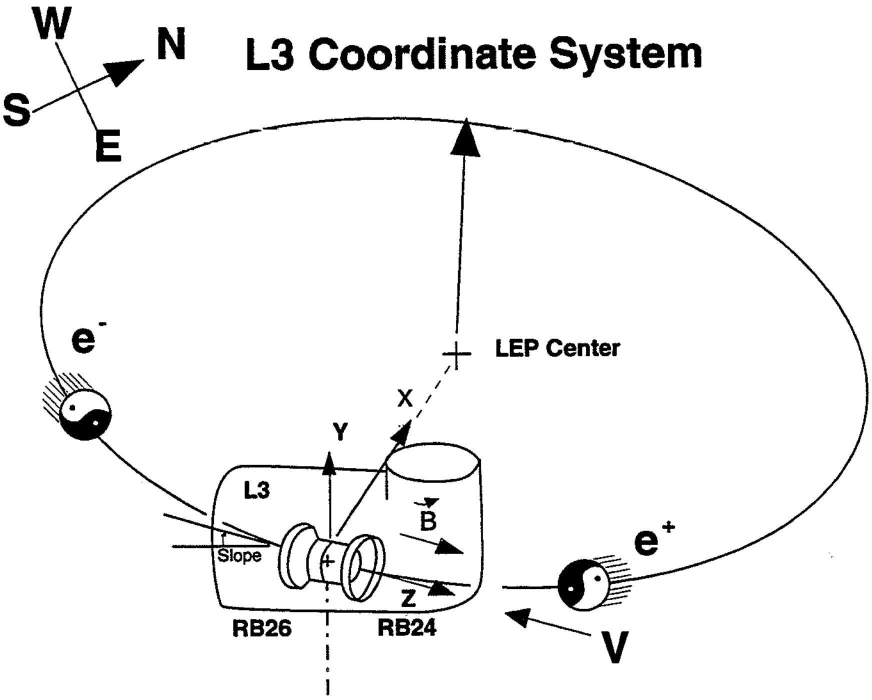


Figure 1.3: Système de coordonnées utilisé dans L3

1.2.2 Le détecteur central de trace

Le détecteur central de trace [5], shematisé sur la Figure 1.4, est constitué d'une chambre à dérive ***Time Expension CHamber (TECH)*** à faible vitesse de dérive, 6 μm/ns, et d'une chambre proportionelle, chambre-Z, permettant de:

- mesurer avec précision la position et l'angle des traces des particules chargées.
- mesurer l'impulsion transverse et identifier la charge des particules jusqu'à une impulsion de 50 GeV /c.
- reconstruire le point d'impact et l'angle des particules chargées à la face d'entrée du calorimètre électromagnétique.
- déterminer la multiplicité des traces provenant du vertex au niveau du système de déclenchement.
- reconstruire la position du point d'interaction ainsi que les vertex secondaires des particules ayant une durée de vie supérieure à $10^{-13}s$.

La TECH comprend deux cylindres concentriques. La chambre intérieure est constituées de 12 secteurs équipés de 8 fils chacun, la chambre extérieure étant composé

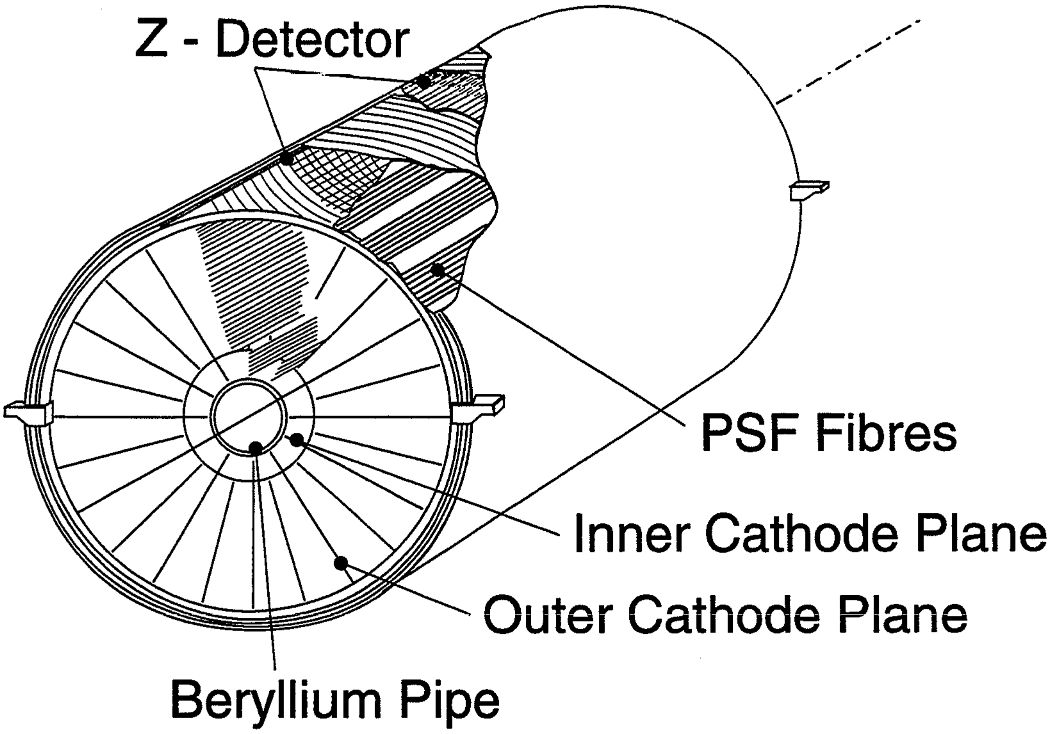


Figure 1.4: Le détecteur central de trace

de 24 secteurs incluant 54 fils. Deux types de fils sont utilisés, les fils d'anodes et les fils de division de charge, ils déterminent respectivement la cordonnée ϕ et la coordonnée selon l'axe Z. De chaque coté de la région d'amplification des groupes de fils de grille permettent de résoudre l'ambiguité gauche droite. Le mélange gazeux permettant d'obtenir une vitesse de dérive lente est constitué de 80% CO_2 et de 20% C_4H_{10}, il est utilisé à une pression de 1.2 bar. Les signaux des anodes sont numérisés par des convertiseurs analogiques digitaux FADC. Pour une trace de grand P_T (45 GeV) les 62 fils de la TECH permettent d'obtenir une résolution simple de 60 μm en moyenne.

Un ruban de 143 fibres scintillantes en plastic (PSF) paralèlle à l'axe du faisceau est utilisé pour la calibration de la vitesse de dérive de la TECH.

Sur la surface extérieure de la TECH la chambre-Z est constitué de deux chambres proportionelles cylindriques, à lecture par la cathode. Elle permettent une mesure plus précise de la coordonnée Z suivant l'axe du faisceau. La chambre Z fonctionne avec un mélange gazeux composé de 80% d'Argon et de 20% de CO_2. La chambre Z a une résolution simple comprise entre 300 μm et 1 mm suivant l'angle polaire de la trace.

1.2.3 Le Calorimètre Electromagnétique

Le calorimètre électromagnétique (ECAL) [6] entoure le détecteur central de traces, il permet de mesurer l'énergie et l'angle des photons et des électrons produits lors des collisions e^+e^-. Il est constitué d'environ 11000 cristaux de BGO (Oxyde de Germanate Bismuth,$Bi_4Ge_3O_{12}$) Figure 1.5. Le BGO est particulièrement bien adapté à la construction d'un calorimètre compact, car il a une faible longueur de radiation, une grande longueur d'interaction et des propriétés chimiques stables. De plus le BGO a une bonne résolution en énergie, (4% à $E = 200$ MeV et 1.3% à $E = 45$ GeV [7]) et une excellente linéarité. Chaque cristal est un tronc de pyramide ayant une section d'entrée de 2 x $2cm^2$ et une section de sortie de 3 x $3cm^2$. Sa longueur et de 24 cm, soit 22 longueurs de radiations. Deux photodiodes associées à une électronique linéaire détectent la lumière scintillante du BGO.

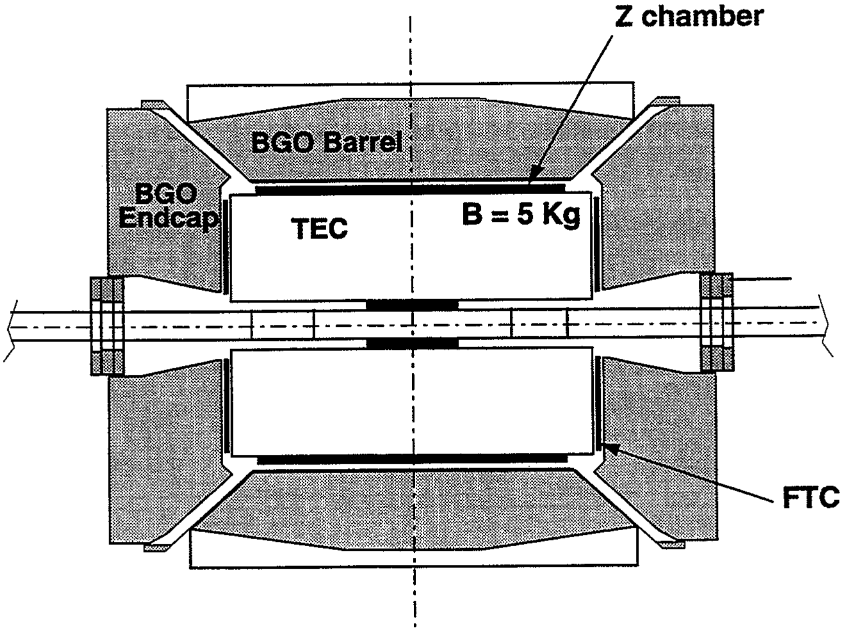


Figure 1.5: Le Calorimètre Electromagnétique

Le calorimètre électromagnétique est composé d'un tonneau (EB) et de deux bouchons (EC) précedés de chambres à dérive (FTC). Le tonneau est constitué de 7680 cristaux, organisés en 48 anneaux de 160 cristaux, couvrant une région angulaire ente 42° et 138°. Le tonneau est mécaniquement divisé en deux demi tonneau symmétriquement par rapport à l'axe $\theta = 90°$, ceci a permis de calibrer chacun des cristaux dans un faisceau d'électrons. Les deux bouchons étendent la couverture angulaire jusqu'à 12° et 168°, chacun contient 1536 cristaux, il furent installés durant l'arrêt hivernal 1990-1991 du LEP.

1.2.4 Les Compteurs à Scintillation

Les compteurs à scintillation [8, 9] sont situés entre les calorimètres électromagnétique et hadronique (Figure 1.6). Les compteurs du tonneau couvrent une région angulaire allant de 34° à 146° par rapport à la direction du faisceau. Ils sont utilisés dans un déclenchement de multiplicité permettant de sélectionner les événements hadroniques et pour distinguer les di-muons des muons cosmiques en mesurant la différence de temps de vol entre des compteurs opposés. Celle-ci est de 6ns pour les muons cosmique et zéro pour les paires de muon.

Les 30 compteurs, constitués de 1 cm de scintillateur plastique, sont courbés afin de s'adapter à la forme du tonneau du calorimètre hadronique Chaque compteur est équipé d'un photomultiplicateur à chaque extrémité. Le temps et l'amplitude du signal sont analysés à l'aide de convertisseurs digitaux TDC et ADC.

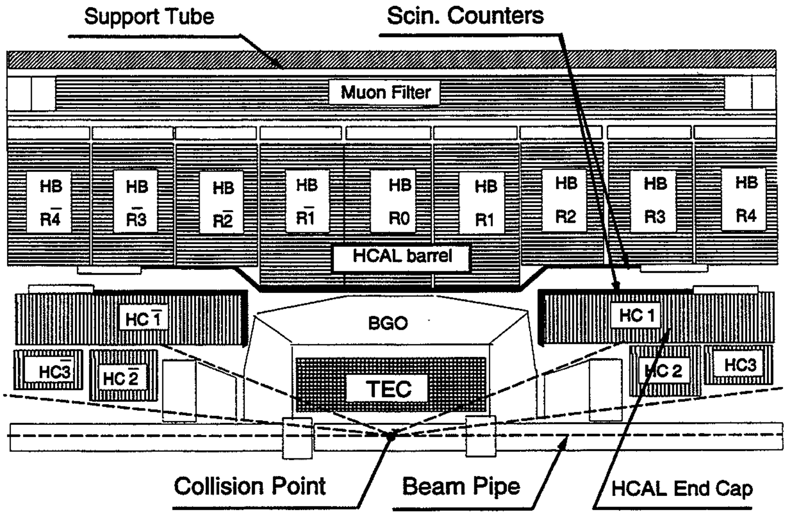


Figure 1.6: Coupe de la partie interne du détecteur L3

1.2.5 Le Calorimètre Hadronique

A l'extérieur du calorimètre électromagnétique, après les scintillateurs, se trouve le calorimètre hadronique (HCAL) [10, 11], Figure 1.6. Il permet de mesurer l'énergie des hadrons provenants des interactions e^+e^- en mesurant, avec l'aide du BGO, l'energie totale. Il agit également comme un filtre à muon.

Le HCAL contient trois parties: Le tonneau (HB), les bouchons (HC) et le filtre à muon (MF). Le tonneau est divisé en 9 anneaux de 16 modules chacun. Le rayon intérieur est de 885 mm pour les trois anneaux centraux et de 979 mm pour les six autres. Il couvre une région angulaire allant de 35° à 145° suivant la direction θ.

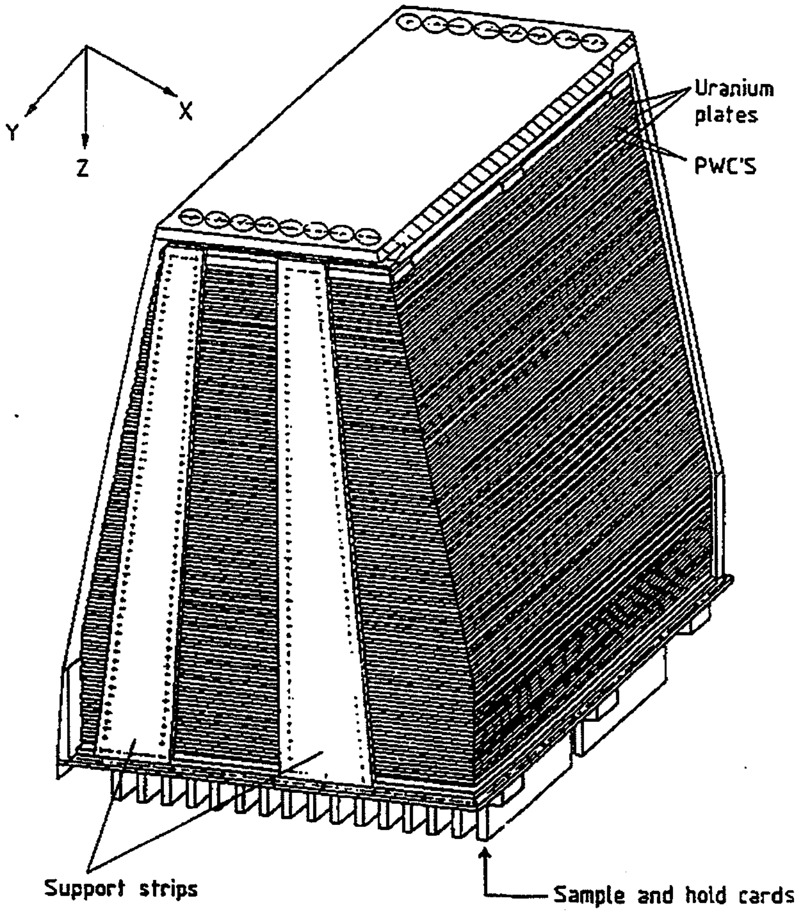


Figure 1.7: Module du calorimètre hadronique

Les modules sont constitués d'empilements de plaques d'absorbeur en uranium appauvri et de chambres proportionnelles. La Figure 1.7 montre un module. Les plans de chambre sont organisés afin que les fils d'un plan soient perpendiculaires aux fils du plan suivant. Les chambres ayant les fils parallèles à la direction du faisceau sont appelées chambres P(hi), les autres sont appelées chambres T(heta). Le mélange gazeux est constitué de 80% d'Argon et de 20% CO_2, il donne un gain tres élevé ($\approx 10^4$ à une tension de 1.6 kV). La résolution totale en energie est de 10% pour des désintégrations hadroniques au Z^0.

Chaque bouchon est constitué de trois anneaux (HC1, HC2 et HC3 Figure 1.6), chaque anneau est divisé en deux parties afin de faciliter l'installation. Les modules des bouchons ont une structure semblable à ceux du tonneau, mais il ont des tailles différentes et comportent moins de plans de chambre. Les deux bouchons étendent la couverture angulaire du HCAL entre 5.5° et 174.5°. Les signaux recuillis par les fils d'anode sont amplifiés ct numérisés à l'aidc de convertiseurs analogiques digitaux Fastbus.

Grace à la densité de l'uranium, l'ensemble des deux calorimètres (ECAL et HCAL) fournissent environ 6 à 7 longueur d'absorbtion. Le filtre à muons est installé sur la face interne du tube support ajoutant environ 1 longueur d'absorbtion supplémentaire. Il améliore l'efficacité de détection des muons en réduisant la pénétration des muons venant de la désintégration des pions. Le filtre à muons est constitué de 8 octants, chacun d'eux étant composé de six plaques de 1 cm de laiton séparés par des plans de chambres proportionelles. Les chambres utilisent le même mélange gazeux que le tonneau du HCAL et fonctionnent à une tension de 1.8 kV.

1.2.6 Le Spectromètre à Muons

Le système de chambres à muons [12, 13] est installé à l'extérieur du tube support. Il est composé de deux roues, chacune étant constituée de huit modules appelés octants. La Figure 1.8 donne une vue globale du Spectromètre à Muons.

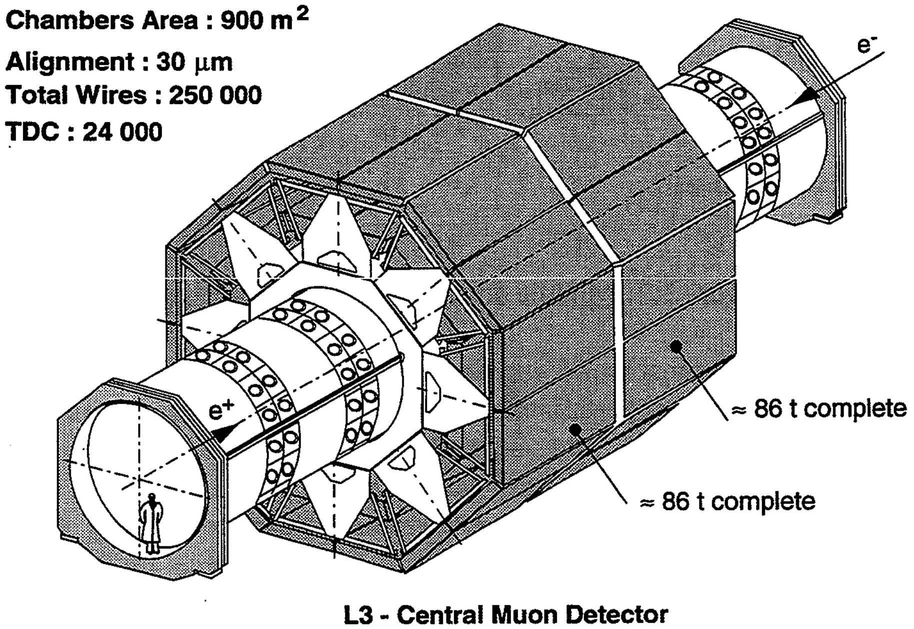


Figure 1.8: Vue globale du Spectromètre à Muon

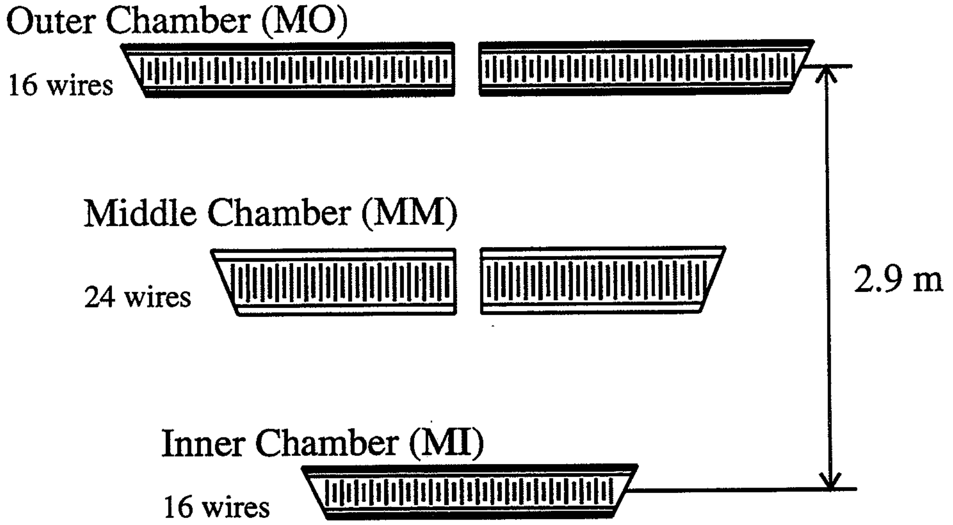


Figure 1.9: Shema d'un octant du système de chambres à muon

Chaque octant (Figure 1.9) est constitué de 5 chambres (P) à dérive: une chambre interne (MI), deux chambres intermédiaires (MM) et deux chambres externes (MO). Elles mesurent les coordonnées des traces dans le plan de déflexion.

Chacune des chambres MI et MO est couverte par 2 chambres-Z dont les fils sont disposés perpendiculairement à la direction du faisceau afin de mesurer la coordonnée suivant l'axe Z. Les signaux des chambres P et Z sont envoyés vers des discriminateurs et des TDC Fastbus permettant de mesurer les temps de dérive, par rapport au signal de référence généré lors du croisemenent des faisceaux.

Le système de chambres à muon est destiné à mesurer l'impulsion des muons avec une precision d'environ 2.5% à 45 GeV. Le mélange gazeux est constitué de 61.5% d'Argon et de 38.5% Ethane pour les chambres-P, de 91.5% Argon et 8.5% de Methane pour les chambres-Z. La région angulaire couverte est $36° \leq \theta \leq 144°$. Elle sera étendue dans les régions avant et arrière en 1994 et 1995.

1.2.7 Les Moniteurs de Luminosités

Les deux moniteurs de luminosité [14, 15] sont installés symmétriquement de part et d'autre du point d'interraction à une distance $z = \pm 2765mm$, Figure 1.2. Ils permettent de calculer la luminosité du faisceau avec une précision de l'ordre de 1% en mesurant la diffusion élastique $e^+e^- \rightarrow e^+e^-$ (Bhabha) à petit angle. Chaque moniteur couvre une région angulaire comprise entre 24.7 *mrad* et 69.3 *mrad*. Il est composé d'un calorimètre électromagnétique en BGO et d'un ensemble de chambres proportionelles placé devant le BGO.

Celles-ci ont été remplacé en 1993 par des détecteurs de traces au silicium permettant d'avoir une meilleure définition du volume fiduciel.

Reference

[1] **LEP Design Report (Vol. II) : The LEP Main Ring**. European Organization for Nuclear Research (CERN), CERN-LEP 84-01, June 1984.

[2] I.Wilson and H.Henke. **The LEP Main Ring Accelerating Structure**. European Organization for Nuclear Research (CERN), LEP Division, CERN 89-09, November 1989.

[3] L3 Collaboration, B.Adeva, *et al.*. **The Construction of the L3 Experiment**. *Nuclear Instruments and Methods in Physics Research A 289*, pages 35–102, 1990.

[4] L3 Collaboration. **Letter of Intent**. European Organization for Nuclear Research (CERN), CERN/LEPC 82-5, March 1982.

[5] D.N.Ren. **The L3 Vertex Chamber Development and Infrastructure**. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, 1990.

[6] L3 Collaboration, F.Ferroni, *et al.*. **The L3 BGO Electromagnetic Calorimeter at LEP**. *Nuclear Physics B 23A*, pages 100–106, 1991.

[7] Jorg Wenninger **Measure de paramètres électrofaibles du Z^0 avec la réaction** $e^+e^- \rightarrow e^+e^-(\gamma)$. PhD thesis, Université de Genéve. Geneve, 1992.

[8] U.Uwer. **Aufbau und Eichung der Szintillationszaehler des L3-Experimentes**. Diploma thesis. Rheinisch-Westfälische Technische Hochschule (RWTH), Aachen, 1990.

[9] U.Uwer. **The L3 scintillation counter system**. L3 Collaboration, L3 Note 1400, March 1991.

[10] L3 Collaboration, A.Arefrev, *et al.*. **Proportional Chambers for the Hadron Calorimeter of the L3 Experiment**. *Nuclear Instruments and Methods in Physics Research A 275*, pages 71–80, 1989.

[11] L3 Collaboration. **Hadron Calorimetry in the L3 Detector**. *Nuclear Instruments and Methods in Physics Research A 302*, pages 53–62, 1991.

[12] L3 Collaboration, B.Adeva, *et al.*. **Muon Detector in the L3 Experiment at LEP**. *Nuclear Instruments and Methods in Physics Research A 277*, pages 187–193, 1989.

[13] Y.Peng. **The Muon Spectrometer of the L3 Detector at LEP**. PhD thesis, University of Amsterdam, 1988.

[14] H.Meinhard. **Precise Measurement of the Luminosity at LEP**. In J.R.Sanford, editor, *26th International Conference on High-energy Physics, Dallas*, Augest 1992.

[15] F.Linde. **Luminosity Measurement at LEP**. In K.Bos and B. van Eijl, editors, *Workshop on detector and event simulation in high energy physics – Monte Carlo 91, Amsterdam*, April 1991.

Introduction

The study of the constituents of matter and the fundamental interactions requires higher and higher particle energy. Most of current searches in elementary particle physics are based on experiments of collision between high energy particles (electrons, protons, and their anti-particles). The ***Large Electron Positron*** collider (LEP) which started to operate in August 1989 provides a unique solution for studying the electromagnetic and weak interactions because the initial state is well defined.

L3 is one of the four experiments installed on the LEP which is situated at CERN (European Organization for Nuclear Research). The ***L3 experiment*** was designed to have an excellent energy resolution measuring electrons, muons and photons. Due to the fast colliding rate of the LEP, a large number of detector cells generate a large amount of data so that its data acquisition system becomes very important. A fast but flexible ***trigger*** system is also required to select the interesting events from among the various sources of background.

This thesis describes the trigger and data acquisition system of the ***L3 experiment***. There are five chapters. The ***LEP collider*** and the ***L3 detector*** are introduced in the first one. The structure of the L3 trigger and data acquisition system is described in the second chapter. Then the other three chapters explain in detail the author's contributions respectively on the ***Hadron Calorimeter readout system***, the ***L3 event building system*** and the ***third level trigger system***.

Chapter 1

LEP and the L3 Experiment

1.1 The LEP

The Large Electron Positron (LEP) collider [1, 2] is built in a tunnel of 26.66 km circumference. The tunnel runs about 100 meters deep under the plain between the Jura mountains and the Geneva lake at the border of France and Switzerland (see Figure 1.1).

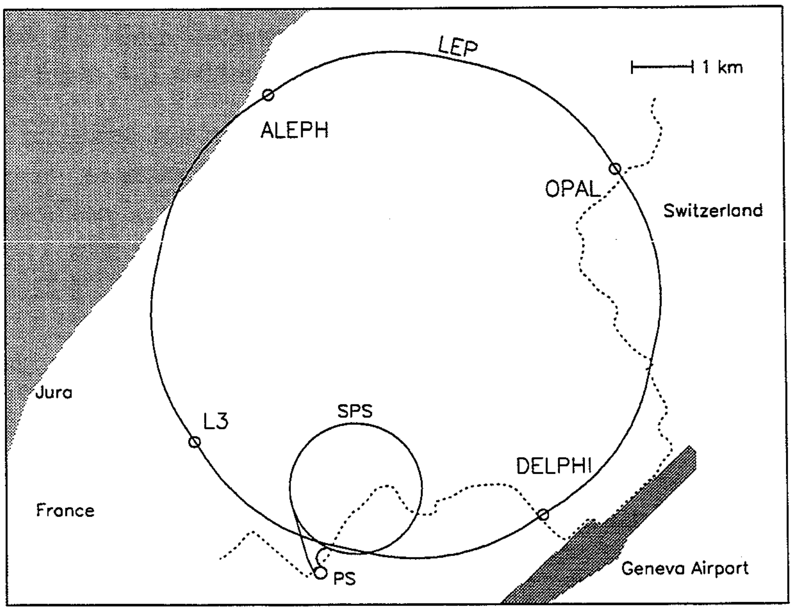


Figure 1.1: The map of LEP

Electrons and positrons circulate in opposite directions inside a beam pipe. More than 3000 bending dipoles hold the particles on their circular orbits, and over 2000 quadrupoles focus the beam. In each interaction points, the beam dimension is 200 μm in the horizontal direction and 10 μm in the vertical direction. The construction of LEP is planned to evolve through several phases corresponding to increasing beam energy. The first phase of LEP has been operating since 1989. Table 1.1 exhibits some parameters of LEP in the first phase.

Table 1.1: The major parameters of LEP (Phase 1)

Circumference	26658.883 m
Dipole bending radius	3096.175 m
Horizontal rms beam radius	200 μm
Vertical rms beam radius	10 μm
Revolution frequency	11245 Hz
Number of bunches per beam	4/8
Number of interaction points	4
Number of RF cavities	128
RF frequency	352.20904 MHz
Injection energy	20 Gev
Maximum beam energy	$\approx$ 60 Gev
Peak luminosity (3 mA beam current)	1.6 $\times 10^{31} cm^{-2} s^{-1}$

The two beams cross each other at 8 places symmetrically distributed along the LEP ring, where 4 of them are used for the experiments named L3, ALEPH, OPAL and DELPHI. They are respectively located in the point No.2, 4, 6 and 8.

Electrons and positrons are grouped in ***bunches*** and accelerated in the LEP from 20 GeV (at the injection time) to 45 GeV within about 15 minutes. At the beginning, LEP was running with 4 bunches of electrons and 4 bunches of positrons, where each bunch has a length of about 15 mm in the beam running direction. To improve the luminosity, the number of bunches was increased to 8 in October 1992.

1.2 The L3 Detector

1.2.1 Introduction

The L3 detector [3, 4] is the largest of the four detectors at LEP, it was designed as a general purpose detector to measure both momentum and direction of muons, electrons and photons with high resolution as well as the vertices and the energy flow of hadronic jets. Figure 1.2 gives a schematic view of the L3 detector.

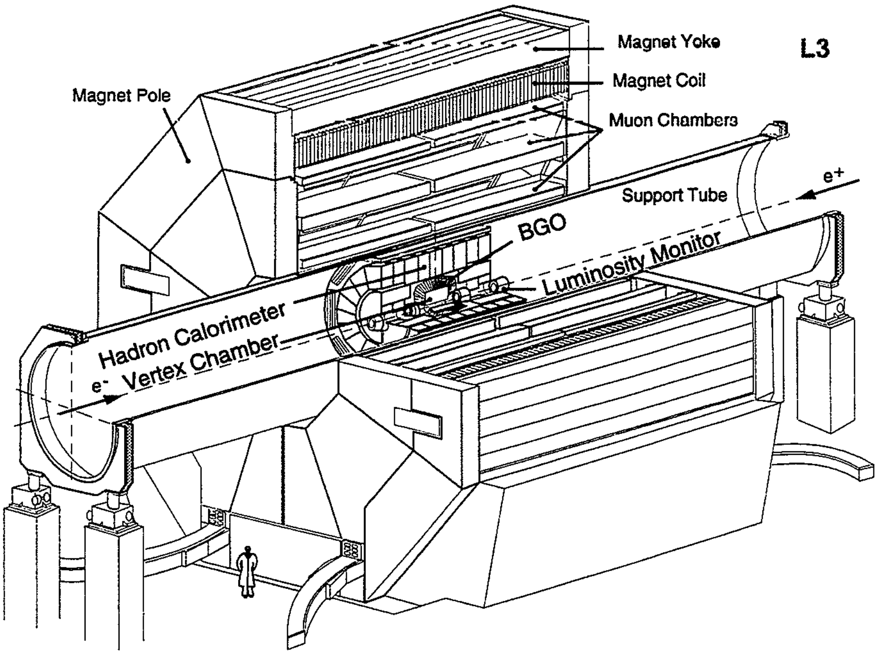


Figure 1.2: The perspective view of the L3 detector

The entire detector resides inside a magnet and is supported by a steel tube with 4.45 m inner diameter and 32 m long. The octagonal solenoid magnet (16 m diameter and 12 m long) consists of a 6700 tons yoke of low carbon steel and a huge aluminum coil weighing about 1100 tons. When the current is 30338 A, the magnetic field is 0.5 Tesla. Starting from the beam line the major components of the L3 detector are:

- The central track detector.
- The electromagnetic calorimeter.
- The scintillation counters.
- The hadron calorimeter.
- The muon spectrometer.
- The luminosity monitors.

Figure 1.3 shows the coordinate system of the L3 detector corresponding to the LEP ring.

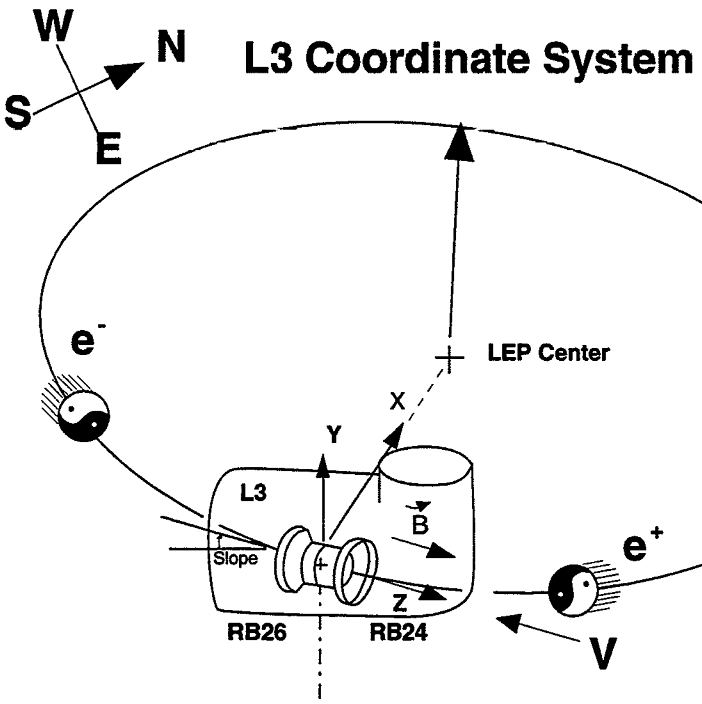


Figure 1.3: The L3 coordinate system

1.2.2 The Central Track Detector

The central track detector [5], as shown in Figure 1.4, consists of a ***Time Expansion Chamber*** (TEC, TECH) with low drift time velocity 6 μm/ns and a proportional Z-Chamber, it was designed with the following goals:

- Measure precisely the direction and the location of charged particle tracks.
- Measure the transverse momentum and the sign of the charge for particles with momentum up to 50 GeV/c.
- Reconstruct the impact point and the direction for charged particles at the entrance of the ***electromagnetic calorimeter***.
- Determine the track multiplicity originating from the interaction region at the trigger level.
- Find the interaction point and secondary vertices for particles with lifetime greater than $10^{-13}s$.

The TEC is composed of two concentric cylinders. The inner TEC consists of 12 sectors with 8 wires each and the outer TEC has 24 sectors with 54 wires each. Two

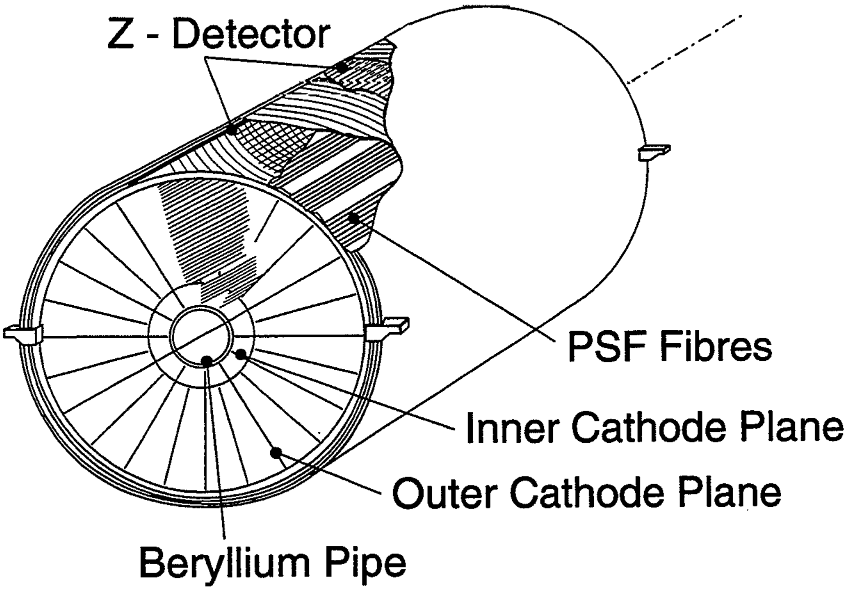


Figure 1.4: The central track detector

types of wires, sense anode wires and charge division wires, determine respectively the ϕ and the Z coordinates. Additionally, groups of five grid wires on each side of the amplification region are used to solve the left-right ambiguity. The gas mixture of 80% CO_2 and 20% C_4H_{10} is used at a pressure of 1.2 bar. The sense wire signals are digitized with analog digital converters (FADC). For a high PT track (45 GeV) the 62 wires provide a single track resolution of 60 μm in average.

A plastic scintillating fiber (PSF) ribbon, with 143 fibers running along the beam direction, are used for the calibration of the TEC.

On the outer shell of the TEC, the Z detector is composed of two cylindrical proportional chambers with cathode strip readout. It provides a more precise measurement of the position along the beam direction. The Z chamber is operated with a gas mixture of 80% Argon and 20% CO_2. The Z-chamber has a single track resolution ranging from 300 μm to 1 mm following the polar angle of the track.

1.2.3 The Electromagnetic Calorimeter

The ***Electromagnetic Calorimeter*** (ECAL, BGO) [6] surrounds the central track detector to measure the energy and the position of photons and electrons produced by the e^+e^- interactions. It is made up of about 11000 BGO (Bismuth Germanate

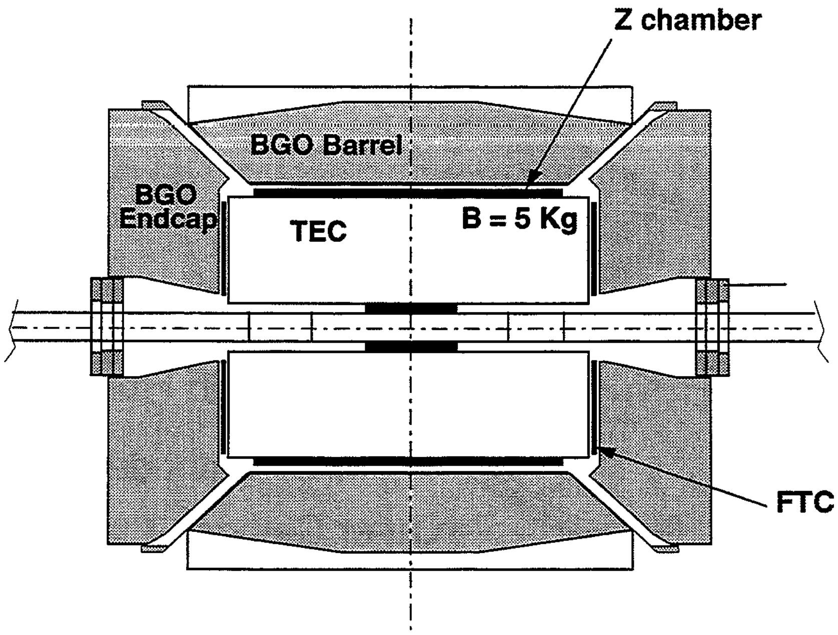


Figure 1.5: The electromagnetic calorimeter

Oxide, $Bi_4Ge_3O_{12}$) crystals, see Figure 1.5. BGO is a particularly attractive material to build a compact calorimeter, for its short radiation length for photons and electrons, large nuclear interaction length and stable chemical properties. Furthermore, BGO has very good intrinsic energy resolution (about 4% at 200 MeV and 1.3% at 45 GeV [7]) and excellent linearity. Each crystal is a truncated pyramid with a cross-section of $2 \times 2cm^2$ at the inner end and $3 \times 3cm^2$ at the outer end. Its length is 24 cm, corresponding to 22 radiation lengths. Two silicon photodiodes and associated linear electronics detect the scintillation light produced by each BGO crystal.

The ***electromagnetic calorimeter*** is composed of a cylindrical barrel (EB), two endcaps (EC) with tracking chambers (FTC) in front. The barrel is made of 7680 BGO crystals, arranged in 48 rings of 160 crystals each, which covers the polar angular region from 42° to 138°. The barrel is mechanically divided into two half-barrels symmetrically around $\theta = 90°$, it allowed the calibration of each crystal in a direct electron beam. The two endcaps extend the polar angular coverage to 12° and 168° and contain 1536 crystals in each. The endcaps were installed during 1990-1991 LEP shutdown.

1.2.4 The Scintillation Counters

The ***scintillation counters*** [8, 9] are located between the electromagnetic and hadronic calorimeters (Figure 1.6). They are used to build a multiplicity trigger for hadronic events and to discriminate di-muon events from cosmic muon by measuring the time-of-flight difference between opposite scintillation counters. The time difference is 6 *nsec* for cosmic ray background and zero for the muon pairs.

The 30 counters, made of 1 cm thick plastic scintillator are bent to suit the shape of the hadron calorimeter barrel. Each counter has one phototube at each end. The time and the amplitude are read out by high precision TDC and ADC. The counters cover the polar angular region from 34° to 146° with respect to the beam line.

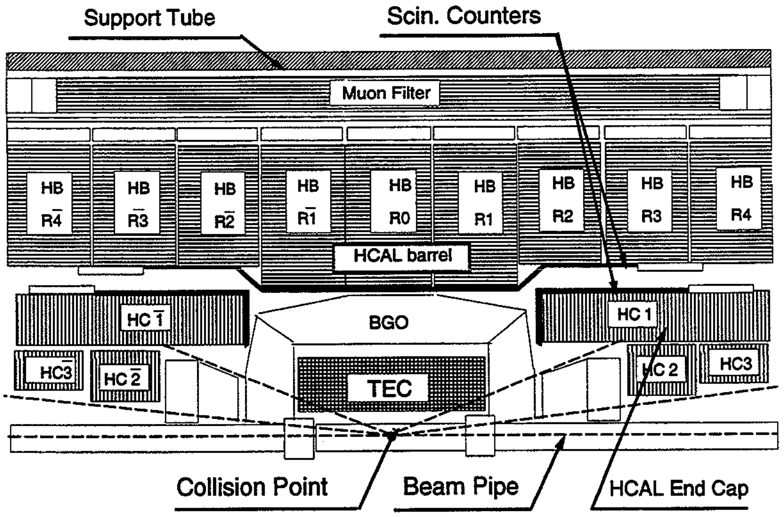


Figure 1.6: The cutaway side view of inner L3 detector

1.2.5 The Hadron Calorimeter

Surrounding the ECAL is the ***Hadron Calorimeter*** (HCAL) [10, 11], see Figure 1.6. It measures the energy of hadrons emerging from the e^+e^- interactions by the total absorption technique, together with BGO crystals. It acts also as a muon filter.

The HCAL contains three parts: the barrel part (HB), the forward-backward part (endcaps, HC) and the muon filter part (MF). The HB part is split into 9 ring with 16 modules in each ring. The inner radius is 885 mm for the central three rings (long modules) and 979 mm for the other six rings (short modules). It covers a range of 35° to 145° in the θ direction.

The modules are constructed with depleted uranium absorber plates sandwiched alternatively with layers of proportional chambers. Figure 1.7 shows a HB module.

The chamber layers in each module are arranged so that the orientation of the wires in one chamber layer are perpendicular to those in the next chamber layer. The chambers with wires along the beam direction are called P(hi) chambers, while the other chambers are called T(heta) chambers. The working gas is a mixture of 80% Argon and 20% CO_2, it has a very large gain coefficient ($\approx 10^4$ under the 1.6 kV working voltage). The total energy resolution is 10% for hadronic decays of Z^0.

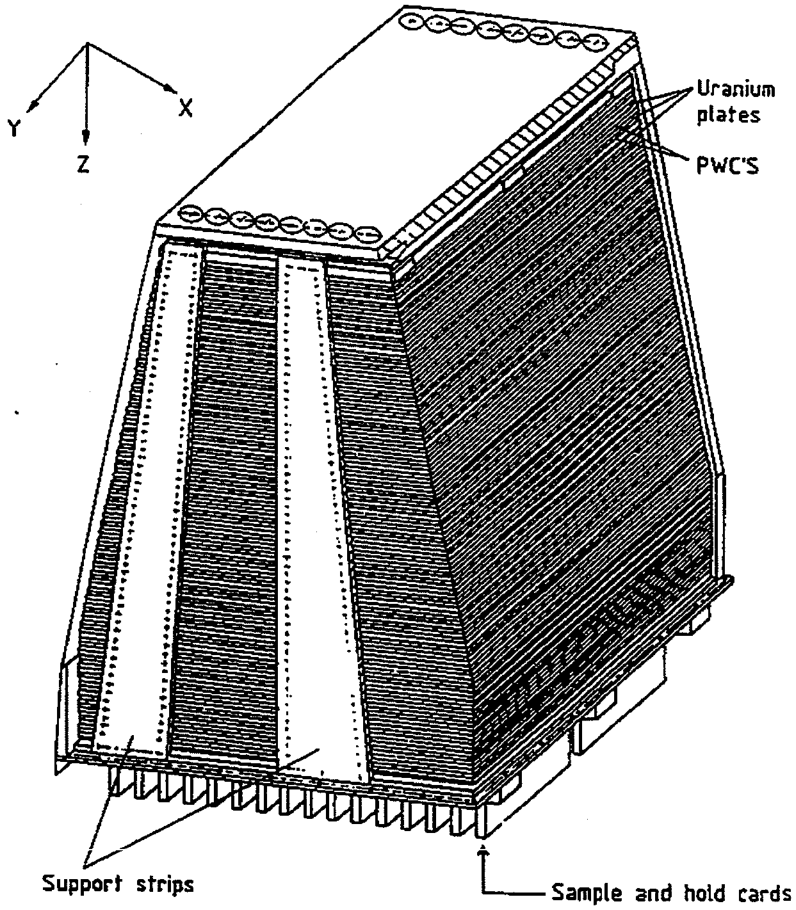


Figure 1.7: Assembly of the HCAL module

Each endcap consists of three rings (HC1, HC2 and HC3, see Figure 1.6), each ring is divided into two half rings to provide easy installation. The modules of the endcaps have a similar structure to the barrel modules but with smaller size and less chamber layers. Both endcaps together extend the angular coverage to a span of 5.5° to 174.5°.

Due to the high density of the uranium, the combined calorimeters (ECAL and HCAL) provide about 6 – 7 nuclear absorption length. The muon filter is mounted on the inside wall of the support tube and adds 1.03 absorption length to the barrel HCAL. It increases the muon detection efficiency by reducing punch-through. The muon filter contains 8 identical octants, each made of six 1 cm thick brass absorber plates interleaved with five layers of proportional chambers and followed by five 1.5 cm thick absorber plate matching the circular shape of the support tube. The chambers use the same gas mixture as the HCAL barrel under 1.8 kV working voltage.

The FASTBUS digitizers are used to produce the data from all of the three HCAL parts.

1.2.6 The Muon Spectrometer

The ***Muon Chambers*** (MUCH) [12, 13] is located outside of the support tube. It is divided into two wheels, each composed of eight modules called octants. Figure 1.8 gives an overview of the whole muon spectrometer.

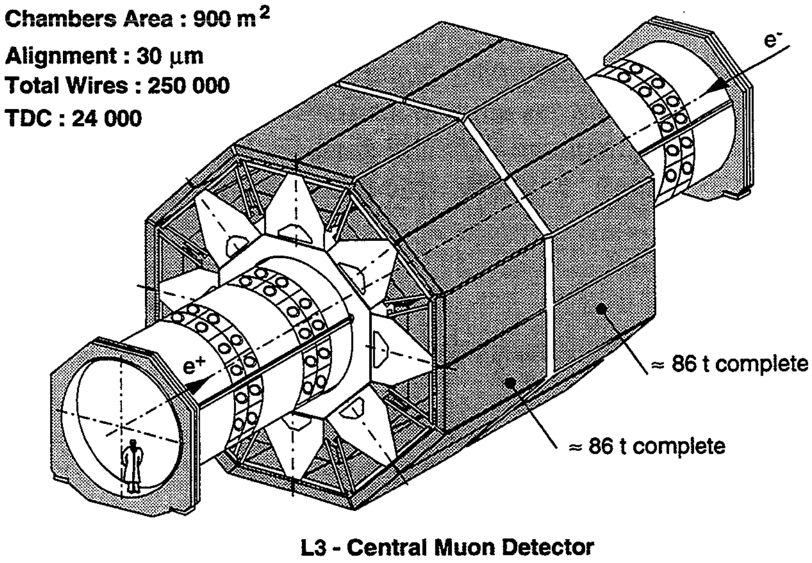


Figure 1.8: The overview of the muon spectrometer

Each octant (see Figure 1.9) is composed of five Precision (P) drift chambers: one inner chamber (MI), two middle chambers (MM) and two outer chambers (MO). They measure the track coordinates in the bending plane. In addition, each of the MI and MO chambers is covered by 2 Z-chambers (respectively called II, IM on MI chamber and OO, OM on MO chamber) with wires perpendicular to the beam direction to measure the Z coordinate. The signals from P and Z chambers are sent to discriminators and FASTBUS time digitizers allowing to measure the drift time in comparison with the beam crossing signal.

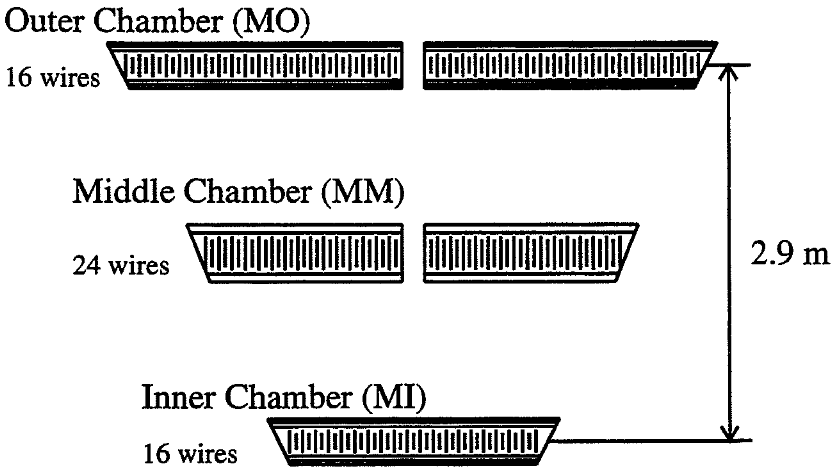


Figure 1.9: The end view of one muon octant

The MUCH is designed to measure muon momenta with a momentum precision of about 2.5% at 45 GeV. The gas mixture is 61.5% Argon and 38.5% Ethane in the P-chambers, and 91.5% Argon and 8.5% Methane in the Z-chambers. The angular coverage is $36^\circ \leq \theta \leq 144^\circ$. It will be extended in the forward-backward region in 1994 and 1995.

1.2.7 The Luminosity Monitors

The two luminosity monitors [14, 15], situated symmetrically on each side of the interaction point, at $z = \pm 2765mm$, are shown in Figure 1.2. They provide beam luminosity information with about 1% precision by using small angle $e^+e^- \rightarrow e^+e^-$(Bhabha) scattering events. Each monitor is composed of a cylindrical BGO electromagnetic detector and a set of proportional chambers.

In 1993 the chambers were replaced by silicon detectors allowing to have a better definition of the fiducial volume. Each monitor covers the forward or backward polar angle between 24.7 *mrad* and 69.3 *mrad*.

Reference

[1] **LEP Design Report (Vol. II) : The LEP Main Ring**. European Organization for Nuclear Research (CERN), CERN-LEP 84-01, June 1984.

[2] I.Wilson and H.Henke. **The LEP Main Ring Accelerating Structure**. European Organization for Nuclear Research (CERN), LEP Division, CERN 89-09, November 1989.

[3] L3 Collaboration, B.Adeva, *et al.*. **The Construction of the L3 Experiment**. *Nuclear Instruments and Methods in Physics Research A 289*, pages 35–102, 1990.

[4] L3 Collaboration. **Letter of Intent**. European Organization for Nuclear Research (CERN), CERN/LEPC 82-5, March 1982.

[5] D.N.Ren. **The L3 Vertex Chamber Development and Infrastructure**. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, 1990.

[6] L3 Collaboration, F.Ferroni, *et al.*. **The L3 BGO Electromagnetic Calorimeter at LEP**. *Nuclear Physics, B 23A*, pages 100–106, 1991.

[7] Jorg Wenninger **Measure de paramètres électrofaibles du Z^0 avec la réaction** $e^+e^- \rightarrow e^+e^-(\gamma)$. PhD thesis, Université de Genéve. Geneve, 1992.

[8] U.Uwer. **Aufbau und Eichung der Szintillationszaehler des L3-Experimentes**. Diploma thesis. Rheinisch-Westfälische Technische Hochschule (RWTH), Aachen, 1990.

[9] U.Uwer. **The L3 scintillation counter system**. L3 Collaboration, L3 Note 1400, March 1991.

[10] L3 Collaboration, A.Arefrev, *et al.*. **Proportional Chambers for the Hadron Calorimeter of the L3 Experiment**. *Nuclear Instruments and Methods in Physics Research A 275*, pages 71–80, 1989.

[11] L3 Collaboration. **Hadron Calorimetry in the L3 Detector**. *Nuclear Instruments and Methods in Physics Research A 302*, pages 53–62, 1991.

[12] L3 Collaboration, B.Adeva, *et al.*. **Muon Detector in the L3 Experiment at LEP**. *Nuclear Instruments and Methods in Physics Research A 277*, pages 187–193, 1989.

[13] Y.Peng. **The Muon Spectrometer of the L3 Detector at LEP**. PhD thesis, University of Amsterdam, 1988.

[14] H.Meinhard. **Precise Measurement of the Luminosity at LEP**. In J.R.Sanford, editor, *26th International Conference on High-energy Physics, Dallas*, August 1992.

[15] F.Linde. **Luminosity Measurement at LEP**. In K.Bos and B. van Eijl, editors, *Workshop on detector and event simulation in high energy physics – Monte Carlo 91, Amsterdam*, April 1991.

Chapter 2

The L3 Trigger and Data Acquisition System

2.1 Introduction

The aims of the L3 Trigger and Data Acquisition (DAQ) system [1, 2, 3, 4, 5] are to select and record the events which have particles coming from the e^+e^- interaction vertex.

The system is achieved by three levels of trigger for the event selection, four subdetector DAQ systems for the signal digitization and the data readout as well as a two level event building system for the data collection. To insure good efficiency, each trigger level has redundant selection criteria which are combined in a logical OR to make the decision. The system is shown in Figure 2.1.

The detector signals provide two data streams. The coarse data are used by the level-1 and level-2 trigger systems to select events. The level-1 trigger decision is made within 11.1/22.2 μsec which is the time between two beam crossings. The precise data are produced by the subdetector DAQ systems only when the event is selected by the first level trigger. By design the only deadtime in the system is introduced during the digitization and readout of the precise data which need at least 500 μsec per event. The second level trigger system collects the trigger data and uses them to reject the background event which accepted by the level-1 trigger.

At the event building stage, the main data stream and the trigger data stream are integrated by the event builders according to the level-2 trigger selection. The events accepted by the second level trigger are sent to the third level trigger system for the final selection. The level-3 trigger selection is based on the online reconstruction by using the precise data. The selected events are transferred to the online VAX computer

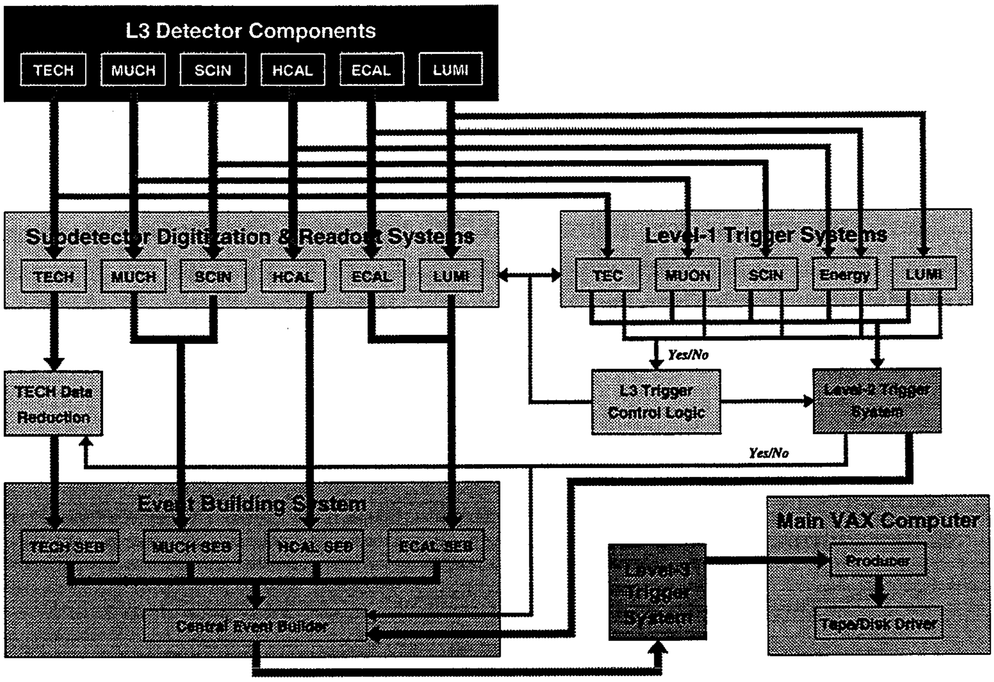


Figure 2.1: The structure of the L3 trigger and DAQ system

for data monitoring and storage.

In order to synchronize different parts of the system, a trigger control logic provides the synchronization signals as well as the event number which is used to assemble the data from the different subdetectors and the trigger system at the event building level.

The different parts of the L3 trigger and data acquisition system are briefly described in the following sections.

2.2 The First Level Trigger System

In the first level, there are five subtriggers based on the signals from the calorimeters, the luminosity monitors, the TEC chambers, the scintillation counters and the muon spectrometer respectively. They use the coarse trigger data from the detector to make the decision at each beam crossing. Since the LEP collision rate is 45/90 kHz, the level-1 trigger decisions have to be made within 22.2/11.1 μsec. Both the trigger raw data and the decisions are sent to the ***level-2 trigger system***.

2.2.1 The Calorimeter Trigger

The ***calorimeter trigger*** [6, 7] is designed to select events with energy in the electromagnetic and hadronic calorimeters as well as in the luminosity monitors, it is also called the ***energy trigger***. The analog sum signals from BGO crystals of both ECAL barrel and endcaps are grouped into 512 channels. The ***luminosity monitors*** provide 32 signals. In the same way, the HCAL analog sum signals are grouped into 384 channels. The signals from the 928 channels, in total, are digitized and converted into energy depositions. An event is selected if one of the following conditions is fulfilled.

- **Total energy trigger :**
  The following thresholds are used for the energy coming from different layers of the calorimeters.
  1. Total energy from all calorimeters layers is larger than 20 GeV.
  2. Total energy from barrel of BGO and Hadronic layers is more than 15 GeV.
  3. Total energy from both barrel and end cap BGO is above 20 GeV.
  4. Total energy from barrel BGO is not less than 10 GeV.
- **Cluster trigger :**
  The trigger cells from the different detector layers (ECAL and HCAL) in the same θ and ϕ coordinates are defined as a **cluster**. The cluster trigger condition is that the sum of energy in one cluster is above 7 GeV. The TEC trigger information allows to use a lower threshold (3 GeV).
- **Hit counting trigger :**
  At least two trigger cells have recorded energy with more than 5 GeV.
- **Single photon trigger :**
  The energy of a BGO cluster is bigger than 80% of the total energy from the whole electromagnetic calorimeter.
- **Luminosity trigger :**
  More than 15 GeV in both luminosity monitors or one monitor energy above 25 GeV and the other not less than 5 GeV.
- **Single tag trigger :**
  The energy from one of the luminosity monitors is above 30 GeV. It is used to monitor the luminosity trigger efficiency.

The ***calorimeter trigger*** is implemented with CAMAC modules. It uses the Fast Encoding and Readout ADC (FERA) to digitize the analog sum signals from the calorimeters and the small angle luminosity monitors.

2.2.2 The TEC Trigger

The ***TEC trigger*** [8] is used to select events with charged particle tracks. It is used as a backup trigger for dimuon and ***energy trigger*** and allows to trigger on two photon events. The decision is obtained in the following steps.

1. Digitize the analog wire signals (14 × 24) of the outer TEC to produce a hit pattern in the projection normal to the beam axis (R_ϕ).
2. Search for tracks from the hit pattern
3. Make the trigger decision based on the total number of tracks and the total number of coplanar pairs of tracks.

The system is based on the FASTBUS standard.

2.2.3 The Scintillator Trigger

The ***scintillator trigger*** [9, 10] is based on the signals from the 30 scintillation counters. It is used to select high multiplicity events. The multiplicity trigger requires a coincidence of 5 out of 16 scintillator pairs. Its information is also used by the ***muon trigger***. The system is implemented in both FASTBUS and CAMAC standards.

2.2.4 The Muon Trigger

The ***muon trigger*** [11, 12, 13, 14] selects events with at least one particle which penetrates the muon chambers. It uses the trigger cells information from the muon chambers and searches for tracks pointing to the interaction region with $P_T > 1$ GeV. The total rate is about 10 Hz. After requiring at least one scintillator counter, it goes down to about 1 Hz, mainly coming from the cosmic rays. There are three subtriggers:

- **Single muon trigger :**
  This trigger defines a muon track as a coincidence of 2 out of 3 P-chamber layers in R_θ plane and as a coincidence of 3 out of 4 Z-chamber layers in R_z plane. At least one octant should have a track identified in both the R_θ and R_z planes. The trigger effective in the region $44° \leq \theta \leq 136°$.
- **Dimuon trigger :**
  The trigger defines a muon track as a coincidence of any two layers in the R_ϕ plane. At least two octants should have a track identified and the second track

should be in one of the five opposite octants in relation to the first track. The angular coverage of this trigger is $36^\circ \leq \theta \leq 144^\circ$.

- **Small angle muon trigger :**
  A muon track is defined, in this case, by a hit in MI chamber in the R_ϕ plane and a hit in II or IM chamber in the R_z plane. Furthermore, if a track is found in the forward half (+z) of the detector, another should be found in the backward half (-z). The angular region $24^\circ \leq \theta \leq 156^\circ$ is covered.

The ***muon trigger*** is composed of over 100 CAMAC modules and some FASTBUS modules located in the MUCH readout system.

2.2.5 The Level-1 Trigger Rates

Figure 2.2 displays the trigger rates versus time in Fill No.1891 which was a high luminosity fill in 1993. The fill was started at about 2:30 and ended at about 21:00.

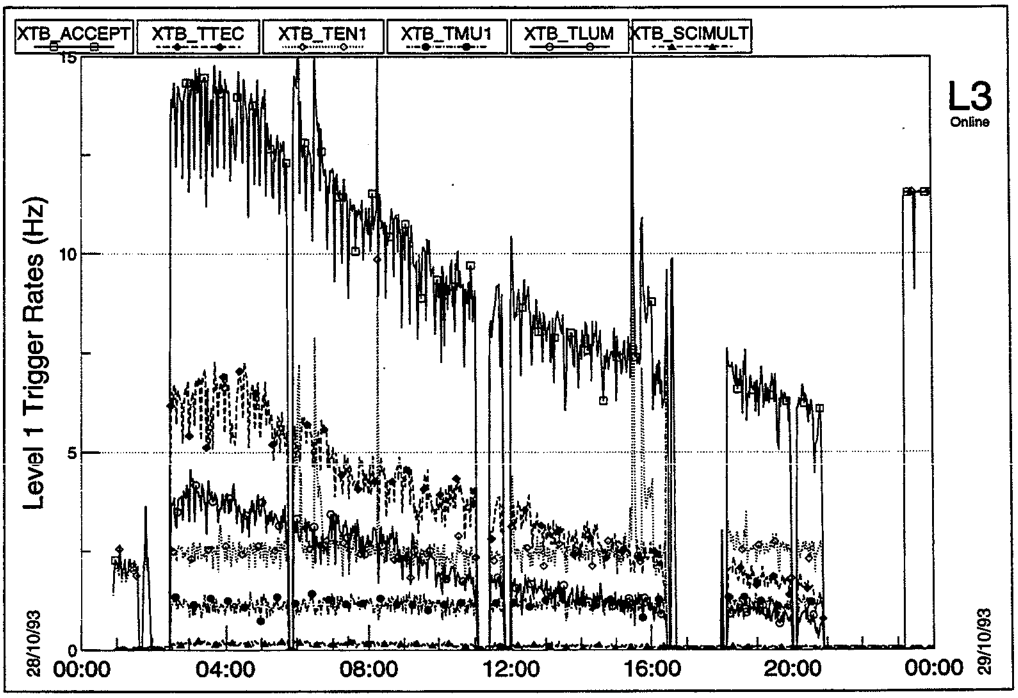


Figure 2.2: The level-1 trigger rates in fill No.1891

XTB_TTEC, XTB_TEN, XTB_TMU, XTB_LUM and XTB_SCIMULT in the firgue indicate respectively the ***TEC trigger***, the ***energy trigger***, the ***muon trigger***, the ***luminosity trigger*** and the ***scintillator multiplicity trigger***. The XTB_ACCEPT is the total rate accepted by the level-1 trigger. It is shown clearly that the ***luminosity trigger*** and ***TEC trigger*** rates are correlated with the beam luminosity which decreases with time, but the ***energy trigger***, ***muon trigger*** and ***scintillator trigger*** are mainly coming from the electronic noise and cosmic ray respectively which are constant in the whole fill.

Table 2.1 lists the typical level-1 trigger rates (in average) from different triggers [15]. The total output rate is about 10 Hz.

Table 2.1: The typical level-1 trigger rates

Group trigger	Rate
TEC trigger	4.0 Hz
Energy trigger	2.5 Hz
(Luminosity trigger)	2.5 Hz
Muon trigger	0.9 Hz
Scintillator Trigger	0.1 Hz
Total rate	10.0 Hz

2.3 The Second Level Trigger System

The second level trigger [16] has two functions; collection of the coarse trigger data coming from the first level and rejection of background events selected by the level-1 trigger. The selection algorithm is based on the trigger data and the level-1 trigger decision. To preserve the first level trigger redundancy, the second level algorithms analyze the following categories of single level-1 trigger events which represent 87% of the first level selected events [16].

- The ***energy algorithm*** is based on the precise energy estimation and rejects events with energy less than the level-2 threshold.
- The ***TEC algorithm*** rejects electronic noise, beam gas or events with tracks not coming from the interaction point.
- The ***muon algorithm*** rejects muon events with a small number of hits in vertex chambers and without scintillator counters in time.

The system [17] is implemented in FASTBUS standard and is shown in Figure 2.3. It is composed of three XOP (*bit-slice microprocessors*) processing farms [18]. A dedicated XOP/FASTBUS interface has been realized to fully integrate FASTBUS into XOP. Three XOP processors work in a ***round robin*** mode. Each one is connected to both input and output FASTBUS segments and acts as a FASTBUS master to take care of the latter three steps in a four steps sequence.

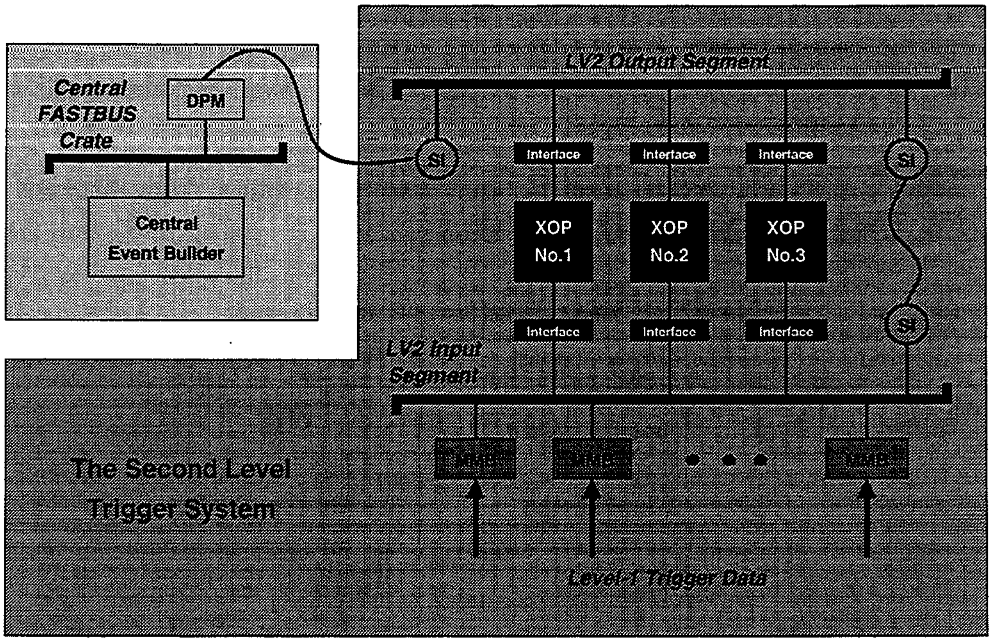


Figure 2.3: The structure of the second level trigger system

1. **Trigger data memorization**
   The trigger data are simultaneously sent to the first level processors and the second level Multiport Multievent input Buffer (MMB) memories. This prevents dead time induced by data transfer between the first and second level. All data (2.5K - 16bit words) from the front end trigger digitizers are stored in parallel to the 48 ECL input port in 12 MMB FASTBUS cards within about 10 μsec. At each beam crossing, the data are overwritten unless a good candidate is identified by ***the first level trigger***.

2. **Trigger event building**
   The MMB acts as an eight events deep FIFO (First In First Out) memory to allow the asynchronous processor operation at high speed. Its real simultaneous random read/write access at a normal value of 60 $nsec$ per word on the ECL port and 125 $nsec$ per word on the FASTBUS port, provides the decoupling between data storage and trigger event building.

   The trigger event building consists of the following steps.

   - Reading the data words of all MMB ports associated to the same beam crossing.
   - Normalizing data with the same energy unit.
   - Suppressing some non significant bits.
   - Recording data.

- Building one structured block managed by pointers, named DPM block.

It has to deal with the fact that several ports mix data from several subdetectors in a non ordered sequence. After typically 360 μsec, the trigger data block is assembled, data are ordered and normalized.

3. **Data processing**
XOP is made of 16 bits integer arithmetic ECL processors, implemented in an internal multibus parallel structure and controlled by 192 bits microcode instructions. Parallelism covers arithmetic operations, data address calculation, data access, loop control, condition checking and next instruction evaluation. Due to parallelism and pipelining, the overall performance is in fact much more higher than the cycle time of 100 $nsec$ would indicate. XOP provides a speed factor around 40 versus 68000 processors as well as good micro instruction field extension facilities. Microcode is the price to pay to reach these performances.

   Processing consists of applying specific algorithms to identify background candidates according to the first level trigger classification and making a decision. The processing time depends strongly on the first level trigger type. Latency time is always less than 5 $msec$.

4. **Transfer data to the central event builder**
As soon as a decision is made, XOP arbitrates for a mastership of the FASTBUS output segment, and writes trigger data and results into the Dual Port Memory (DPM) acting as the input buffer of the ***central event builder***.

   Since there is a time dispersion of several milliseconds introduced by the different event processing time, one disturbs the event ordering produced by the first level. As the ***central event builder*** operates on the various buffer memories according to the original first level sequence, it is mandatory to recover that sequence at the DPM order.

   A DPM is partitioned by software in 64 events. The six less significant bits of the event number define the DPM addresses in which the event is stored. A flag associated to each event controls alternatively write access by XOP processors and read access by the central event builder.

The second level trigger was designed to accept up to 500 Hz of input rate. It processes only **single** or combinations of **single** level-1 trigger with scintillation counters. At the Z^0 peak, its global reduction is about 50%. More than 99.95% of selection efficiencies are achieved.

2.4 The Readout and Event Building Systems

The data from each subdetector come to its front end electronics. The four subdetectors (TECH, ECAL, HCAL and MUCH) use different systems to make digitization and readout according to the detector properties, see [19, 20, 21, 22, 23]. Some of the characteristics of the subdetector DAQ systems are listed in Table 2.2.

Table 2.2: The major subdetector readout systems

Detector Name	Number of Channels	System Standard	Digitizer		Data Size	
			Device	Resolution	Raw	Reduced
TECH	2000	VME	Flash ADC	1024 × 6 bits	2 Mbytes	5 Kbytes
ECAL	12000	VME	ADC	12 bits	80 Kbytes	8 Kbytes
HCAL	30000	FASTBUS	ADC	12 bits	120 Kbytes	4 Kbytes
MUCH	21000	FASTBUS	TDC	10 bits	2 Kbytes	2 Kbytes

Each subdetector system can be operated in two modes. Firstly, it should be able to perform the subdetector calibrations and system debugging independently from other branches. This is called a ***local mode***. Secondly, in order to be embedded into the main L3 DAQ stream, the system is operated as a branch of the whole DAQ system and is running in parallel with other branches. This is called a ***global mode***. In the ***global mode***, the data from each subdetector branch are stored in the FASTBUS event buffers at the event building level. Since the auther has been working on the HCAL readout system, the HCAL system will be described in detail in the next chapter.

The ***L3 event building system*** [2] consists of two stages, the ***subdetector event builders*** and the ***central event builder***. Each of the four ***subdetector event builders*** merges the corresponding subdetector data from the buffers into a logical unit and sends it to the ***central event builder***. The ***central event builder*** completes the final combination of the event and checks the level-2 decision. The good events are sent to ***the third level trigger system*** to make further selection. The system will be described in detail in Chapter 4.

2.5 The Third Level Trigger System

The third level trigger [24] rejects the background events selected by the ***level-1*** and the ***level-2*** using the precise data from the subdetectors. Several algorithms are used to analyze the events.

- The **energy algorithm** reconstructs energy in the electromagnetic and hadronic calorimeters. As the calculations are based on the fine digitization, the thresholds can be more precisely defined than in level-2.

- The **muon algorithm** requires a muon track to have a scintillator hit with a time of ±10 *nsec* with respect to the beam crossing.

- The **TEC algorithm** reconstructs the TEC tracks by using the R_z trigger data to check if the tracks originate from the vertex. It checks also whether the tracks in the ϕ coordinate are correlated with some energy deposit or with hits in the scintillation counters.

The system is fully embedded in the DAQ tree and is based on several powerful processors. It consists of three parts, the input interface, the processor units and the output interface. The output data are sent to a ***data producer*** which stores events in a memory buffer on the main computer. The buffer is controlled by a Memory Buffer Manager (MBM) software package [25] (developed by CERN ECP division). From this buffer, all events are recorded and selected events are dispatched to monitoring programs.

The third level trigger system accepts up to 20 Hz as input rate without introducing any additional dead time to the system. It provides a reduction of about 50% of the input rate. The system is described in detail in Chapter 5.

2.6 The Trigger Control Logic

2.6.1 Overview

The ***trigger control logic*** [26] makes the synchronization of the different triggers and subdetector components of the DAQ system with the LEP beam. It provides the following functions.

- Provide timing signals for starting level-1 trigger processors and data acquisition modules of the subdetectors.

- Make a final level-1 trigger decision to accept or abort an event.

- Distribute event information to subdetector readout systems if the event is accepted.

- Monitor different background and trigger rates, system lifetime etc. to check the running status of the whole DAQ system.

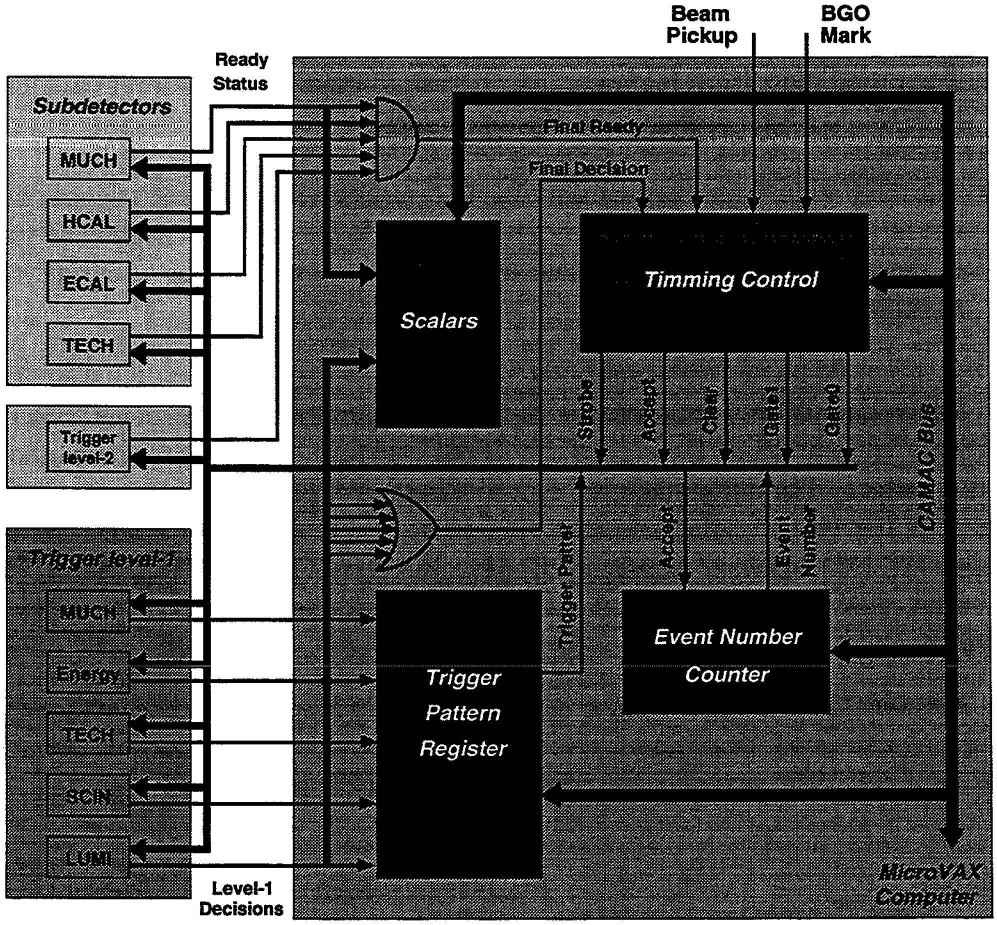


Figure 2.4: The block diagram of the trigger control logic

The ***trigger control logic*** is completely implemented by CAMAC modules. The CAMAC bus is only used for system initialization and monitoring. The timing logic is hardwired using either ECL or NIM standards. The schematics of the ***trigger control logic*** is shown in Figure 2.4.

The ***trigger control logic*** is fully computer controlled. A program running on a MicroVAX computer provides the link between the ***L3 run control*** and the ***trigger control logic***. Parameters and commands received from the ***L3 run control*** are passed through CAMAC bus to hardware. To monitor the running status of the trigger and DAQ system, different trigger rates and important single rates are recorded and constantly monitored. The program provides direct operator access through a terminal. There it displays all trigger control parameters and scaler readout. It also accepts keyboard inputs to change all parameters and to select different display items. Thirty selected trigger rates are written every 100 seconds to a disk file and presented on graphic display as a past history.

2.6.2 Timing of the Trigger and DAQ System

The positron bunch passing by a beam pickup device induces a very narrow signal which is discriminated by a tunnel diode discriminator. All other timing signals of the system are derived from this signal. The absolute time zero of the system is defined as the time when the beam bunch passes the interaction point.

The discriminated beam crossing signal which appears at the discriminator output at +75 $nsec$ is available for different trigger and DAQ components in both ungated and gated forms. The ungated one comes every 22.2/11.1 μsec (4 × 4 bunches/8 × 8 bunches) as long as there are beams, it is called GATE0. The gated one is disabled when DAQ system is busy and is called GATE1.

An accurate digital delay generator is triggered by the discriminated beam crossing signal and generates a delayed signal which precedes the next beam crossing by 1.7 μsec. Upon receiving this timing signal, ***trigger control logic*** may perform two actions depending on the current DAQ status.

1. If an event has been accepted, the ***trigger control logic*** checks the READY status of the subdetectors. It does nothing and waits for the next cycle if not all the subdetectors are ready. When all subdetectors have finished readout and buffering of the event and become ready again, it sends a CLEAR to reset all the trigger and subdetectors to start a new data acquisition cycle.

2. If a new data acquisition cycle has been started, the time of the delay is used for ***the first level trigger*** to make decisions. Then the ***trigger control logic*** checks and combines the results from different level-1 trigger processors. If the final level-1 trigger decision is positive, it sends an ACCEPT signal to all subdetectors and inhibits further data acquisition cycles. Otherwise it sends out a CLEAR to fast clear the trigger and DAQ system and starts a new cycle.

Although the final level-1 decision generally uses a simple logic OR of all level-1 trigger results, coincidence can be performed between different triggers. Individual triggers can be included into or excluded from the final level-1 decision according to the trigger mask word which is obtained from the ***L3 run control*** during initialization.

Two 16 bit words are sent to all subdetectors together with a STROBE signal which is in fact the ACCEPT signal delayed by about 200 $nsec$. One word contains the event number which is incremented by ACCEPT, the other is a trigger pattern in which every level-1 trigger has its corresponding bit set if its decision is positive. The trigger pattern provides the information for the processing in the second and third trigger levels. At the event building stage where data from different subdetectors are assembled together, the event number is verified to be able to assemble data belonging to the same event.

Subdetectors can use either beam crossing or CLEAR signal to generate their ADC gate or TDC stop. The former one has better accuracy and the latter one comes about 1.7 μsec before next beam crossing. After the ACCEPT signal is sent, the following CLEAR and gated beam crossing signals are disabled to allow subdetectors to finish the data conversion and buffering. The trigger control logic waits for at least 300 μsec before checking the ready status of the subdetectors. During this time the subdetectors should set their status to busy and should return it to ready when the data conversion and buffering are completed. Any bottleneck in the system will propagate the busy status backwards to the ***trigger control logic*** which will hold a new DAQ cycle until all detectors are ready. This ensures the data integrity in all circumstances.

The second level trigger system receives trigger data from all ***level-1 triggers*** during the level-1 trigger processing and put them into an event buffer. If an ACCEPT is received, it moves the pointer to the next buffer and uses the data for level-2 processing. Otherwise, the event buffer will be overwritten by the next event.

A signal called BGOMARK, which is scaled down from machine RF signal to beam crossing frequency, is used as another timing source for the detector calibration and test run. When there is beam, this signal is synchronized to beam and used by ***electromagnetic calorimeter*** as its timing base.

Table 2.3 lists the signals generated by the ***trigger control logic***, and Figure 2.5 shows the timing relationship between signals.

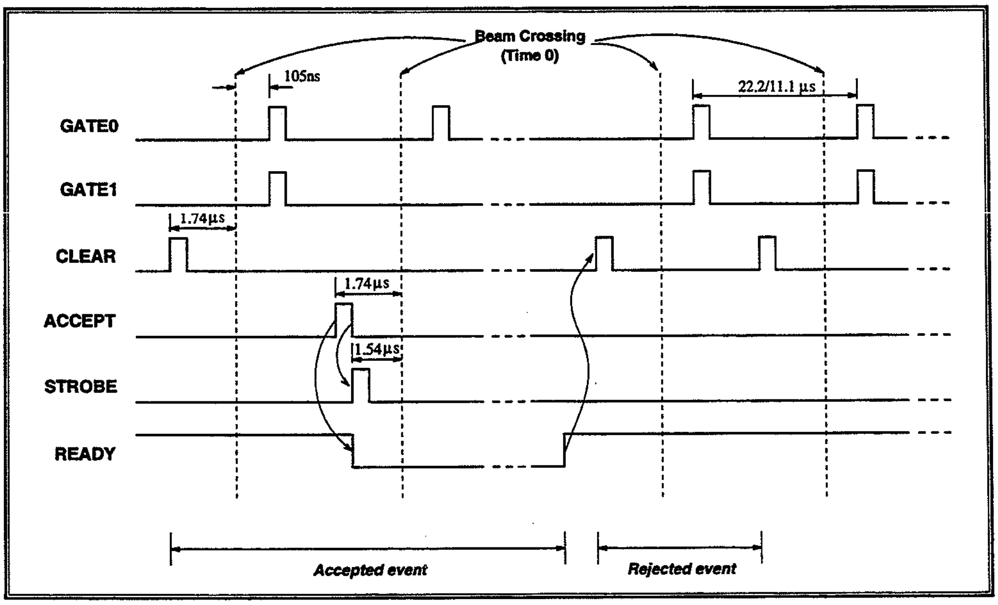


Figure 2.5: The level-1 trigger timing

Table 2.3: The signals from the trigger control logic

Signal name	Signal form	Signal width	Timing	Standard
GATE0	ungated pulse	100 $nsec$	+105 $nsec$	NIM/ECL
GATE1	gated pulse	100 $nsec$	+105 $nsec$	NIM/ECL
CLEAR	pulse	100 $nsec$	-1.74 μsec	NIM/ECL
ACCEPT	pulse	100 $nsec$	-1.74 μsec	ECL
STROBE	pulse	100 $nsec$	-1.54 μsec	ECL
data words	level		-1.74 μsec	ECL

Reference

[1] L3 Collaboration, B.Adeva, *et al.*. **The Construction of the L3 Experiment.** *Nuclear Instruments and Methods in Physics Research A 289*, pages 35–102, 1990.

[2] B.Bertucci, S.Falciano, G.Medici, and D.Linnhöfer. **The L3 FASTBUS Data Acquisition System.** In *The Computing in HEP Conference, Santa Fe, New Mexico, USA*, April 1990.

[3] L3 Collaboration. **Trigger and Data Acquisition System of L3.** European Organization for Nuclear Research (CERN), CERN/LEPC 84-5 (LEPC/PR3/L3), January 1984.

[4] M.Fukushima *et al.*. **The Trigger and Data Acquisition System of the L3 Experiment.** L3 Collaboration, L3 Note 520, September 1987.

[5] M.Fukushima *et al.*. **L3 Data Acquisition System.** L3 Collaboration, L3 Note 371, March 1985.

[6] R.Bizzarri, F.Cesaroni, S.Gentile, G.Lunadei, M.Fukushima, G.Herten, and T.Hebbeker. **The L3 Energy Trigger.** *Nuclear Instruments and Methods in Physics Research A 283*, pages 799–802, 1989.

[7] R.Bizzarri, F.Cesaroni, S.Gentile, G.Lunadei, M.Fukushima, and T.Hebbeker. **The First Level Energy Trigger of L3 Experiment: Description of the Hardware.** *Nuclear Instruments and Methods in Physics Research A 317*, pages 463–473, 1992.

[8] P.Béné, M.Bourquin, J.H.Field, G.Forconi, A.Léger, J.Perrier, N.Produit, and J-P.Richeux. **First-level Charged Particle Trigger for the L3 Detector.** *Nuclear Instruments and Methods in Physics Research A 306*, pages 150–158, 1991.

[9] U.Uwer. **Aufbau und Eichung der Szintillationszaehler des L3-Experimentes.** Diploma thesis. Rheinisch-Westfälische Technische Hochschule (RWTH), Aachen, 1990.

[10] U.Uwer. **The L3 scintillation counter system**. L3 Collaboration, L3 Note 1400, March 1991.

[11] T.S.DAI. **Heavy Flavor Production in Z^0 decays**. PhD thesis, Massachusetts Institute of Technology (MIT), Cambridge, 1991.

[12] M.Fukushima. **L3 Level-1 Muon Trigger**. L3 Collaboration, L3 Note 383, October 1985.

[13] M.Fukushima. **L3 Level-1 Muon Trigger**. L3 Collaboration, L3 Note 515, Augest 1987.

[14] T.S.Dai and M.Fukushima. **L3 Level-1 Muon Trigger**. L3 Collaboration, L3 Note 668, May 1988.

[15] G.Herten. **Trigger Parameters for the 1991 Run**. L3 Collaboration, L3 Note 930, April 1991.

[16] S.P.Beingessner, J.J.Blaising, F.Chollet-Leflour, A.Degré, G.Dromby, G.Forconi, G.Goy, J.Lecoq, R.Morand, M.Moynot, G.Perrot, and S.Rosier-Lees. **The Second Level Trigger of the L3 Experiment: The Event Selection**. *Nuclear Instruments and Methods in Physics Research A **340***, pages 322-327, 1994.

[17] Y.Bertsch, J.J.Blaising, H.Bonnefon, F.Chollet-Leflour, A.Degré, G.Dromby, J.Lecoq, R.Morand, M.Moynot, G.Perrot, and X.Riccadonna. **The Second Level Trigger of the L3 Experiment: The Implementation**. *Nuclear Instruments and Methods in Physics Research A **340***, pages 309-321, 1994.

[18] J.Lecoq and M.Moynot and G.Perrot and P.Bähler and C.Ljuslin. **The XOP Processor in FASTBUS**. European Organization for Nuclear Research (CERN), CERN 85-15, 1985.

[19] M.Fukushima. **The Trigger and Data Acquisition System of the L3 Experiment**. L3 Collaboration, L3 Note 957, June 1991.

[20] The Chapter 3 of this thesis.

[21] Philip E. Kaaret. **Forward Production of J/ψin Hadronic Interactions Calibration of a Large BGO Electromagnetic Calorimeter**. PhD thesis, Princeton University, June 1989.

[22] X.Lü. **The Time Expansion Chamber — Performance in a Testbeam**. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, 1991.

[23] G.A.Hu. **Muon Chamber FASTBUS Readout System**. L3 online documents, 1993.

[24] C.Dionisi, S.Falciano, A.Fucci, D.Linnhöfer, C.Luci, L.Ludovici, L.Luminari, B.Martin, F.Marzano G.Medici, and G.Mirabelli. **The Third Level Trigger System of the L3 Experiment at LEP**. *Nuclear Instruments and Methods in Physics Research A 336*, pages 78–90, 1993.

[25] C. Boissat *et al.*. **MODEL a software suite for Data Acquisition**. In *Computing in High Energy Physics Conference, Oxford*, April 1989.

[26] S.X.Wu. **L3 Trigger Control**. L3 Collaboration, L3 Note 1383, February 1993.

Chapter 3

The Hadron Calorimeter Data Acquisition System

In this chapter, the Hadron Calorimeter (HCAL) data acquisition system is described. The system is mainly implemented in the FASTBUS standard [1]. The design provides a three layer structure both on hardware and software level in order to allow the necessary flexibility for the different requirements. The author's main contributions in this part are in the structure design and the programming of all three layers of software.

As described in Section 1.2.5, the HCAL consists of three parts; barrel, endcaps and the muon filter. Hardware and software for the barrel and the muon filter are fully integrated together in all layers. The endcaps use a similar system which can be run either independently or together with the barrel and the muon filter part. This chapter deals only with the system for the barrel and the muon filter.

3.1 General Description

3.1.1 The System Requirements

It was required to operate HCAL electronics in both the ***local mode*** and the ***global mode***. The ***local mode*** is used for calibrations and system debugging. In the global mode, the system works under the control of the L3 DAQ system for the data taking and global tests.

Event buffers are needed to adapt the data flow between the data readout by the front end electronics and the data processing either by the ***L3 event builder*** or by the local controller.

In the ***global mode***, the HCAL electronics has to be synchronous with other subdetectors. This is done by means of the signals from the ***L3 trigger control logic*** which provides the synchronization with the LEP beam. To minimize the dead time of global data taking, the readout time is required to be within 500 μsec.

Since no resources can be used from the L3 DAQ system in the ***local mode***, all synchronization signals are generated locally instead of using the signals from the ***L3 trigger control logic***. The trigger signals are distributed to each digitizer.

Cosmic rays can be taken to calibrate the whole L3 detector when there is no beam in the LEP machine. In this condition, a wider integration time for digitization is used.

In the ***local mode***, an event processor is needed for the following categories of local operations.

- **Detector calibration.**
  The noise due to the radioactivity of the uranium plates of the HCAL generates a constant background signal. Once the relation between the signal and the energy deposited by beam particles is established, the uranium noise can be used for the calibration of the calorimeter. It is called ***uranium noise calibration*** and is done once a month.
- **Calibrations of electronics.**
  Two types of local runs are used to calibrate both the pedestal of the charge integrator in the digitizer and the gain of the preamplifier. The former one is called the ***pedestal calibration*** and needs no signals from the calorimeter. Since the pedestals are very important to scale the absolute signal value and are sensitive to the environment, the calibration is regularly done once a day. The latter one needs a synchronous test pulse at the input of amplifiers, therefore is called ***test pulse calibration***. The amplitude of the test pulse is controlled by the Digital to Analog Conversion (DAC) modules. It is done by the experts to monitor the performance of the amplifiers and the digitization chain.
- **System debugging**
  A variety of test functions are used by the experts to debug the system. It is possible to test each part of the system independently to locate and fix hardware problems.

3.1.2 System Structure

The architecture of the whole system is shown in Figure 3.1. To fulfill the system requirements, the following three layers are implemented.

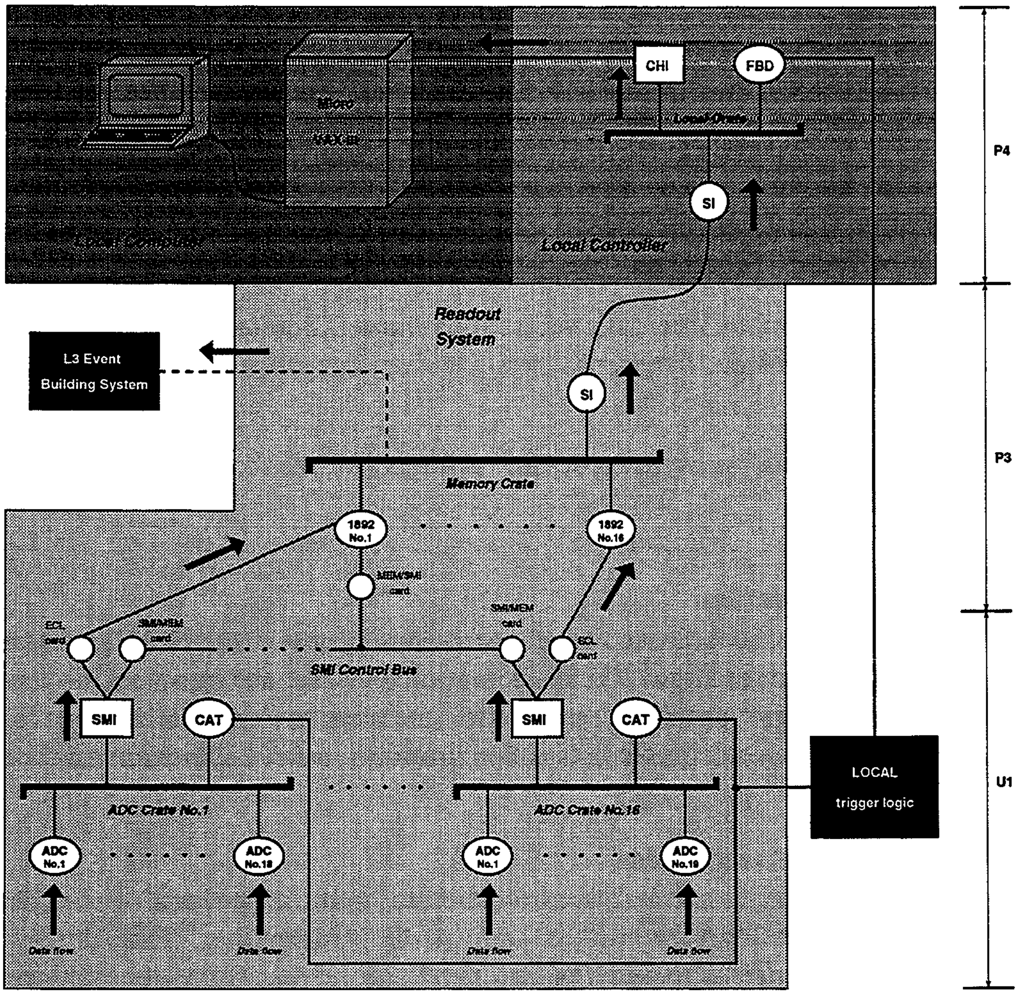


Figure 3.1: The hadron calorimeter readout system

- The first layer is the **readout system** which processes the HCAL signals in four steps — digitization, readout, data compression and buffering. It is composed of 16 FASTBUS crates to house Analog to Digital Converter (ADC) modules (called ***ADC crates***) and one FASTBUS crate to mount 16 event buffer memories (called ***memory crate***). This is the common part for both the ***global*** and ***local*** modes. The event buffers can be read either by the ***local controller*** or by the ***L3 event building system*** according to the operational mode. The timing of digitization and readout is controlled by a ***local trigger logic*** which generates the synchronization signals in the ***local mode*** and distributes the local or global trigger to each ADC module in both modes.

- The second layer is the **local controller** which realizes the central control of ***ADC crates***, ***memory crate*** and the local trigger logic to deal with initialization, trigger setting and the data processing in the ***local mode***. The system is implemented by an intelligent FASTBUS master, CHI (CERN Host Interface), Struck STR330. It is based on a Motorola MC68030 microprocessor and is located in a FASTBUS crate called ***local crate***.

- The third layer is the **local computer** which is a MicroVAX-III computer to provide the human interface and the storage of the data and software. It is also the host of the CHI module. The CHI is connected to the DRQ-11 bus of the computer via a dedicated interface.

3.1.3 System Location

The front end electronics is located in a place, called ***U1 block house***, about 40 meters away from the detector. The ***ADC crates*** and the ***local trigger logic*** of HCAL are installed there. The other parts are located in four counting rooms about 70 meters distance from ***U1***. The ***memory crate*** as a part of the L3 event building system is located in the third counting room (***P3***) and the ***local controller*** and the ***local computer*** are in the fourth counting room (***P4***). Figure 3.2 illustrates all locations.

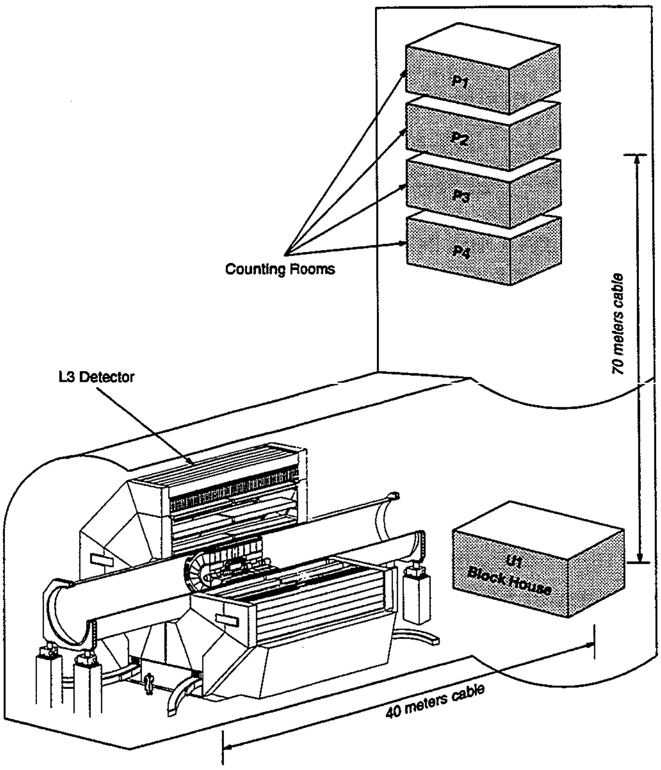


Figure 3.2: The locations of the L3 electronics

3.2 Hardware

3.2.1 Digitization

The analog signals from the HCAL are amplified using LeCroy model 2724 amplifiers, which are mounted on the outer surface of the calorimeter modules.

The signals to the front panel of each ADC are fed via a ***paddle card*** which is mounted in front of the ADC. The ***paddle cards*** house the terminating network and the coupling transformers as well as the rearrangement of the input signals for the trigger purpose. Transformer coupling allows to use different grounding points in readout system and preamplifiers in order to avoid the low frequency noise from detector.

The 1882F ADC [2] is a one slot wide FASTBUS module. It was chosen as the digitizer of the HCAL system due to the following properties.

- It has a high capacity, 96 channels are integrated into one module. Each channel has 12-bit resolution with 50 fC/count of sensitivity.
- Its wide charge integration time ranges from 50 $nsec$ up to 2 μsec. The HCAL gate width is 400 $nsec$ in the LEP beam condition and 2 μsec for cosmic rays.
- 24 ungated current sum signals are generated by 96 inputs (each signal adds 10% of four input signals). They are sent for the ***L3 energy trigger*** via a dedicated ***analog sum card*** (see Appendix A).
- A FIFO (First In First Out) buffer with 8 events deep can be read concurrently with the conversion of following events at the maximal speed of 10 megawords per second.
- A dedicated time interval, called the Measure Pause Interval (MPI), and a fast clear signal are provided and can be used to control the conversion according to the trigger decision.

The barrel part of the calorimeter is constructed from 9 rings (numbered from 1 to 9) and each ring is further built from 16 modules (numbered from 1 to 16), see Section 1.2.5. The signals from the modules with same number in all rings are processed by one FASTBUS crate which is numbered by using the corresponding module number decreased by one (*i.e.* from 0 to 15). The signals from one module are digitized by two ADC's. In total 23040 signals from HCAL barrel and additional 6144 signals from Muon Filter are digitized by 292 LeCroy 1882F ADC modules which are installed in 16 FASTBUS crates (4 crate contains 19 ADC's and other 12 have 18 ADC's). In the four crates which have 19 ADC's, the 19th module is dedicated to the muon filter only.

The other 18 modules in each of the four crates are organized in the same way as the ADC modules in the rest 12 crates. All 96 channels in the middle 6 ADC's among these 18 modules are used for the ***long modules*** of HCAL barrel [see Section 1.2.5]. The rest 12 ADC's are split to two parts, the upper 72 channels are used for the HCAL ***short modules*** [see Section 1.2.5] and the lower 24 channels are used for the muon filter.

3.2.2 The Readout System

A Segment Manager Interface (SMI), LeCroy 1821 [3], is used to readout the data from the ADC buffers because it can provide the high speed FASTBUS operations and has an on board pedestal subtraction system. It is a two slot wide module based on a fast programmable ECL sequencer. Each 64-bit instruction of sequencer contains multiple fields to perform different functions in parallel. Using a 60Mhz clock, the sequencer has an instruction execution cycle of 33 *nsec* only.

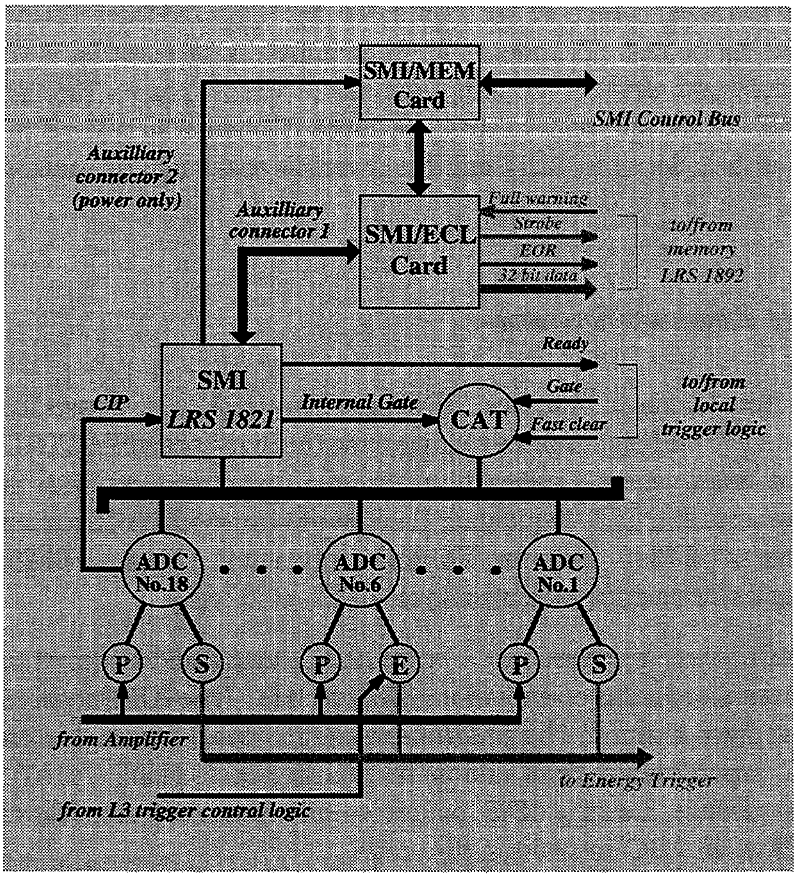
Each ***ADC crate*** becomes a subsystem and is organized as shown in Figure 3.3. Each subsystem is under the control of one SMI module. The ADC modules digitize the analog signals from detector. The ***paddle card*** and the ***analog sum card*** are connected to each ADC respectively in front and on the auxiliary backplane. A CAT (Calibration And Timing) module, LeCroy 1810, is used to distribute the gate, fast clear and the common MPI to all ADC modules in the crate.

The SMI can be either a master or a slave on the FASTBUS crate segment. The operations of SMI are fully controlled by an intelligent host via the eight 16-bit I/O registers which are accessible via the host I/O interface. The following two handshake signals are used to control the readout procedure.

- **ADC conversion in progress (CIP).**
  The signal is taken from the front panel of the last ADC in the crate and sent to the ECL input connector INP1 in the SMI front panel. It is checked by the sequencer and is used to start the data readout procedure.

- **Ready signal.**
  This signal is generated by the sequencer microprogram to indicate the readout state, either it is ready for receiving data or it is busy on the readout. The signal is sent to the ***local trigger logic*** via the ECL output connector OUT2 in the front panel of the SMI.

Besides these handshake signals, a signal is sent from the OUT3 connector of the SMI to the ECL gate input on the CAT front panel. It is used to provide a special trigger for the system test which is explained later.



• P — Paddle card **• S — Analog sum card**

• E — Modified analog sum card for receiving event number

Figure 3.3: The organization of the SMI readout system

The data path

The data which are read from the ADC modules can either go through or bypass the on board pedestal subtraction system according to the control from the host. Through the system, the pedestal values stored in a $8K \times 10bit$ pedestal memory are subtracted from the data coming from the corresponding channels. The data are compressed after the subtraction of the pedestals. Only the data which are greater than a predefined threshold are sent out from this system. The data destination after the pedestal subtraction system can be selected by the host. It is either a $4K \times 32bit$ internal data memory or an external buffer via the auxiliary connector. In this system, the LeCroy memories, model 1892 [4], was chosen as the event buffers by the L3 event building system and are shared by the HCAL ***local controller***. The target of the data from the HCAL readout system is therefore the LRS1892 memories.

The LRS1892 is a $256 \times 32bit$ dual port memory. The front panel input port allows the 16/32 bit input data in differential ECL standard. The FASTBUS port supports both data read and write operations. The events are stored into data blocks in a record structure. Each record has a header, called ***memory header***, which is located in the first word of the record and contains the current record number and the address of the ***memory header*** in the next record. The ***memory header*** is generated automatically by the EOR (End Of Record) signal at the end of recording a block of data.

The ***SMI/ECL personality card*** [5], made by LeCroy, is used as a interface in each subsystem to drive data from SMI to LRS1892 memory over 70 meters. Two operation modes are supported by the ***SMI/ECL card***. The daisy chain mode allows that the data from several SMI modules are sent to one LRS1892 memory in a predefined order. It saves the memory modules in the case of that each SMI has only a small amount of data. The HCAL system uses the direct mode in which each SMI has a dedicated memory. Thus 16 SMI modules send data to 16 memories in parallel in order to save the readout time. In total 18 memories are installed in the ***memory crate*** while 2 of them are used for the HCAL endcaps.

Besides 32 data inputs, 6 control and handshake signals are provided between ***SMI/ECL card*** and LRS1892 memory. A ***ready*** signal is sent by the memory to tell the SMI that the front panel port is enabled. It is not used in the HCAL system. The SMI sequencer can check either ***full*** or ***full warning*** signal provided by the LRS1892, to protect against the overwriting. In the case of the HCAL system, the ***full warning*** is selected. A ***strobe*** signal is needed from the ***SMI/ECL card*** to record each data word in the memory. It is generated when the sequencer pipes the data word to the ***SMI/ECL card***. The data block can be completed by the ***end of record*** signal or can be aborted by the ***abort*** signal from the ***SMI/ECL card***. Both signals are controlled by the sequencer microprogram.

The SMI host interface

There are two ports provided by the SMI for interface to the hosts [3, 5]. One is on the front panel and the other on the main auxiliary connector. In HCAL system, the ***SMI/ECL card*** which is connected in the main auxiliary connector of SMI provides not only the data path, but also converts the SMI control signals to a port which is pin-to-pin compatible with the front panel port.

The port is implemented by a 34-pin connector in the normal single-end TTL standard. Besides the 16 ***data*** lines, the following signals realize I/O operations. The 4 ***module select*** lines can choose one out of 16 SMI modules. The 3 ***register address*** lines can select one of the 8 I/O registers. The I/O direction is controlled by ***read*** and ***write*** signals. An ***acknowledge*** pulse is sent from SMI to the host to complete the current I/O ***strobe*** which is generated by the host. The ***take control*** signal, which selects the port, and the ***bypass*** signal, which makes the broadcast, exist only on the front panel port.

In the host side, the LRS1892 memory provides a port which is fully compatible with the SMI front panel I/O port. This allows another intelligent FASTBUS master to control the SMI modules. The data input or output through the port is performed by means of the read or write operation on the CSR10 of LRS1892. The port is therefore called the CSR10 port. The CSR10 bit map is shown in Table 3.1.

Table 3.1: The bit map of CSR10 of LRS1892

Bit number	Meaning
0 – 15	16-bit Data
16 – 18	Register Address
19	Register Address Inhibit (1 — Inhibit)
20	Read Operation (1 — Read)
21	Write Operation (1 — Write)
22 – 23	Unused Bits
24 – 27	SMI Address for Module Select
28	Take Control (1 — Enable Back Panel)
29	Bypass (1 – Broadcast mode)
30	Handshake Control (0 — Pulse Clock, 1 — level)
31	Generate S1 Strobe Signal (1 — Generate S1)

One memory can control up to 16 SMI modules which can be addressed from 0 to 15. Due to the 70 meters distance between the ***ADC crates*** and the ***memory crate***, a ***SMI control bus*** in the differential TTL standard is used to connect all 16 SMI modules to a single memory CSR10 port.

The specially designed boards [6] have been developed to convert the single-end TTL signals to the differential TTL signals. One board, called ***MEM/SMI card***, fits to the auxiliary backplane of the dedicated memory for connecting the ***SMI control bus*** to its CSR10 port. The others, called ***SMI/MEM cards***, are mounted on the second auxiliary connector of each SMI for connecting its I/O port on the ***SMI/ECL card*** to the ***SMI control bus***. The detail description of these two kind of boards is given in Appendix B.

The operations based on the SMI host I/O registers can be found in the SMI manual [3]. The main function of each register is listed here.

- Register 0 is a configuration register for the program operations.
- Register 1 is used to address the program memory.
- Register 2 contains the data for the program memory.
- Register 3 is to configure the data bus of SMI.

- Register 4 is the least significant 16 bits of data word.
- Register 5 is the most significant 16 bits of data word.
- Register 6 is used to address the data.
- Register 7 returns the SMI states in the read cycle. In the write cycle, it is used to generates the strobes for latching the data or address and starting the sequencer microcode.

3.2.3 Local Controller

The ***local controller*** supervises the whole readout system which contains 16 SMI modules, 16 event buffers (LRS1892) and the ***local trigger logic***. In the ***local mode***, it reads out data from event buffers and processes them. The local controller has also an interface to its host, the ***local computer***.

The CHI (Struck STR330) [7] was chosen to implement the ***local controller*** because it is a general purpose powerful FASTBUS processor with the following main parts.

- A microprocessor MC68030 driven from a 16/25 Mhz clock with at least 1 Mbyte program memory.
- Up to four I/O interface boards can be attached to it and many optional boards are provided by the manufacturer to allow CHI to be interfaced to the different systems.
- A FASTBUS port can act either as a slave or as a master. In both cases, all FASTBUS operations are supported. The FASTBUS data space is connected to the CHI data memory.
- A 1 Mbyte, at least, multi-port data memory can be accessed from processor, I/O interface and both FASTBUS master and slave ports.

In order to control and communicate with other parts of the system, the following links are established for the CHI.

- **The host interface.**
  We use the STR330/QAP interface to connect the CHI to the DRQ-11 bus on the ***local computer***. The CHI becomes an I/O device of the computer which is under the control of computer software.

- **The link between CHI and memories.**
  In order to allow the CHI in the ***local crate*** to access the event buffers in the ***memory crate***, a standard FASTBUS cable segment (*HCAL cable segment*) is implemented to connect two crates together by means of two segment interconnect (SI) modules, one for each crate. Each SI has two ports, a crate port and a cable port, which can pass the FASTBUS operations from one port to another according to the routing initialization. Through this link the CHI controls the 16 LRS1892 memories and the 16 SMI modules via the dedicated memory.

- **The trigger control path.**
  A FASTBUS to CAMAC Branch Driver (FBD), Struck STR320, is installed in the ***local crate***. It drives a standard CAMAC parallel branch highway. One branch can support up to 7 CAMAC crate controllers. In this way, the CHI controls the HCAL local trigger system.

3.2.4 The Local Trigger Logic

The ***local trigger logic*** has three main tasks; synchronization with the L3 trigger, generation the local trigger signals and distribution of the trigger signals to all ADC modules. The system is implemented through one NIM crate, two CAMAC crates and 16 CAT modules (one in each ADC crate). The diagram of the logic is given in Figure 3.4. According to the functions, the system can be divided into the following parts.

The trigger generator

The following three types of trigger are provided by this part.

- The **global trigger** uses the CLEAR0 signal directly from the ***L3 trigger control logic*** to synchronize with the L3 trigger in ***global mode***. It can be vetoed in the ***local mode***.

- The **fast trigger** is started by the ready signal from the ready logic part to run the ***readout system*** at its maximum speed.

- The **random trigger** is generated by two NIM timing modules. One simulates the CLEAR0 signal which comes every 22 μsec from the ***L3 trigger control logic***. The other uses the typical output rate of the L3 level-1 trigger. It is used specially for the pedestal calibration because the pedestal values are sensitive to the trigger timing.

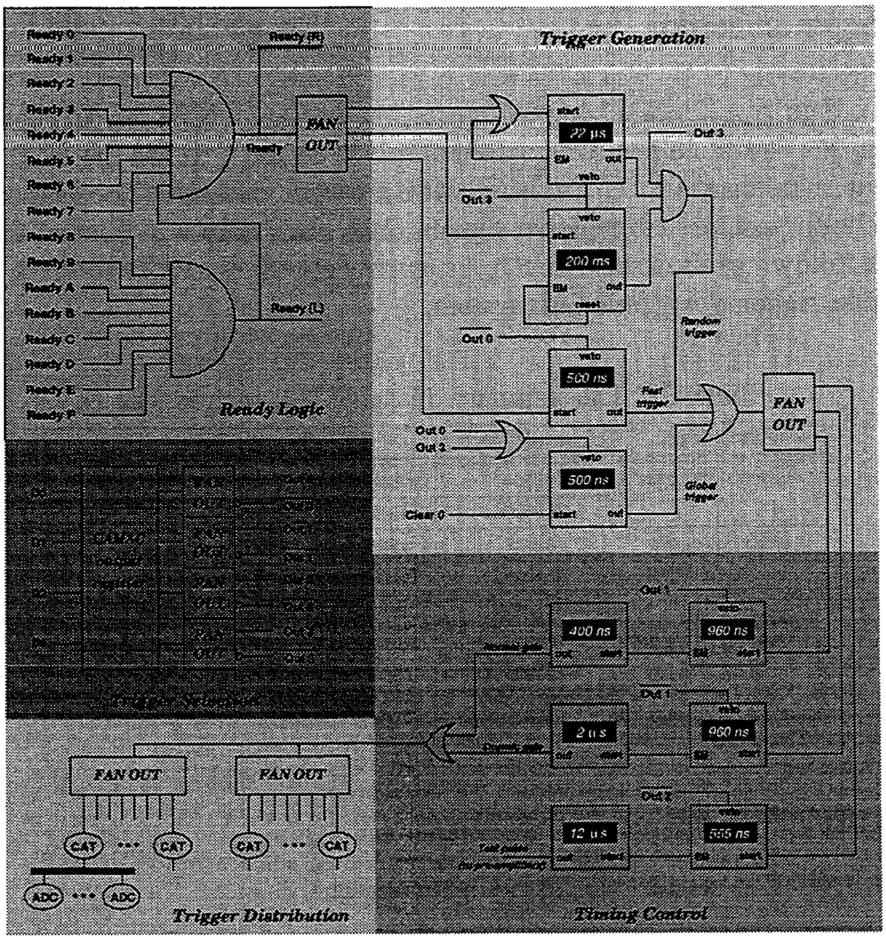


Figure 3.4: The HCAL local trigger logic

The timing control

This part controls the delay and the gate width according to the trigger from the local trigger generator. The following three pulses are provided and each pulse is controlled by two NIM timing units, one for the delay and another for the pulse width.

- The **gate signal** is 400 *nsec* wide with a 960 *nsec* delay in case of the 8 bunches mode of LEP.
- The **cosmic gate** is 2 *μsec* wide with the same delay as the normal gate. It is used in the cosmic ray calibration.
- The **test pulse** is 12 *μsec* wide with 555 *nsec* delay. It is used to trigger the DAC module to generate the pulse signals for test pulse calibration.

The trigger selection

It is done by a CAMAC output register module and a NIM fanout module. The output register has 8 outputs which are controlled by the CHI via the FBD. Each of the 8 least significant bits is correlated with a output channel. The fanout module provides both positive and negative signals from each output to control the trigger type. The following bits of the output register are defined currently and Table 3.2 sums the possible trigger types for HCAL system.

Table 3.2: The summary of the HCAL trigger type

Categories	Control Word	Trigger Type
Global Trigger	Value 0	Global trigger with normal gate
	Value 2	Global trigger with Cosmic gate
Fast Trigger	Value 1	Fast trigger with normal gate
	Value 3	Fast trigger with Cosmic gate
	Value 5	Fast trigger with test pulse
Random Trigger	Value 8	Random trigger with normal gate
	Value 10	Random trigger with cosmic gate
	Value 12	Random trigger with test pulse

- Bit 0 (OUT 0) controls the Fast Trigger.
- Bit 1 (OUT 1) selects the Cosmic gate output.
- Bit 2 (OUT 2) enables the Test Pulse output.
- Bit 3 (OUT 3) chooses the Random Trigger.

The trigger distribution

The two group of ***ADC crates*** are located in two place, left and right sides in U1 block house. Each group contains 8 ***ADC crates*** and a corresponding CAMAC crate used for the local trigger purpose.

A fanout module with 8 output channels in each CAMAC crate distributes one of the two inputs, either the ***normal gate*** or the ***cosmic gate*** depending on the trigger selection. The outputs are sent to the CAT modules in all ADC crates belonging to this group. Furthermore, the CAT in each crate distributes the gate to all ADC module via the FASTBUS backplane.

In the same way, another fanout with 8 output channels distributes the CLEAR0 signal to the ADC modules via the CAT modules. The CLEAR0 signal comes from the L3 trigger control logic directly and is used as the fast clear signal of the ADC.

In addition, each CAT provides a common MPI signal via the FASTBUS for all ADC modules in the same crate. The duration of the MPI is set to 32 μsec by the SMI at the starting time.

The ready logic

The 16 ready signals from each crate are integrated through two CAMAC logical AND/OR modules which have 16 ECL inputs, one NIM output and 4 ECL outputs. One of them combines the 8 signals from the left side group. This primary result is sent to both the ***L3 trigger control logic*** and another module in the right side CAMAC crate. The module in the right side combines the 8 ready signals in right side group with the additional input of the left side result to make a final result. This result is not only sent to the ***L3 trigger control logic*** but also used in the ***local trigger logic*** to control the local trigger generation.

The design of this module allows that the state of each input signal can be read or masked via the CAMAC bus. This provides both the flexibility to use the part of the HCAL system and an easy way to check the ready states from each subsystem.

The scalers

A scalar module, LeCroy 4434E, in the right side CAMAC crate is used to monitor the running status of the HCAL system. It has 32 input channels, but only the following channels are used currently.

- Channel 0 counts the local gate either from the fast trigger or from the random trigger.

- Channel 1 counts the gate signal from the gate fanout in the right side CAMAC crate. This gives the information of the total number of gates which have been sent to the ADC's.

- Channel 3 counts the total number of the fast clear signals from the fast clear fanout in the same CAMAC crate.

- Channel 4 counts the total number of ready signals which is the final signal from the output of the ready logic.

- Channel 5 counts the CIP signal from one ADC in one of the subsystems which is used to indicate the number of conversions done by the ADC module.

The internal trigger

Besides the above trigger logic, a special trigger is provided by the SMI so that each subsystem can be operated in a stand alone mode. Therefore it is called the ***internal trigger***. The gate is sent from the ECL output OUT2 on each SMI front panel to the CAT in the same crate directly. To generate this gate, the SMI has to use a dedicated software. The ready signal is always set to the busy status to disable the other signals from the ***local trigger logic***.

The test pulse

On each side CAMAC crate, there is one DAC module with 16 output channels. The analog output level of each DAC channel is programmable via the CAMAC bus. The output is triggered by the test pulse signal from the timing control.

The 24 amplifiers are mounted in one board for the signals from one HCAL modules. Each board has a input port for the test pulse. The 32 test pulse signals are sent to 144 amplifier boards respectively.

3.3 Software

3.3.1 Software Structure

Corresponding to the hardware layers, the software consists of three layers respectively running on the SMI's, the CHI and the VAX computer. Figure 3.5 shows the organization of the software.

The operators have two possibilities to control the system either to use a standalone program from any terminal or use the HCAL run control program. In the HCAL run control only the standard procedures are provided. The standalone program provides all possible functions used by experts for system debugging and tests.

The RPC (Remote Procedure Call) package [8] provides the software link between VAX and CHI. It makes the program on the **called** machine work as a subroutine on the **calling** machine. The **calling** routine and the **called** routine are respectively named as **RPC client** and **RPC server**. The main communication functions between the CHI and the VAX in this system are implemented by a RPC routine, ***command server*** which sends the command from the VAX to the CHI and returns the result by means of CHI data memory after the command is executed.

On the CHI, the software is organized in two categories; the applications corre-

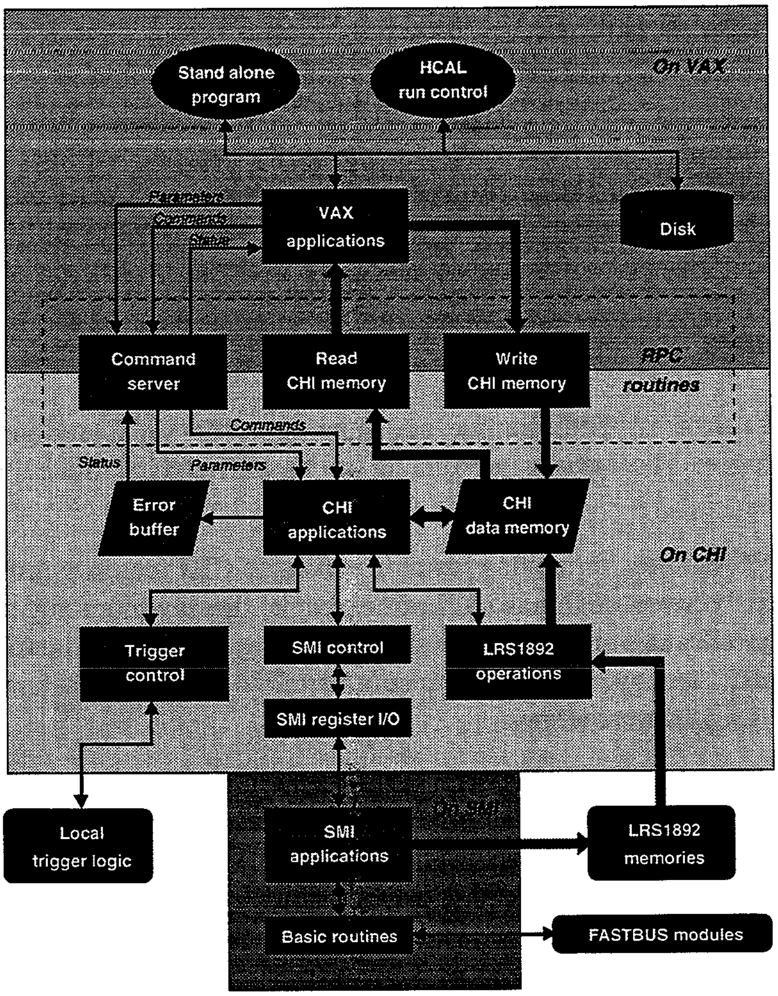


Figure 3.5: The HCAL software structure

sponding to the VAX commands and the basic routines which are interface to the SMI modules, the event buffer memories and the ***local trigger logic***. The applications break down the VAX commands to the necessary steps by using the basic routines.

As described in Section 3.2.2, the software link between CHI and each SMI is through the ***SMI control bus*** by means of operations on the host I/O registers of the SMI. Therefore the most important basic routines on the CHI are the input/output routines of the SMI registers through which all functions are realized.

Similar to the CHI software, the SMI software consists of the applications and the basic routines. The basic routines implement the essential FASTBUS operations which can be called by the applications. The applications are provided according to the requirements of the operations in each subsystem.

3.3.2 The VAX Software

The software on the VAX is written in FORTRAN language. All applications are based on the three RPC routines; command server, reading and writing CHI memories. The applications can be classified in several categories; cold start procedures, data taking procedures, calibration and test procedures as well as some utility routines.

RPC routines

The ***clients*** of all RPC routines are running on the VAX and are linked with the VAX software. The ***servers*** are running on the CHI and are linked with the CHI software.

The RPC routines for reading from or writing to CHI memory provide the way of data transfer between CHI and VAX, for example, the downloading and uploading the ADC pedestal values and reading back the running results from the CHI and so on. Two arguments; starting address and the number of bytes to be transferred have to be given before using these routines.

The ***command server*** is the main RPC routine through which the VAX software controls the CHI actions. Three arguments; status, command and the parameter array are required. Each command is defined by a four byte word while each byte contains an ASCII character. The first character is used to indicate either the barrel and the muon filter part by the character **B** or the endcaps part by the character **E**. The other three characters define the commands. Table 3.3 lists the currently implemented commands.

Table 3.3: The commands defined by the command server

Command	Procedure	Command	Procedure
INI	Initialization	TPD	Pedestal Calibration
STA	Start Data Taking	TUN	Uranium Noise Calibration
STO	Stop Data Taking	TEV	Test event numbers
DLP	Download Pedestal Values	TME	Test Memory
ULP	Upload Pedestal Values	VAL	Test Validation

At most sixteen 32-bit parameters can be sent to the CHI via the parameter array. The most important parameters are listed in Table 3.4.

An error buffer with 256 32-bit words is used to stack the error codes for the CHI operation errors. Before the command is sent, the buffer is cleared. The error messages are stacked by the CHI whenever an error occurs during the command execution. The first word of the buffer is used as a counter to count number of error messages in the buffer. It is taken by the command server as the status argument. After getting this returned message, VAX checks it. The command is successfully executed when the

Table 3.4: The running parameters

Parameter Name	Meaning
Number of event	The number of event to be processed
Trigger type	The trigger selection
Code type	The SMI microcode selection
Data mode	The SMI Data bus operation mode
Ready mask	Defines which ready signals are used
HB offset	The pedestal offset for the HCAL barrel
HM offset	The pedestal offset for the muon filter
Threshold	The threshold for the data compression

status is zero. Otherwise VAX reads and decodes the information in the CHI error buffer and then reports errors to the operators.

Cold start

The cold start is necessary especially because the RPC software links have to be established when the program is started. Two steps are provided — reset procedure and the cold initialization. The reset procedure includes the following items.

1. Perform the CHI reset and the bus reset on both the ***local crate*** and the ***memory crate***.

2. Initialize the hardware link (SI) between the ***local crate*** and the ***memory crate***.

3. Download the CHI software and start it.

If the CHI software has already been started and the link between both crates has been initialized, it is possible to start from the cold initialization. The following items are included in this step.

1. Establish the RPC link between both the CHI and the VAX.

2. Perform a warm initialization procedure which is explained in the following pages.

3. Read the pedestal value from file, download them to the CHI data memory and then send them to the pedestal memory of each SMI by CHI software.

Data taking

Three procedures — warm initialization, start and stop data taking control the data readout from the HCAL system. These procedures can be used not only for the global operations but also in the ***local mode*** for the system tests with different trigger selections. Especially the warm initialization has to be performed before any calibrations or system tests. The warm initialization is done in the following two steps.

1. Send command to the CHI to initialize the event buffer memories, the SMI modules, the ***ADC crate*** and the ***local trigger logic***. The module positions in each ***ADC crate*** and the relationships between the event buffers and the SMI modules are returned to VAX.
2. Compare the information from the CHI with hardware configuration stored in the file and report if any differences are found.

The start and stop procedures are mainly implemented by the CHI. The VAX sends command with the selected parameters to the CHI. The CHI sets the trigger type, data compression modes and threshold to the local trigger system and the SMI's respectively then starts the selected application in all SMI modules. Afterwards the execution status is returned to the VAX to complete the procedure.

Calibrations and system tests

The main tasks of the ***local controller*** and the ***local computer*** are to perform uranium noise, pedestal and test pulse calibrations. In addition, some test functions are provided for the experts to allow the system debugging.

Because of the low data transfer rate between the VAX and the CHI, the main online event processing work is done by the CHI during the local calibrations and system tests. Only the floating point calculations are performed by the VAX based on the CHI results in order to save the processing time.

The uranium noise in each channel is described by a spectrum illustrated in Figure 3.6. In order to obtain the good statistics results, at least 1000000 events are required. The fast trigger is used to minimize the total calibration time. The results can be selected in two resolutions with different processing time. In the fast one, each channel is described by sixteen 16-bit energy bins which contain the counts of hits. The first fifteen bins correspond to amplitude from 0 in the linear steps of 16. The last bin corresponds to the amplitude above 240. The spectra of all ADC channels can be processed and stored in the CHI data memory at the same time. The high resolution spectrum for each channel refers to a 1024 16-bit energy bins. Due to the limit of CHI

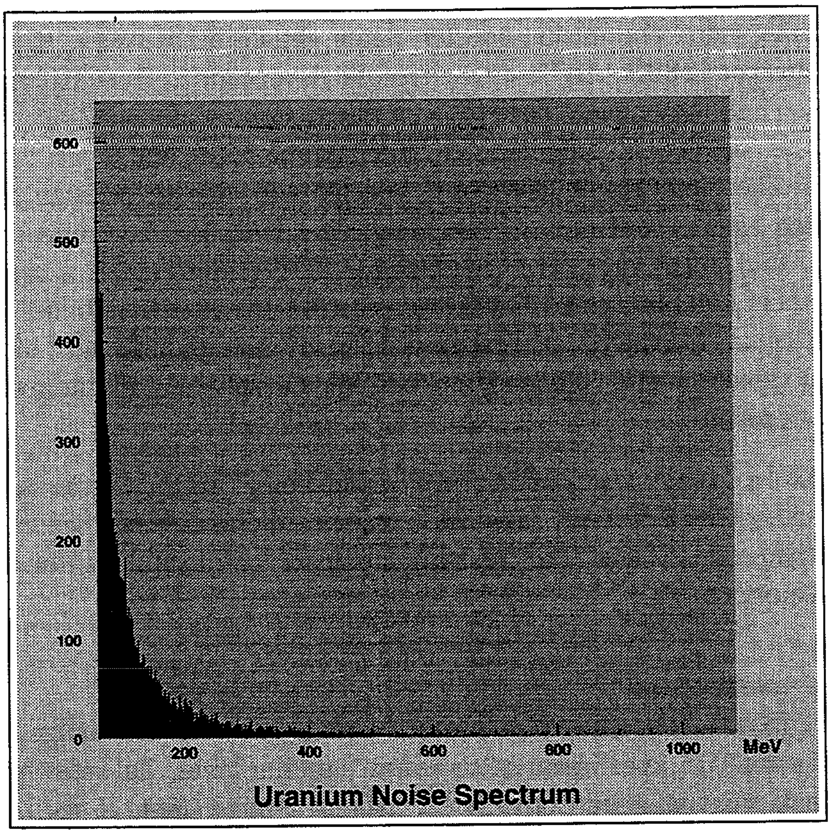


Figure 3.6: The uranium noise spectrum

memory, only the channels from two predefined ADC modules can be processed. All processing task is done by CHI. The VAX reads the results after the given number of event are processed by CHI and then stores them to disk files.

The pedestal and test pulse calibrations use almost the same procedure except an additional trigger bit and the DAC level setting required only by the test pulse calibration. Before the global operation, the pedestal calibration needs to use the random trigger to simulate the global trigger condition. In other cases, the fast trigger can be used. In the calibrations, the mean value and the root mean square deviation for each channel are calculated by the following equations.

$$m = \frac{\sum_{i=1}^{n} x_i}{n}$$

and

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - m)^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n} - m^2}$$

Where m is the mean value, σ is the root mean square deviation and x_i is the data in the ith event. The mean pedestal values should be between 100 and 500 as the specification of 1882F ADC and the typical σ is around 2. The CHI calculates the sum ($\sum_{i=1}^{n} x_i$) and the square sum ($\sum_{i=1}^{n} x_i^2$) and stores them as 32-bit words. The VAX reads the results and completes the calculations in the double precision floating data format. The final results are stored in a file with integer pedestal mean values and root mean square deviations in the single precision floating point format. The sequence of the data storage corresponds to the ***ADC crate number***, slot number and channel number. A maximum of 2000 events can be processed due to the 32 bits limit of the square sum. Normally 1000 events are already enough in the most of cases. The new pedestal values are downloaded into the SMI pedestal memory automatically after pedestal calibrations.

In addition, many test functions are implemented. In the test mode, the events are processed by the CHI only. The VAX prepares the parameters, sends commands and reports the errors if any. The following are three very useful test functions which are used quite often.

- **Event number test**
  In the global mode, the event number word is used to synchronize the different parts in the L3 experiment at the event building level. In order to check the received event number from the ***L3 trigger control logic***, a test procedure is provided. The test allows that the system is running in the ***local mode*** and all trigger signals and information are coming from the ***L3 trigger control logic***. The CHI checks the validation of the event number in each event.

- **Memory test**
  In this test, each SMI sends test data blocks to the corresponding event buffer continuously. The CHI check the correctness of data words event by event to test data transfer between SMI and its event buffer especially the front panel input part of the event buffer.

- **Validation test**
  Any trigger type and data taking microcode for the SMI can be used in this test mode. The CHI checks the validations of ADC slot numbers and channel numbers continuously whatever the events are sent with or without data compression and data headers. The ***local trigger logic*** and the data path can be tested in this way.

Utility routines

Some common routines are used by the above procedures. The error decoding is a very important routine. The number of errors in the CHI error buffer is returned as the execution status by the CHI after the command has been completed. If this status is not a zero, the decoding routine reads the error buffer from the CHI and decodes the contents. Besides the CHI errors, the errors from the VAX software are decoded in the same way. The 32-bit error word consists of the error code in the most significant 16 bits, the SMI's address or the slot number of event buffer in the bits 8 – 15 and the number of words followed in the least significant 8 bits which allows the more information to be given. The alarm level can be predefined in the decoding routine, either warning or fatal error.

The pedestal download, upload and pedestal checking routines are also provided. From the HCAL run control, the pedestal checking is automatically done after each pedestal calibration. It compares the differences between the new pedestal values and the old one which are read from the previous file and checks the root mean square deviations. If the differences or the deviations in many channels are above the predefined limits, an alarm is given to warn the operator. In the standalone program, the operator can perform these three functions separately by means of typing the corresponding command.

Other routines such as the file operations, display and online helps *etc.* are provided for the operators. The log files record the operations and the error messages from the HCAL run control which can be checked by the experts whenever it is necessary.

3.3.3 The CHI Software

The MoniCa monitor [9], developed at CERN for the MC68000 microprocessor family, resides on the CHI which provides the input/output management, debugging facilities, runtime libraries and the standard RPC link with the host computers via the interface. The CHI software of HCAL is written in the MC68030 assembly language which is compiled by using the cross assembly tools supported by CERN on VAX VMS computers [10, 11] and linked with RPC and standard FASTBUS libraries [12]. The final code is produced in the Motorola absolute object format (S-records) which can be loaded via the standard RPC link and started either from the Monica or by the RPC link.

The four sets of the basic routines perform the essential operations respectively on the event buffers, SMI modules and trigger CAMAC crates as well as control the CHI internal hardwares. The applications break down the VAX commands to the steps based on these routines. Since MC68030 provides 8 data registers (D0 – D7) and 7 address registers (A0 – A6) for the general purpose uses, we use register variables as much as possible in order to minimize processing time.

Memory operations

Three kind of memory operations are provided in the CHI with the following steps.

- **Memory initialization**

  1. Scan memories in the ***memory crate*** and perform the following steps on them one by one.
  2. Check the memory device identification number.
  3. Select the memory operation mode which includes the linear or circular mode, the block operation mode and the front panel input enable (CSR0).
  4. Set the end of memory (CSR13) and the offset (CSR14) for the full warning.
  5. Reset the front panel pointer (CSR11) and the end of block register (CSR15).
  6. Clear the header pointer (CSR16) and the record number (CSR12).
  7. Set the memory addressing pointer (NTA) to the beginning of the memory (address 0).

- **Find the relationship between memory and SMI modules**
  The procedure is for checking the hardware configuration.

  1. Start the SMI code to send the corresponding identifier of the ***ADC crate*** and a test word to event buffers.
  2. Check the correctness of the test word which has been sent by the SMI.
  3. Read and check the identifier from each memory then add the slot number of the memory to a table which is organized in the order of the ADC crate numbers.

- **Read a block of event**
  The procedure is used for the local event processing.

  1. Find a memory which has a valid event.
  2. Make a SKIP operation on this memory which put the NTA to the beginning of the event block and the CSR15 to the end of the block.
  3. Start the DMA for a block transfer to read the event.
  4. Poll on the DMA complete flag to wait for the end of block transfer.
  5. Set a flag for this memory and check if all memories are read in the current event.
  6. Go back the first step for the next memory if not all memories are completed. The memory can be disabled automatically if no valid event was found within a predefined timeout.

SMI control

All SMI operations are based on the input/output operations of the SMI host I/O registers. Therefore routines; INP21 (for reading from the registers) and OUT21 (for writing to the registers) play the most important roles in the CHI software. The INP21 and OUT21 routines are described here.

- **INP21 routine.**

  1. Combine the SMI address in bits 24 – 27 of D0 with the register number in bits 16 – 18 of D1.
  2. Clear the least significant 16 bits for the input data.
  3. Set the strobe bit and the read bit.
  4. Write the word to the CSR10 of the dedicated memory, it passes the control word to the SMI via the ***SMI control bus***.
  5. Wait about two microseconds for the data ready.
  6. Read back the word from CSR10 of the memory in which the least significant 16 bits contain the data to be read.
  7. Send zero to the CSR10 to clear the SMI control bus.

- **OUT21 routine.**

  1. Combine together the SMI address in bits 24 – 27 of D0, the register number in bits 16 – 18 of D1 and the data in the least significant 16 bits of D1.
  2. Set the strobe bit and the write bit.
  3. Write the word to the CSR10 of the dedicated memory which passes the word to the selected SMI via the ***SMI control bus***.
  4. Wait about one microsecond for completion of the operation.
  5. Send zero to the CSR10 to clear the SMI control bus.

Since the SMI program memory with 256 instructions size [3] is not enough to keep the whole software, an overlay technique is used. In each overlay, the software is controlled within the 256 instructions. Each SMI routine has not only the starting address but also the overlay number. The SMI software is stored in the CHI memory. By default, the first overlay is downloaded to SMI at the beginning. Whenever a SMI microcode is to be executed, the program checks both the current overlay in the SMI and the overlay number of the selected microcode to decide if it is necessary to download the selected overlay to the SMI modules. This is done in the following steps by the start SMI code routine (EXEC21).

1. Check the overlay numbers and download the required overlay if necessary.
2. Configure the data bus mode.
3. Enable the sequencer to check the host flag bit in the register 0 during the running.
4. Set a 16-bit running parameter to the SMI register 2.
5. Write the starting address of the selected code to the SMI register 1.
6. Start the SMI via the register 7.

The pedestal values can be downloaded or uploaded word by word via the SMI registers 3, 4, 5, 6 and 7. No block transfer is provided by the SMI interface. After the pedestal download, an upload is performed automatically. The uploaded values are compared with the downloaded values in order to check the validation. In the same way, the threshold can be download to all SMI modules by a dedicated routine.

Trigger control

The trigger control routines are based on the FBD operations [13]. The secondary address cycle of the FBD forwards the least significant 17 bits to the CAMAC bus [14] to address module and to perform the function code with the data in the followed data cycle of the FBD. The contents of these 17 bits are listed in Table 3.5.

Table 3.5: The CAMAC address word from FBD

Bits	Symbol	Meaning
0 – 3	A	Sub-address
4 – 8	N	Module address
9 – 11	C	Crate address
12 – 16	F	Function code

The block transfer is permitted by the FBD. The CAMAC Q and X responses can be obtained and controlled via the CSR0 of the FBD by means of single FASTBUS operation. The following routines are provided to control the local trigger system.

- **FBD and CAMAC initialization**
  This routine resets the FBD and the CAMAC crates and performs a bus clear via the crate controller in each CAMAC crate.

- **Trigger selection**
  It writes (F=16) the trigger type to address 0 (A=0) in the CAMAC output register module.

- **Set ready mask**
  A 16-bit mask is given by the VAX software, in which the bit number is used as the ADC crate number. This routine divides it into two words according to the crates in two groups. The two mask words are sent to the ready control logic modules in the two CAMAC crates respectively. If the bit is one, it means the corresponding crate has been disabled.

- **Read ready status**
  It reads the ready status from these two ready control modules and then combines them into a single 16-bit word in which each bit indicates the status from a corresponding subsystem. The non-zero bits give the ready subsystems and the others are busy.

- **Clear scalars**
  This routine sets the clear bit (CL) in the scalar module address A=0 via the function code F=16.

- **Scalars readout**
  This routine first controls the data load from the counters to the buffer via F=16 and A=0 and then read the scalar counters via F=2 and A=0.

- **Set test pulse DAC level**
  It sets the test pulse level via the CAMAC DAC modules. The digital data level is written to the modules via the function code F=16. The 16 channels are addressed by the sub-address (A) from 0 to 15.

CHI internal operations

The CHI internal operations include the following items.

- **Interrupt service**
  It is used to handle the interrupts and to set necessary flags according to the interrupt source which can be used in the main procedure.

- **Bus error handler**
  This routine handles the fatal error of the CHI and gives the error messages, then the routine terminates the current procedure and returns the error codes to the VAX.

- **Error record**
  It is called whenever an error occurs. The error word is given via the register D1 and is stored into the current point of the error buffer. The pointer is moved to the next word and the number of error words is increased by 1. If the error buffer is full, the procedure is terminated by this routine.

- **Disconnect**
  It clears the FASTBUS and frees the mastership. It is normally used at the end of an application.

Applications

According to the VAX commands, the following applications are implemented in the CHI with the given steps.

- **Warm initialization**
  1. Initialize all event buffers to be ready for receiving data from all SMI modules.
  2. Read the register 7 in each SMI and check the power bits to make sure each ***ADC crate*** is powered on.
  3. Perform a simple read/write test in the register 2 of each SMI to check the connections over the ***SMI control bus***.
  4. Download the default overlay to each SMI.
  5. Perform a bus clear in all ***ADC crates*** via the SMI microcode.
  6. Through each SMI send the crate identifier and a test word to the corresponding event buffer.
  7. Check the test word to see if there is any problem in the data path.
  8. Check identifier and set the memory slot number into the relationship table according to the identifier.
  9. Scan each ***ADC crate*** via SMI code to find the hardware locations.
  10. Initialize the FBD and the CAMAC crates and clear the scalar module.
  11. Set ready mask to the ready logic.
  12. Initialize the event buffer again for the followed data taking or calibration procedure.
- **Start data taking**
  1. Set the trigger type.
  2. Load the threshold to all SMI modules.
  3. Get the code address and overlay number.
  4. Start SMI code execution.
- **Stop data taking**

1. Set the host flag bit in each SMI register 0 to let the sequencer send out all events in the ADC buffers. The ADC readout is stopped by the sequencer.
2. Read the trigger counts from the scalar module.

- **Uranium noise calibration**

  1. Set the trigger type.
  2. Load the threshold to all SMI modules.
  3. Start the SMI code.
  4. Read ADC data from the event buffers and fill the energy bin map.
  5. Terminate the procedure if the given number of events have been processed.

- **Pedestal and test pulse calibration**

  1. Set the trigger type.
  2. Start the SMI code.
  3. Read ADC data from the event buffers and calculate the sum and the square sum for each channel.
  4. Terminate the procedure if the given number of events has been processed.

- **System test**

  1. Set the trigger type.
  2. Start the SMI code.
  3. Read ADC data from the event buffers and check the data validations according to test mode.
  4. Terminate the procedure if the given number of events has been processed.

3.3.4 The SMI Software

The SMI sequencer uses the special 64-bit instructions. Since the compiler from the manufacturer is not general enough, a dedicated compiler [15] is developed in FORTRAN language which can be used on the most of general computers.

There are only the following special registers [3] which are provided by the SMI.

- 32-bit data register is used for the FASTBUS operations.

- 32-bit T-POLL register is to store the result from the T-pin scan.

- 8-bit TCNT register is used with the T-POLL register together. The sequencer can control the shift and the increment simultaneously in the two register.
- 8-bit N register can be used for the data comparison.
- 16-bit PDATA register is to communicate with the host.

Due to the limited SMI program memory size and the sequencer operation codes, the functions are implemented by the program blocks with the fixed parameters for the dedicated purposes. After compilation the programs are converted into format compatible to CHI assembly language which can be kept in the CHI memory together with the CHI software. Each program has an address and an overlay number. Each overlay is organized in two parts. The first part is a static part common to all overlays and consists of the basic FASTBUS operation subroutines and macros. The second part contains different application program blocks.

Microcode compiler

The 64-bit instruction [3] word contains the 7 fields — 4-bit operation code (OP), 8-bit condition code(CC), 4-bit bus definition (BD), 8-bit constant instruction data (HS), 8-bit strobes (ST), 4-bit data control (DC) and 12-bits FASTBUS protocols (FB). In the source code, each field of the instruction word is separated by a space at least. The fields are indicated by its mnemonics with the brackets except the operation code and condition code. Here is an example of the source code of a SMI instruction for this compiler.

```
[CJMP N.2048NS BD(NREG) HS($C8) ST(TCNT,32BREG) DC(BYTE0) FB(AS=1)]
```

Where the ***CJMP*** is an operation code which means the conditional jump operation. The condition is true if the internal ***2048 nsec*** timer is NOT (*N.*) exhausted. The source of the I/O bus is the ***N register*** and the source of HSDATA bus is the instruction word (by default). The data from the instruction word is a hexadecimal byte ***C8***. The data from the ***N register*** and the instruction word are respectively stored into the ***32 bit register*** and the ***TCNT register***. The data goes to the ***BYTE 0*** of the 32 bit register. The ***AS*** signal in the FASTBUS is set to high (***AS=1***).

All mnemonics are defined by the SMI manual. All fields have their default values which are listed in Table 3.6. The fields can be omitted if only default values are desired.

The macros are allowed by this compiler. A file defines all input file names for the programs and the macros. The compiler searches them to include the macros in the

Table 3.6: The default value in the SMI microcode compiler

Instruction Section	Default Definitions
Operation Code	NOP
Condition Code	None
Bus Definition	Instruction data and the FASTBUS
Instruction Data	0
Strobes	None
Data Control	Word
FASTBUS Protocol	None

correct positions. An input file is required to define the final sequence of the programs in the basic part and the overlays. The output is written to a file in a hexadecimal ASCII format. Each instruction is described by 12 hexadecimal numbers. The instructions in each overlay are grouped in one block with the overlay number as the header.

Basic subroutines

The following subroutines [16] can be called by any programs. They are included in the static part of the SMI code.

- **FB_IL**
  An idle loop executed when the sequencer is not running a user code. At the end of each user code, the control must be transferred back to this loop.

- **FB_PA**
  It performs the FASTBUS primary address to a slot given by the TCNT register. The MS code must be set and the timer should be reset in the calling sequence.

- **FB_WR**
  It performs a FASTBUS write cycle for either the secondary address or the data write depend on the MS code. The data is taken from the 32 bit register. The MS code must be set and the timer should be reset in the calling sequence.

- **FB_RD**
  It performs a FASTBUS read cycle for either the secondary address or the data read according to the MS code. The result is stored in the 32 bit register. The MS code must be set and the timer should be reset in the calling sequence.

- **FB_SS**
  It reads the current SS response and sends it to the data destination via 32 bit register. The data destination is normally the external event buffer which is connected to the ***SMI/ECL card***.

- **FB_CB**
  It performs a bus reset to release all signals on FASTBUS.

- **FB_WT**
  It controls the sequencer wait for N microseconds. The N is given by the TCNT register at the calling time.

The timer in the FB_PA, FB_WR and FB_RD is used to control the timeout which is 4 μsec. The error code is generated and sent to the data destination via the 32 bit register whenever a timeout occurs. The error codes are defined in Table 3.7.

Table 3.7: The error codes during the FASTBUS operations

Error Code	Description of Errors
11 (hex)	$AK \neq 0$ after $AS = 0$ during primary address cycle
12 (hex)	$AK \neq 1$ after $AS = 1$ during primary address cycle
20 (hex)	$DK \neq 0$ after $DS = 1$ or $DK \neq 1$ after $DS = 1$ during write operation of secondary address or data cycle
30 (hex)	$DK \neq 0$ after $DS = 1$ or $DK \neq 1$ after $DS = 1$ during read operation of secondary address or data cycle

Basic macros

The following macros can be included at any place by compiler.

- **PAUSE**
  It makes a delay of 1 – 256 microseconds. An argument should be followed to give the time for delay.

- **WAITMS**
  It makes a delay of 1 – 256 milliseconds. An argument should be followed to give the time for delay.

- **LD32**
  It loads the 32 bit register from the instruction word. A 32-bit hexadecimal word has to be given as the argument.

- **ZERO32B**
  It clears the 32 bit register.

Applications

The application routines can be classified into the following three groups.

- **Initialization.**

  - **FBCLR**
    It calls the FB_CB subroutine to releases all signals from FASTBUS and clears the bus.
  - **MTPIN**
    This routine performs a broadcast to find out all modules in the crate and reads the identifiers of the modules. The identifiers are sent to the event buffer in the sequence of the slot numbers. The empty slots are presented as zeros.
  - **CREMEM module**
    It sends a test word and the crate identifier to event buffer which is used to check data path and the relationship between memories and the ***ADC crates***. The crate identifier is read from the personality card connected to the ADC in the slot number 6 of each ***ADC crate***.

- **Readout.**
  Several readout routines [16] are used for different applications. The name of the routines are defined by 4 characters which are listed in Table 3.8. A short description is given here. The global data taking routine (BTMD) is illustrated in detail later.

Table 3.8: The SMI readout routines

Character	Symbol	Meaning
The first	B	Using ADC buffer mode
	N	Using ADC non-buffer mode
The second	T	Trigger and headers from L3 trigger control logic
	L	Trigger from local trigger logic and headers from SMI
	I	Both trigger and headers generated by SMI
The third	M	Normal Memory header (one event as a record)
	O	Only one memory header (the whole memory as a record)
The fourth	D	Normal data headers and SS codes are sent
	N	No data headers and SS codes are sent

  - **BTMD**
    It is used for the global data taking. It needs both real trigger and headers

provided by the ***L3 trigger control logic***. The trigger is distributed via the ***local trigger logic***.

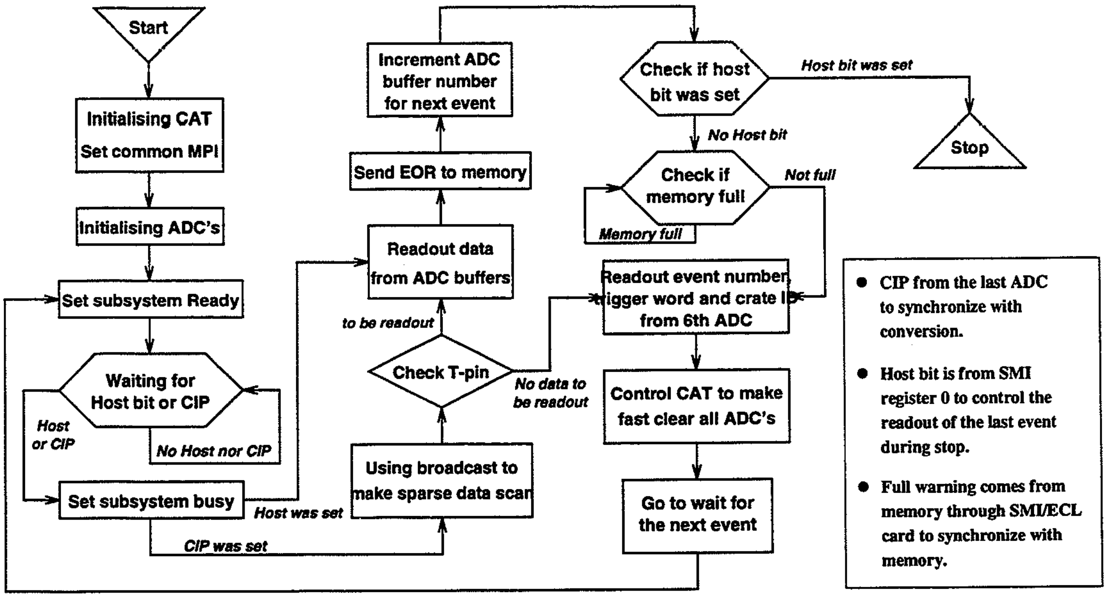
  - **BLMD**
    It simulates the global running condition in the ***local mode***. The trigger is generated by the ***local trigger logic***. The 8-bit event number is generated by SMI. The trigger pattern is a dummy word. The crate identifier is read out from the personality card connected to the ADC in the slot number 6 of each ***ADC crate***.
  - **BIMD**
    It gives a possibility to run each crate individually by using the ***internal trigger***. The headers are generated in the same way as the routine BLMD.
  - **BLMN**
    It is used for the calibrations. The trigger comes from the ***local trigger logic*** and data is sent to the memory without any headers and SS codes in order to reduce the processing time.

- **System test.**
  Many of test functions use the readout routines to provide the ADC data for the tests, for example, the event number test and the data validation test. Only the memory test function needs a special SMI code. In this test, the dedicated code generates an event with 512 32-bit words. Each event consists of two data blocks. In the first block, the least significant byte is incremented from 0 to 255, the bits 8 – 15 contains a 8-bit event number and the most significant 16 bits are cleared. In the second block, the least significant 16 bit are cleared, the most significant 8 bits contains the event number and the bits 16 – 23 are incremented from 0 to 255. The two bytes which have data simulate the changes as the slot number and the channel number in the ADC words. A RDOC is sent to the event buffer at the end of each event so that a memory headers is generated by the LRS1892 memory automatically.

Data taking procedure

Here is the detail description of the main macro BTMD [16] which is used for L3 data taking. The flow chart of the macro is shown in Figure 3.7.

Three important input signals are monitored by the program via the condition code. The CIP (from the INP1) allows the SMI to synchronize with the ADC conversion and is used to start the data transfer of the previous event in the ADC buffer. The HOST bit from the register 0 controls the stop sequence. Whenever the SMI gets the HOST flag, the SMI reads out and sends the last event in the ADC buffer to the memory and then the control is transferred back to the idle loop. The full warning (from the external event buffer via the ***SMI/ECL card***) protects against the overwriting in the memory.


Start
Initialising CAT
Set common MPI
Initialising ADC's
Set subsystem Ready
Waiting for Host bit or CIP
Host or CIP
No Host nor CIP
Set subsystem busy
Host was set
CIP was set
Increment ADC buffer number for next event
Send EOR to memory
Readout data from ADC buffers
to be readout
Check T-pin
No data to be readout
Using broadcast to make sparse data scan
Check if host bit was set
Host bit was set
Stop
No Host bit
Check if memory full
Not full
Memory full
Readout event number, trigger word and crate ID from 6th ADC
Control CAT to make fast clear all ADC's
Go to wait for the next event

- CIP from the last ADC to synchronize with conversion.
- Host bit is from SMI register 0 to control the readout of the last event during stop.
- Full warning comes from memory through SMI/ECL card to synchronize with memory.

Figure 3.7: The flow chart of the SMI data taking routine

The ready signal is generated via OUT2 which is controlled by the FASTBUS protocol code in instructions. It is set before waiting for the CIP signal and is cleared after a CIP is received.

Since the CIP comes from the last ADC only, a sparse data scan is performed after receiving a CIP in order to check whether all ADC's have the data in their buffers. Whenever the data are ready in all ADC modules, the transfer is started from the ADC's one after the other by means of the FASTBUS block transfer. The FASTBUS slave response code (SS code) is sent to the destination after the block transfer from each ADC module. After the readout, the ADC buffer number is incremented for the next event. The whole readout time is less than 500 μsec. The timing is illustrated in Figure 3.8.

It is important to realize that the readout is always delayed by one event according to the ADC conversion because the readout and the conversion are started concurrently. That means the first cycle has no data being readout due to the fact that buffer is empty. The first event is readout in the second CIP duration. The event headers (event number, trigger pattern and the crate identifier) are sent after the readout of the previous event. It is synchronized with the data conversion. The last event can be sent out by means of setting the HOST bit in register 0 of each SMI by the CHI.

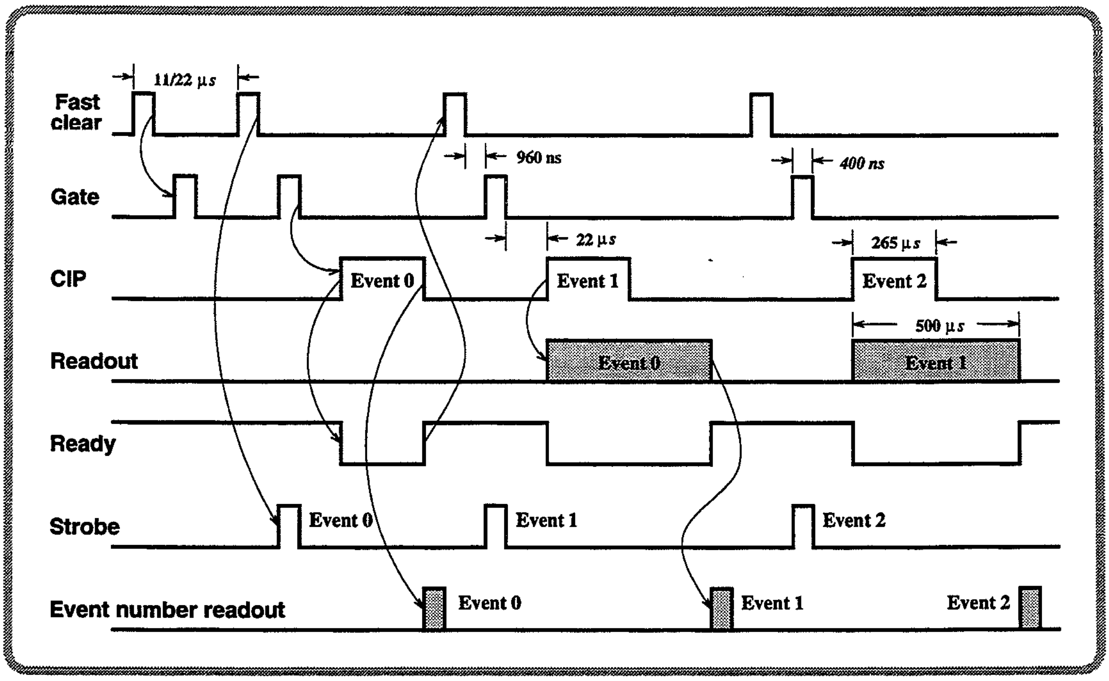


Figure 3.8: The readout timing in the HCAL system

3.3.5 Data Format

The event of HCAL data [17, 16] consists of a memory header, three data headers and 18 or 19 ADC data blocks. The memory header is generated by the memory automatically when an end of record is received. The three data headers are the event number, trigger pattern and the crate identifier.

The 12-bit ADC word is combined with the slot number, channel number and buffer number which are defined in Table 3.9.

Table 3.9: The bit map of the ADC word

Bits	Meaning
0 – 11	ADC data
12 – 15	unused
16 – 22	Channel number
23	Range bit for the 1885F only
24 – 26	Buffer number
27 – 31	Slot number

Each ADC data block is followed by a Slave Status (SS) response code [1] which indicates the status of the block transfer. The normal SS code should be the $SS = 2$ which means that the block transfer was successfully ended. If the data compression is enabled, it is possible to have only a SS code from one ADC. The average number of data from each subsystem is about 100 words. The event structure is shown in Figure 3.9.

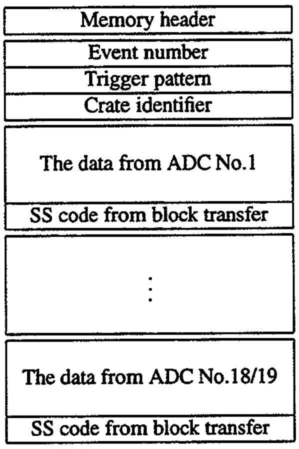

Memory header
Event number
Trigger pattern
Crate identifier
The data from ADC No.1
SS code from block transfer
⋮
The data from ADC No.18/19
SS code from block transfer

Figure 3.9: The HCAL data format in each memory

Since the FASTBUS crate has 26 slots (from 0 to the 25), we use the slot number greater than 25 to indicate the special words: headers, the SS code and the error codes. The contents of these words are defined in Table 3.10.

Table 3.10: The format of control words

Word Name	Identifier Value	Information Bits	
	(Bits 27 – 31)	Bits	Information
The First Header Word	26	0 – 15	Event Number
The Second Header Word	26	0 – 15	Trigger Pattern
The Third Header Word	26	0 – 3	Crate Identifier
SS Response Word	27	0 – 2	SS Response
Error Code Word	28	0 – 15	Error Code

The error codes reports the FASTBUS operation errors. It can be in any place of the data block. The error codes are listed in Table 3.7.

3.4 System Performance

The HCAL DAQ system has already been used for 5 years. The system has a high reliability and its failure rate is less than 0.5% during LEP in physics condition (including the failures from HCAL endcaps system). The most of faults are caused by real hardware problems.

The ADC conversion time is about 265 μsec. The total readout time is less than 500 μsec. Thus the dead time is about 0.5% at the 10 Hz level-1 trigger rate during the global data taking.

During the pedestal and test pulse calibrations, the processing rate of the CHI is about 4 Hz. The total calibration time is about 5 minutes for 1000 events. The uranium noise calibration takes about 20 minutes for a 16 bin spectrum using one million triggers.

Reference

[1] **An American National Standard — IEEE Standard FASTBUS Modular High-Speed Data Acquisition and Control System**. ANSII/IEEE Std 960-1993, December 1993.

[2] LeCroy Corporation. **Model 1882F and 1885F FASTBUS ADC's (Preliminary Information)**, October 1987.

[3] LeCroy Corporation. **1821 User's Manual**, March 1987.

[4] LeCroy Corporation. **1892 Users Manual**, April 1987.

[5] LeCroy Corporation. **Manual 1821/ECL**, July 1985.

[6] D.N.Mao, V.K.Gupta and M.M.Ilyas. **The Design of the SMI 1821 Interface**. FBLJA/L3, World Laboratory. FBLJA Internal Report, May 1989.

[7] H.Müller, T.Berners-Lee, R.Divià, and D.Burckhart. **CERN Host Interface User's Manual (CHI-P Processor and FASTBUS Port)**. European Organization for Nuclear Research (CERN), May 1989.

[8] T.J.Berners-Lee. **RPC User Manual**. European Organization for Nuclear Research (CERN), CERN/DD/OC, February 1989.

[9] Horst von Eicken. **Software Support for Motorola 68000 Microprocessor at CERN — MoniCa Symbolic Debugger**. European Organization for Nuclear Research (CERN), CERN 86, January 1986.

[10] Horst von Eicken. **Software Support for Motorola 68000 Microprocessor at CERN — M68MIL Cross Macro Assembler**. European Organization for Nuclear Research (CERN), CERN 83-13, December 1984.

[11] J.Blake. **Microprocessor Cross Software under VAX VMS**. European Organization for Nuclear Research (CERN), GEN 46, March 1989.

[12] U.S. NIM Committee. **FASTBUS Standard Routines**. U.S. DOE/ER-0325, June 1989.

[13] Dr.B.Struck. **Technical Manual — STR320 FASTBUS to CAMAC Branch Driver (FBD).**

[14] Institute of Electrical and Electronics Engineers (IEEE). **CAMAC instrumentation and interface standards**. IEEE Std 583-1975, 595-1976, 596-1976, 683-1976, 1976.

[15] D.N.Mao and M.M.Ilyas. **A Dedicated Microcode Compiler for SMI 1821**. FBLJA/L3, World Laboratory. FBLJA Internal Report, January 1989.

[16] X.Cai and M.M.Ilyas. **The SMI 1821 Resident Programs for the Hadron Calorimeter Data Acquisition System**. FBLJA/L3, World Laboratory. FBLJA Internal Report, May 1989.

[17] M.M.Ilyas. **The Hadron Calorimeter Data Format**. FBLJA/L3, World Laboratory, FBLJA Internal Note, 1988.

Chapter 4

The Event Building System

4.1 Introduction

The task of the L3 event building system [1, 2, 3, 4] is to combine the data from the five individual sources (TECH, ECAL, HCAL, MUCH and trigger) into a single event. Each branch of the data stream should be easily included in or excluded from the main DAQ procedure, global data taking. The system should be able to receive input data at the level-1 trigger rate and to produce the output events at the level-2 trigger rate.

Therefore the system is composed of two stages implemented in the FASTBUS standard [5]. The first stage is composed of four subsystems housed in four FASTBUS crates. Each subsystem has an event builder, called the subdetector event builder (SEB), to assemble the subdetector data from the source memories. The SEB sends the merged data to the FASTBUS cable port of the dedicated event buffer at the second stage.

In the second stage, five dual port memories are used respectively to receive the data from the four SEB's as well as the trigger data from the level-2 trigger processor. They are housed in one FASTBUS crate, called the ***central crate***, together with a central event builder (CEB). The CEB combines the data from all memories according to the L3 event structure. Upon the level-2 trigger decision, the completed event is sent to the third level trigger system via the FASTBUS cable segment.

The whole system is shown in Figure 4.1. It was first set up in 1989. At the beginning, the event builders were composed of two intelligent modules; the FASTBUS Block Mover (BM), F930 [6], and the FASTBUS General Purpose Master (GPM), STR500 [7]. The LeCroy LRS1892 memory [8] is used as the event buffer in the SEB stage and the Dual Slave Memory (DSM), STR351 [9], is chosen as the

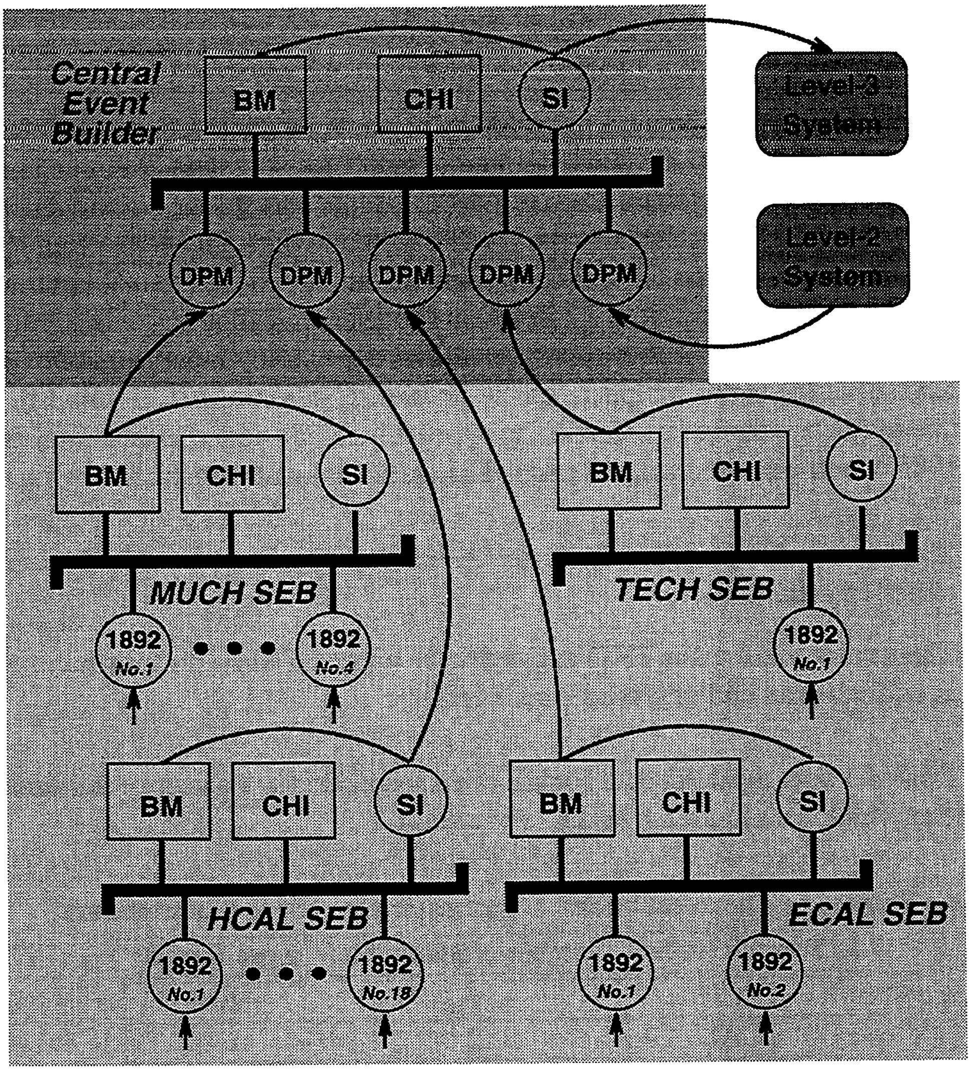


Figure 4.1: The L3 event building system

dual port memory in the CEB stage. Due to the problem of reliability and processing capability, the GPM and DSM were replaced respectively by the CHI (CERN Host Interface) [10] and the DPM (Dual Port Memory) [11] in 1992.

The author's contributions in this part is on the software update based on the upgrade of the CHI and DPM as well as the FASTBUS segments configuration which has been changed at the beginning of 1994. The following sections describe the upgraded system in detail.

4.2 System Composition and Data Flow

The SEB's process the subdetector's data in parallel at the level-1 trigger rate. The data from each subdetector are delivered into several LRS1892 memories concurrently by the dedicated readout system. The LRS1892 memory was chosen as the event buffer at this stage because it has not only the FASTBUS port which allows the data readout by the event builder via the FASTBUS backplane, but also a front panel input port in the ECL standard which can fit to the different subdetector readout systems. Following the data produced by the readout system, it allows either 16 bit input format for TECH and ECAL [12, 13] or 32 bit input format for the HCAL and MUCH [14, 15]. The 16-bit mode can be selected automatically if the cable for the upper 16 bits is absent [8]. Two 16-bit words are automatically merged into a 32-bit word with the following rules:

1. The first word is placed in the lower 16 bits and the second word is placed in the upper 16 bits.
2. The hexadecimal value FFFF is filled to the upper 16 bits if the number of words is odd when an end of record is received.

Table 4.1 lists the input data format of each subdetector as well as the number of memories which are required by each subdetector.

Table 4.1: The event buffer memories required by the subdetectors

Detector Name	Number of Memories	Input Format
TECH	1	16 bit
ECAL	2	16 bit
HCAL	18	32 bit
MUCH	4	32 bit

The DPM [11] has replaced the DSM module as the event buffer in the CEB stage. It was designed by LAPP with some features which are needed in the L3 environment. It is a real dual port memory which allows the access to two ports concurrently. Its FASTBUS cable port is used for the input data flow from either the SEB or the level-2 trigger processor. The FASTBUS crate port is used for the output data flow read by the CEB. A common FASTBUS control space register CSR10 is provided to protect against the overwriting of unread events in the memory. There is no additional header added to the records by the DPM besides a marker at the end of each record. This marker generates the FASTBUS slave response $SS = 2$ to indicate the end of block during the block transfer in the read operation.

The BM was chosen as a part of the event builder due to its advantage of two master ports on both FASTBUS crate and cable segments. The BM [6] is based on an

8-bit microcode sequencer — Motorola MC10904. During the data transfer, it holds the mastership on both segments and performs a simultaneous read and write block transfers from the source to the destination. Each data word goes through a single word pipeline register as a memory buffer which introduces only a 50 *nsec* delay per data cycle. Besides the data transfer, it is normally working as a slave under the control of the CHI. The slave port can be selected to be on either crate or cable segment depending on a DIP switch.

The microprogram of the BM is stored in a $2K \times 32bit$ of static RAM and controls the BM operations. The microcode is started by the CHI via the CSR0 of the BM. After the program is ended, the control is returned to the CHI via the FASTBUS interrupt. A $1K \times 32bit$ memory in the CSR space can be used to keep the user defined information for the operations which are loaded by CHI before the microprogram is started. A dedicated microcode compiler [16] was developed in the VAX FORTRAN language by the L3 Rome group.

The CHI [10], which has already been introduced in Section 3.2.3, is used as the main device of the event builder due to its powerful microprocessor. The CHI is used not only as the GPM to supervise the events and to start the BM for the data transfer, but also can transfer data by itself. Therefore there are two possible data transfer modes, with or without the BM. The mode is selected by the software configuration tool and is loaded into the CHI by the initialization process.

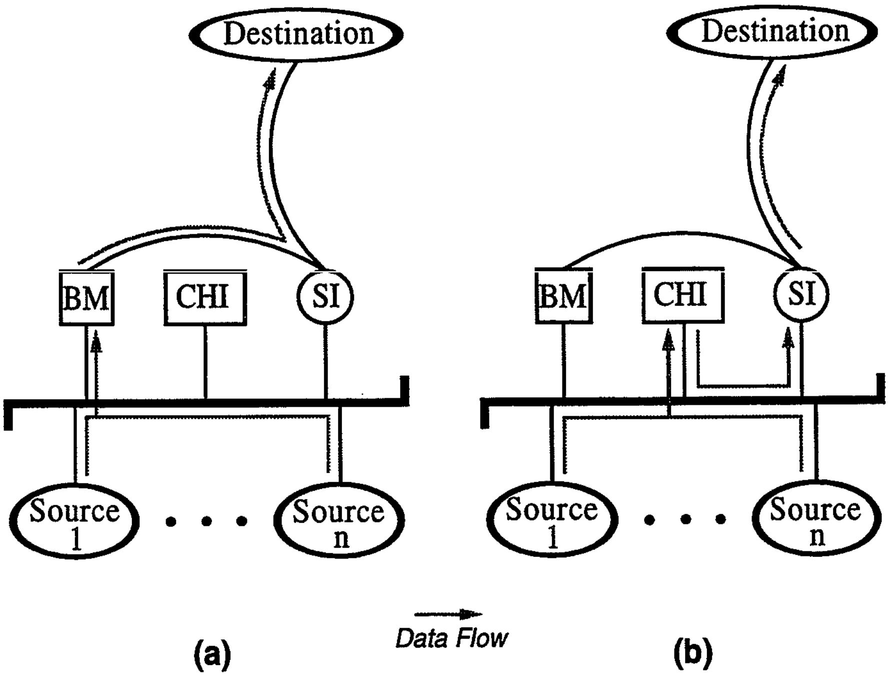


Figure 4.2: The data flow in the event builder system

All event builders, the SEB's and the CEB, have the same architecture in which all source memories are connected to the FASTBUS crate segment and the destination is on the FASTBUS cable segment. Figure 4.2 shows the data flow in both data transfer modes, with (a) or without (b) the BM.

Using the BM, the data flow goes from the sources to the destination directly through the BM. Without the BM, the CHI transfers data in two steps. Firstly it reads the data into its internal data memory from all source memories. Then the assembled data are sent out to the destination via a SI module. The block transfer of the CHI is done by an on-board DMA controller. The CHI data memory allows about 20 Mbytes/sec access rate. Due to the double accesses of the data memory, the whole data transfer rate is about 10 Mbytes/sec.

4.3 The L3 Event Structure

Since the offline software is based on the ZEBRA memory management package [17], the linear ZEBRA FZ format is used to build the L3 event [1, 18, 20].

The ZEBRA format makes easier the building of data structures and the transport among different computers. ZEBRA relies on the concept of ***bank*** which is defined as an area of contiguous memory storage for a logical data unit. A ***base address*** gives the actual location of the bank in the memory and a set of control words specifies the contents of the bank. Several banks can be organized and linked to each other via the ***link address***.

Each subdetector delivers its data in several LRS1892 memories. The data is in the hardware generated format and is fully transparent to the event builders. The SEB combines the data from its source memories into a single logical data unit, *i.e.* the corresponding subdetector bank, and stores the bank into the dedicated DPM. In the same way, the trigger raw data are assembled by the level-2 trigger processors and are delivered into the trigger DPM as the trigger bank. The CEB creates an event bank which describe the general information about the event and then link it with the subdetector banks and the trigger bank. The structure of the six banks in the final L3 event is illustrated in Figure 4.3.

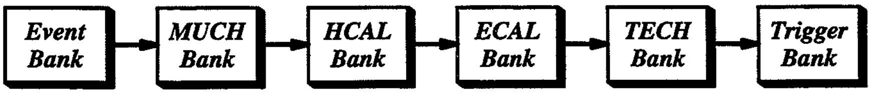


Figure 4.3: The L3 data banks

Each bank, as required by the ZEBRA format, has both numeric and Hollerith identifiers. The identifiers provide flexibility when there are missing banks in the event.

The names and the identifiers of L3 data banks are listed in Table 4.2.

Table 4.2: The L3 data banks

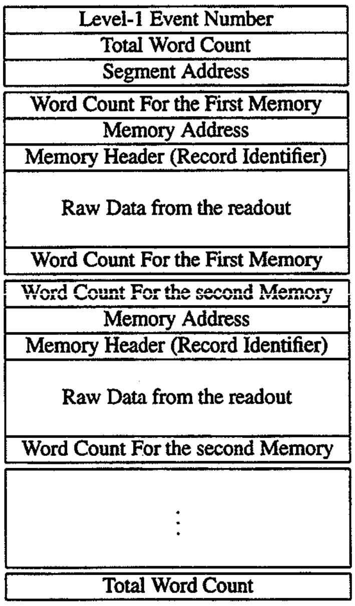
Name	Bank Identifier		
	ASCII	Hollerith (Hex)	Numeric (Dec)
Event Bank	EVVE	45565645	1000
MUCH Bank	MUUM	4D55554D	1001
HCAL Bank	HCCH	48434348	1002
ECAL Bank	ECCE	45434345	1003
TECH Bank	TEET	54454554	1004
TRIG Bank	JTTJ	4A54544A	1005

Each bank is controlled by 10 header words which are listed in Table 4.3.

Table 4.3: ZEBRA headers

1	I/O control/offset	6	Hollerith bank identifier
2	Address of the next bank	7	Total number of links (NL)
3	Address of the up bank	8	Number of structural links (NS)
4	Address of the original bank	9	Number of data words in this bank (WC)
5	Numeric bank identifier	10	Status of the data

Besides the ZEBRA header, each subdetector bank consists of three common words, data blocks from LRS1892 one after another and the total word count at the end. Figure 4.4 shows the data organization.



Level-1 Event Number
Total Word Count
Segment Address
Word Count For the First Memory
Memory Address
Memory Header (Record Identifier)
Raw Data from the readout
Word Count For the First Memory
Word Count For the second Memory
Memory Address
Memory Header (Record Identifier)
Raw Data from the readout
Word Count For the second Memory
⋮
Total Word Count

Figure 4.4: The organization of the subdetector bank

The event number is generated by the L3 trigger control logic and is delivered by each subdetector readout system. The total word count does not include the ZEBRA header words and the event number word. The segment address is used to indicate the corresponding memory crate. The trigger data bank is built by the level-2 trigger processors in the same format as the subdetector bank.

Besides the ZEBRA header, the event bank consists of 35 words to describe the event. Some of them are filled by the CEB and the rest are filled by the third level trigger processor and the ***producer*** on the VAX. The event bank is defined in Table 4.4.

Table 4.4: The event description bank

1	Run Number	19	Word Count of ECCE Bank
2	Event Number	20	Word Count of TEET Bank
3	L3 Status	21	Word Count of JTTJ Bank
4	Date	22	Total Word Count
5	Time	23	Level-3 Interface Address
6	Event Error Code	24	Event Count for Level-3
7	Level-1 Trigger Pattern	25	Level-3 Flag
8	Level-2 Trigger Pattern	26	TECH Status (1)
9	Level-3 Trigger Pattern	27	TECH Status (2)
10	Level-3 Processing Time	28	TECH Status (3)
11	Beam Energy in MeV	29	TECH Status (4)
12	Unused	30	TECH Status (5)
13	Unused	31	TECH HV Status
14	Level-1 Event Number	32	Unused
15	Level-2 Event Number	33	Unused
16	Word Count of EVVE Bank	34	Unused
17	Word Count of MUUM Bank	35	Unused
18	Word Count of HCCH Bank		

4.4 The Event Building Software

The event building software was originally written in FORTRAN and was upgraded to the C language by Mr. G.A.Hu in 1993 in order to easily include assembly language in the source code. The source and executable files of the software are stored on the online VAX computer. The CERN cross assembly tools and GNU-C compiler are used to produce the CHI executable file in the Motorola S-record format. The programs are downloaded into the CHI's and are started during the initialization.

The event building procedure

The SEB and the CEB have almost the same procedure to build up the events. The main loop of the event building contains the following steps.

1. Check the level-2 decision from the trigger DPM (for the CEB only).

2. Check one by one if the source memories have valid data.

3. Check if the partial events in all memories are synchronized by a common event number attached to the data. It compares the numbers not only between each memory but also with a internal number in the CHI which is incremented at the beginning of the loop.

4. Check the destination. The SEB checks the read pointer in the DPM to find if enough space is left to store the event. In the CEB, the program checks if there is a ready device of the third level trigger to be able to receive the event.

5. Whenever both sources and destination are ready, the CHI starts the data transfer either by the BM or by itself.

6. If the BM is used, the CHI waits for the data transfer to be completed by the BM and then checks for any errors logged during the transfer.

7. Set the new read pointers in the source memory to free the buffers which are located by the processed event. The CEB has to start the corresponding level-3 processor and reset the level-2 trigger flag.

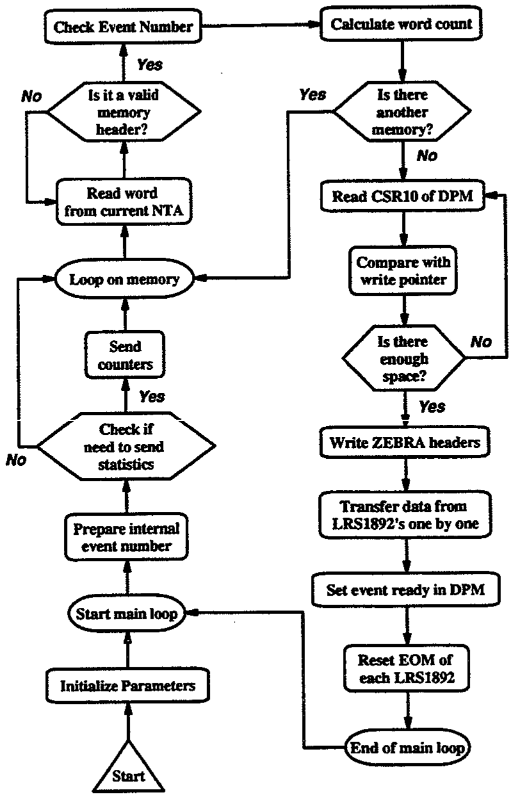


Figure 4.5: The flow chart of the subdetector event builder

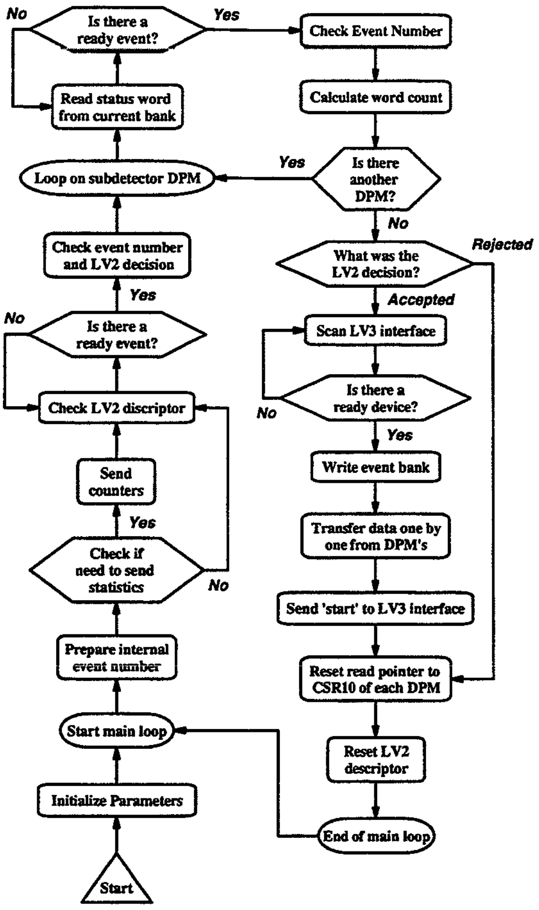


Figure 4.6: The flow chart of the central event builder

Because of the different device types of the event builders on both sources and destination, the CEB uses a somewhat different algorithm from that of the SEB's. The flow charts of both programs are respectively shown in Figure 4.5 and Figure 4.6.

The information from the L3 run control and the system configuration is needed for each event builder to know the data transfer condition and environment. Each event builder has a default parameter file stored on the VAX which is read by the initialization process and modified according to the current requirements. The parameters are downloaded to a dedicated area in the data memory of each CHI during the initialization.

The system test

Besides building up the events for the normal data taking, the event building programs provide a test function for the system. The test function mode is selected from the L3 run control. In this mode, neither the subdetector readout electronics nor the trigger systems are used. The level-3 trigger system is transparent to the data. The data is generated by each SEB and filled into each source LRS1892 memories. Then the SEB's and the CEB build up the event in the normal way and finally send the events to the producer.

Currently a ***random pattern test*** and a ***speed test*** are implemented. In the random pattern test mode, the data check can be performed in each of the following levels, the SEB's, the CEB, the level-3 trigger process or the producer. This allows the checking of the whole DAQ chain from the LRS1892 memories up to the producer except the trigger data partition. It is very useful for diagnosing problems in the system. The data check is done by the CHI only in each event builder so that the BM has to be disabled in this case. The speed test is used to test the throughput of the system.

In the test mode, the procedure in the CEB is the same as for normal data taking, moreover the data check is enabled. At the SEB level, the behavior is different following the test mode which is selected.

- **The random pattern test.**
  In this mode, the CHI generates the random data in a large array in the CHI memory before starting the event building. Then the CHI generates two random numbers for each LRS1892 memory in each event, one is the pointer of the array and another is the data length. The two parameters for each memory locate a block of random data in the array. The data block is headed by a memory header which is generated by the CHI as well and is written into the corresponding memory. After the data blocks in all memories are ready, the event builder starts the normal procedure to build up the event and to send it to the destination. If the data check is enabled, the CHI checks the data validation before sending data to the destination.

- **The speed test.**
  In this mode, the CHI fills the whole memory of each LRS1892 with random data before starting the event building. The data structure is the same as the one generated by the front end electronics. The average record length is the mean data size of the subdetector electronics. Then the event builders build up the events in the normal way. Since the data is always ready in the memories, the event building is performed as fast as possible. It is used to test the system performance.

The data transfer of the BM

When the BM is enabled, it executes the data transfer. The procedure is programmed in microcode. Whenever the BM is started, the sources and the destination have been already checked by the CHI and are assumed to be valid. The following five steps are then performed in the BM.

1. Send either the ZEBRA headers (in case of the SEB), or the event bank (in case of the CEB) to the destination. The ZEBRA headers or the event bank have been prepared and loaded into the BM memory by the CHI before the BM is started.
2. Pipe the data to the destination from the source memories one after another according to the bit map of the active memories.
3. Write the word count to the corresponding locations in the destination. The word count has been counted automatically during the block transfer.
4. Set the ready word in the destination DPM in case of the SEB, or send a start command to the level-3 trigger interface in case of the CEB.
5. Send an interrupt via the FASTBUS interrupt message (FIM) in the CSR space of the CHI to return the control back to the CHI, then the microprogram terminates.

4.5 System Configuration

In order to change the hardware configuration and the software parameters easily, some tools have been developed. The hardware configuration tool is used to manage the FASTBUS segments. It defines the path to allow each event builder to access its destination on the FASTBUS cable segment. The software configuration tool defines the parameters which are needed by the event builders.

The hardware configuration

One of the basic elements of FASTBUS [5] is the ***segment***. It is an autonomous bus interconnecting the modules and can be implemented as a backplane card in a crate (***crate segment***) or as a cable (***cable segment***) with the same functional characteristics. A Segment Interconnect (SI) [21] can provide a link between a cable segment and a crate segment. Each segment needs an address to be identified. The segment address is implemented as the group address (GP) in the most significant byte of the address word [5]. If GP is zero, the address is accepted by the ancillary logic in the current segment which controls the attribution in the segment. The address with non-zero GP can be passed to the module in the segment through several SI modules if the GP is defined and enabled in the routing tables on those SI modules.

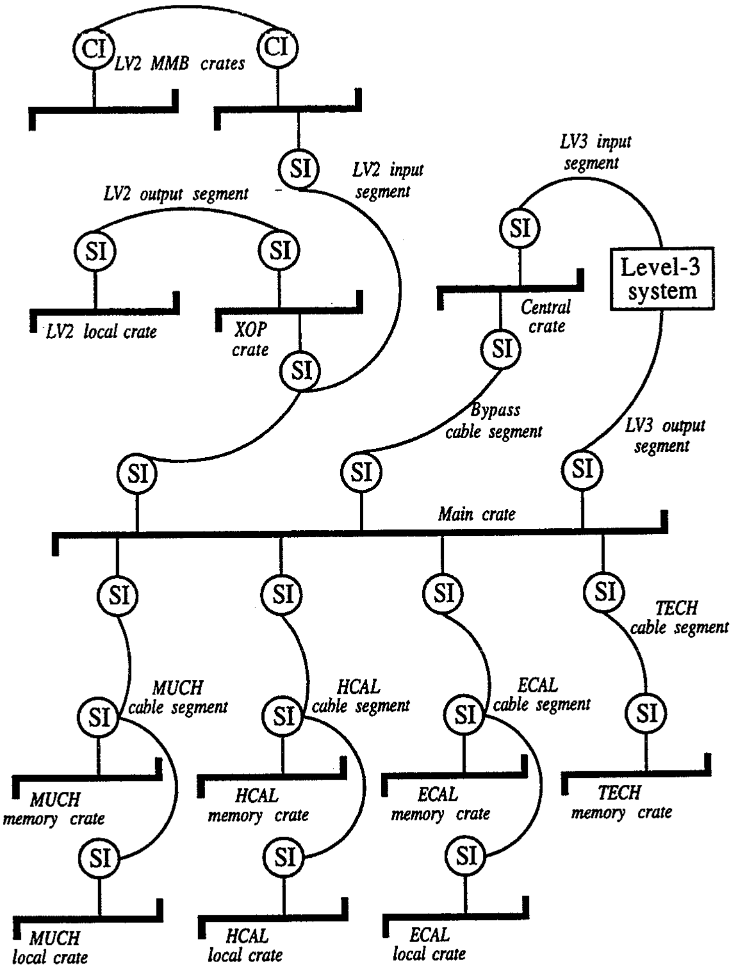


Figure 4.7: The original configuration of the L3 FASTBUS segments

In the L3 FASTBUS system, the segments were set up as shown in Figure 4.7. Since the GPM has no direct link with the VAX computer, all segments are initialized via the CHI on a crate, called the ***main crate***, which is used for the level-3 trigger readout. Therefore the segments are organized in a tree structure. In the ***main crate***, the entries to all segments are provided.

Due to the upgrades of the CHI and the new level-3 trigger system, the CHI in the ***main crate*** became unnecessary so that the configuration of the system was changed at the beginning of 1994. The new setup is shown in Figure 4.8. The system is separated into several distributed subsystems to reduce the interferences between each other. Each subsystem has at least one CHI to be linked with the L3 online VAX computers. Another advantage is that the same setup is used for both the global and local mode. This avoids the enabling or disabling of the SI's in the main crate during the changes of local and global.

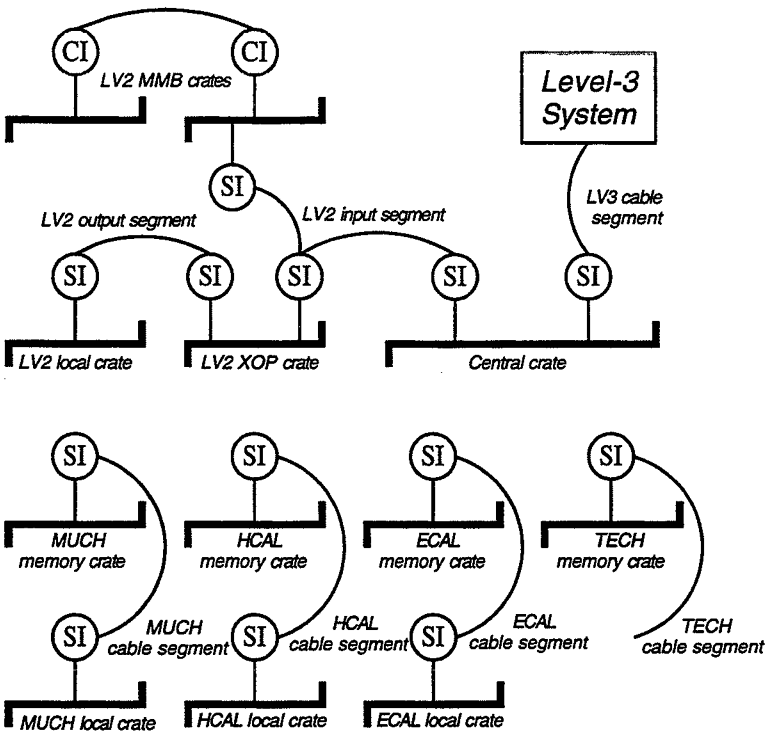


Figure 4.8: The new configuration of the L3 FASTBUS segments

In the new setup, each subdetector has a ***memory crate*** to keep the LRS1892 memories and its SEB as well as a ***local crate*** for the local activities except the TECH system which does not need a local FASTBUS crate. A ***cable segment*** links the two crates together and provides the access to the cable port of the dedicated DPM. The level-2 trigger XOP's need two FASTBUS cable segments, one for the input and another for the output. The ***input cable segment*** is connected to two memory crates containing the MMB modules, they are called the ***MMB crates***. The ***output cable segment*** connects to the DPM for sending the trigger data to the CEB. The ancillary logic of the output segment is located in the ***local crate***. The FASTBUS interfaces of the XOP's are kept in a crate which is called the ***XOP crate***. The new level-3 system needs only a ***cable segment*** for the input data. The CEB is located in the ***central crate***. The GP address of each segment is given in Table 4.5.

The FASTBUS System Management (FSM) package [22] which is based on the ORACLE database system [23] is used to manage the hardware configuration. The module locations, segment addresses and the path definitions are stored in the database. Each subsystem is defined to become an environment. Two files can be generated by the FSM database for each environment. A data file contains the segment initialization procedure which is used for the FSM run-time library to initialize the SI's and the segments in the subsystem according to the definitions. A parameters file keeps all

Table 4.5: The L3 FASTBUS segments

Subsystem Name	Segment Name	Segment Address
Central Event Builder	Central Crate Segment	20000000 (Hex)
Level-3 Trigger	Cable Segment	B2000000 (Hex)
Level-2 Trigger	Input Cable Segment	A1000000 (Hex)
	Output Cable Segment	B1000000 (Hex)
	XOP Crate Segment	41000000 (Hex)
	Local Crate Segment	31000000 (Hex)
	MMB Crate 1 Segment	51000000 (Hex)
	MMB Crate 2 Segment	52000000 (Hex)
HCAL	Cable Segment	A2000000 (Hex)
	Memory Crate Segment	42000000 (Hex)
	Local Crate Segment	32000000 (Hex)
MUCH	Cable Segment	A3000000 (Hex)
	Memory Crate Segment	43000000 (Hex)
	Local Crate Segment	33000000 (Hex)
ECAL	Cable Segment	A4000000 (Hex)
	Memory Crate Segment	44000000 (Hex)
	Local Crate Segment	34000000 (Hex)
TECH	Cable Segment	A5000000 (Hex)
	Memory Crate Segment	45000000 (Hex)

the module addresses with the GP addresses which can be searched for by the FSM run-time library whenever they are needed.

The software configuration

The software configuration is done by a program written in VAX FORTRAN language. It is based on the Window Manager (WM) [24] and the Cluster Process Communication (CPC) [25] libraries of the L3 online system. The WM is used to provide the human interface of the program. The CPC is used to keep the configuration in a cluster-wide database (CLUSCOM) which is a mapped common block. The cluscom of the configuration is updated by this program whenever a change is made and can be read by other processes on the same computer cluster. The following items can be configured.

- **The CHI serial numbers**
  Since the CHI is the only connection with the computer via the Ethernet, the corresponding ethernet address has to be given in order to access a subsystem through the CHI. The address is defined to use the protocol with addition of the CHI serial number issued by the manufacture so that the address for the subsystem is changed whenever the CHI is changed. Therefore the serial number of the CHI in each subsystem is kept in the cluscom.

- **The event building programs and parameters**
The program and the default parameter file can be selected for each event builder. The directories for both program and parameters can also be given. This makes the software debugging more flexible.

- **The BM selection.**
Because the CHI can be operated with or without the BM, each BM can be disabled or enabled by this program. For the same purpose, the microprogram for each BM can be selected to be different. The file directory can also be changed.

- **The LRS1892 memories selection.**
It is essential that each event builder knows the number of source memories and their locations. The sources for the CEB depends on the partitions which are in the global mode. The source memories for each SEB have to be configured by this program. The LRS1892 memory addresses are taken from the FSM parameter files. Every memory can be enabled or disabled. A 32-bit word is given to each SEB for the bit mask according to the memory locations. The bit number is used as the slot number in the memory crate when the bit is set.

- **The test mode definition**
The test function of the event builders can be set up in this program. The operation is defined by two words which are used in the event builders and the producer. One is the ***test mode*** which tells the system which kind of test should be done. Currently the random pattern test and the speed test are implemented. Another is the ***check mode*** which decides at which level the data check should be performed. The data can be check at the SEB's, the CEB, the level-3 processors, the producer or any combination of these.

4.6 The Initialization Process

The CHI is not a standalone computer so that a host computer is needed. In this system, all CHI modules are connected to ethernet and can be accessed from the online VAX cluster via ethernet. A dedicated process, L3_INIT_B, in the L3 run control is running on the main online VAX computer and is used to initialize the whole event building system.

Two major procedures are implemented and called the ***cold*** and ***warm*** initializations. The process initializes only the partitions, TECH, ECAL, HCAL, MUCH or trigger, which are in the global mode. Each partition can be easily included in or excluded from the system. To produce the output events, the system requires at least one partition in the global mode which provides the input data.

The cold initialization has two steps. Firstly, it initializes the FASTBUS segments of each partition one by one via the event builder CHI. The procedure uses the information in the FSM database and is controlled by the FSM run-time libraries. Secondly, the selected program for each event builder is downloaded into its CHI program memory by means of the RPC (Remote Procedure Call) [19] routines supported by the MoniCa, a CHI monitor program. The cold initialization has to be done at the beginning when the system is started as well as whenever any partition is included in or excluded from the global system.

The warm initialization has to be done before any data taking procedure. It performs the following steps.

1. **Get running parameters from the Run Control.**
   Since all parameters are set from the L3 Run Control, it is necessary to know under which condition the system will run. The following parameters are needed in the event builders.

   - ***Run Type*** is used to indicate the running purpose which can be for physics, cosmic rays study, calibrations or tests. It is important for the event builders to know whether it will run in the test mode or in the normal event building mode. A special run type called ***TEST_FB*** is defined to set the event builders in the test mode.
   - ***Detector Mask*** gives the information about which partitions are in the global operation mode. It is used by both cold and warm initializations to control which part of the system should be initialized. This information is also sent to the CEB to indicate the active data sources.
   - ***Level-3 Mask*** tells about which level-3 trigger processors are in use. It controls the initialization of the level-3 trigger interfaces and is used by the CEB to select the destinations.

2. **Initialize the LRS1892 memories.**
   Through each SEB CHI, all LRS1892 memories are initialized to be ready to receive data from the subdetector readout systems.

3. **Initialize the DPM memories.**
   Both the cable and crate ports of DPM's are initialized. The cable ports are done by means of each SEB CHI and the crate ports are done via the CEB CHI.

4. **Load microcode into the BM.**
   According to the configuration, the process downloads the selected microprogram from the VAX to the enabled BM modules via the corresponding CHI's.

5. **Initialize the level-3 interfaces.**
   The operation mode and the channel selection are sent to the level-3 trigger

interfaces. The channel selection of each interface is decided by the level-3 mask parameter. The operation mode is explained in the next chapter and is set by the software configuration tool.

6. **Download the running parameters to the CHI.**
   The process reads the default parameter file for each event builder according to the configuration and modifies the parameters corresponding to the information from the L3 run control. Then the parameters are loaded into the CHI data memory of each event builder. The first 10 words of the parameters are the ZEBRA headers of the data bank.

7. **Start CHI.**
   After all the above steps are successfully performed, the process starts all event builders.

Since each CHI has started to look at the sources and the destination after a warm initialization, the FASTBUS segments are locked by the CHI most of the time. A CHI reset function is implemented via a dedicated module in the central crate which was developed by the L3 Rome group. The module provides 6 TTL pulse output which are compatible with the reset input in the front panel of the CHI and are controlled by the FASTBUS operation of the modules. The reset is done before each cold initialization and the subdetector initializations as well as whenever any partition changes from the global mode to the local mode to free the FASTBUS segments for the local activities. The subdetector initializations are always done before the warm initialization of the L3_INIT_B process.

4.7 The Monitoring Program

It is important for the operators to know the execution states of all event builders. Two tools are provided by the L3 event building system. One is a ***display server*** which displays the messages from the four SEB's and from the CEB as well as from the producer during the test mode. Another is a monitoring program, called ***FASTBUS monitor***, which displays the statistics of the running states of all event builders.

In order to allow the ***display server*** and the ***FASTBUS monitor*** to run on the different computers, a ***communication server*** was implemented. It interfaces to all CHI's via ethernet to get the display messages and the statistics. The statistical results are written into a shared common. The display messages are sent to the ***display server*** via DECNET. All communication protocols are based on the RPC package. A server is provided to read the data from the shared common and send them to the ***FASTBUS monitor*** which can run on any computer in the cluster. It is called the ***copy server***. The relationship of these servers and processes is illustrated in Figure 4.9.

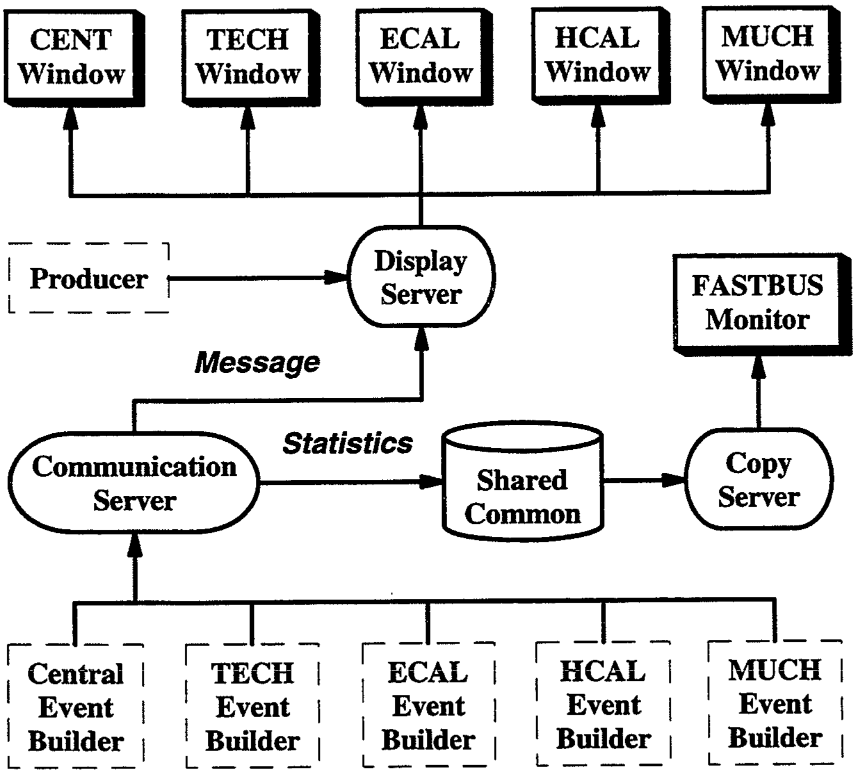


Figure 4.9: The event building monitoring processes

The communication server

The program of the ***communication server*** is written in FORTRAN language. The functions are mainly performed by the following three RPC routines.

- **Transmit_Message.**
  It is running as a RPC server to receive the messages to be displayed from all clients, the event building CHI's. The RPC is established over ethernet. Besides the text message, the CHI identifier and a flag are sent together with the message. The CHI identifier indicates the message source which is used to display the message on the corresponding window by the ***display server***. The flag contains the information about the destination of the message. The targets of the message can be the display window, log file and/or the L3 run control.

- **Send_Counter.**
  This is a RPC server routine and is used to receive the data block of the statistics from the clients. The data block contains the number of events processed and the number of each kind of error such as the event number error, word count error and so on. The CHI sends the information when a certain number of events

are processed. After receiving the data, the communication server updates the results in the shared common for the ***FASTBUS monitor***.

- **DSP_CHI.**
  This is a RPC client routine to send the display messages together with the CHI identifier and the destination flag to the ***display server*** over DECNET.

The display server

The ***display server*** contains a RPC server, DSP_CHI, to receive the message either from the ***communication server*** for the event builders or from the ***producer*** directly to indicate the results of the data check during the test operation. The messages are respectively displayed on five windows, four for the SEB's and one for the CEB. The messages from the ***producer*** are sent to the CEB window.

The software can be run in both the VWS or DECWindow environments. The program checks the bit mask in the destination flag to decide where the message have to be sent. On both the display window and the log file, each message becomes a line with a prompt which is the event builder name.

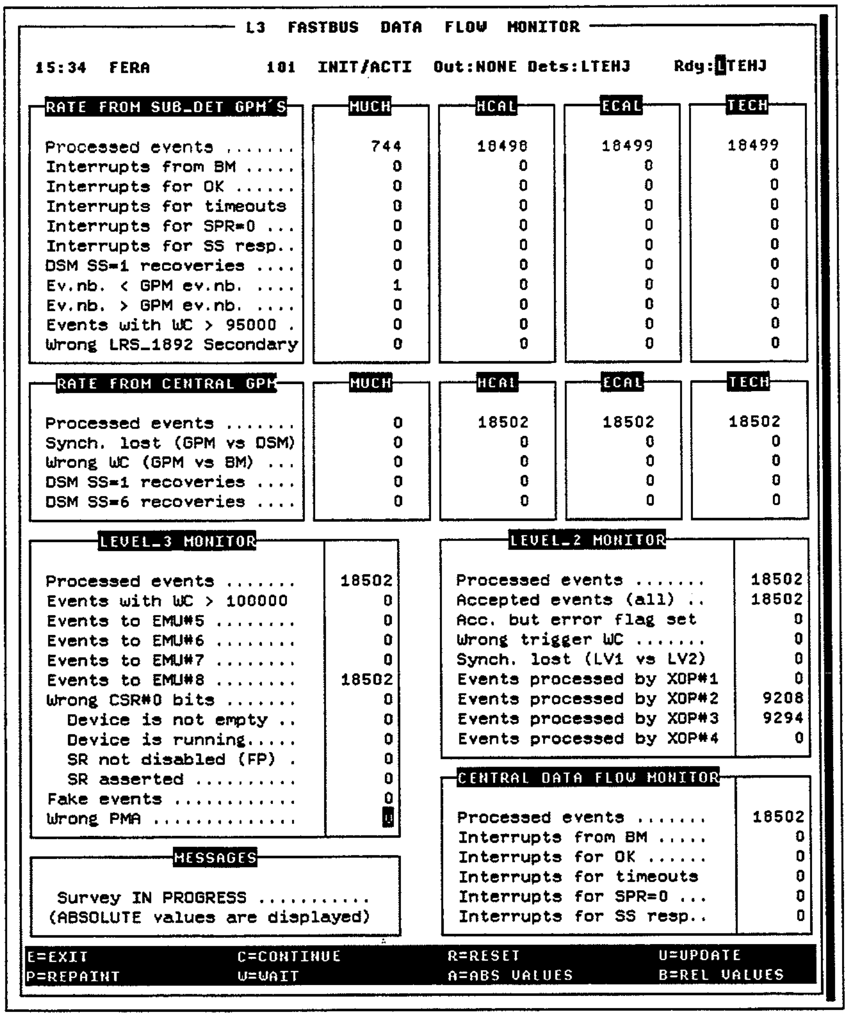
The FASTBUS monitor and the copy server

The ***copy server*** contains a RPC server routine which allows the data in the shared common to be read by the client, the ***FASTBUS monitor***. The ***FASTBUS monitor*** is written in FORTRAN and based on the SMG library for the VMS system. The SMG library provides the routines to control the display on the normal terminals. The FASTBUS monitor displays the statistics on the following windows as shown in Figure 4.10.

- **The SEB window** displays the information from each SEB, the number of events processed, the number of errors and the number of fake events which were sent by the SEB to replace the bad event due to the unrecovered errors.

- **The CEB window** gives the number of events processed by each SEB and the number of errors from the corresponding DPM.

- **The level-2 trigger window** displays the trigger data statistics; the total number of processed events, the number of events from each XOP processor, the number of accepted events, the number of rejected events and the number of errors from the trigger data.



```
                      L3  FASTBUS  DATA  FLOW  MONITOR

15:34  FERA            101  INIT/ACTI  Out:NONE Dets:LTEHJ    Rdy:LTEHJ

RATE FROM SUB_DET GPM'S        MUCH        HCAL        ECAL        TECH

Processed events .......        744       18498       18499       18499
Interrupts from BM .....          0           0           0           0
Interrupts for OK ......          0           0           0           0
Interrupts for timeouts           0           0           0           0
Interrupts for SPR=0 ...          0           0           0           0
Interrupts for SS resp..          0           0           0           0
DSM SS=1 recoveries ....          0           0           0           0
Ev.nb. < GPM ev.nb. ....          1           0           0           0
Ev.nb. > GPM ev.nb. ....          0           0           0           0
Events with WC > 95000 .          0           0           0           0
Wrong LRS_1892 Secondary          0           0           0           0

RATE FROM CENTRAL GPM          MUCH        HCAL        ECAL        TECH

Processed events .......          0       18502       18502       18502
Synch. lost (GPM vs DSM)          0           0           0           0
Wrong WC (GPM vs BM) ...          0           0           0           0
DSM SS=1 recoveries ....          0           0           0           0
DSM SS=6 recoveries ....          0           0           0           0

        LEVEL_3 MONITOR                        LEVEL_2 MONITOR

Processed events .......    18502    Processed events .......    18502
Events with WC > 100000         0    Accepted events (all) ..    18502
Events to EMU#5 ........        0    Acc. but error flag set         0
Events to EMU#6 ........        0    Wrong trigger WC .......        0
Events to EMU#7 ........        0    Synch. lost (LV1 vs LV2)        0
Events to EMU#8 ........    18502    Events processed by XOP#1       0
Wrong CSR#0 bits .......        0    Events processed by XOP#2    9208
  Device is not empty ..        0    Events processed by XOP#3    9294
  Device is running.....        0    Events processed by XOP#4       0
  SR not disabled (FP) .        0
  SR asserted ..........        0      CENTRAL DATA FLOW MONITOR
Fake events ............        0
Wrong PMA ..............        0    Processed events .......    18502
                                     Interrupts from BM .....        0
            MESSAGES                 Interrupts for OK ......        0
                                     Interrupts for timeouts         0
 Survey IN PROGRESS ...........      Interrupts for SPR=0 ...        0
(ABSOLUTE values are displayed)      Interrupts for SS resp..        0

E=EXIT           C=CONTINUE        R=RESET           U=UPDATE
P=REPAINT        W=WAIT            A=ABS VALUES      B=REL VALUES
```

Figure 4.10: The FASTBUS monitor display

- **The level-3 trigger window** displays the statistics from the destination operations of the CEB which includes the number of events which are sent to each interface, the number of errors detected during the data transfer to the level-3 interfaces and the number of fake events for the replacement of the bad events by the CEB.

4.8 The Diagnostic Tool

A diagnostic tool is provided for the operators to find out the reason whenever the operation of the DAQ system is hung up. The program is called FBDIAG, it is written in FORTRAN by the author of this thesis and is regularly updated and improved.

It displays the possible source of the problem and gives the recommendation of the recovery procedures. The diagnosis includes the following steps.

1. Interrogate the L3 run control to know which partitions are operated in the global mode. It indicates the parts of the system to be checked.

2. Get the ready status of each partition from the L3 trigger control logic to know which the subdetector readout systems are still busy. If the corresponding SEB is waiting for the data from a busy system, the problem is possibly on this readout system.

3. Read the information from the CEB and each SEB one by one to check the status of each event builder. The information is located in a dedicated place in the data memory of each CHI and are read via the MoniCa RPC routines. The following information is provided by each event builder.

   - ***The execution position*** tells the diagnostic tool on which part the event builder is working. It is important to indicate the source of the problem.
   - ***The current event number*** gives the information to point out the possible problem partition which has a less number of events processed.
   - ***The primary address*** indicates on which FASTBUS module the event builder is working. This is used to find the problem module in some cases.
   - ***The secondary address*** gives the information which is used, in some cases, by the experts to locate the problem in the bad module.

4. Reads the statistical results from each event builder to see if any fatal error took place.

5. According to all of the above information as well as the knowledge and the experience of the system, the program makes the diagnosis and gives the conclusion and the advice to the operator.

All of the information are recorded into a log file which can be checked by the expert afterwards.

4.9 System Performance

This system has been running since 1989 [20]. In 1991, all of the GPM and the DSM modules were replaced by the CHI and the DPM modules respectively. Since 1993, most of the BM modules have been disabled in most of time due to the unreliable cable port of the BM. The failure rate has become less than 0.5% during LEP in physics condition. Most of faults are caused by the hardware problems and the instability of CHI and DPM.

The throughput of the system depends on the number of the source memories. There are two typical cases in the L3 system. In the first case, the TECH data has been already collected by its readout system and requires only one LRS1892 memory with the event size of about 10 Kbytes. Thus the TECH SEB is doing block transfer in most of time so that it has a very high throughput. On the contrary, the HCAL data are put into 18 memories with about 100 words per event in each memory so that the HCAL SEB has a very low throughput. Because it has to make a lot of single word transfers to check the event in the memories. The throughput of the CEB depends on the number of subdetectors in the global mode. If all partitions are in global mode, the overhead of the data transfer is almost the same as the MUCH SEB which has 4 source memories. Table 4.6 lists the throughput of the system from the SEB up to the producer while the level-3 trigger is in a transparent mode.

Table 4.6: The throughput of the event building system

Conditions	Throughput	
TECH + CEB without BM's	102 Hz	1.1 Mbytes/sec.
TECH + CEB with BM's	112 Hz	1.2 Mbytes/sec.
MUCH + CEB without BM's	113 Hz	0.3 Mbytes/sec.
MUCH + CEB with BM's	200 Hz	0.5 Mbytes/sec.
HCAL + CEB without BM's	27 Hz	0.1 Mbytes/sec.
HCAL + CEB with BM's	57 Hz	0.3 Mbytes/sec.
All + CEB without BM's	27 Hz	0.8 Mbytes/sec.
All + CEB with BM's	48 Hz	1.3 Mbytes/sec.

Without the BM's, the system can process data in 27 Hz which is enough to handle the current level-1 trigger rate (10 – 15 Hz). To increase the system throughput for future beam condition, the main work will be to speed up the HCAL SEB which is the bottleneck of the system. The following solutions are under investigation.

- Use new event buffer memories in the HCAL memory crate. The new memories should support the sparse data scan and other broadcast function of the FASTBUS. This can reduce a lot of single word transactions by the HCAL SEB.
- Increase the CHI data transfer rate. The main problem is the double access of the CHI data memory while one access cycle for the data reading and another for the data writing. The solution is either to use some special CHI interface to transfer data from the SEB CHI to the CEB CHI directly, or to design a new master module which transfer data from the source memories to the ECL port of the master directly in the same cycle.

- Reduce the software overhead which comes from the FASTBUS standard routines. A new FASTBUS library is being written to provide new routines which should have less software overhead.

Reference

[1] B.Bertucci, S.Falciano, G.Medici, and D.Linnhöfer. **The L3 FASTBUS Data Acquisition System**. In *The Computing in HEP Conference, Santa Fe, New Mexico, USA*, April 1990.

[2] L3 Collaboration. **Trigger and Data Acquisition System of L3**. European Organization for Nuclear Research (CERN), CERN/LEPC 84-5 (LEPC/PR3/L3), January 1984.

[3] M.Fukushima *et al.*. **The Trigger and Data Acquisition System of the L3 Experiment**. L3 Collaboration, L3 Note 520, September 1987.

[4] M.Fukushima *et al.*. **L3 Data Acquisition System**. L3 Collaboration, L3 Note 371, March 1985.

[5] **An American National Standard — IEEE Standard FASTBUS Modular High-Speed Data Acquisition and Control System**. ANSII/IEEE Std 960-1993, December 1993.

[6] KineticSystems Corporation. **Model F930 Block Mover II Instruction Manual**.

[7] H.Müller and R.Vachon. **The GPM General Purpose FASTBUS Master/Slave User Manual**. European Organization for Nuclear Research (CERN), CERN EP 85-02, October 1985.

[8] LeCroy Corporation. **1892 Users Manual**, April 1987.

[9] Dr.B.Struck. **Technical Manual — STR323 Dual Slave Memory (DSM)**.

[10] H.Müller, T.Berners-Lee, R.Divià, and D.Burckhart. **CERN Host Interface User's Manual (CHI-P Processor and FASTBUS Port)**. European Organization for Nuclear Research (CERN), May 1989.

[11] B.Bertucci, S.Falciano, W.Koutsenko, D.Linnhöfer, G.Perrot, and S.X.Wu. **A Dual Port Memory DPM**. L3 Internal Note, May 1990.

[12] Philip E. Kaaret. **Forward Production of J/ψin Hadronic Interactions Calibration of a Large BGO Electromagnetic Calorimeter**. PhD thesis, Princeton University, June 1989.

[13] X.Lü. **The Time Expansion Chamber — Performance in a Testbeam**. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, 1991.

[14] The Chapter 3 of this thesis.

[15] G.A.Hu. **Muon Chamber FASTBUS Readout System**. L3 Online Documents, 1993.

[16] S.Falciano. **A FORTRAN program to compile BM microcode**. L3 Online Note, 1989.

[17] European Organization for Nuclear Research (CERN). **The ZEBRA System**, 1993.

[18] S.Banerjee and F.Bruyant. **The L3 Event Data Structure**. L3 Collaboration, L3 Note 748, May 1990.

[19] T.J.Berners-Lee. **RPC User Manual**. European Organization for Nuclear Research (CERN), CERN/DD/OC, February 1989.

[20] T.Angelov, B.Bertucci, S.Falciano, M.Fukushima, D.Linnhöfer, B.Martin, and G.Medici. **Performances of the Central L3 Data Acquisition System**. *Nuclear Instruments and Methods in Physics Research A **306***, pages 536–539, 1991.

[21] Dr.B.Struck. **Technical Manual — STR400 Segment Interconnct (SI)**.

[22] E.M.Rimmer. **FASTBUS System Manager (FSM) User Guide and Reference Manual**. European Organization for Nuclear Research (CERN), CERN/DD, December 1989.

[23] Michael Bronzite. **Introduction to Oracle**. New York : McGraw-Hill, 1991.

[24] T.Wenaus. **The L3 Online Window Manager Facility (WM)**. L3 Collaboration, L3 Note 807, April 1989.

[25] T.Wenaus. **The L3 Cluster Process Communication Facility (CPC)**. L3 Collaboration, L3 Note 806, April 1989.

Chapter 5

The Third Level Trigger System

5.1 Introduction

The third level trigger is designed to select good events after analysis of the data from the different subdetectors. The event reconstruction and analysis requires high computing power.

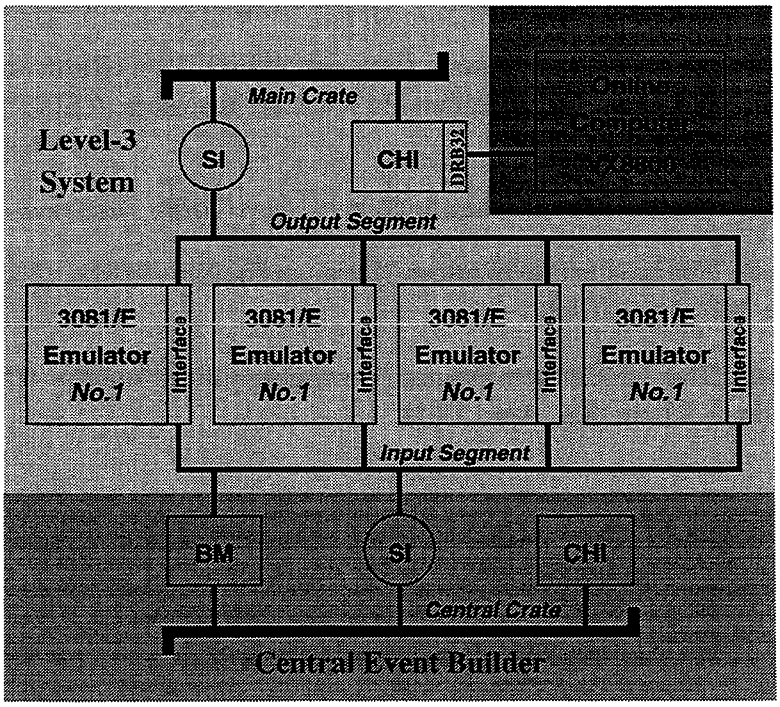


Figure 5.1: The old level-3 trigger system

The old system

Before 1993, the system was composed of four IBM 3081/E emulators [1] which worked in a **round robin** mode. The emulators provided the computing power (> 1 unit of IBM 370/68 each) to process the events assembled by the CEB. A dedicated interface was used by each emulator to allow its memory to be accessed from the two FASTBUS ports on the cable segments. One port on the ***input segment*** was used to receive data from the ***CEB***. Another port on the ***output segment*** was used to allow the data to be read from the emulator by the CHI on the ***main crate***. The readout CHI sent the good events to the ***producer*** on the main online VAX computer via the DRB32 interface. The system is shown in Figure 5.1.

This system had been used for four years since 1989. As the cost and the performance of the workstations become more and more attractive, a workstation based upgrade was made in 1993 to increase the computing power for the future LEP beam condition using industrially supported components.

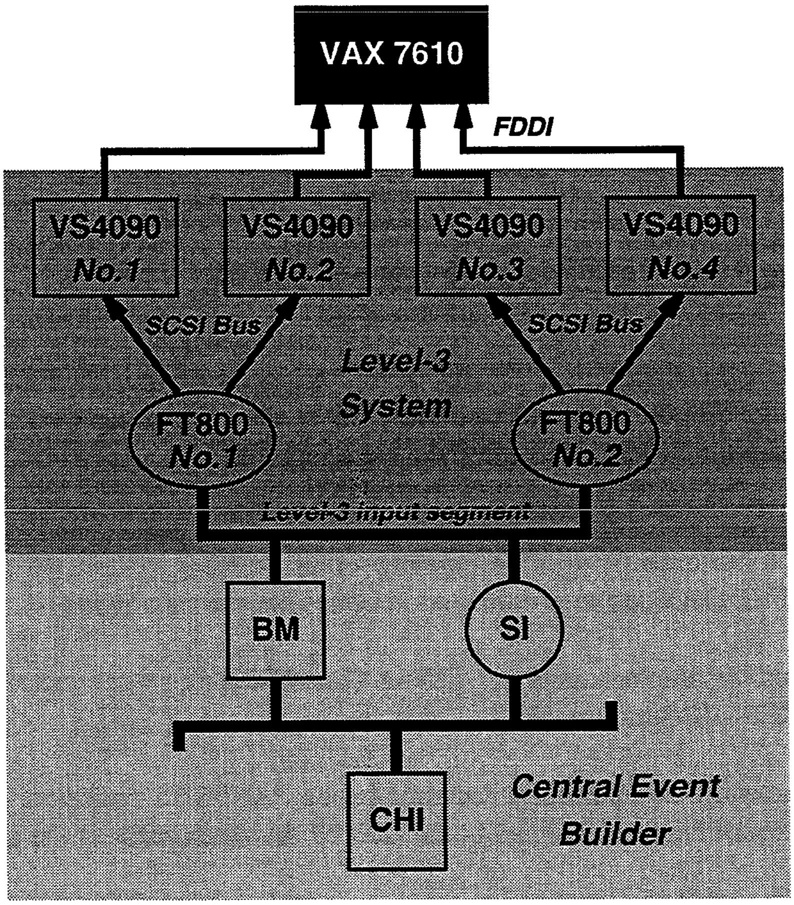


Figure 5.2: The new level-3 trigger system

The new system

The L3 online computing system is composed of a VAX cluster. The VAXstation 4000/90 was chosen as the processors in the system since it maintained system compatibility and also offered an upgrade path to the more powerful alpha VAX technology. The system consists of four such VAXstations with two dedicated interfaces which were designed by the collaboration of the L3 Rome group and the CERN ECP division. The interface is based on the INMOS transputer T800 network and connects the FASTBUS system of the ***CEB*** to the VAXstations. It is called FT800 (Fastbus to T800 transputer) interface [2]. Each FT800 distributes the data to two VAXstations via the SCSI [3] ports. The two FASTBUS ports of the interface provide the flexibility for both input and output data flow of the system. Currently the data are received by the cable port of the FT800 from the ***CEB*** and are sent to one of the free VAXstations by means of the transputer network and the SCSI interface. The main level-3 algorithm is running on the VAXstation to make the decision. The good events are shipped from the VAXstations to the***producer*** on the main VAX computer via the FDDI (Fibre Data Distributed Interface). The system is shown in Figure 5.2.

The following sections describe the hardware and the software of the new level-3 trigger system. Emphasis is on the interface part because the software for transputers and SCSI handler were developed by the author.

5.2 The FASTBUS Interface FT800

The main development of the new level-3 trigger system is the FT800 interface. The task of the FT800 is to interface the ***CEB*** to the computing workstations. The following main functions of this interface were required.

- It has to be embedded in the existing event building system which requires a slave FASTBUS cable port on the interfaces for the data input.
- It should be able to connect to different manufacturing workstations via a common SCSI I/O standard [3].
- For backward compatibility, it should have a FASTBUS crate port which allows the processed data to be read by the CHI on the ***main crate*** as the old system.
- Since workstations will continue to evolve during the lifetime of the experiment, it is required to have enough flexibility for further upgrades with a minimum of hardware and software investment.

The transputer [4] family of components from INMOS is designed for embedded controllers supporting parallel processing and reliable communication over serial links. A transputer network can be coupled to different bus standards through a variety of commercially supported interfaces and it is possible to accomodate almost any target platform with a minimum of effort. Therefore the T800 transputer was selected as the main component of the FT800 interface.

In the FT800, two FASTBUS slave couplers are implemented on both crate and cable segments under the control of two respective T800 transputers. Two SCSI TRAM's [5, 6] provide the interface to two VAXstations. A transputer network links the transputers for the FASTBUS couplers and the SCSI TRAM's to control the data flow. The block diagram of the FT800 interface is shown in Figure 5.3.

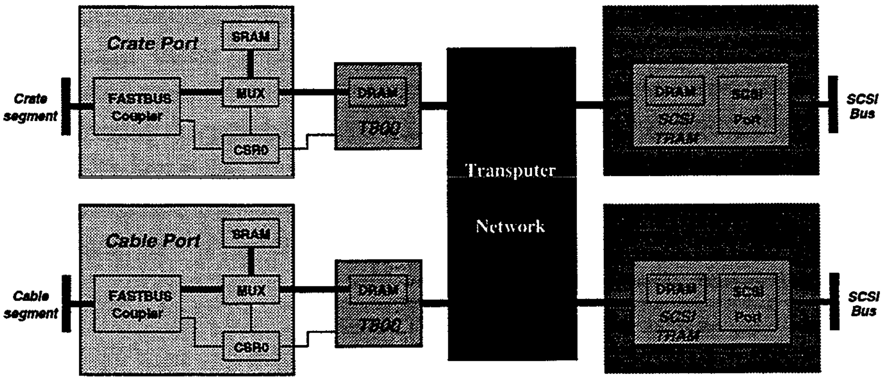


Figure 5.3: The block diagram of the FT800 interface

The FASTBUS coupler

The FASTBUS slave couplers of the FT800 are designed with the same features and philosophy as the existing interface for the 3081E emulators as seen from the software point of view. This was done deliberately to facilitate migration of the control algorithms from one system to the next.

Each FASTBUS port connects to a 512 Kbytes SRAM memory block which is shared with the dedicated T800 transputer. Since control flow is from FASTBUS to the embedded T800, it is only FASTBUS that controls the shared memory attribution and it is FASTBUS that informs the T800 if it's the current master of the SRAM. It is a well defined master slave relationship. All these control functions are implemented by the standard CSR0 register. The important control bits are listed in Table 5.1. All bits can be set and cleared by the FASTBUS master. The error flags, user code and DEVICE FULL/EMPTY bits can be set by the T800 transputer to handshake with the FASTBUS master.

Table 5.1: The CSR0 of the FT800 interface

Bits	Actions	Meaning
Bit 2	write	START which tells the T800 to start the data processing
	read	RUN which indicates if the T800 is running
Bit 6	write	MUX when TRUE it switches the SRAM to the T800
	read	indicate the multiplexer of SRAM
Bit 9	write/read	DEVICE FULL when the bit is set
Bit 10	write/read	DEVICE EMPTY when the bit is set
Bit 0, 7 and 8	read	The error flags from the T800
Bit 11, 12 and 13	write	The user code to the T800
	read	The user code from the T800

The transputer network

The INMOS transputer T800 [7] is a 32 bit CMOS microprocessor with a 64 bit on-chip floating point co-processor and four bidirectional serial links. The standard INMOS communication links support the communication between INMOS transputers by direct point-to-point connections. The network technology is used for the interconnection of the transputers. Each transputer is a node of the network and operates in parallel with other partners. The data exchange path between transputers relies on the design of the network. In the FT800, the transputer network was designed according to the following requirements.

- The input data stream comes from the FASTBUS cable port of the FT800. This allows to keep the same philosophy as the old system.

- The two VAXstations are connected to the SCSI TRAM's so that through the network it should be possible to alternatively dispatch the data from the transputer in the cable port to two SCSI TRAM's.

- After the event processing in the VAXstation, the output data stream were considered to have two possibilities. Firstly, the data can be sent back to the crate port of the FT800 to be read by the CHI in the ***main crate***. This allows the system to be operated in the same way as the existing system. Secondly, the data can be sent directly from the VAXstation to the main online VAX by using one of a wide choices of fast communication standards between computers. The network should be designed to operate the system in both modes.

- In addition to the data and control flow, the transputer network requires a bootstrap from either a TRAM with EPROM or an external computer. Since the system is still under development, the EPROM solution is not convenient for the system debugging. The INMOS B300 ethernet adapter can connect the transputer network to almost any kind of computers via ethernet and therefore was chosen to link the FT800 with its host, *i.e.*, one of the four VAXstations.

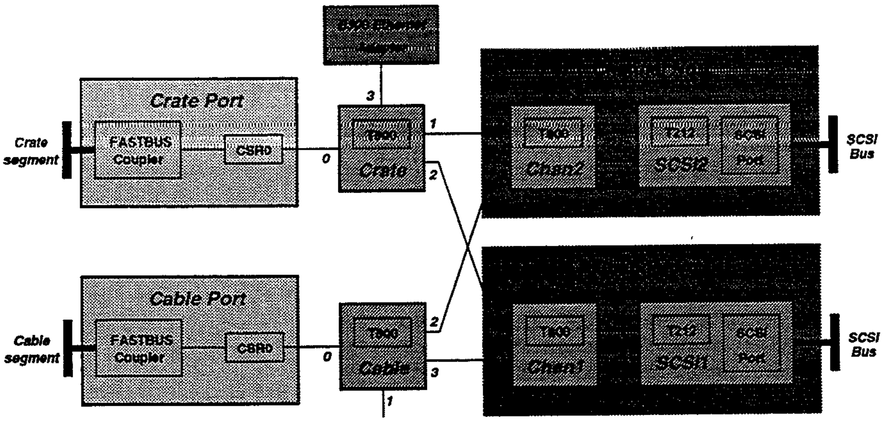


Figure 5.4: The transputer network configuration

The transputer network of the FT800 is shown in Figure 5.4. There are six transputer nodes on this network. The T800 transputers which are connected to the FASTBUS couplers are called respectively the ***cable*** node and the ***crate*** node. Besides these two nodes, a T800 and a SCSI TRAM in the upstream of the board provide the path to the VAXstation in the channel which is numbered as 1. Therefore the nodes are called the ***CHAN1*** and the ***SCSI1***. In the same way, the other two nodes in the downstream of the board are called the ***CHAN2*** and ***SCSI2*** for the VAXstation in the channel 2.

Each of ***CHAN1*** and ***CHAN2*** has a connection to both ***cable*** and ***crate*** nodes. Each of the ***SCSI*** nodes requires two links from the corresponding ***CHAN*** node to obtain maximum performance. The link 0 in both ***cable*** and ***crate*** nodes is used to connect the CSR0 of each port. The link 3 of the ***crate*** node is connected to the B300.

The SCSI TRAM is based on the T212 transputer. It supports the full functions of the SCSI standard. It can be either the initiator which controls all the operations or the target which is under control of the initiator. The SCSI software package is provided by INMOS to realize the functions on the SCSI bus. Since the VAXstation 4000/90 generic software only supports a single initiator of the SCSI bus, the SCSI TRAM has to work as a target.

5.3 Software Structure

The software structure in the level-3 trigger system consists of multiple stages from the data input at the FT800 up to the data output to the ***producer*** which is running on the main online VAX computer. It is shown in Figure 5.5.

The following functions are implemented by each stage of the software.

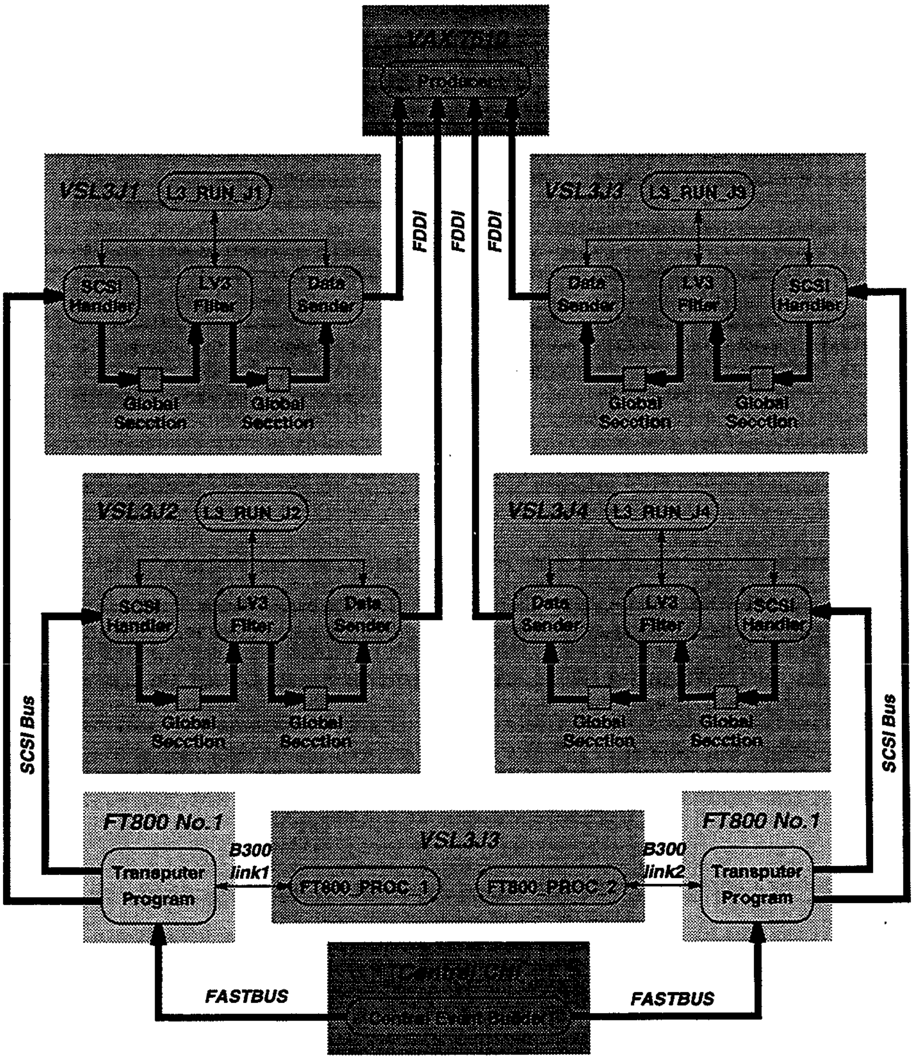


Figure 5.5: The structure of the level-3 trigger software

- The ***transputer software*** is running on the transputer network of the FT800 to receive data from the ***CEB***. The events are distributed to the VAXstations according to the configuration. The software on two FT800 interfaces are booted and controlled by two processes, the ***FT800_PROC_1*** and the ***FT800_PROC_2***, which are on one out of four VAXstations. The connection between the transputer programs and theses processes is established by the B300 ethernet adapter. The messages from the transputers are recorded in the log files.

- The ***SCSI handler*** process is running on the VAXstation to receive data from the interface via the SCSI bus. The data are sent to a block of shared memory which has been allocated by the writing process, which is the ***SCSI handler*** here, and can be read by another process. In this system, we call it a ***global section***. More than 20 events can be buffered in the ***global section***.

- The ***LV3 filter*** process is the FORTRAN language program for physical algorithm of the online reconstruction. The events are read from the ***global section*** of the ***SCSI handler***. After the processing, the bad events are discarded and the good events are written into another ***global section*** which has been allocated by the ***LV3 filter***.
- On request of the ***producer***, a ***data sender*** process is running on the VAXstation. It reads the processed events from the ***global section*** of the ***LV3 filter*** and sends the events to the ***producer*** on the main online VAX via the FDDI interface.
- In the ***producer***, the channels are assigned to the respective FDDI ports to all VAXstations. The ***producer*** requires the data from the ***data senders*** via the dedicated channel. After the data has been successfully received, another event is required.
- The parent processes, ***L3_RUN_J1, L3_RUN_J2, L3_RUN_J3*** and ***L3_RUN_J4***, run on the VAXstations and distribute the ***L3 run control*** command to the child processes, the ***SCSI handler***, the ***LV3 filter*** and the ***data sender***.

5.4 The Transputer Software

OCCAM2 [8, 9] is the assembly language of transputers. It supports parallel as well as sequential execution and provides automatic communication and synchronization between concurrent processes. It can easily describe concurrent algorithms and as such was chosen for the transputer software on the FT800.

The transputers are designed to allow the multiple processes with different priorities. The processes can be placed either on one transputer as shown in case (a) in Figure 5.6 or on different transputers as in case (b) in Figure 5.6. The unidirectional communication channels between processes can be assigned either to physical links in the case (b), or as arguments through transputer memory in the case (a). Only two channels in the opposite directions are allowed over one physical link. There is no limit for number of channels between the processes on the same transputer.

The operation modes

In order to have enough flexibility, the software for the FT800 transputer network is designed to be able to operate in the following three modes.

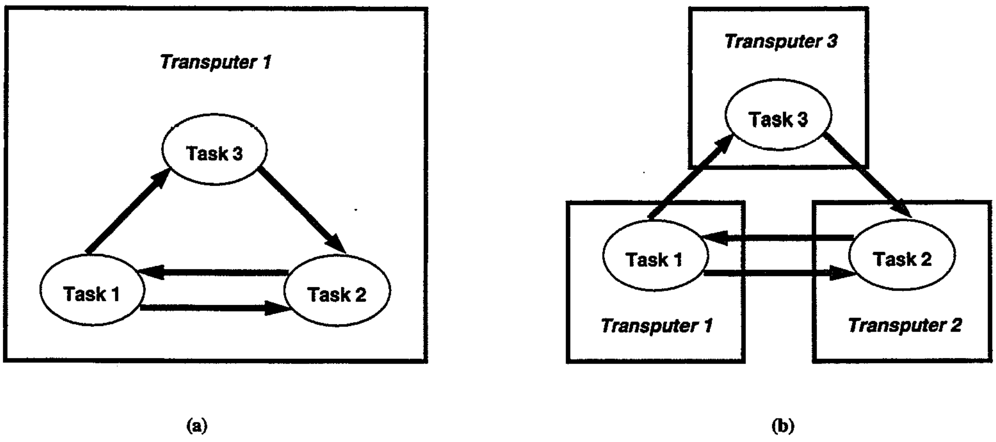


Figure 5.6: The multiple tasks on the transputer network

1. **Cable – crate mode**
   This is a test mode to pass data from the ***cable*** node to the ***crate*** node without sending it to the VAXstations via the SCSI bus. The data is sent to the ***crate*** node via both the ***CHAN1*** and ***CHAN2***. The CHI in the ***main crate*** reads out the data from the SRAM in the FASTBUS crate segment. This mode allows to check the functionality of the transputers without the SCSI path.

2. **Cable – VS mode**
   This mode is currently employed. The data from the ***cable*** node is distributed either through the ***CHAN1*** and ***SCSI1*** to one VAXstation or through the ***CHAN2*** and ***SCSI2*** to the other VAXstation. The channels can be enabled or disabled from the ***L3 run control***. The channel mask is set by the FASTBUS master during the initialization.

3. **Cable – VS – crate mode**
   The data is distributed to the VAXstations in the same way as the second mode. After the processing, the VAXstations send the processed data back to the ***crate*** node via the ***SCSI1*** or ***SCSI2*** respectively. The CHI in the main crate is used to readout the data and interface to the ***producer***.

The operation mode can be set by the event building software configuration tool. The selected mode is passed to the transputer software via the SRAM in the cable node during the initialization.

The processes on the transputers

On the SCSI TRAM, the INMOS software package [10] is used so that there is no user program on it. The INMOS program is placed on both SCSI1 and SCSI2 and requires one or two physical links from the corresponding CHAN1 or CHAN2 transputer. Two links are used in the FT800 to speed up the data transfer [10]. In this case, the data and control channels are separated.

Due to the symmetry of the system, the ***CHAN1*** and ***CHAN2*** use the same program which is called ***SCSI server***. On the other two nodes, the main processes are called ***cable*** process and the ***crate*** process according to the node names. The software for the FT800 network is illustrated in Figure 5.7.

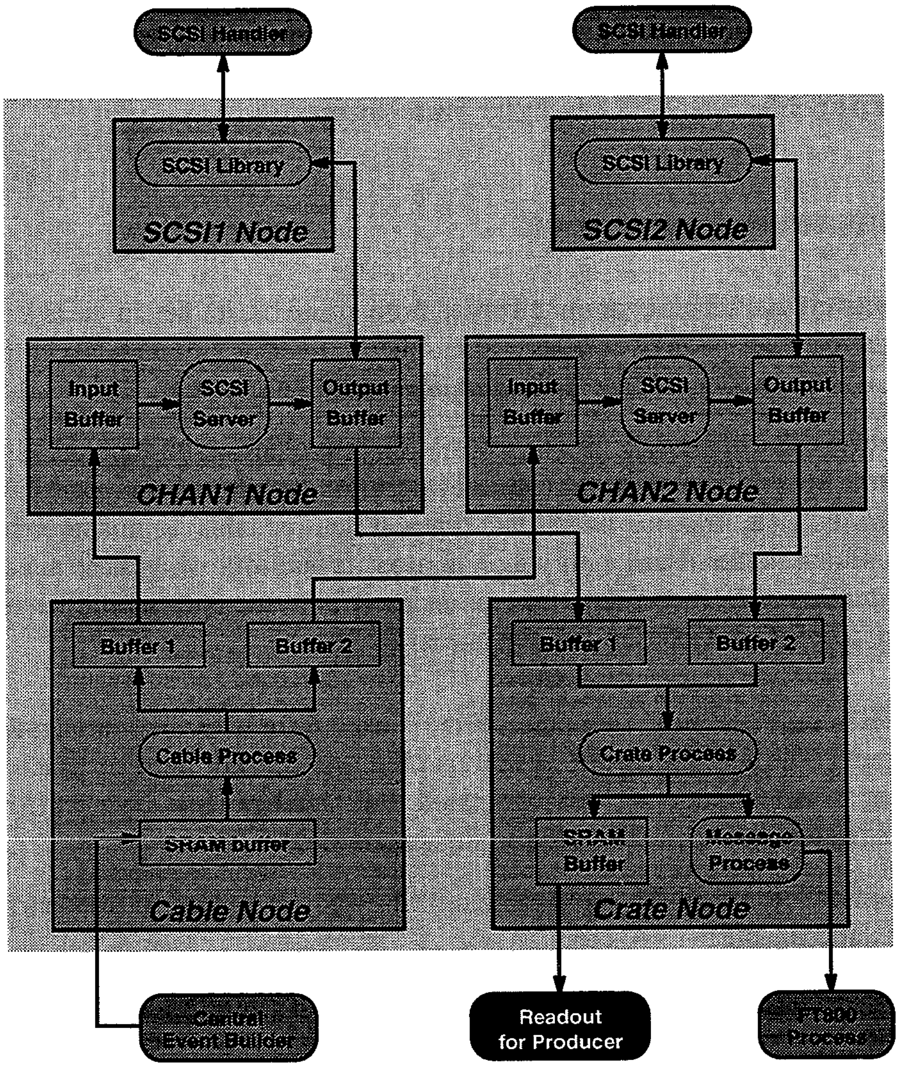


Figure 5.7: The FT800 software structure

In order to notify the running states on each process to the operators, the messages are needed during the running. The messages are sent to the host computer by means of the B300. Therefore a ***message*** process on the ***crate*** transputer is used to communicate with the B300.

Traffic on the FT800 consists of events flowing according to the selected mode and control messages flowing through the ***cable*** process and the ***SCSI servers*** to the ***crate process***. The data and control flows are mixed together and are sent through both channels alternatively since there is only one physical link between the ***cable*** process and each ***SCSI server*** as well as between the ***crate*** process and each ***SCSI server***. By way of exception, the control flow goes through the unused channel if only one channel is selected for the data flow. This is checked by the program automatically.

In each user transputer node, two buffers are provided. In the ***cable*** and ***crate*** processes, the buffers are used for the two channels respectively. In the ***SCSI server***, one buffer is dedicated for the input data from the ***cable*** process and another for the output data to either the SCSI bus or the ***crate*** process.

The communication protocol

There are two kind of communication protocols to be defined in the system. One is between the transputer processes and another is over SCSI bus between the ***SCSI server*** on the transputer and the ***SCSI handler*** on the VAXstation.

Between the transputer processes, the main problem is to distinguish the data and control flows. Therefore the data blocks to be transferred are headed with two control words. The first one is the data type. It can be data, command or message. The second one is the data length which is used for the following data transfer. The receiver process checks the data type to decide where the data should be sent. The following three commands are defined currently:

- **INIT** command with two parameters is used to transfer the initialization messages. The first parameter is the operation mode and the other is the channel mask. After receiving this command, the target process changes to the new parameters and then delivers it to the next nodes.
- **Request** command has no parameter followed. It is used for the handshake of the data transfer.
- **Ready** command without parameter is to reply the **request** command.

Due to the oriented data flow, the data source and target are fixed. Between the ***cable*** process and the ***SCSI servers*** the data source is the ***cable*** process and the targets are the ***SCSI servers***. Between the ***SCSI servers*** and the ***crate*** process, the sources are the ***SCSI servers*** and the target is the ***crate*** process. It is assumed that there is no data sending back from target to source except the defined handshake reply. The following handshake protocol is defined for the communications between the transputer processes.

1. The target should be always able to receive the control flow (commands and messages) except when it is waiting for a data block for which the target has been required.
2. Whenever the source process has data to be transferred, a request command needs to be sent to the target process. After that, the source process have to wait for the reply from the target.
3. After receiving a request from the source process, the target replies a ready command only when it is ready to receive the data from the source. Afterwards the target has to wait for the following data transfer.
4. After receiving the ready command, the source sends the data block to the target. There is no feedback message from the target when the transfer is completed.

Over the SCSI bus, the SCSI commands has already been defined (see [3]). Two types of SCSI command are composed of either 6 words or 10 words depending on the data length to be transferred. The 6 words command uses one byte to describe the data length and the 10 words command uses four bytes to express the data length.

In this system, the ***SCSI server*** is operated as a RAM disk. The ***SCSI handler*** on VAXstation is the master which sends the SCSI command to control what kind of data is needed and reads the data from the ***SCSI server*** on the transputer via the SCSI TRAM. The following SCSI commands are used in the system.

- **Inquiry 6** is a 6 words command. It is used to test the communication between the VAXstation and the transputers. The ***SCSI server*** gives the inquiry data to the ***SCSI handler*** after receiving this command. The data contains the device information which can be checked by the ***SCSI handler***.
- **Read 6** is a 6 words command. The ***SCSI handler*** reads the event length by means of this command. The ***SCSI server*** either sends the event length to be transferred when the data is ready in the output buffer, or gives a zero if no data is ready.
- **Read 10** is a 10 words command which is used to transfer the events. The maximum data size is 8 Kbytes per block which is limited by the SCSI TRAM. If the event size is larger than the limit, the ***SCSI handler*** has to divide the event into several blocks and to combine them again after all blocks have been received.
- **Write 6** is a 6 words command which is used only in the Cable – VS – Crate mode. The ***SCSI handler*** informs the ***SCSI server*** to be ready for receiving the processed event by using this command. The data contains the number of blocks to be transferred.

- **Write 10** is a 10 words command which is also used for the Cable – VS – Crate mode to send back the processed event to the transputer.

The processing procedures

The Cable – VS mode is currently used in the system. In this mode, the procedures of all user transputers are described in the following.

The cable process

The cable process has three tasks. Firstly, it obtains the operation mode and the channel mask which are passed by the CHI and sends them to other user processes. Secondly, it has to handshake with the ***CEB*** to get the events into the SRAM. Finally, according to the channel mask it ships the event from the SRAM to a selected ***SCSI server*** via the corresponding buffer in the DRAM.

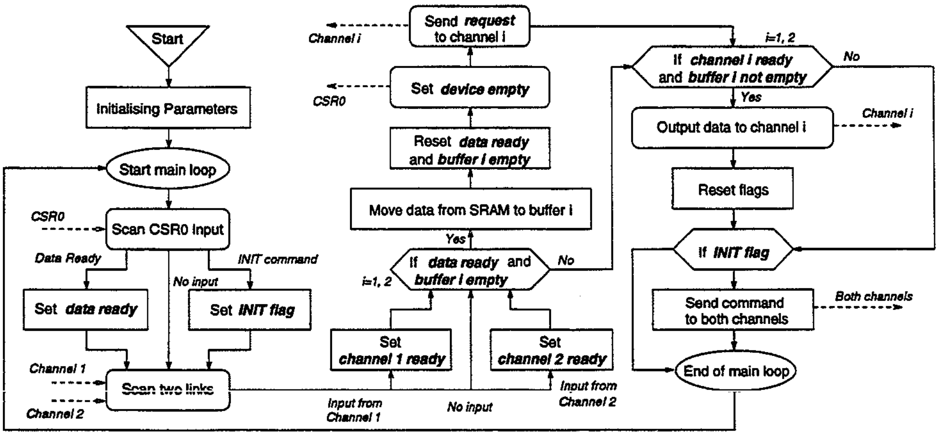


Figure 5.8: The flow chart of the cable process

Several global flags are used to avoid the process waiting for a certain input channel. The process scans the input channels with a very short timeout by using the priority technology of the transputer. If one channel receives an input signal, the communication is established. After the input is finished, a corresponding flag is set to allow the process proceeding the data afterwards. Otherwise, the input is skipped. The flow chart of the cable process is shown in Figure 5.8.

The SCSI server process

One of the tasks of the ***SCSI server*** is to service the SCSI commands from the VAXstation. In the current working mode, the SCSI commands, **inquiry 6**, **read 6** and **read 10**, are used. But SCSI commands from the VAXstation always requires a response to prevent a timeout. Therefore the ***SCSI server*** waits for the SCSI commands and processes other tasks between two SCSI commands.

The other tasks of the ***SCSI server*** are to receive the data from the ***cable process***. According to the data type, the messages are sent to the ***crate*** process and the events are sent to the VAXstation via the SCSI bus. The global flags are used in the same philosophy as the ***cable*** process. The flow chart of the ***SCSI server*** is illustrated in Figure 5.9.

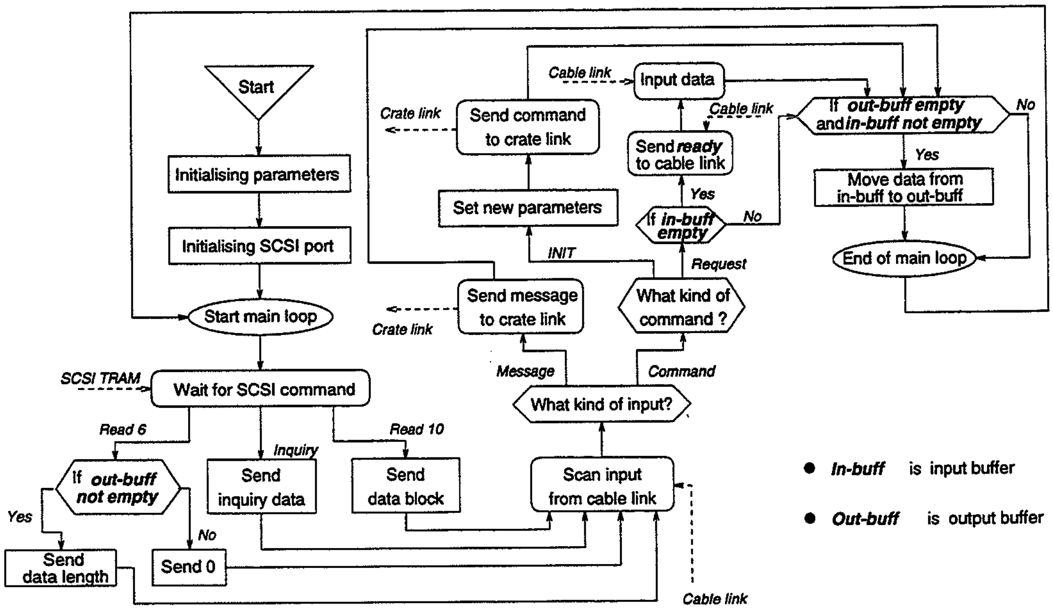


Figure 5.9: The flow chart of the SCSI server

The crate process

The ***crate*** process is working as a multiplexer in this operation mode. It scans two input channels from both ***SCSI servers*** and sends the messages to the ***message*** process.

The message process

The task of the ***message*** process is to communicate with the host computer. The messages received from the ***crate*** process are sent to the host via B300. The necessary handshake is completed by it.

The diagnostic tools

In order to trace the system failures, some diagnostic tools were developed. The main difficulty is to find why and in which transputer the program fails in a transputer network. Therefore a ***dump*** program is provided to dump the necessary information from each transputer.

In the software design, each process puts the program state and the important parameters in a fixed place in the transputer memory. The ***dump*** program reads this information as well as the first 10 words in each data buffer. Analysis of the information requires expert knowledge of hardware and software of the system.

A ***health*** test tool for the FT800 interface is under development. The tool makes the general check of each part of the FT800 interface which can be performed by any operators. It includes the following steps.

1. Test the SRAM of the FASTBUS port by writing data to the SRAM, reading back and then checking the data.
2. Test the CSR0 function between the CHI and the transputers to check the communications.
3. Test the transputer memories.
4. Test the transputer links by checking the data after the transfer over the links.
5. Test the SCSI interface between ***CHAN*** nodes and the VAXstations.

5.5 The SCSI Handler

The ***SCSI handler*** is running on each VAXstation to read the event from the interface and send it to the ***LV3 filter*** process via the ***global section***. The SCSI commands are realized by the standard QIO function which is provided by DEC to make the input/output operations over the device on the VAX computer.

A generic driver is loaded at the computer starting time to operate the SCSI device in a standard way. If the device is not booted correctly, the QIO operation can not be performed.

The program is designed for the Cable – VS mode only. Therefore only **inquiry 6**, **read 6** and **read 10** are used. Three flags can be changed when each time the initialization takes place. They are used to enable or disable the debugging, global section and the data check. The flags are loaded from a description file and the data check flag can be set in the software configuration tool for the event builders. It provides the flexibility to debug the system. The flow chart of this program is given in Figure 5.10.

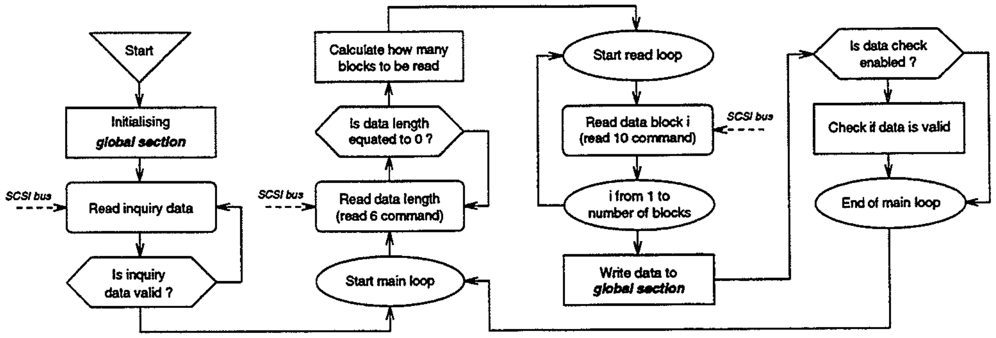


Figure 5.10: The flow chart of the SCSI handler

5.6 System Performance

The system was set up at the beginning of 1993. The failure rate is very low (less than 0.5% during LEP in physics condition). The most of faults are caused by the software on the VAX. The throughput of the system has been tested under different conditions, the result [2] is listed in Table 5.2.

Table 5.2: The throughput of the new level-3 trigger system

Running Condition	Throughput
FASTBUS to cable SRAM	6.6 Mbyte/sec
Copy from SRAM to DRAM	4.0 Mbyte/sec
From cable to SCSI server (link speed)	1.7 Mbyte/sec
From FASTBUS to one SCSI handler on VAXstation	0.67 Mbyte/sec
LV3 filter processing rate	0.9 Mbyte/sec
The data sender to the producer (FDDI speed)	1.3 Mbyte/sec
The whole system with one FT800 and one VAXstation	0.67 Mbyte/sec
The whole system with one FT800 and two VAXstations	1.25 Mbyte/sec
The whole system with two FT800's and four VAXstations	2.4 Mbyte/sec

The throughput of the system with one FT800 and two VAXstations is already enough to manage the current data taking rate. To run with higher rate, the improvements in both SCSI TRAM and the SCSI interface of the VAXstation are needed. The incoming T9000 transputer technology will allow a higher link speed and processing capability. It is already considered as the option for the future changes of the FT800 interface.

Reference

[1] P.F.Kunz *et al.*. **The 3081/E Processor**. European Organization for Nuclear Research (CERN), CERN/DD 84-04, 1984.

[2] B.Martin, S.Falciano, C.Luci, L.Luminari, F.Marzano, G.Medici, and X.Cai. **FT800 : A FASTBUS to Transputer Interface**. In *RT93 Conference Record of the Eighth Conference on Real-Time Computer Applications in Nuclear, Particle and Plasma Physics*, 1993.

[3] *American National Standard for Information System — Small Computer System Interface (SCSI)*. ANSI X3.131-1986, June 1986.

[4] INMOS Limited. *The Transputer Databook (2nd Edition)*, 1989.

[5] INMOS Limited. *Dual-In Line Transputer Modules (TRAMs)*. INMOS Technical Note 29.

[6] INMOS Limited. *IMS B422 SCSITRAM iq systems Databook*, 1991.

[7] INMOS Limited. *Engineering Data — IMS T800 Transputer*, April 1987.

[8] INMOS Limited. *OCCAM2 Toolset User Manual*, March 1991.

[9] INMOS limited. **OCCAM 2 Reference Manual**. Computer Science. Prentice Hall International, 1988.

[10] INMOS Limited. *IMS F002A SCSI OCCAM Library*, November 1990.

Conclusions and Prospects

The L3 trigger and data acquisition system is implemented in CAMAC, VME and FASTBUS standards. It was set up in 1989 and was ready to handle the first LEP events.

Three parts of the L3 trigger and DAQ system are described in detail in this thesis. The author has made contributions and therefore is a specialist on these parts. Since 1989, upgrades were made in the event building and third level trigger systems. The HCAL readout system was running well so that there is no upgrade expected.

The further upgrade of the event building system will be related to the event buffers and the event builder masters, the CHI modules. The new event buffer should have a fast access speed and support the sparse data scan, a standard FASTBUS broadcast function. The master should have a faster and more reliable way of transferring data between the two event building stages as well as between the central event builder and the third level trigger system.

A future upgrade of the level-3 trigger system may be based on the more powerful Alpha VAX technology. The Fastbus to T800 transputer interface will be upgraded by using the new transputer T9000. Also the communication over the SCSI bus has to be improved.

In addition, the author will be involved in the next major upgrade in the L3 trigger and data acquisition system which is relative to the second level trigger system. The trigger data memories (MMB modules) will be replaced by a transputer controlled event buffer. A transputer T9000 based event farm is under development in order to replace the current XOP processors. A FASTBUS interface similar to the FT800 will provide a slave port for reading the processed events by the central event builder. The upgrade is expected to be done during the 1994 winter shutdown.

Appendix A

The Analog Sum Card

To prepare the signals for the ***L3 calorimeter trigger***, each HCAL barrel module is divided into two part [1, 2, 3, 4]. The inner part corresponds to the chambers in the first interaction length. It is called layer A. The outer part is called layer B. The first module from each side has only the layer B because the angles are covered by the layer A from other modules (include the HCAL end caps). Figure A.1 illustrates the HCAL layer structure.

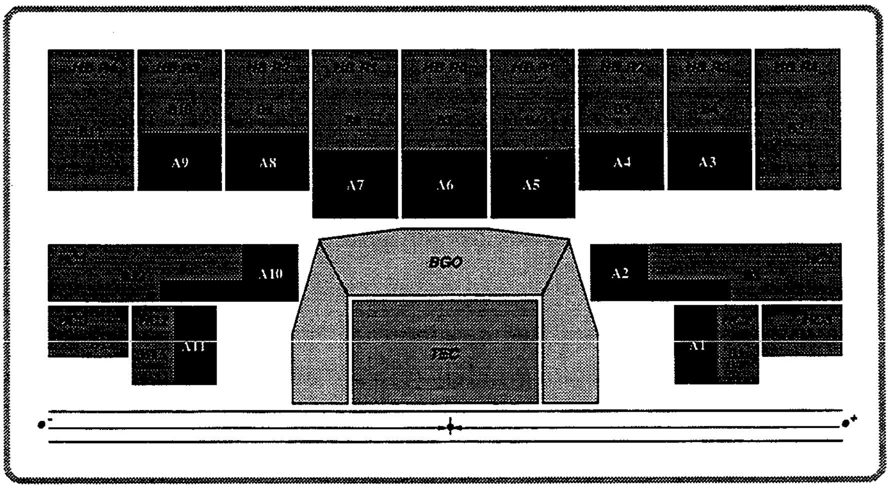
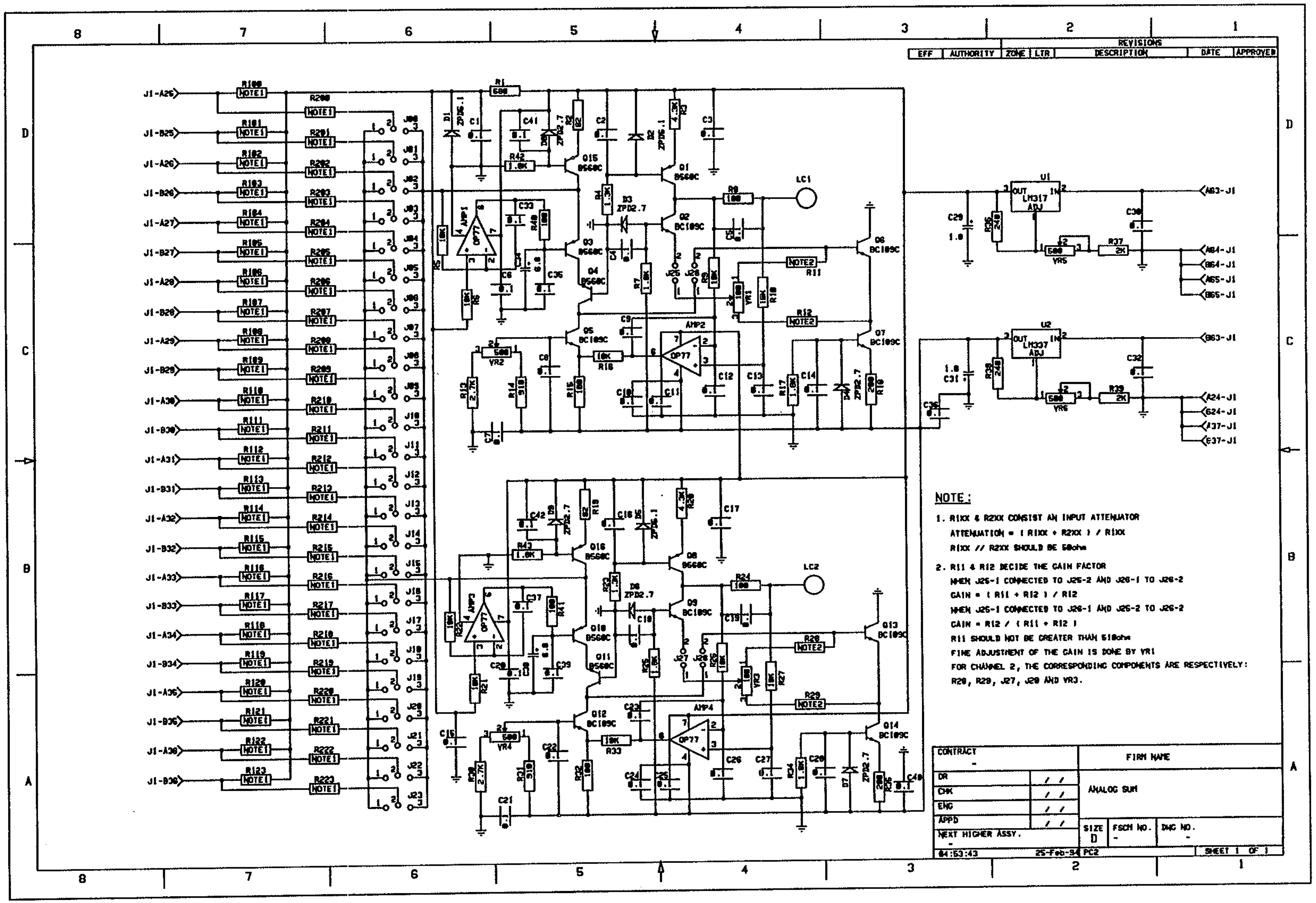


Figure A.1: The HCAL layers definded by the calorimeter trigger

Two signals are needed by the ***L3 energy trigger*** from each HCAL module [1]. One comes from the layer A and the other comes from the layer B. Since the signals from each module are processed by two ADC modules, an analog sum card [5] was designed to combine the 48 current sum signals from two ADC's into two analog signals. The main board is connected to one ADC. A child board which contains only the input attenuation network and the jumpers is connected to another ADC and sends



NOTE:

1. R1XX & R2XX CONSIST AN INPUT ATTENUATOR
ATTENUATION = (R1XX + R2XX) / R1XX
R1XX // R2XX SHOULD BE 50ohm

2. R11 & R12 DECIDE THE GAIN FACTOR
WHEN J25-1 CONNECTED TO J25-2 AND J26-1 TO J26-2
GAIN = (R11 + R12) / R12
WHEN J25-1 CONNECTED TO J26-1 AND J25-2 TO J26-2
GAIN = R12 / (R11 + R12)
R11 SHOULD NOT BE GREATER THAN 510ohm
FINE ADJUSTMENT OF THE GAIN IS DONE BY VR1
FOR CHANNEL 2, THE CORRESPONDING COMPONENTS ARE RESPECTIVELY:
R28, R29, J27, J28 AND VR3.

Figure A.2: The circuit diagram of the analog sum card

the signals to the main board. The current sum signals are declared into the auxiliary connector of the ADC module. The circuit diagram of the main board is shown in Figure A.2.

Reference

[1] L3 Collaboration, B.Adeva, *et al.*. **The Construction of the L3 Experiment**. *Nuclear Instruments and Methods in Physics Research A **289***, pages 35–102, 1990.

[2] R.Bizzarri, F.Cesaroni, S.Gentile, G.Lunadei, M.Fukushima, G.Herten, and T.Hebbeker. **The L3 Energy Trigger**. *Nuclear Instruments and Methods in Physics Research A **283***, pages 799–802, 1989.

[3] R.Bizzarri, F.Cesaroni, S.Gentile, G.Lunadei, M.Fukushima, and T.Hebbeker. **The First Level Energy Trigger of L3 Experiment: Description of the Hardware**. *Nuclear Instruments and Methods in Physics Research A **317***, pages 463–473, 1992.

[4] L3 Collaboration. **Hadron Calorimetry in the L3 Detector**. *Nuclear Instruments and Methods in Physics Research A **302***, pages 53–62, 1991.

[5] S.X.Wu **The Memorandum of the Analog Sum Card for Hadron Calorimeter.**. L3 Collaboration, L3 Internal Node, 1988.

Appendix B

The Interfaces of the SMI Control Bus

Two interfaces [1] are used to convert the differential TTL signals of the ***SMI control bus*** to the standard TTL signals on the connectors of LSR1892 memory [2] and each SMI [3]. The connectors for both LRS1892 and SMI are fully compatible. The pin map [3] is shown in Figure B.1.

control data bit 0	1	2	*control data bit* 1
control data bit 2	3	4	*control data bit* 3
control data bit 4	5	6	*control data bit* 5
control data bit 6	7	8	*control data bit* 7
control data bit 8	9	10	*control data bit* 9
control data bit 10	11	12	*control data bit* 11
control data bit 12	13	14	*control data bit* 13
control data bit 14	15	16	*control data bit* 15
$\overline{\textit{acknowledge}}$	17	18	*register address A*1
*register address A*2	19	20	*register address A*4
register address inhibit	21	22	$\overline{\textit{write}}$
$\overline{\textit{read}}$	23	24	$\overline{\textit{take control}}$
*strobe S*1	25	26	*ground*
ground	27	28	*ground*
*module select SEL*0	29	30	*module select SEL*1
*module select SEL*2	31	32	*module select SEL*3
$\overline{\textit{bypass}}$	33	34	*ground*

Figure B.1: The pin map of SMI 1821 I/O connector

The differences of the two interfaces, the SMI/MEM card (for the SMI end) and the MEM/SMI card (for the LRS1892 end), are due to the different direction of the control flow. The data lines are bidirectional signals. The control signals are mostly from the master (the LRS1892) to the slave (the SMI) except the ***acknowledge*** signal which is in opposite direction. Figure B.2 and Figure B.3 give the schematic diagrams of the respective the SMI/MEM card and the MEM/SMI card.

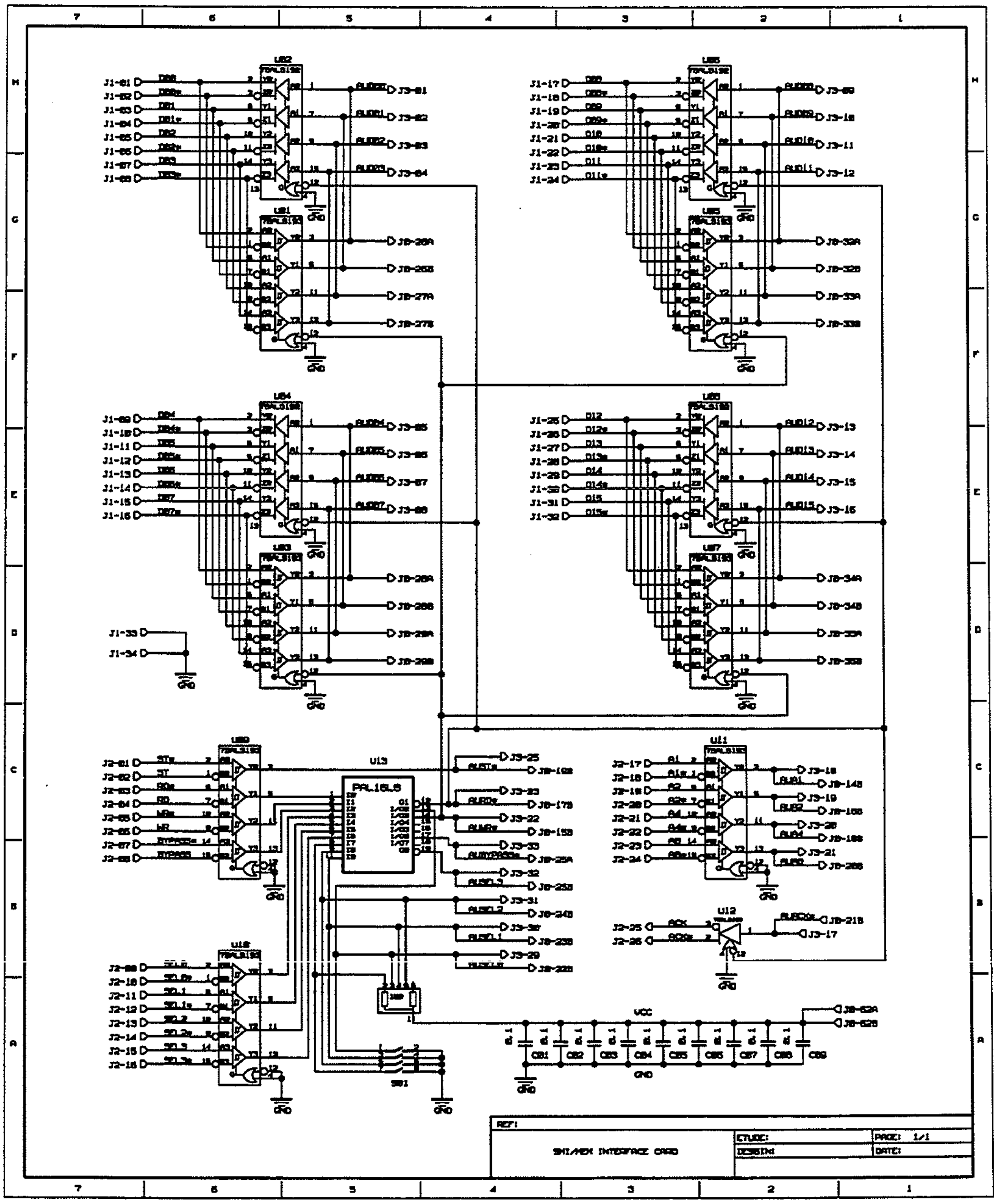


Figure B.2: The schematic diagram of the SMI/MEM card

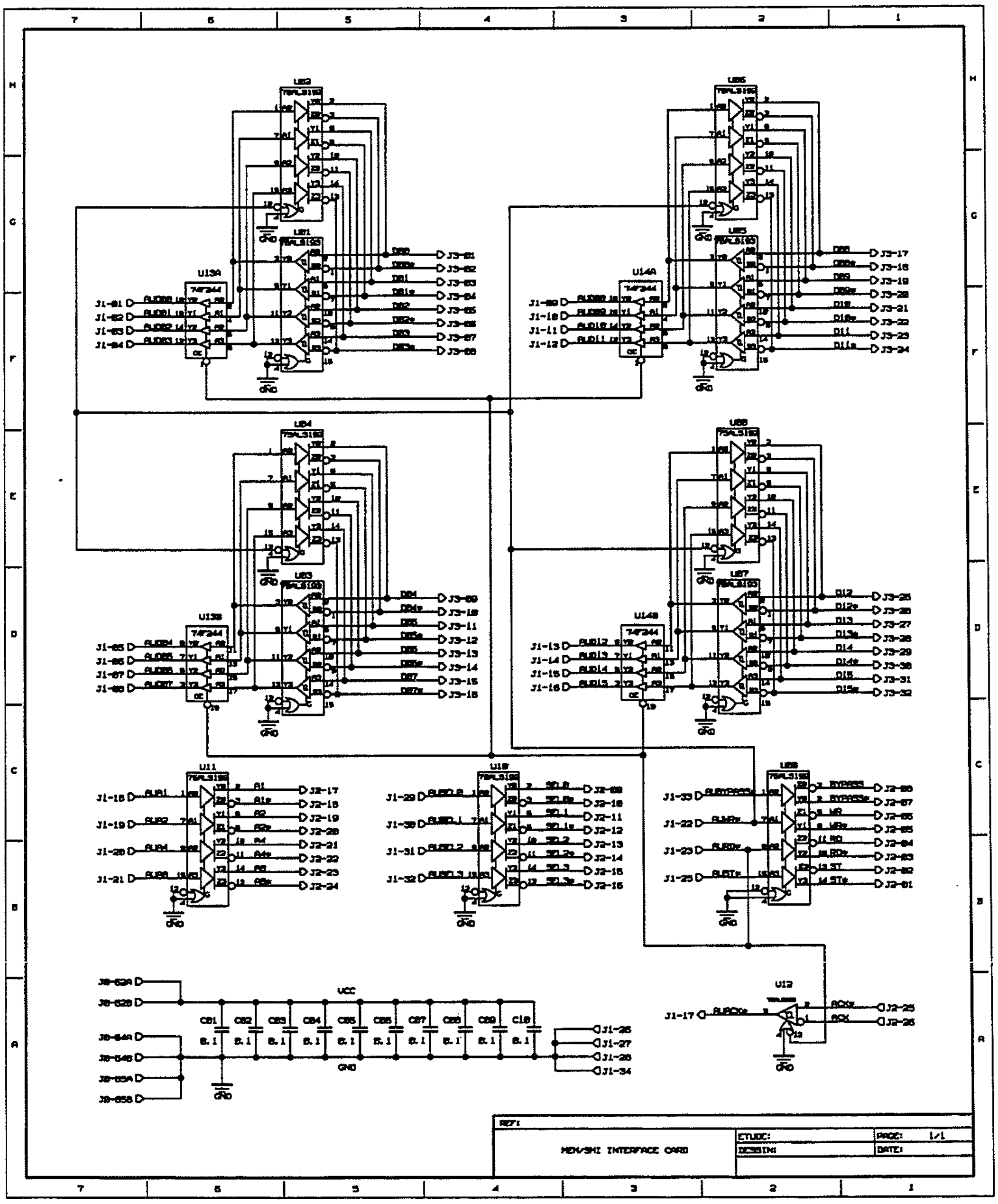


Figure B.3: The schematic diagram of the MEM/SMI card

Reference

[1] D.N.Mao, V.K.Gupta and M.M.Ilyas. **The Design of the SMI 1821 Interface.** FBLJA/L3, World Laboratory. FBLJA Internal Report, May 1989.

[2] LeCroy Corporation. **1892 Users Manual**, April 1987.

[3] LeCroy Corporation. **1821 User's Manual**, March 1987.

Acknowledgements

I would like firstly to thank my thesis supervisor, Prof. J.J.Blaising for not only guiding me to finish my thesis but also for his great help towards my work and life here. Also the French introduction and the French version of the first chapter are written by him.

I am very grateful to Prof. S.C.C.Ting and Prof. A.Zichichi for providing the opportunity for me to work in the ***L3 experiment*** as a fellow of the FBLJA project.

I thank especially Dr. M.M.Ilyas, my instructor in the FBLJA project. I leant a great deal about the FASTBUS and the HCAL system from him. I also thank Dr. M.Fukushima and Dr. D.Linnhöfer for their enthusiastic help when they were here.

Special thanks should go to Dr. M.Capell for his suggestions and corrections to my thesis.

I would also like to acknowledge Prof. C.Dionisi, Prof. S.Falciano, Dr. C.Luci and B.Martin for the memorable collaboration on the third level trigger system and the very helpful suggestions on my thesis writing.

Many thanks to Mr. G.A.Hu and Mr. S.X.Wu for their software and hardware support. Also I want to thank Dr. H.S.Chen, Prof. C.Y.Chien, Dr. T.S.Dai, Prof. Y.Galaktionov, A.Klimentov, Dr. L.Niessen, Dr. Y.J.Pei, Prof. J.M.Qian, G. Schwering, Dr. Y.F.Wang, Dr. C.G.Yang, Prof. B.Zhou, Dr. J.F.Zhou for their help and advice on my work.

Finally, I would like to acknowledge my wife Hongyi and my son Yibo for their constant love and support. Also I want to thank my parents for their encouragement.