LAPP-T-96/02

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

THÈSE de DOCTORAT

présentée pour obtenir le titre de

Docteur en Sciences
Spécialité : Physique Experimentale & Instrumentation

par

Vincent CAPONY

Etude et développement du transfert des données pour l'expérience NA50 : le réseau de transputers du détecteur de multiplicité.

Soutenue le 12 juillet 1996 devant la Commission d'Examen

L. Foulloy
C. Baglin
G. Juanole
L. Ramello
S. Cittolin

Table des matières

Chapitre 1

ELEMENTS DE LA PHYSIQUE DU PLASMA DE QUARKS ET DE GLUONS

1.1 Introduction

La matière dite *ordinaire*, est un assemblage stable de hadrons. Le terme de hadron désigne deux types de particules : les baryons (nucléons) et les mésons (pions, kaons) constitués respectivement de 3 et 2 quarks. La cohésion des quarks, qui forment un hadron, est assurée par des échanges de gluons : quarks et gluons sont confinés en hadrons. Cet agencement est décrit par le modèle des quarks, et la théorie de la ChromoDynamique Quantique(QCD). Le modèle des sacs est un moyen simple pour se représenter ces assemblages : chaque hadron est assimilé à un sac renfermant 2 ou 3 quarks. La matière hadronique est donc un assemblage de sacs. La densité moyenne d'une telle matière est de l'ordre de 0.15Gev/fm^3.
La QCD prévoit cependant que sous des conditions extrêmes de température ou de densité d'énergie, cette matière confinée peut subir une transition vers un nouvel état : le Plasma de Quarks et de Gluons(QGP). Dans une telle phase, les quarks et gluons évolueraient librement pour former un plasma : c'est le déconfinement de la matière haronique, on a interpénétration des sacs de quarks.

Lorsque la température et la densité de cette matière hadronique sont suffisantes, des calculs de QCD sur réseau font apparaître l'existence d'une transition de phase qui conduit au déconfinement de la matière [1].
La brusque variation de la densité d'énergie ϵ du système est une des manifestations de cette transition. Si l'on modélise les deux phases séparées par cette transition (gaz de hadrons et plasma), par des gaz parfaits (pas d'interaction entre les particules), les grandeurs thermodynamiques comme la densité d'énergie ϵ et la pression P peuvent s'écrire sous la forme [2] :

$$\epsilon = 3\frac{\pi^2}{30}T^4 \tag{1.1}$$

$$P = \frac{\epsilon}{3} = 3\frac{\pi^2}{90}T^4 \tag{1.2}$$

pour la phase hadronique et,

$$\epsilon = 37\frac{\pi^2}{30}T^4 + B \tag{1.3}$$

$$P = 37\frac{\pi^2}{90}T^4 - B \tag{1.4}$$

pour le plasma.

Dans ces expresssions, B correspond à la pression qu'exerce le vide sur le plasma. Il s'agit de la constante de sac.
La figure 1.1 représente le diagramme des phases de la matière nucléaire. On peut noter que la matière, sous forme de noyaux, se situe à température nulle et à densité nucléaire $\rho = \rho_o = 0.17GeV/fm^3$.

La température critique T_c, au delà de laquelle il semble possible de changer de phase à densité quasi nulle dépend fortement des hypothèses de départ, et notament du nombre de saveurs de quarks N_f considérées dans le système [3]. La figure 1.2 nous montre [3] que pour $N_f = 2$ (présence des quarks u et d), on obtient une température critique T_c de l'ordre de 150 MeV.

La transition de phase qui apparait pour cette température $T_c = 150\ MeV$ est illustrée par la figure 1.3 et nous montre clairement les brusques variations de la densité d'énergie ϵ et de la pression P autour de cette température critique. Cet exemple représente une transition de phase du second ordre correspondant à une densité d'énergie de 1 à $2.5GeV/fm^3$ [4].

1.2 Le déconfinement de la matière

Le phénomène de déconfinement de la matière hadronique repose sur le comportement du potentiel de confinement en fonction de la température.
Lorsque la température est nulle, la forme la plus générale [5] du potentiel d'interaction entre deux quarks séparés d'une distance r peut se réduire [5] à :

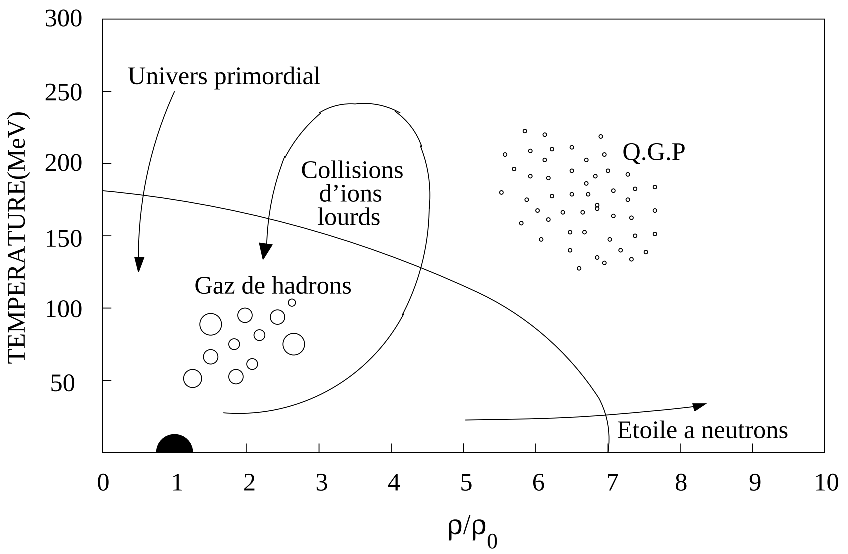


FIG. 1.1 - *Ce diagramme des phases de la matière nucléaire, nous montre la "frontière" entre les deux état : QGP et gaz de hadrons. La matière ordinaire correspond à* $T = 0$ *et* $\frac{\rho}{\rho_o} = 1$

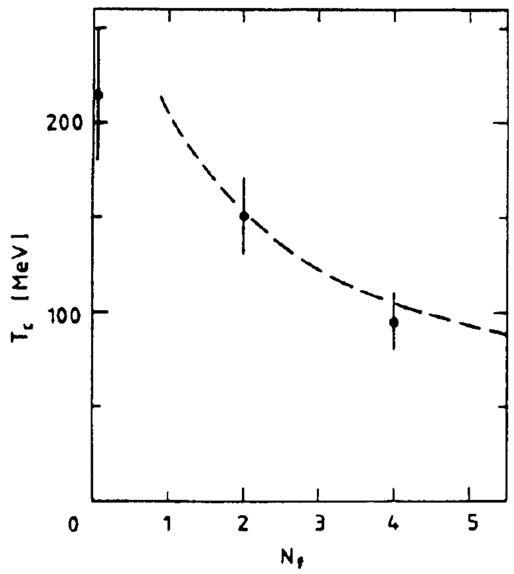


FIG. 1.2 - *La température de transition de phase Tc en fonction du nombre de quarks N_f, nous montre que pour $N_f = 2$, on obtient $T_c \simeq 150$ MeV*

$$V(r, T = 0) = \sigma r^{\beta} - \frac{\alpha}{r^{\beta}} + V_0 \tag{1.5}$$

où,
r représente la distance entre les quarks, σ la tension de corde, α le couplage qui caractérise l'échange de gluons à courte distance, β un paramètre à ajuster pour décrire le potentiel de manière réaliste (fonction du modèle choisi pour décrire le système étudié et de la saveur des quarks : [7], [8]), et enfin V_0 qui représente un potentiel de normalisation constant.

La partie du potentiel en σr^{β}, caractérise la liberté asymptotique de l'intéraction : la force d'attraction due aux échanges de couleurs diminue à courte distance. En revanche, le terme en $\frac{1}{r}$ décrit un comportement de type coulombien du potentiel pour les intéractions à très faibles distances.

Lorsque la densité de quarks augmente, et pour une température non nulle, la charge de couleur subit un phénomène d'écrantage dû à la présence de nombreuses autres charges environantes. Pour de telles conditions, le potentiel d'intéraction quarks- quarks prend la forme [9] :

$$V(r, T) = \left(\frac{\sigma(T)}{\mu(T)^{\beta}}\right)\left(1 - e^{-(\mu(T)r)^{\beta}}\right) - \left(\frac{\alpha}{r^{\beta}}\right) e^{-(\mu(T)r)^{\beta}} + V_0 \tag{1.6}$$

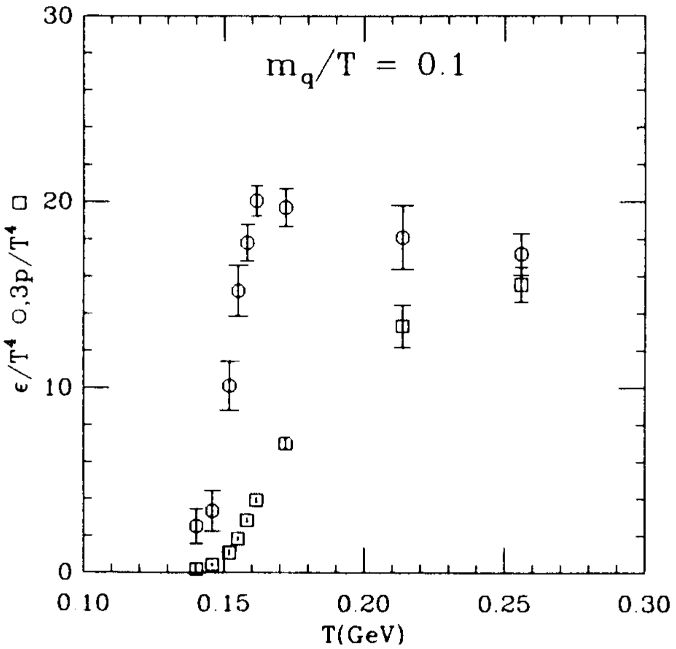


FIG. 1.3 - *Ce diagramme nous montre qu'il existe une brusque variation de la densité d'energie ϵ au voisinage de la température critique T_c. De plus les deux plateaux, situés de part et d'autre de cette transition, montrent bien la dépendance de ϵ en T_4.*

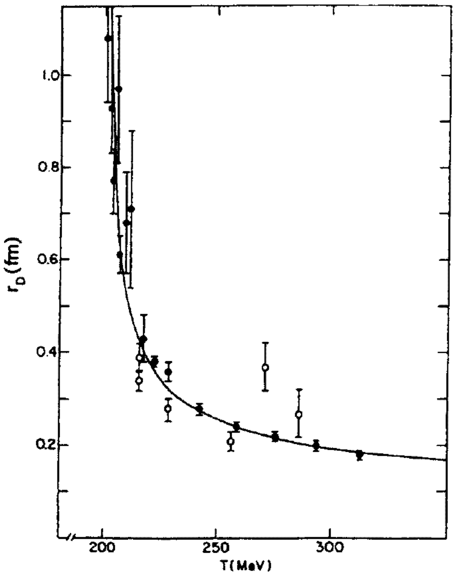


FIG. 1.4 - *Le rayon de Debye est une quantité décroissante avec la température T du système.*

où,

$\mu(T) = \frac{1}{r_D(T)}$ est la masse d'écrantage, inversement proportionelle au rayon de Debye.

La figure 1.4 nous montre la dépendence en température du rayon de Debye $r_D(T)$.

Il s'agit d'une simulation sur réseau [10] qui met en évidence la forme décroissante de l'expression $\frac{1}{\mu(T)}$. Au delà d'une température critique T_c, l'expression du potentiel se résume à :

$$V(r, T \geq T_c) = \frac{\alpha}{r^\beta} e^{-(\mu(T) r)^\beta} \tag{1.7}$$

qui en fonction de $r_D(T)$ donne,

$$V(r, T \geq T_c) = \frac{\alpha}{r^\beta} e^{-(\frac{r}{r_D(T)})^\beta} \tag{1.8}$$

la disparition du terme relatif à la tension de corde σ ne laisse subsister qu'une forme coulombienne du potentiel, atténuée par le terme d'écran $e^{-(\frac{r}{r_D(T)})^\beta}$. Les quarks sont alors quasi-libres, c'est le plasma QGP.

Créer en laboratoire des conditions suffisantes pour explorer cet état de la matière représente un défi entamé dès 1986 auprès des grands accélérateurs tels que le SPS, de l'AGS. Afin d'explorer cette transition de phase de la matière hadronique, nous provoquons des collisions d'ions lourds ultrarelativistes. Les plus violentes de ces réactions doivent nous permettre d'atteindre des températures et des pressions suffisamment élevées pour engendrer la formation de plasma de quarks et de gluons.

1.2.1 Les collisions d'ions lourds ultrarelativistes

D'après le diagramme des phases de la matière hadronique 1.1, nous voyons qu'il existe plusieurs chemins pour franchir la frontière qui sépare le gaz de hadrons du QGP. Les collisions d'ions lourds ultrarelativistes peuvent-t-elles permettre de passer cette frontière?

Observée depuis le repère lié à leur centre de masse, une collision d'ions lourds apparait comme la rencontre frontale de deux ellipsoïdes de matière. Cette forme "écrasée" des noyaux est due au facteur γ de la contraction de Lorentz, qui vaut environ 10 pour une collision à 200 GeV/nucléon sur cible fixe.
Pour des énergies élevées (de l'ordre de 100 à 200 GeV/nucléon), les noyaux présentent entre eux une certaine transparence et se croisent en laissant derrière eux une région trés excitée, dépourvue de nucléon. L'énergie déposée dans cette région (appelée région centrale) se trouve sous la forme d'un gaz de hadrons (essentiellement des pions) de densité baryonique nette égale à zéro ($n_q = n_{\bar{q}}$).

Ces croisements laissent apparaître 3 régions principales schématisées sur la figure 1.5:

- la région centrale.
- la région de fragmentation de la cible.
- la région de fragmentation du projectile.

A partir de ce modèle de collisions, Bjorken [11] a évalué la densité d'énergie ϵ dans la région centrale à:

$$\epsilon = \left(\frac{dE_t}{dy}\right)_{y=0} \times \frac{1}{\tau S_T} \tag{1.9}$$

où, E_T est l'énergie transverse des particules par unité de rapidité autour de y=0, τS_T est le volume de réaction, S_T étant la surface de recouvrement des deux noyaux et τ le temps de formation des partons lors de l'interaction ($\tau \simeq 1fm/C$).

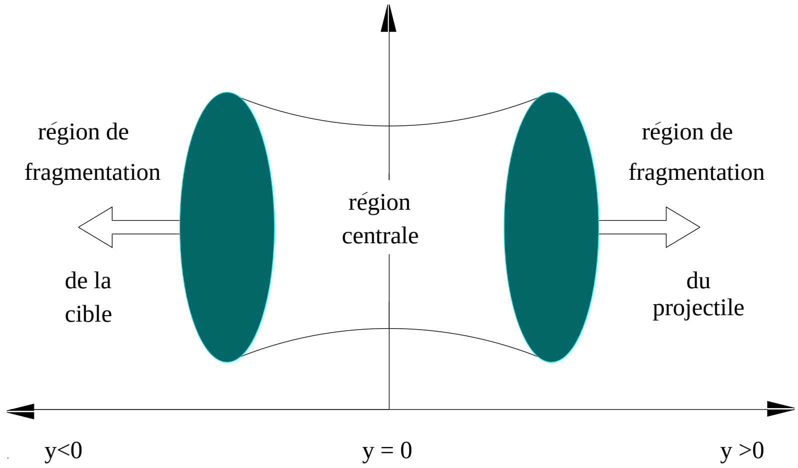


FIG. 1.5 - *Ce diagramme nous montre que juste après la collision, 3 régions peuvent être considérées. La région centrale possède la plus grande densité d'énergie ϵ, C'est dans cette zone que l'on recherche le plasma QGP.*

Pour une particule donnée, la rapidité y est définie par :

$$y = \frac{1}{2} ln \left(\frac{E + P_z}{E - P_z} \right) \tag{1.10}$$

où, E est l'énergie totale de la particule et P_z son impulsion longitudinale. Il s'agit d'une variable cinématique qui nous permet d'estimer l'évolution longitudinale d'un événement. Cette variable est additive dans une transformation de Lorentz. De plus, si la masse propre de la particule est négligeable devant son énergie totale, on utilise la variable qui définit la pseudo-rapidité η par :

$$\eta \simeq y \simeq -\frac{1}{2} ln \left[tan \left(\frac{\theta}{2} \right) \right] \tag{1.11}$$

où, θ est l'angle entre l'impulsion de la particule et l'axe du faisceau incident.

Le terme en $\frac{1}{S_T}$ de l'équation (1.9) représente le nombre de nucléons impliqués dans la collision, par unité de surface [11]. Par suite, la densité d'énergie ϵ est directement corrélée au paramètre d'impact b de la collision.
Dans un phénomène de choc, le paramètre d'impact b est défini comme étant la distance b qui sépare, avant la collision, le centre du noyau au repos de la direction de la vitesse initiale de la particule incidente. Ce paramètre est illustré sur la figure 1.6.

On parle aussi de centralité de la réaction : une collision d'ions lourds sera d'autant plus centrale que son paramètre d'impact b sera faible. Les collisions non centrales seront

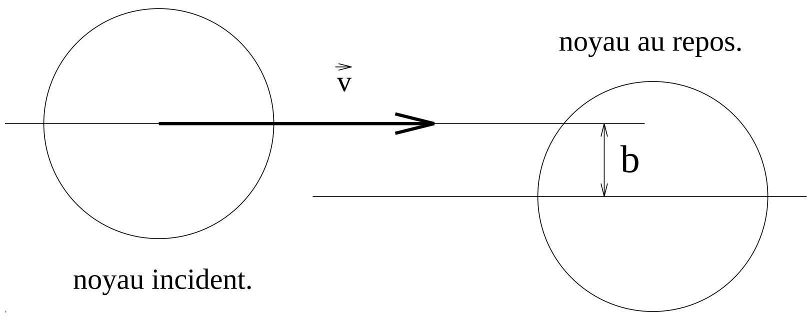


FIG. 1.6 - *Le paramètre d'impact b, traduit le décentrement entre projectile cible. On parle de collisions centrales lorsque b est faible et périphériques dans le cas contraire.*

dites périphériques.

C'est donc pour les collisions les plus centrales, que nous aurons la plus grande densité d'énergie ϵ, et donc un milieu favorable à la transition de phase vers un QGP.

1.3 Evolution de la collision noyau-noyau

Si les conditions expérimentales sont telles que la transition de phase peut être atteinte, alors, l'évolution de la collision noyau-noyau est décrite par le diagramme d'espace-temps 1.7. Sur ce schéma, les temps propres τ sont définis par des hyperboles d'équation : $\tau = \sqrt{t^2 - z^2}$. La collision a lieu au point de coordonnées $(t = 0, z = 0)$.

A partir de l'instant où les noyaux entre en collision, on distingue les étapes suivantes dans l'évolution de la collision :

- Au point d'impact, la rencontre des deux objets formés de nucléons provoque des cascades d'interactions hadroniques, créant ainsi un grand nombre de partons : c'est la thermalisation.
- La phase de plasma QGP qui suit correspond à un milieu thermalisé de quarks et de gluons déonfinés, supposés à l'équilibre chimique.
- Après une courte période ($\simeq 10^{-23}s$), le plasma se détend et se refroidit : c'est l'hadronisation (retour à la phase hadronique). S'il s'agit d'une transition de phase du premier ordre, elle sera caractérisé par une coéxistence mixte de gaz de hadrons et de résidu de plasma.
- Le gaz de hadrons issu de la précédente transition se détend à son tour jusqu'au point appelé "Freeze-Out" où les hadrons se découplent.

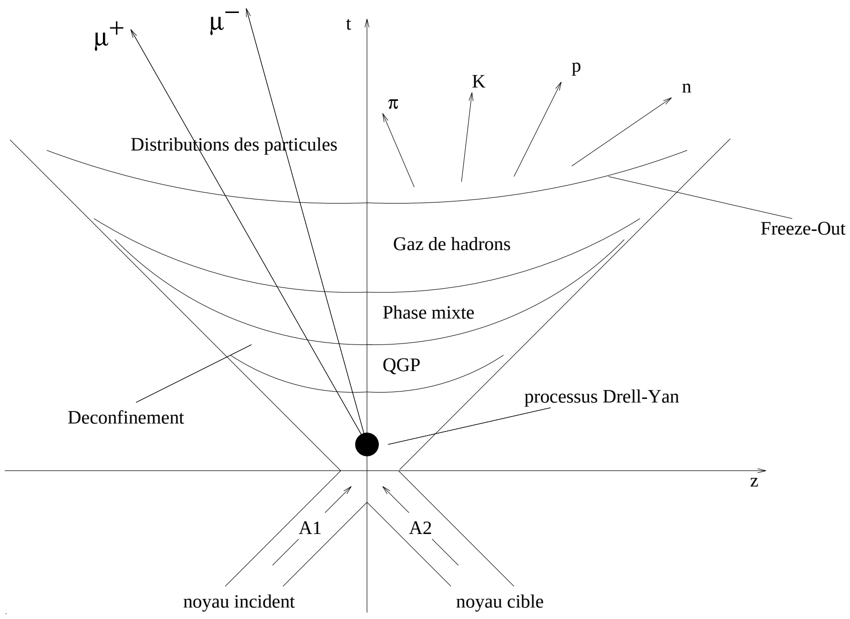


FIG. 1.7 - *Cette figure nous montre l'évolution spacio-temporelle de la collision. Les des ions A1 et A2 entrent en collisions pout* $(t = 0, z = 0)$. *Des particules* $\mu^+\mu^-$ *produite, dès les premiers instants de la collision, par le processus Drell-Yan se retrouvent dans l'état final, et ce quel que soit l'évolution de la région centrale.*

- L'état final de la collision, observé dans les détecteurs est formé d'une multitude de particules (mésons, nucléons, e^-, photon...).

Du fait de leur très faible durée, les quatres premières étapes ne peuvent être accéssibles directement par un quelconque détecteur. En revanche, toutes les particules produites dans la dernière étape, par leurs caractéristiques (énergie, impulsion, multiplicité, distribution...) peuvent permettre une reconstitution du scénario de la collision. Il convient de trouver, parmi tous ces produits de réaction, ceux qui auraient pu "garder la mémoire" d'un état transitoire déconfiné. On parlera de signatures.

La suite de ce chapitre décrit brièvement quelques signatures potentielles d'un plasma de quarks et de gluons, recherchées par l'expérience NA50 :

- la baisse de production du charmonium en présence d'un QGP.
- la production élevée de particules étranges.

1.4 La supression des résonances de saveurs lourdes

En 1986, il a été prédit [12] que la présence d'un milieu dense déconfiné doit interdire la formation de résonances de saveurs lourdes. Le modèle de Cornell [7], [8] suppose le potentiel de la liaison $c - \bar{c}$ non relativiste décrit par :

$$V_{cor}(r, T = 0, \beta = 1) = \sigma r - \frac{\alpha}{r} \tag{1.12}$$

Lorsque la densité de charges de couleur augmente, et pour $T \neq 0$, ce potentiel prend la forme :

$$V_{cor}(r, T \neq 0, \beta = 1) = -\frac{\alpha}{r} e^{-\frac{r}{r_D(T)}} \tag{1.13}$$

L'accroissement de la température provoque une diminution du rayon de Debye et tend à annuler $V_{cor}(r)$. Les quarks c et $\bar{c}$ apparaissent alors associés à d'autre saveurs telle que u et d beaucoup plus abondantes dans le milieu.

Si l'on prend comme exemple celui de la résonnace J/ψ qui correspond à l'état lié $c - \bar{c}$ à $3.1 Gev/C^2$, la présence d'un plasma dense de type QGP va empécher sa formation.
Il faut cependant prendre en compte les paramètres cinématiques des états liés $c - \bar{c}$ pour décrire un peu plus précisément ce phénomène de suppression. En effet, si l'on considère qu'une paire $c - \bar{c}$ n'est supposée former une résonance J/ψ qu'après un temps $\tau_c \simeq 1 fm/c$, il n'est pas exclu que cette paire parvienne à s'échapper du milieu déconfiné, avant que n'expire ce temps de formation : la liaison $c - \bar{c}$ évite ainsi une dissolution dans le plasma. En tenant compte de l'expansion longitudinale du volume déconfiné, il apparaît que ce sont les paires $c - \bar{c}$ de grande impulsion transverse qui ont la plus forte probabilité de ne pas être "dissoute". Les phénomènes de suppression des résonances de saveurs lourdes seront donc plus marqués pour les faibles P_T.

Toutefois, pour pouvoir observer une quelconque diminution des productions de ces résonances en présence d'un éventuel plasma QGP, il faut être capable de comparer ces taux de production avec un phénomène totalement décorrélé du QGP. La base de référence utilisé est le processsus Drell-Yan. Il s'agit là d'un processus dur qui produit des paires $\mu^+\mu^-$ au tout début de la collision [13] par intéraction $q - \bar{q}$:

Du fait de la très faible interaction de ces paires de muons avec la matière, produits avant la formation du QGP, ces dimuons Drell-Yan se retouveront "intacts" en fin de réaction (cf figure 1.7) : ils forment un continuum de référence. La diminution de la production d'une résonance sera exprimée par rapport au taux de production de ces paires $\mu^+\mu^-$.

Chapitre 2

LE DISPOSITIF EXPERIMENTAL

2.1 Introduction

Comme nous l'avons vu dans le chapitre précédent, lors de collisions d'ions lourds, la création d'un plasma de quarks et de gluons se traduit entre autre par la baisse du taux de production des résonances $c\bar{c}$. On peut observer indirectement ces états liés par la mesure de leurs produits de désintégration : les dimuons $\mu^+\mu^-$. L'objet du dispositif expérimental NA50 est d'explorer la matière produite par des collisions d'ions Pb^{208} sur cibles fixes de plomb, l'observation étant déclenchée sur l'apparition de signaux dimuoniques $\mu^+\mu^-$.

L'ensemble du dispositif expérimental dédié à ces observations est situé dans la zone nord[1] du SPS au CERN. La figure 2.1 montre une coupe longitudinale de cet ensemble. La cible est constituée de cellules en plomb, sur laquelle se déverse périodiquement des paquets d'ions Pb à 160 GeV/nucléon. Elle est située avant l'essentiel de l'appareillage.

Nous cherchons avant tout à détecter des muons. Ces particules ont la particularité de n'intéragir que très faiblement avec la matière. Nous exploitons cette propriété en plaçant un absorbeur principal, en aval du lieu de l'interaction. Cet élément arrête ainsi toutes les particules à l'exception des muons qui le traversent sans interaction. L'espace situé derrière ce bloc est donc réservé à l'étude de ces particules $\mu^+\mu^-$: c'est le spectromètre à muons.

Le schéma 2.1(a) nous montre qu'il est composé de quatre hodoscopes (R1-R4) et huit chambres à fils (CP1-CP8) disposés autour d'un aimant. Ce spectromètre permet de repérer les traces laissées par les muons, et assure les déclenchements, lorsque celles-ci sont corrélées à une même interaction. Les courbures des trajectoires, induites par l'aimant,

1. NA50 pour North Area 50

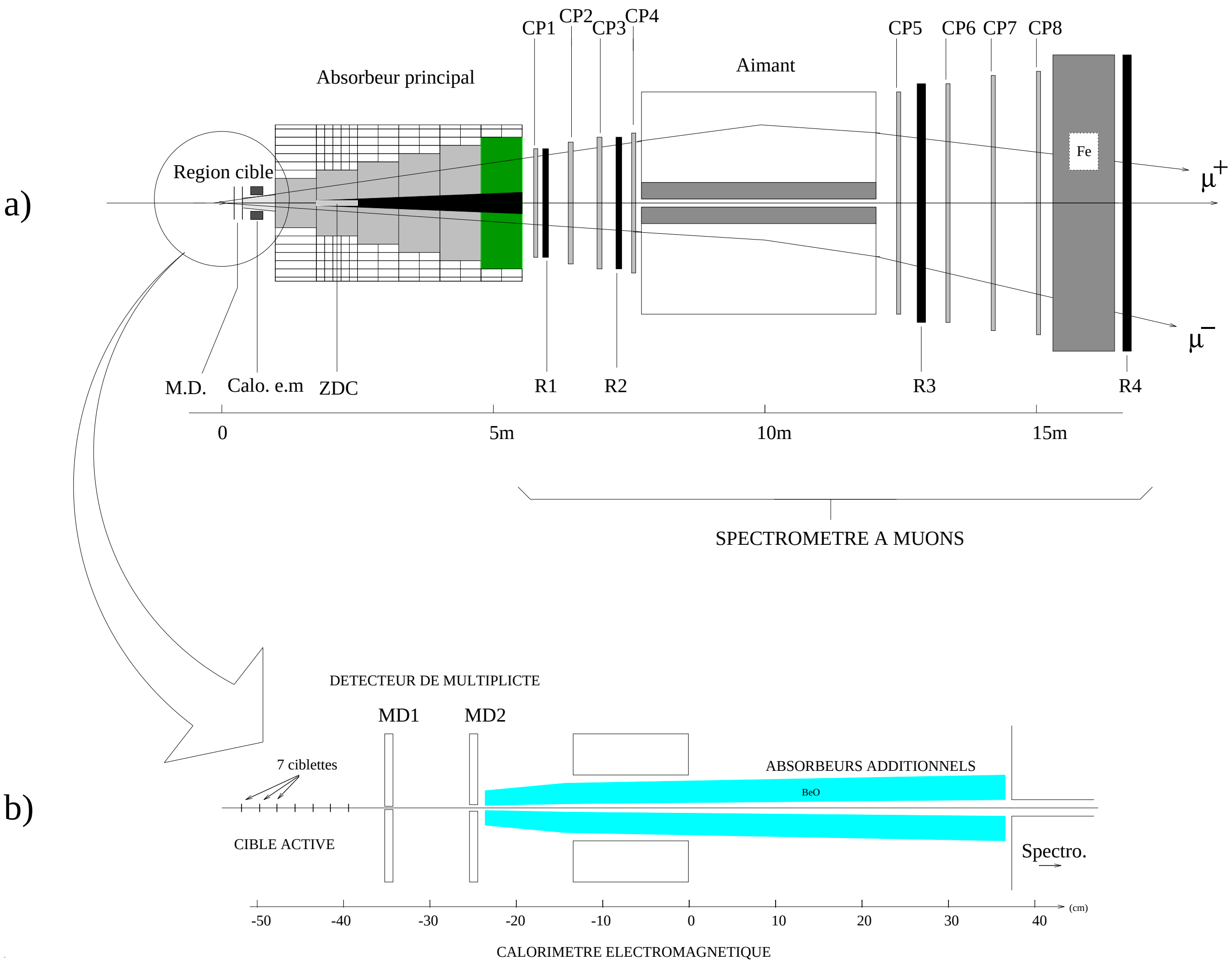


FIG. 2.1 - *Ce schéma représente une coupe longitudinale de l'ensemble du dispositif expérimental de NA50 (a) et une coupe plus détaillée de la région cible (b). Sur la partie (a), on remarque le spectromètre à muons, placé derrière l'absorbeur principal. Il est chargé de repérer les traces de muons en vue d'effectuer les déclenchements. Dans la région cible (b), on observe la présence du calorimètre électromagétique et du détecteur de multiplicité chargée. Un calorimètre à zéro degré est inséré dans l'absobeur principal.*

sont reliées à l'impulsion transverse des muons.

Le spectromètre possède l'avantage de pouvoir opérer à très haute intensité.
En effet, le détecteur doit être capable de mesurer des paires de muons dans le domaine de masse du J/ψ, centré sur 3.1 GeV/c^2 avec le maximum d'éfficacité. La production de telles paires est un phénomène rare. Pour des raisons de statistique évidentes, le détecteur doit donc accepter un faisceau aussi intense que possible.

Pour toute paire de muons corrélée, deux calorimètres mesurent d'une part l'énergie des constituants neutres issus de l'interaction (calorimètre électromagnétique) et d'autre part l'énergie des fragments spectateurs de l'interaction (calorimètre à zéro degré ou ZDC). Le premier, visible sur la figure 2.1(b) est situé derrière la cible. Le second est inséré dans l'absorbeur principal le long de la ligne du faisceau.

Toujours dans la région cible, un détecteur de multiplicité étudie la production des particules chargées.
Cet élément nouveau, basé sur la technologie des micro-bandes de silicium sera opérationnel dans sa totalité pour novembre 1996. Une partie de mon travail de thèse a résidé dans le développement d'un système de transfert pour les données de ce détecteur ; c'est pourquoi il fera l'objet d'une description plus détaillée.

Dans ce chapitre, nous decrivons l'ensemble de ce dispositif expérimental en insistant plus particulièrement sur la description des nouveaux détecteurs (par rapport à NA38), le reste de l'appareillage ayant déja été largement décrit dans d'autre thèses [17].

2.2 Le faisceau

L'expérience NA50 est placée sur la ligne ECN3 du SuPerSychrotron (SPS). Deux types de faisceaux sont utilisés pour alimenter ce dispositif expérimental :

- un faisceau de 10^{11} protons à 450 GeV/nucléon
- un faisceau de 5.10^7 ions plomb Pb^{208} à 400 GeV/nucléon, soit 160 GeV/nucléon.

Lors d'une période de prise de données (de 20 à 40 jours), des paquets d'ions (appelés *bursts*) sont déversés de façon cyclique sur la région cible. La périodicité de ce phénomène est définie par deux grandeurs (cf figure 2.2) : d'une part, la durée du burst t_{Burst}, et d'autre part, la durée de l'inter-burst $t_{IntBurst}$.

Ces deux valeurs fixent respectivement le temps d'acquisition des données, et le temps maximum accordé au transfert de ces informations. Nous verrons par la suite que le système d'enregistrement sur support magnétique, impose un temps de transfert inférieur à $t_{InterBurst}$.

Comme le montre le tableau 2.1, les cycles de déversement sur la cible ainsi que l'intensité de chaque bouffée sont différents selon le type de prise de données.

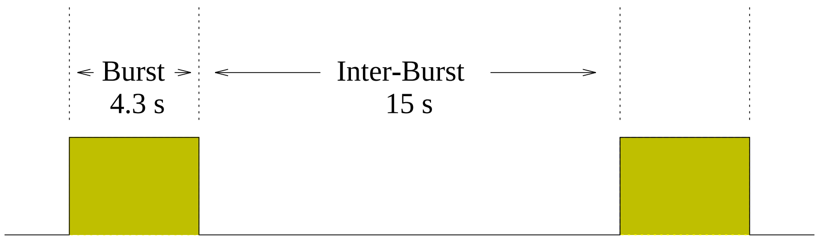


FIG. 2.2 - *Schéma du cycle faisceau Pb^{208} délivré par le SPS au CERN. Les ions plomb sont déversés sur la cible pendant une bouffée de 4.3 s (burst). Ce déversement est renouvelé toute les 14.8 s.*

Prise de données	I (partic./sec)	t_{Burst} (sec)	$t_{InterBurst}$ (sec)
proton	10^{11}	2.5	11
plomb	5.10^{7}	4.3	14.8

TAB. 2.1 - *Caractéristiques faisceau selon les prises de données. La notation t_{Burst} désigne le temps de déversement (burst) tandis que $t_{InterBurst}$ est la période durant laquelle aucune particule ne pénètre la cible.*

2.3 La cible active

Lors de la collision, on observe une multiplicité élevée en pions et kaons. L'utilisation d'une cible longue induirait un taux de réinteractions non négligeable des fragments et particules secondaires produites lors de la première collision. Dans cette configuration, il serait impossible de savoir si un événement est issu de la première collision ou bien s'il est le produit d'une réinteraction. De plus, le vertex d'interaction doit être connu avec précision puisqu'il sagit d'un paramètre essentiel à la détermination de l'énergie transverse de la collision. La cible dite active doit donc :

- présenter une longueur d'interaction aussi grande que possible tout en minimisant les interactions des particules secondaire.
- permettre d'identifier avec précision le vertex d'interaction.

Pour cela, la cible active de plomb est fractionnée en 7 ciblettes. La taille de chacune d'entre elles est de : $1.5 \times 1.5 \times 0.5\ mm^3$. Elles sont placées le long de la ligne du faisceau à intervalles réguliers de $\delta z = 25$ mm (cf figure 2.3). L'ensemble repésente 9% de longueur d'interaction.

Dans la région cible, le taux de radiation (plusieurs dizaines de Mrad/mois avec 10^7 ions Pb/s) a conduit à abandonner le dispositif de détection de vertex basé sur des scintillateurs (NA38) qui, pour une même géométrie ne supportaient pas cet environnement hostile.

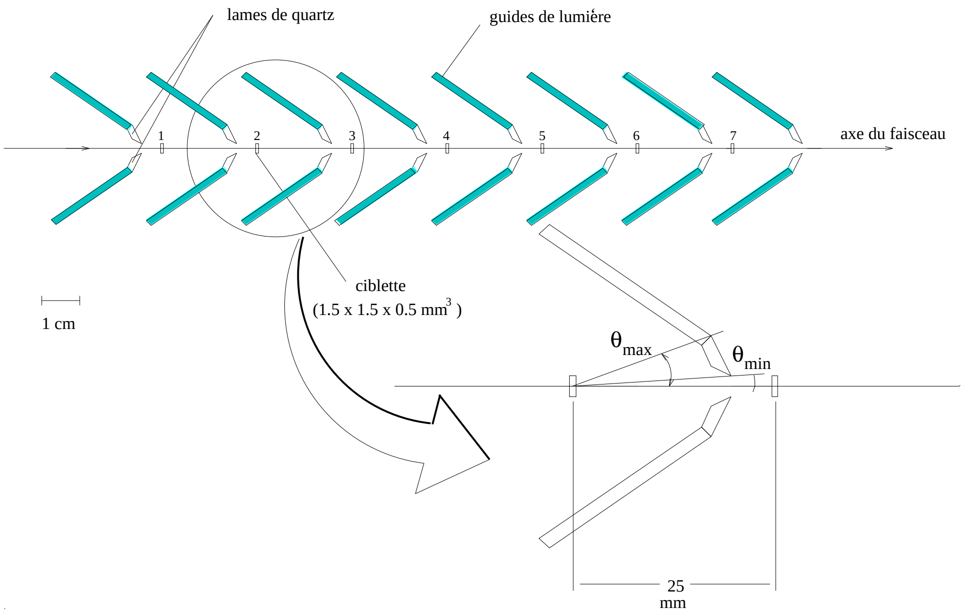


FIG. 2.3 - *Ce schéma de la cible fragmentée, montre les 7 ciblettes le long de la ligne de faisceau, ainsi que les paires de lames de quartz associées à chacune d'entre elles. Ces lames permettent d'isoler les ciblettes qui ont abrité les collisions. L'ensemble représente 9% de longueur d'interaction.*

Pour localiser le vertex d'interaction, on utilise deux lames de quartz, disposées de part et d'autre de chaque ciblette de plomb. Le quartz a été choisi pour sa très bonne tenue aux radiations et ses propiétés non scintillantes. Ces lames collectent la production de lumière Cerenkov, induite par les pions chargés de grande vitesse issus de la collision, et permettent ainsi de signer le lieu de l'interaction (cible i avec $1 \leq i \leq 7$). L'épaisseur des lames (1 mm) minimise la production de particules secondaires. Leur couverture angulaire (relative à leur cible propre) s'étend de $\theta_{min} = 50$ mrad à $\theta_{max} = 350$ mrad. Les deux quartz d'une même paire sont distant de 15 mm à θ_{min}.

2.4 Le spéctromètre à muons

2.4.1 Introduction

Rappelons que l'on cherche à isoler parmi les collisions, celles qui engendrent des particules lourdes telles que ρ, ω, ψ et ψ'. La méthode d'identification utilisée est la détection de paires $\mu^+\mu^-$ énergiques, qui signent par leur présence l'existence antérieure de tels états liés ou bien de diffusion Drell-Yan : c'est le rôle du spectromètre à muons.

Il permet d'appréhender les paramètres cinématiques des paires de muons produites par les collisions d'ions lourds. L'ensemble de son appareillage, situé derrière un absorbeur,

est agencé autour d'un aimant toroïdal qui courbe les trajectoires des particules chargées. Quatre hodoscopes associés à une logique cablée permettent de signer le passage d'une paire de muons corrélés, et constituent ainsi le système de déclenchement. Huit chambres à fils repèrent les trajectoires exactes empruntées par ces muons. L'absorbeur hadronique (ou absorbeur principal) placé en front du spectromètre stoppe mésons et faisceau résident, sources principales de bruit de fond pour la détection des muons.
L'ensemble de ce spectromètre, s'étend sur 18 mètres, le long de l'axe du faisceau (cf figure 2.4).

2.4.2 Les absorbeurs

Lors des collisions d'ions lourds, on observe une production importante de mésons chargés π et K. Leurs désintégrations engendrent l'apparition de muons qui constituent une source de bruit considérable pour la détection de muons. Trois types d'absorbeurs sont placés le long de la ligne de faisceau pour arrêter ces mésons avant leur désintégration.

L'absorbeur principal est formé de trois corps :

- Un centre lourd à géométrie conique : le "plug". Situé derrière le calorimétre à zéro degré (ZDC), il est composé de tungstène et d'uranium et mesure 3.6 mètres de long. Son rôle est d'absorber l'intégralité des ions du faisceau n'ayant pas intéragi avec la cible : ceci représente environ 80% du flux initial. Ce noyau d'éléments lourds est placé à 165 cm du centre de la cible pour éviter la création de nouvelles paires de muons dans l'accéptance du spectromètre.
- Ce centre conique est enveloppé par une couronne d'une longueur de 4.8 mètres, qui couvre l'accéptance de l'appareillage aval (32 à 116 mrad). Le rôle de cette enveloppe est d'arrêter les produits non muoniques de la réaction, sans pour autant affecter la trajectoire des muons qui la traversent. Il en resulte une couronne faite de cuivre, de carbone et d'oxyde de berylium. Elle représente 22 longueurs de radiation, 11 longueurs d'interaction et correspond à 15 longueurs d'absorption pour les π.
- Les composants précédent sont insérés à l'intérieur de blocs de fer et de béton pour piéger toute autre particule secondaire.

Un autre absorbeur, de dimensions moindres, est placé à la sortie du calorimètre électromagnétique. Son rôle est d'éliminer le plus tôt possible les π et les K chargés, avant leur désintégration en μ, pour les mêmes raisons que précédemment.

Un dernier mur de fer, placé entre la chambre à fils CP8 et l'hodoscope R4, absorbe les mésons qui auraient traversé le spectromètre, et permet ainsi d'éviter les déclenchements non dimuoniques. Sa position en aval de la dernière chambre à trace évite de pertuber la résolution en masse.

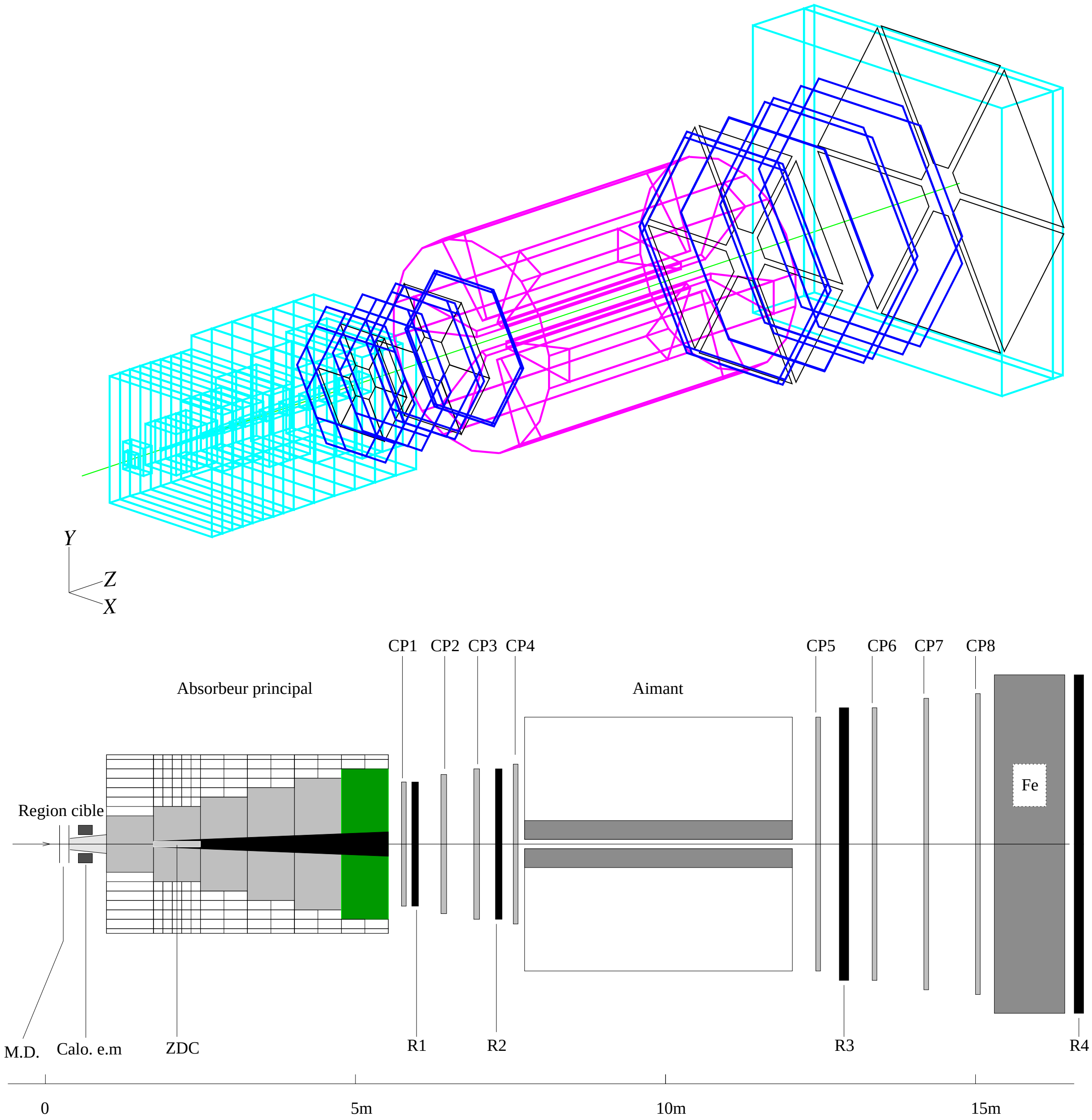


FIG. 2.4 - *Coupe longitudinale du spectromètre, depuis la région cible jusqu'à l'hodoscope R4. La figure supérieure, par sa repésentation 3d, renseigne sur les volumes occupés par le dispositif complet.*

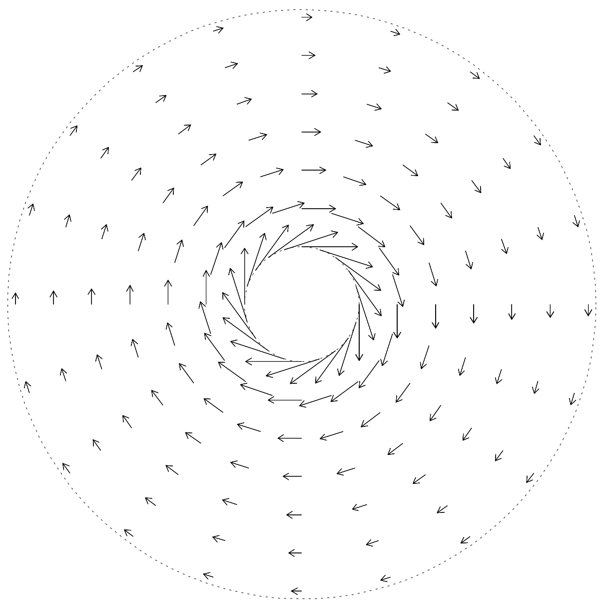


FIG. 2.5 - *Diagramme représentant l'allure du champ magnétique* $\vec{B}(r) = \frac{B_0}{r}\vec{e_\phi}$ *créé par l'aimant, dans un plan orthogonal à l'axe du faisceau. Une telle disposition des lignes de champ, laisse les particules chargées dans le même plan azimutal lors de la traversée de l'aimant. Seule une déviation* θ_μ *leur est imposé, avec au premier ordre :* $\theta_\mu \propto \frac{1}{p_T}$.

2.4.3 L'aimant

Cet aimant est positioné au centre du spectromètre. De symétrie hexagonale autour de l'axe du faisceau, il possède 4.8 m de longueur pour un diamètre de 2 m. Chaque sextant est un bobinage autour d'un noyau de fer qui couvre 18 degrés. Sa région utile se situe entre 30 et 154 cm de rayon ce qui constitue l'acceptance du spetromètre.

Il délivre un champ magnétique toroïdal $\vec{B}$, pulsé à la fréquence du SuPerSynchrotron (SPS). Cette propriété de champ toroïdal laisse les particules chargées dans le même plan azimutal lors de la traversée de l'aimant. La reconstruction des traces chargées est ainsi grandement simplifiée.
De plus, la valeur du champ magnétique est inversement proportionnelle à la distance à l'axe du faisceau (cf fig. 2.5) :

$$\vec{B}(r) = \frac{B_0}{r}\vec{e_\phi}$$

$\vec{e_\phi}$ est le vecteur unitaire du plan azimutal,
et $B_0 = 22$ T.m pour une intensité I = 4000 A.

Ainsi, les muons subissent à la traversée de l'aimant, une déviation θ_μ, qui au premier ordre est inversement proportionnelle à l'impulsion transverse de la particule : $\theta_\mu \propto \frac{1}{p_T}$.

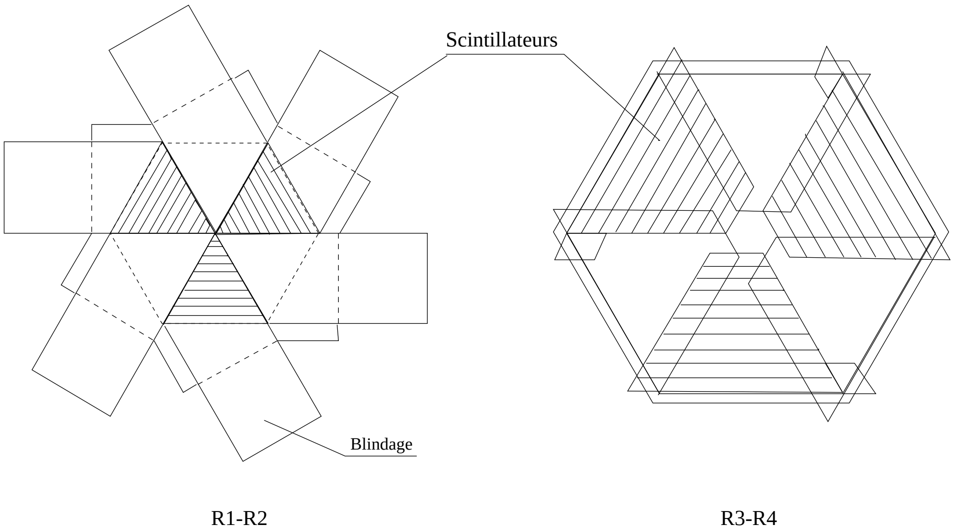


FIG. 2.6 - *Schémas des hodoscopes R1-R2 et R3-R4. Ces coupes transversales nous montrent que ces hodoscopes sont composés de bandes de scintillateurs, répartis dans une structure hexagonale. Ces 4 détecteurs placés de part et d'autre de l'aimant central, permettent de repérer rapidement des traces cohérentes de paires $\mu^+\mu^-$, et sont à l'origine des déclenchements.*

2.4.4 Les hodoscopes R1, R2, R3 et R4

Les hodoscopes R_i sont des plans hexagonaux de scintillateurs plastiques qui ont pour but le déclenchement de la prise des données.

Les plans R_1 et R_2 sont situés entre l'absorbeur principal et l'aimant ; les plans R_3 et R_4 sont placés après l'aimant, de chaque coté du mur de fer.
Un plan hodoscope correspond à 6 sextants de 32 bandes[2] de scintillateurs plastiques de 1 cm d'épaisseur, disposées parallèlement à leurs bords extérieurs (figure 2.6).
La largeur des scintillateurs des deux premiers plans croît en fonction de leur distance par rapport à l'axe du faisceau. De plus, le plan R_2 est l'image homothétique du plan R_1 par rapport à la position de la ciblette centrale. Grâce à cette géométrie, une trajectoire passant par les bandes R_1^i et R_2^i (ou R_2^{i-1}) pointe en direction du vertex de l'interaction.
Les bandes de scintillateurs de R_3 et R_4 ont des largeurs constantes (5.5 cm), et fournissent une approximation de la trajectoire des particules après déflexion dans l'aimant.
Le sextant j d'un plan donné couvre le même intervalle en angle azimutal ϕ que le sextant j des trois autres plans hodoscopes.

Un déclenchement sera ordonné si, à un instant donné, les impacts hodoscope permettent de construire, de part et d'autre de l'aimant, dans deux sextants différents,

2. R_3 n'est constitué que de 24 bandes de scintillateurs.

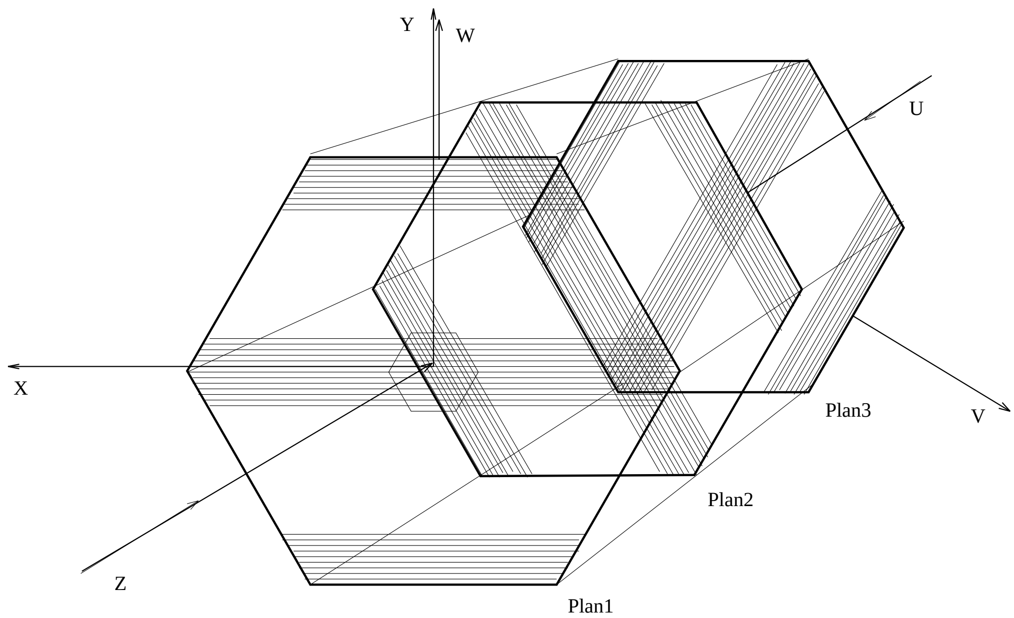


FIG. 2.7 - *Schéma des 3 plans Y, U, V d'une chambre à fils, décalés entre eux de 60 degrés. Ces chambres, au nombre de huit, localisent avec prsécision les traces des paires* $\mu^+\mu^-$ *lorsqu'un événement potentiel à été déclenché par les 4 hodosopes Ri.*

une paire de traces corrélées à un même point d'interaction. Une logique cablée, traite systématiquement les signaux générés par ces quatre plans d'hodoscopes pour identifier de tels événements dimuoniques : c'est le système de déclenchement inter-sextant. Le fonctionnement de cette logique est expliqué en détails dans la référence [15].

L'ensemble des quatre plans permet aussi d'avoir *en ligne* une estimation de l'impulsion transverse des muons individuels qui est inversement proportionnelle à angle de déviation dans l'aimant.

Deux autres plans hodoscopes P_1 et P_2, de part et d'autre de l'aimant, réalisent un déclenchement indépendant et permettent d'estimer l'efficacité du déclenchement principal.

2.4.5 Les chambres proportionnelles à fils

Pour connaître les positions spatiales des muons, nous disposons de 8 chambres proportionnelles multifils CP_i.

Les quatre premières chambres (CP_1 à CP_4) permettent de reconstruire les traces des muons avant leur déflexion dans l'aimant ; ces chambres ont un rayon de 1.3 mètre. Les chambres arrières (CP_5 à CP_8), situées après l'aimant, ont un rayon de 3 mètres et permettent la reconstruction des traces après déflexion des particules dans l'aimant.

Chaque chambre est constituée de trois plans de fils U,V et Y (cf figure 2.7). Dans un plan, les fils sont espacés de 3 mm et, deux fils de deux plans différents d'une même chambre font un angle de 60^o entre eux. Les projections U,V,Y définissent ainsi 3 directions dans l'espace. Une telle géométrie n'autorise qu'une seule combinaison de fils touchés pour 2 impacts de particules dans une chambre.

La géométrie des chambres à fils conduit à une précision sur la direction des traces des particules, de l'ordre de quelques dixièmes de mrad.

2.5 Le calorimètre électromagnétique

Comme nous l'avons vu dans le premier chapitre, l'énergie transverse neutre E_T^0 produite par l'interaction, est un des paramètres essentiels à la determination de la densité d'énergie ϵ atteinte lors de l'impact.
Le rôle du calorimètre électromagnétique (figure 2.8) est de mesurer l'énergie E_T^0 qui provient essentiellement des hadrons neutres π^0 issus de la collision. On ne mesure pas directement l'énergie déposée par ces π^0 mais celle laissée par leurs produits de désintégration (cascade de γ,e^+,e^-):

$$\pi^0 \rightarrow \gamma\gamma \rightarrow 2e^+2e^-...$$

Ce détecteur est centré sur l'axe du faisceau à 32 cm derrière la ciblette centrale (cf figure 2.14). Il possède une symétrie héxagonale pour un fractionnement en sextants. Chacun des sextants est constitué de 5 cellules.
Une cellule est une succession de feuilles de plomb et de fibres scintillantes parallèles au faisceau dans un rapport de volume 2:1 (cf figure 2.8).
Lors de la traversé des feuilles de plomb, les photons produits par la réaction $\pi^0 \rightarrow \gamma\gamma$ engendrent des gerbes électromagnétiques. Celles-ci excitent les fibres scintillantes qui créent alors un signal lumineux. Des guides de lumière et des fibres optiques dirigent 30% du flux lumineux de chacune des 30 cellules vers des photomultiplicateurs. Ces PM seront lus par des modules de conversion ADC. L'intensité des signaux receuillis est proportionelle à la quantité d'énergie neutre déposée. Ce rapport de proportionnalité est évalué par des séances de calibration en faisceau d'électrons d'énergie connue.

Un trou central de 2.4 cm de rayon laisse passer le faisceau qui n'a pas interagit dans les ciblettes. La longueur et le rayon du calorimètre sont de 12 cm. L'intervalle de pseudo-rapidité couvert est compris entre 1.7 et 4.1 dans le référentiel du laboratoire.
Les dimensions du calorimètre correspondent à 15.7 longueurs de radiation, soit 90% de l'énergie d'un γ, et 0.3 longueur d'interaction. Donc, le calorimètre est aussi un absorbeur pour les π et les K chargés. La contribution de ces particules chargées à l'énergie mesurée est de 25%.

Pour une cellule i donnée, l'énergie transversale receuillie est : $E_{T_i} = E_i sin\theta_i$ où θ_i est l'angle entre l'axe du faisceau et l'axe vertex d'interaction-cellule i, et E_i est l'énergie

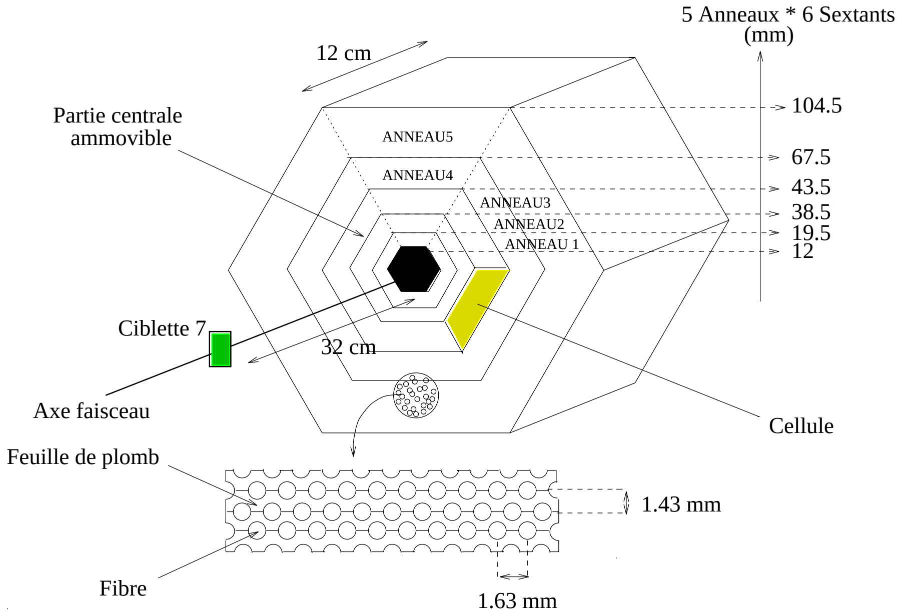


FIG. 2.8 - *Schéma du calorimètre electromagnétique, avec détails de la structure interne. Placé sur l'axe du faisceau, à 32 cm derrière la ciblette centrale, il est formé de fibres scintillantes insérées dans du plomb. Ce détecteur permet de mesurer l'énergie transversale neutre des π^0 produits par les collisions.*

associée au signal receuilli par la cellule i.
Par suite, l'énergie transversale neutre totale peut être définie comme :

$$E_T^0 = \sum_{i=1}^{30} \alpha_i E_{T_i}$$

α_i est le facteur corrigeant la contribution hadronique ; ce facteur dépend de la position du vertex.

Une bonne mesure de E_T^0 dépendra donc de la bonne estimation du vertex d'interaction.

La résolution en énergie de ce calorimètre est environ $\frac{\sigma(E)}{E} = \frac{30\%}{\sqrt{E}}$. Elle est suffisante pour pouvoir estimer des valeurs moyennes de E_T^0 sur les intervalles d'au moins 10 GeV de largeur.

2.6 Le calorimètre à zéro degré

Installé pour la première prise de données en ions plomb (novembre 1994), ce calorimètre à zéro degré (ZDC) est utilisé sur NA50 pour mesurer l'énergie E_{ZDC} des fragments n'ayant pas pris part aux interactions. Cette quantité, permet de calculer l'énergie déposée dans la réaction E_{dep} :

$$E_{dep} = E_{faisceau} - E_{ZDC}$$

Il est ainsi possible d'estimer le nombre de nucléons participant, qui pour un faisceau de plomb à 160 GeV/nucléon est donné par :

$$N = 208 - \frac{E_{ZDC}}{160GeV}$$

Ces valeurs traduisent la centralité de la réaction.

De part sa position, le long de l'axe du faisceau, cet appareillage offre un contrôle sur le centrage du faisceau. En outre, la distribution des énergies ZDC est directement liée au nombre d'ions incidents présents dans la fenêtre d'acquisition (empilement d'ions). Ces deux derniers points viennent compléter l'hodoscope de faisceau BH et le détecteur BI.

Le ZDC est un calorimètre hadronique (mesure des interactions nucléaires), placé à l'intérieur de l'absorbeur principal, à 180 cm derrière la cible, sur l'axe du faisceau.
Sa partie active est en forme de parallèlépipède. La section perpendiculaire au faisceau est un carré de $5 \times 5\ cm^2$ pour une longueur totale de 65 cm, soit 5.5 longueurs d'interaction. Ce volume est composé de 30 lames de tantal (Ta) de 1.5 mm d'épaisseur (cf figure 2.9), dans lesquelles sont insérées 900 fibres optiques de quartz de 365 μm de diamètre, le tout dans rapport de volume 17:1. L'espacement minimum entre deux fibres représente 1.5 mm. Elles sont disposées parallèlement à l'axe du faisceau. Cette disposition, qui réduit de 50% le signal collecté par rapport à l'orientation optimum à 45 degrés, a été choisie en raison des contraintes mécaniques imposées (insertion du calorimètre dans l'absorbeur principal).

La partie active est divisée longitudinalement en quatre quadrants de 225 fibres. Les extrémités des fibres sont directement connectées à des photomultiplicateurs, situés à un mètre de l'axe du faisceau. Les signaux ainsi générés sont transportés jusqu'à des modules QDC, dans la salle de comptage, via 200 m de cables.

Les données des mémoires des QDC sont lues par un réseau de transputers pour être transférées vers les mémoires finales.

L'ensemble de ce dispositif conduit à une résolution en énergie $\frac{\sigma}{E}$ de 30% (pour des faisceaux hadronique de protons ou de pions), et de 5% pour des faisceaux de plomb.

2.7 Le détecteur de multiplicité

Le calorimètre électromagnétique nous informe sur la production des particules neutres. Il nous renseigne mal sur le phénomène de création des particules chargées, qui lors

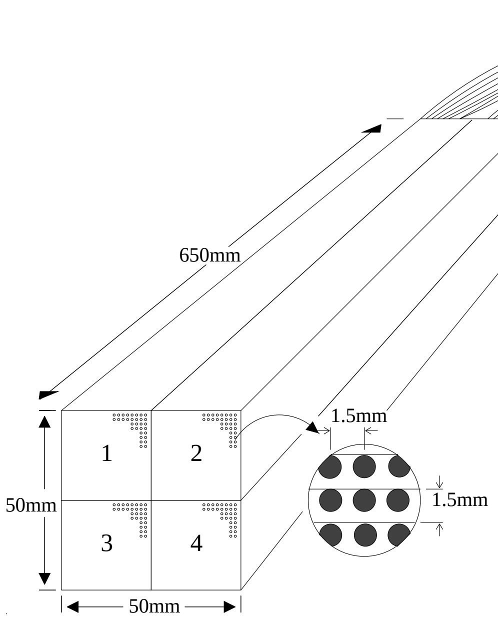


FIG. 2.9 - *Schéma et dimensions du calorimètre à zéro degré ZDC. Placé derrière la cible, sur l'axe du faisceau, il est chargé de mesurer l'énergie résiduelle E_{ZDC} des interactions Chacun de ses quadrants est composé de fibres scintillantes incluses dans un bloc de tantal. Quatre photomultiplicateurs PMi conduisent les signaux sortant vers des modules ADC.*

des collisions d'ions lourds à 160 Gev/nucléon sont principalement les hadrons π^+ et π^-.

Nous avons donc mis en place un détecteur de multiplicité qui constitue un des nouveaux éléments apportés au précédent dispositif expérimental NA38.

Son rôle est de :

- mesurer la multiplicité chargée N_{ch} globale, résultant de la collision. Celle-ci peut nous conduire à une classification des événements selon leur centralité []. D'autre part la mesure d'un rapport $\frac{N_{ch}}{N_{neutres}}$ différent de $\frac{2}{3}$ peut nous renseigner sur la présence éventuelle d'un plasma.
- mesurer la distribution en rapidité (information en θ) des hadrons chargés.
- déterminer avec une résolution angulaire suffisante la position des particules chargées dans le plan R-ϕ. Aucune direction privilégiée ne doit apparaître dans une collision parfaitement centrale.

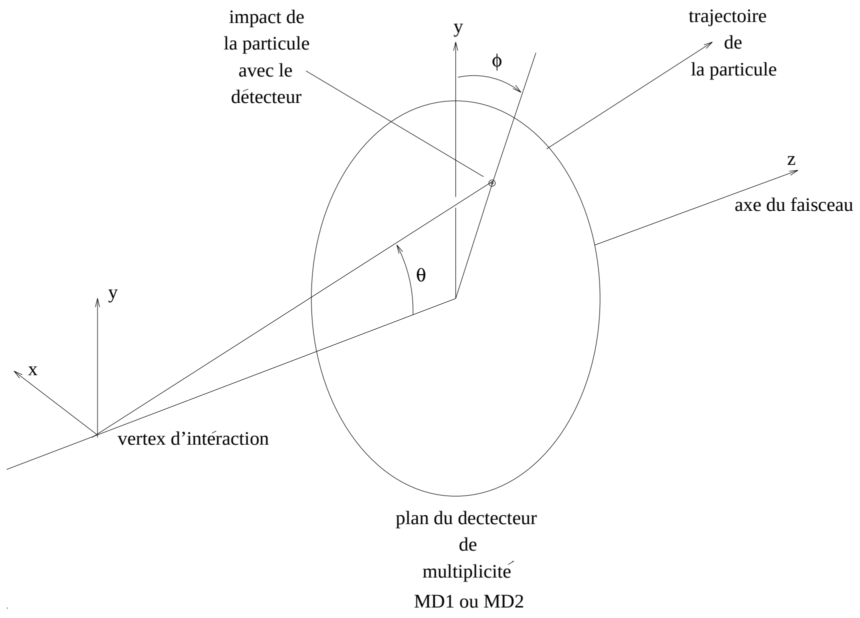


FIG. 2.10 - *Schéma du repérage d'un impact sur le détecteur de multiplicité, en fonction des angles θ et ϕ.*

2.7.1 Architecture du détecteur

Il sagit d'un détecteur à bandes de silicium à symétrie circulaire, composé de deux "sous-détecteurs" identiques MD1 et MD2 en forme de disques, avec pour axe, celui du faisceau d'ions. La figure 2.10 montre les repérages angulaires utilisés pour localiser les impacts des particules sur ces disques de détections.

Un disque (MD1 ou MD2) est un assemblage de deux couronnes :

- une couronne interne :
  La surface sensible de cet élément interne s'étend radialement de 0.44 cm à 3.44 cm depuis l'axe du faisceau. Elle est constituée de 36 secteurs azimutaux, de 10 degrés en ϕ. Chacun d'entre eux contient 128 bandes de silicium, de 350μm d'épaisseur. Ils représentent les plus petits éléments de détection : à toute bande est associé un signal binaire qui signera ou non le passage d'une particule chargée.

- une couronne externe :
  En revanche, la couronne externe, qui est découpée selon les mêmes secteurs que la couronne interne, n'est composée que de 64 bandes par secteurs. Sa surface couvre les rayons de 3.44 cm à 8.64 cm.

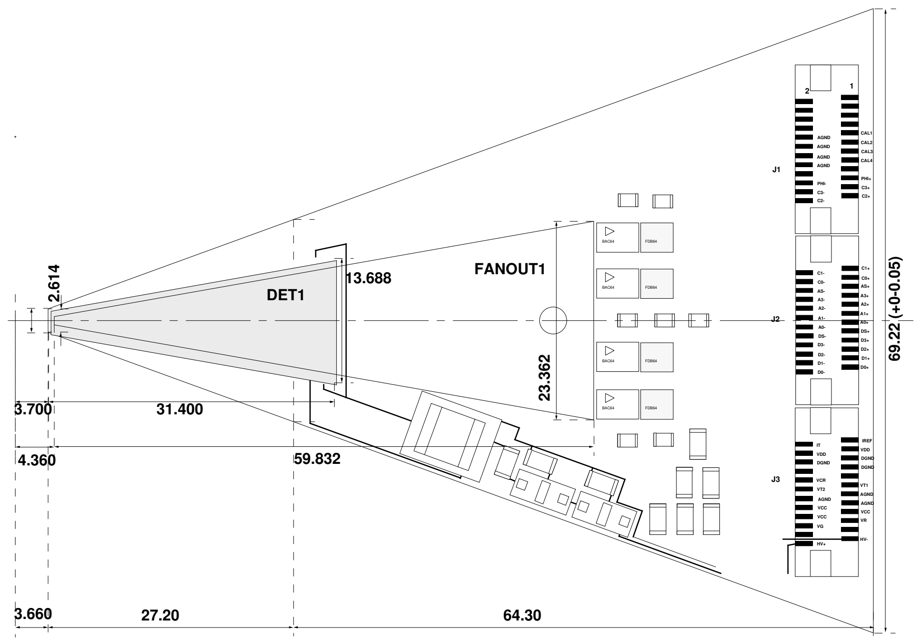


FIG. 2.11 - *Schéma et dimensions (mm) de la carte support des secteurs internes : Board1. La partie grisée (notée DET1) représente la surface de détection en silicium correspondant à deux secteurs internes contigus, et couvre 2×10^o en ϕ. Cette portion de détecteur renferme 2×128 canaux. On note, en prolongation de la partie sensible, l'emplacement de la plaque de verre supportant les pistes de sorties des bandes de silicium (FANOUT1). l'ensemble de cette carte support occupe 40^o en ϕ.*

Les secteurs sensibles, qu'ils soient internes ou externes, sont supportés par des cartes (communément appelées *Board1* et *Board2*) sur lesquelles sont implantés tous les circuits nécessaires à la lecture des bandes de silicium : chaque carte supporte deux secteurs contigus, soit 20^o en ϕ de surface active. Cependant, du fait de l'électronique de lecture, ces cartes, possèdent un emcombrement supplémentaire de 10^o en ϕ, de part et d'autre de la surface active (cf fig. 2.11 et 2.12) .

La superposition des ces deux cartes *"Board1"* et *"Board2"* permet donc d'obtenir une surface active sur 8.64 cm en r.

Comme on le voit sur la figure 2.13, neuf superpositions Board1+Board2, placées côte à côte, ne suffisent pas à couvrir de façon continue les 360^o en ϕ. Pour combler les trous, il faut accoler deux disques comme celui représenté sur la figure 2.13(a), décalés de 20^o en ϕ l'un par rapport à l'autre. Le résultat obtenu est montré vu de face sur la figure 2.13(b).

Chaque détecteur MD1 ou MD2 est le résultat de cet assemblage.
Le détecteur MD1 est situé à 3.4 cm en aval de la dernière ciblette, tandis que MD2 est

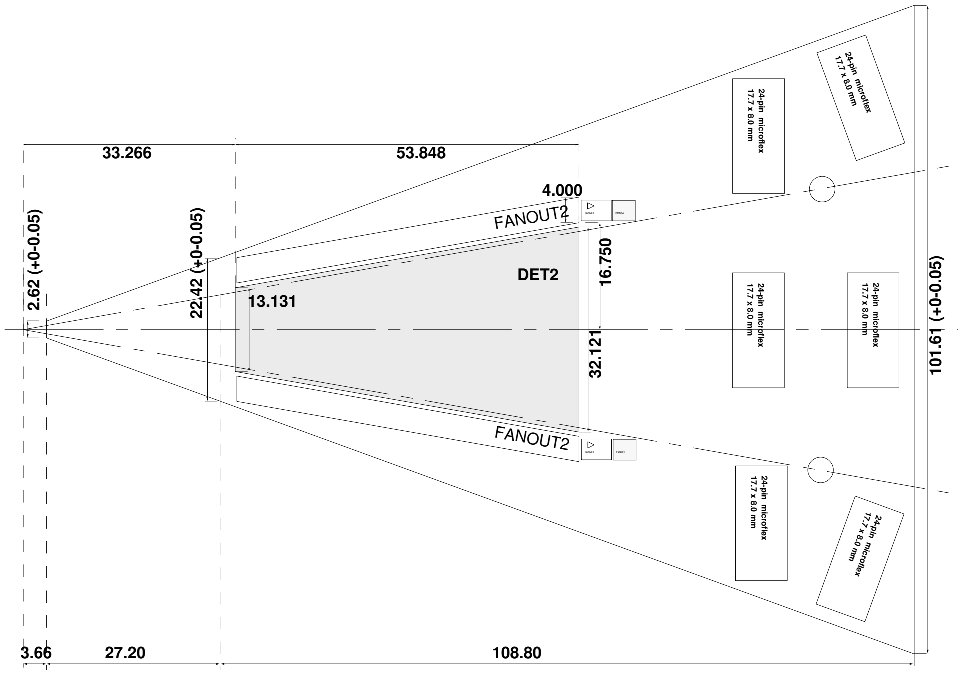
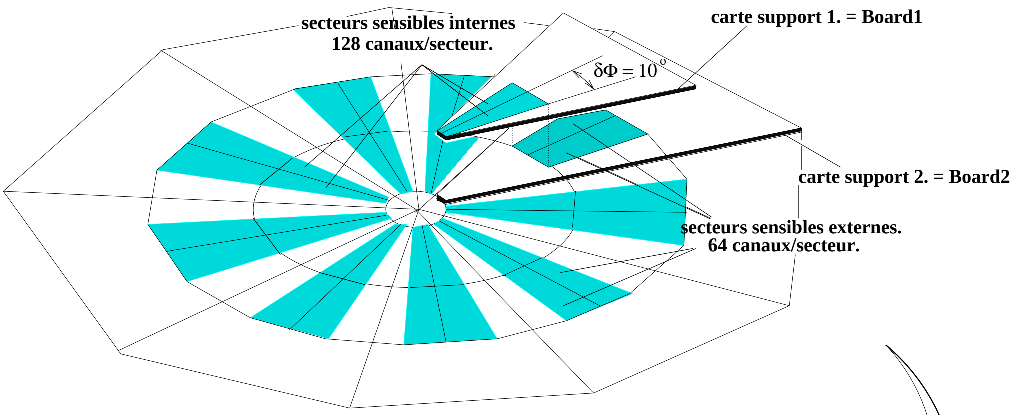


FIG. 2.12 - *Schéma et dimensions (mm) de la carte support des secteurs externes : Board2. La portion de détecteur (grisée notée DET2) est définie par 2×64 canaux. Les couvertures angulaires de la carte support et de la zone sensible (grisée) sont identiques à celles de la carte interne. Ici les plaques de verres supportant les pistes de lectures de bandes de silicium sont disposées de part et d'autre de la partie sensible (FANOUT2).*

a)



b)

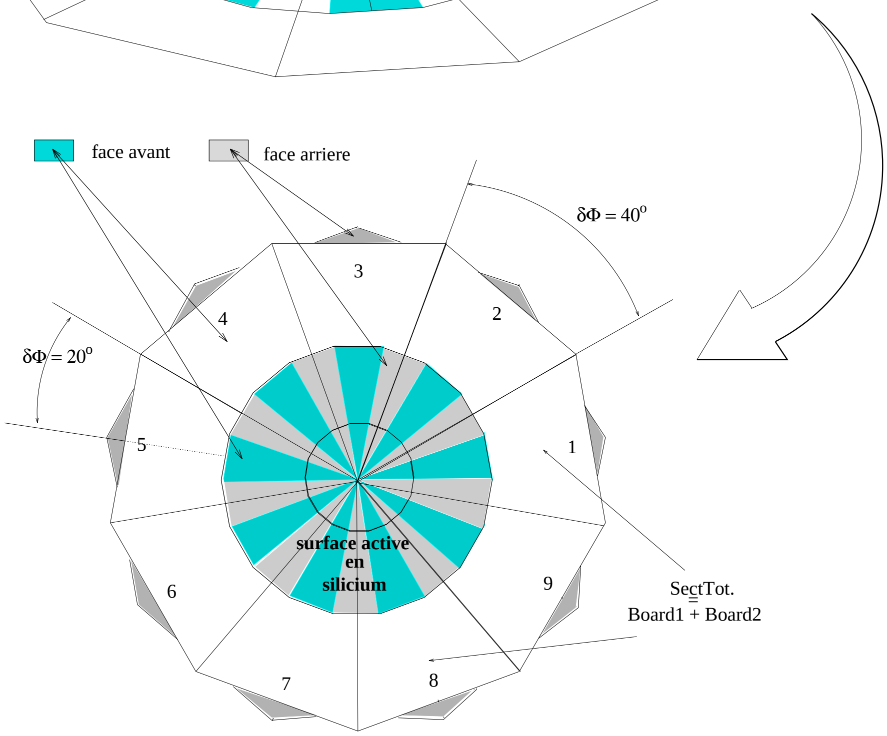


FIG. 2.13 - *Ce schéma nous montre qu'il est nécessaire de superposer deux cartes Board1 et Board2 pour former une surface active complète de 20 degrés en ϕ. Etant donné l'encombrement de* 40^o *en ϕ des cartes support, neuf superpositions Board1+Board2, placées côte à côte, ne suffisent pas à couvrir de façon continue les* 360^o *en ϕ. Pour combler les trous, il faut accoler deux disques comme celui représenté sur la figure (a), décalés de* 20^o *en ϕ. Le résultat obtenu est montré vu de face sur la figure (b) : il s'agit d'un disque, formé de 2 couronnes (interne et externe). La résolution en ϕ est de* 10^o. *Chaque détecteur MD1 ou MD2 est le résultat de cet assemblage.*

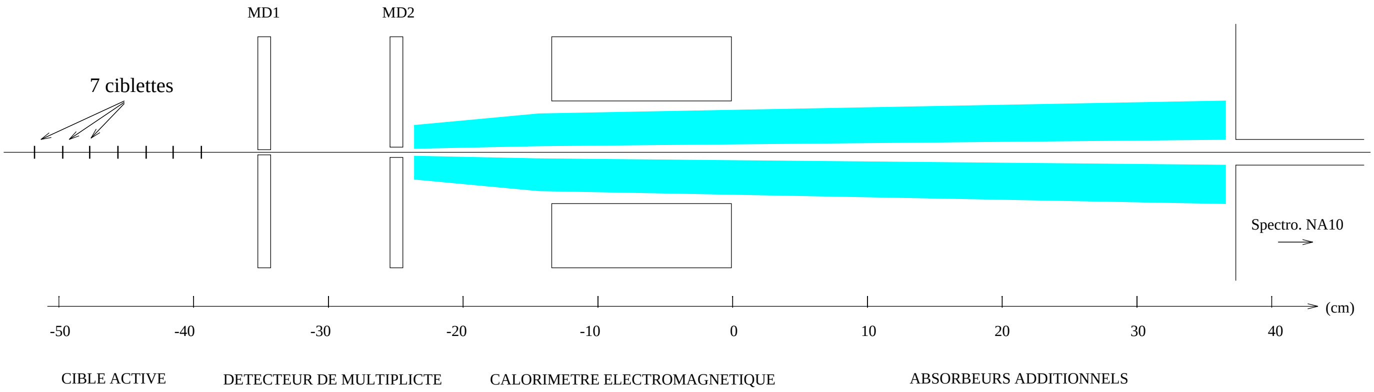


FIG. 2.14 - *Cette coupe longitudinale de la région située autour de la cible active nous montre la position du détecteur de multiplicité (MD1+MD2) à 3 cm en aval de la 7eme ciblette. Les dimensions sont en cm.*

Couronne	R_{int} (mm)	R_{ext} (mm)	η cible 1	η cible 4	Nb canaux par couronne
1	4.4	34.4	2.4-4.4	1.9-3.9	4608
2	34.4	86.4	1.5-2.4	1.0-1.9	2304

TAB. 2.2 - *Ce tableau résume les caractéristiques du sous détecteur MD1. Il montre en particulier les rayons intérieurs et extérieurs des deux couronnes, ainsi que les couvertures en rapidité, assurées pour les ciblettes 1 et 4. De plus, nous voyons que MD1 totalise 6912 canaux de détection.*

placé à 10 cm derrière MD1 (cf figure 2.14). Le premier disque permet de compter les particules chargées issues des interactions Pb-Pb dans les premières ciblettes. Le second joue le même rôle pour les interactions dont le vertex est localisé dans la deuxième moitié de la cible. Cette répartition du comptage sur deux éléments identiques, rend plus homogène la détection en θ que par un disque unique.

Les principales carractéristiques de cet ensemble sont résumées dans les tableaux 2.2 et 2.3.

La largeur des bandes de silicium des secteurs internes et externes (au nombre de 128

Couronne	R_{int} (mm)	R_{ext} (mm)	η cible 4	η cible 7	Nb canaux par couronne
1	4.4	34.4	2.5-4.6	1.8-4.1	4608
2	34.4	86.4	1.6-2.5	1.1-1.8	2304

TAB. 2.3 - *Caractéristique du sous détecteur MD2.*

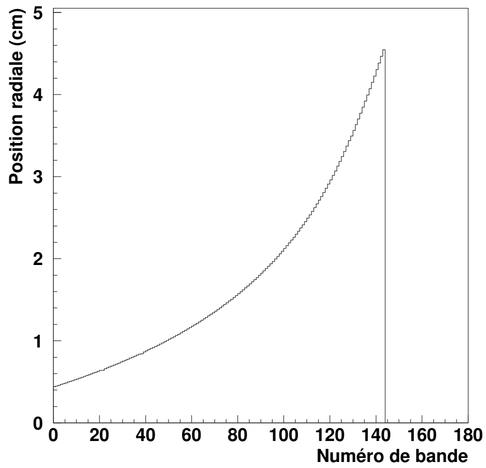


FIG. 2.15 - *Cette courbe représente les positions radiales (distance à l'axe du faisceau en cm) des 144 premières bandes de silicium pour un secteur donné. La concentration des bandes est d'autant plus importante que l'on est proche de l'axe du faisceau.*

et 64 resp.) a été optimisée à l'aide du programme GEANT. Elle est représentée sur la figure 2.15, pour les 144 premières bandes de silicium, en partant de l'axe du faisceau. On observe que la concentration des bandes est d'autant plus importante que l'on est proche de l'axe du faisceau.

Cette segmentation particulière donne au détecteur une résolution d'autant plus grande que l'on est proche de l'axe de référence. Cela est dû à la concentration préférentielle des particules secondaires chargées à faible r.

Les signaux générés par le détecteur sont finalement conduits jusqu'aux circuits de lecture par des pistes d'aluminium de 12 μm disposées sur une plaque de verre de 300 μm d'épaisseur [14].

Pour chaque carte support, ces plaques sont, soit :

- superposées aux secteurs sensibles de la couronne interne. La figure 2.11 montre une plaque notée *FANOUT1*.
- situées de part et d'autre des secteurs sensibles de la couronne externe. Elles sont notées *FANOUT2* sur la figure 2.11.

La contrainte d'ordre mécanique due à l'utilisation de ces plaques, ne permet pas une résolution inférieure à 10 degrés en ϕ. En effet, en dessous de cette valeur, l'agencement

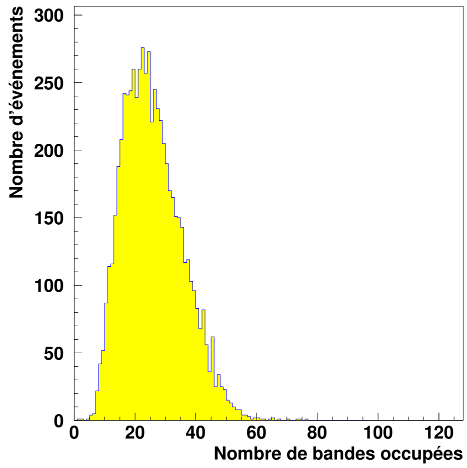


FIG. 2.16 - *Occupation moyenne des secteurs de la couronne interne de MD1. Le nombre total de bandes de silicium étant de 128 pour les secteurs internes, cette distribution montre clairement une occupation maximum de 25 bandes sur 128. La couronne interne, opère donc très fréquemment avec 19% de taux d'occupation.*

des pistes de lecture du silicium (fanout) devient inextricable.

Le taux d'occupation de ce détecteur est défini comme étant la proportion de bandes de silicium touchées par les particules secondaires lors d'une collision. Les particules chargées, produites par les collisions Pb-Pb, conduisent à un taux d'occupation du détecteur de l'odre de 19% pour les collisions périphériques. Il peut atteindre la valeur de 40% pour des collisions centrales.

La figure 2.16 illustre ce taux d'occupation pour les secteurs de la couronne interne, pouvant rescencer chacun au maximum 128 impacts. Cette distribution a été obtenue avec la prise de données de novembre 1995.

2.8 Les hodoscopes additionnels

2.8.1 L'hodoscope de faisceau

Nous avons vu que le signal de déclenchement de cette expérience est une paire de muons détectée dans le spectromètre. Cette double trace doit cependant signer une interaction unique dans la cible.
Pour un événement donné, l'empilement d'ions dans le faisceau est un phénomène qui peut être à l'origine d'interactions multiples : le rôle de l'hodoscope de faisceau est d'identifier les événements *empilés*.

Dans le faisceau incident, avant l'interaction, deux ions successifs sont séparés par un intervalle de temps δt. On parle d'empilement d'ions dans le faisceau (ou "Pile Up Faisceau (PUF)" [17]), lorsque δt est inférieur au pouvoir de séparation de l'ensemble des détecteurs, soit 40 ns. Une telle répartition en temps des ions incidents favorise des interactions multiples que l'on ne pourra pas séparer. Il faut donc être en mesure de signer des événements PUF.

Pour contrôler la répartition en temps des ions dans le faisceau incident, on utilise l'hodoscope de faisceau, noté BH pour "Beam-Hodoscope". Ce détecteur, est placé à 33 mètres en amont de la cible. Il est constitué de deux plans P1 et P2 accolés, de 16 et 14 scintillateurs (cf Fig 2.17), perpendiculaires à l'axe du faisceau.

Tout ion traversant ces plans signe son passage d'une impulsion lumineuse dans un élément de P1 et/ou de P2. Les scintillateurs du plan P2 sont décalés par rapport à ceux de P1, afin d'augmenter l'efficacité de détection. Un troisième plan de scintillateurs, amovible, permet de mesurer l'efficacité de l'ensemble qui atteint 95%. Les scintillateurs sont lus par des convertisseurs numériques en temps (TDC), d'une profondeur de 60 ns et d'une résolution de quelque pico-secondes. Ce module permet de localiser *la position en temps* des ions dans cet intervalle de 60 ns.

Dans la pratique, si l'on considère un flux incident de 5×10^7 particules/s, et pour une porte d'échantillonage de 40 ns, 46% des échantillonages sont réalisés sans PUF, 54% avec PUF (double,triple et quadruple) [17].
L'empilement des interactions est de 10% pour cette même fenêtre.

2.8.2 L'hodoscope d'interaction

Etant donnée l'intensité du faisceau utilisé (5.10^7 ions/s), un certain pourcentage de particules interagissent au sein même de l'hodoscope de faisceau précédemment décrit. Les fragments engendrés par ces collisions polluent le faisceau qui dessert la cible et perturbent ainsi les mesures. Cet hodoscope (BHI pour "Beam Hodoscope Interactions"), situé juste derrière l'hodoscope de faisceau a donc pour but de mesurer l'éventuelle apparition de particules secondaires créées par ces interactions parasites. Il est constitué de bandes de scintillateurs positionnées hors de l'axe du faisceau.

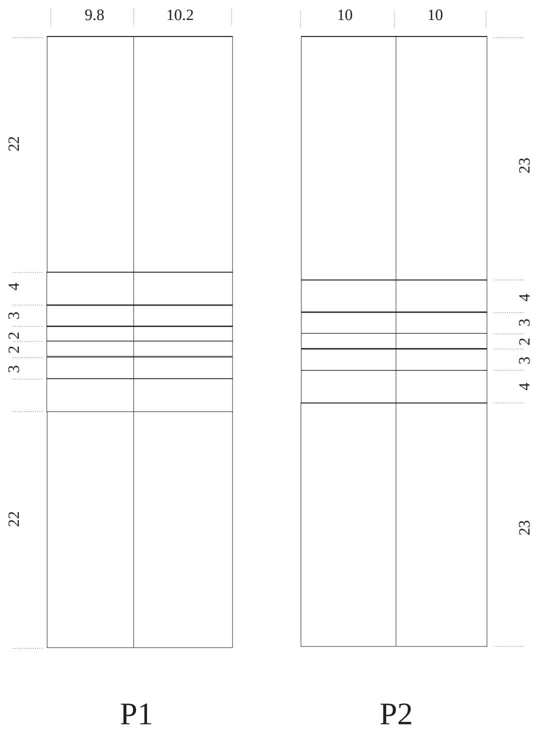


FIG. 2.17 - *Schéma et dimensions (mm) des deux plans P1 et P2 de l'hodoscope de faisceau (BH pour Beam Hodoscope). Ces deux plans de détection sont placés à 33 m avant la cible, sont perpendiculaires à l'axe du faisceau. Ils permettent de détecter les empilements d'ions dans le faisceau qui faussent les mesures effectuées en aval.*

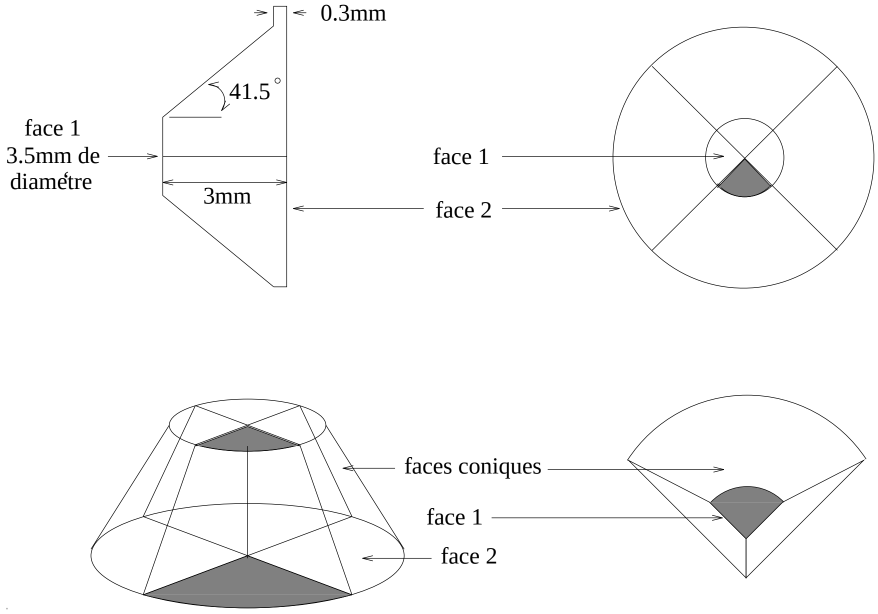


FIG. 2.18 - *Schéma et dimensions du détecteur Beam In (BI). Ce compteur Cerenkov en quartz, situé 90 mm avant la cible permet de connaître le centrage du faisceau sur celle-ci.*

2.8.3 L'hodoscope d'entrée

Nous avons vu que chaque élément de la cible fragmentée, présente à l'axe théorique du faisceau une surface de $1.5 \times 1.5\ mm^2$. Ajoutée à cette faible valeur, la longueur totale de la cible égale à 15 cm, impose un alignement sévère du faisceau le long de l'axe des 7 ciblettes.

Pour contrôler cette visée, nous utilisons un détecteur centré sur l'axe de la cible : le BI pour *Beam In*, localisé à 90 mm avant la cible active. Il s'agit d'un compteur Cerenkov en quartz (cf fig 2.18), formé de quatre quadrants, qui présente dans sa globalité une géométrie de cône tronqué. Chaque quadrant, au passage des particules du faisceau, receuille de la lumière par effet Cerenkov. Une production de signal plus élevée dans certains quadrants traduit donc un mauvais centrage de l'axe du faisceau. Ce dispositif assure ansi un suivi *en ligne*[3] de la position du faisceau. Une étude approfondie du fonctionnement de ce détecteur est disponible dans le document [16].

3. En ligne (on-line) signifit durant une prise de données

Chapitre 3

LE TRANSFERT DES DONNEES DANS NA50

3.1 Introduction

D'une manière générale, les détecteurs du dispositif expérimental NA50, sont échantillonnés par des circuits de lecture qui placent les informations receuillies en mémoire pour constituer des événements. Cette mémoire tampon est ouverte sur un dispositif de transfert. Celui-ci permet de faire le lien avec une mémoire finale déstinée à servir un système d'enregistrement sur support magnétique (cassette, disque). La mémoire tampon peut contenir de un à plusieur milliers d'événements. La figure 3.1 résume le chaînage de ces divers éléments.

Dans ce chapitre, nous allons voir qu'il existe plusieur chaînes, identique à celle décrite figure 3.1, qui peuvent fonctionner en parallèle. Une description du circuit transputer nous permettra de voir qu'il s'agit d'un composant tout à fait adapté à ce type de structure. Enfin, nous donnerons l'exemple du réseau de transputer qui contrôle le transfert de la moitié des données sur NA50.

3.2 Vers un système géré par transputers

3.2.1 La répartition des données

Avant toute description du système de transfert, il est important répertorier les sources de données. En effet, ce sont elles qui vont alimenter les mémoires tampon. Le nombre de sources et leur débit respectif vont donc influencer la structure même du dispositif de transfert.

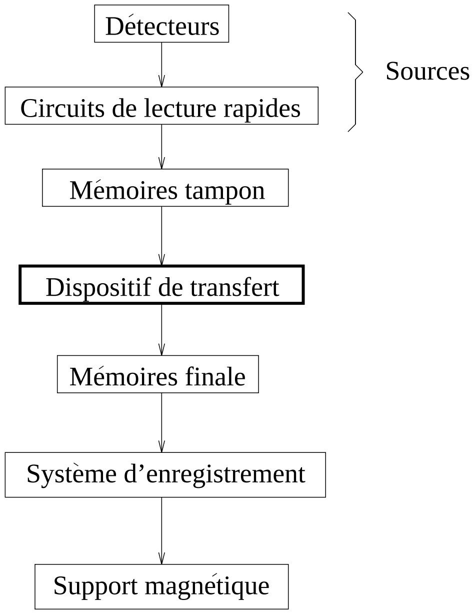


FIG. 3.1 - *Cette figure illustre le chaînage général des éléments permettant d'assurer les prises des données sur l'expérience NA50.*

L'ensemble des informations générées par le dispositif expérimental NA50, provient de six sources différentes :

- 1 les quatre hodoscopes R1-R4 du spetromètre à muons,
- 2 les huit chambres à fils CP1-CP8 du spetromètre à muons,
- 3 le calorimètre électromagnétique,
- 4 le calorimètre à zéro degré ZDC,
- 5 les détecteurs additionnels,
- 6 le détecteur de multiplicité,

En comparaison avec NA38, l'utilisation d'un faisceau d'ions Pb^{208} plutôt que O^{16} ou S^{32}, entraîne un acroissement d'un facteur 10 de la fréquence des déclenchements (plus grande luminosité). De plus, l'apparition des nouveaux détecteurs (multiplicité et ZDC) double le volume des données à prendre en compte.

Ainsi, la charge imposée au dispositif de transfert, qui sur NA38 s'élevait à 800ko/bouffée tous détecteurs confondus, peut atteindre sur NA50 les valeurs de :

- 2Mo/bouffée pour les chambres proportionnelles à fils (MWPC).

- 2Mo/bouffée pour les hodoscopes du spectromètre.
- 4Mo/bouffée pour les détecteurs additionnels, le calorimètre électromagnétique et le calorimètre à zéro degré.
- 8Mo/bouffée pour le détecteur de multplicité.

Toutes ces valeurs correspondent aux volumes maximum attendus pour 5000 événements par bouffée, soit un tolal de 16Mo d'informations (cycle SPS renouvelé toute les 15 s environ).

Sur NA50, la philosophie essentielle est de transférer toutes ces données vers les mémoires finales, bouffée par bouffée et non pas événement par événement. Ce choix élimine les temps morts, introduits à chaque événement par le système de transfert, lors des accés aux mémoires tampon. Ces mémoires tampon doivent cependant être capables de contenir l'intégralité d'un déversement. D'autre part, pour les dispositifs de transferts installés, nous avons choisi de dupliquer toutes les mémoires finales : un groupe de mémoires *ACQUISITION* et l'autre *ESPION*. Cette dualité nous permet de réserver le bloc *ACQUISITION* au dispositif d'enregistrement sur bandes, et de disposer du bloc *ESPION* pour réaliser des études de données en-ligne. Ce dernier point est particulièrement intéressant puisqu'ils permet aux groupes d'analyse d'étudier les détecteurs pendant les prises de données. Il est ainsi possible d'effectuer d'éventuelles corrections hardware, tant au niveau de l'électronique de lecture qu'au niveau des parties actives des détecteurs, et ce, sans attendre le dépouillement définitif des données. L'accés au données en-ligne permet aussi de mettre au point les algorithmes d'analyse. Dans la suite de ce document nous verrons que le bloc *ACQUISITION* est formé de cartes *B016ACQ* tandis que le bloc *ESPION* est formé de cartes *B016SPY* (voir figure ??).

Pour tout le dispositif expérimental, quatre catégories de mémoires tampon sont utilisées afin stocker les données d'un déversement complet :

- La mémoires tampon du spectromètre à muons, formée de deux modules MEMRMH [20] visibles VME. L'un d'entre eux est dédié aux données des quatre hodoscopes, et l'autre aux données des huit chambres à fils. Ces mémoires ont la capacité de conserver l'intégralité d'un déversement dans une limite de 5000 événements. Ce n'est qu'en fin de burst que ces mémoires seront prises en charge pour être transférées vers le système d'enregistrement.
- La mémoire des deux calorimètres, et de quelque détecteurs additionnels. Elle est formée par les mémoires FIFO [1] des modules de conversions QDC et TDC chargés de numériser les signaux de ces détecteurs. Ces FIFO peuvent absorber 3000 événements. Pour assurer la totalité des conversions en charges (QDC) et en temps (TDC), 27 modules de 8 FIFO sont nécessaires. Comme pour la MEMRMH, ces mémoires sont accéssible par le bus VME.

1. pile mémoire First In First Out

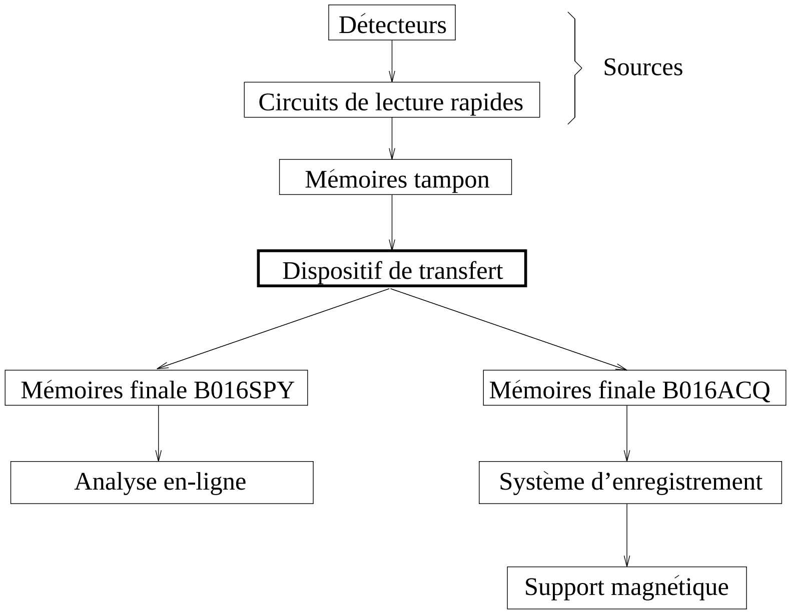


FIG. 3.2 - *Comme le montre cette figure, sur NA50, nous avons choisi de dupliquer les mémoires finales en deux blocs ACQ et SPY tout deux formés de cartes B016 (décrite au chapitre 5 paragraphe 8).*

- La mémoire des échelles. Pour vérifier le bon fonctionnement des détecteurs, il est nécessaire de procéder à des opérations de comptage. Ceci permet en particulier de connaître le nombre d'interactions par particule du faisceau, de contrôler la stabilité du flux de particules, de compter le nombre de déclenchements, le nombre de fronts montants etc... On utilise pour cela des modules compteurs VME (communément appellés échelles ou scaler) : ils comptent le nombre de transitions logiques appliquées en entrée du module. Les résultats sont enregistrés dans la mémoire du module et sont accéssibles via le bus VME : c'est la catégorie des mémoires échelles. L'ensemble de cette mémoire est constitué de 40 échelles VME de 16 compteurs chacune.

- La mémoire du détecteur de multiplicité. Formée de 6 blocs répartis sur 6 modules BUSIF, nous verrons qu'elle n'est accessible en lecture que par les circuits transputers. Elle a la capacité de recevoir 5400 événements multiplicité.

Pour faire le lien entre ces mémoires tampon et les mémoires finales, nous devons donc gérer le transfert des données provenant d'une soixantaine de modules indépendants les uns des autres. Cette multiplicité des mémoires sources et leur carractère indépendant appel à une gestion simultané des transferts : construction de plusieur lignes (ou branches) de communications parallèles, entre mémoires tampon et mémoires finales. Il s'agit simplement d'une parallélisation du schéma 3.1 proposé en introdution. Pour chaque cycle SPS, les lignes ainsi établies doivent cependant être en mesure de supporter la charge de transfert globale de 16 Mo.

Pour de tels mouvements de données, nous allons voir que le transputer est un circuit tout à fait adapté. En effet, il intègre toute les fonctionnalités permettant d'aboutir à la réalisation de structures parallèles, capables de soutenir des communications intenses ; c'est une des voies que j'ai exploré sur NA50 avec les circuits transputers.

3.3 Intérêt du transputer

La caractéristique principale d'un circuit comme le transputer est d'être, de part sa construction particulièrement dédié au parallélisme. Ce concept renferme cependant deux types d'approches :

- le parallélisme réel.
- le parallélisme virtuel ou pseudo-parallélisme.

Le premier point est fondé sur des architectures multiprocesseurs, où chaque transputer est dédié à une tâche : on parlera de réseau de transputers. Un ensemble d'opérations s'exécutent ainsi de façon simultanée procurant un gain de temps par rapport à un système monoprocesseur. Le second point est, quant à lui monoprocesseur, et permet un développement multitâches procurant une souplesse de programmation due en partie à l'asynchronisme qu'il introduit. Le développement d'une architecture transputers englobe donc ces deux approches du parallélisme. Nous y reviendrons aux paragraphes 3.5 et 3.6. D'autre part, dans la suite de ce document, nous parlerons indifféremment de processeur ou de transputer. De même, le terme de processus sera employé pour désigner une tâche implantée sur un transputer.

Un transputer est un microprocesseur multitâches disposant d'une mémoire locale interne, et de quatre liens. Les liens permettent d'établir des connections point à point entre transputers : ce sont les voies de communications entre les circuits.
L'utilisation des liens permet de construire des réseaux quelconques de transputers, de tailles et de topologies arbitraires.

La figure 3.3 schématise les principaux types de réseaux [27], que l'on peut construire au moyen de transputers. Nous verrons aux paragraphe 3.9 et chapitre 5 que, sur NA50, le transfert des données est géré par deux réseaux distincts : le réseau dit *général* et le réseau de *multiplicité*. La figure 3.4 nous montre que l'un s'apparente au type cellulaire tandis que l'autre est un réseau en arbre.

Les principaux avantages qui découlent de l'utilisation de réseau de transputers sont que :

- l'utilisateur n'a pas à se préoccuper du mécanisme de communication, celui-ci est intégré de façon hardware au niveau du composant.

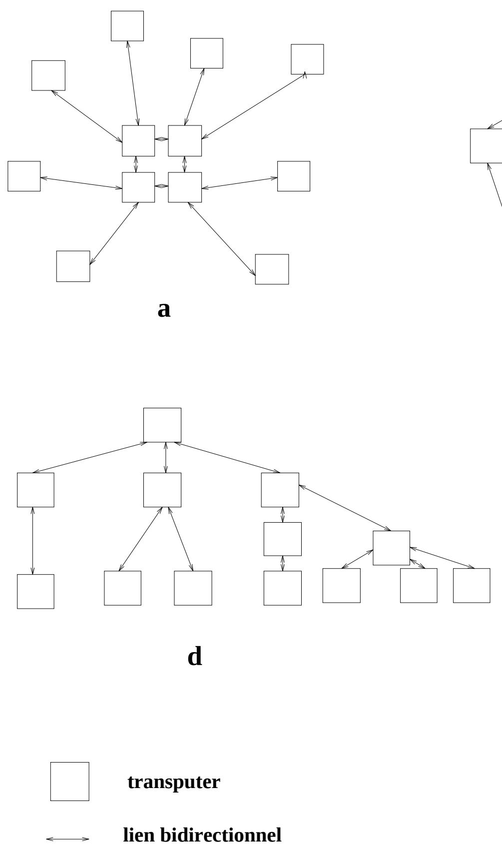


FIG. 3.3 - *Exemples de réseaux de transputers. Les différentes topologies représentées sont : le réseau en étoile (a), le réseau maillé (b), le réseau en boucle (c), le réseau en arbre ou hiérarchique (d), et le réseau cellulaire (e). Les transputers sont les neouds du réseau, et les mailles sont matérialisées par les liens.*

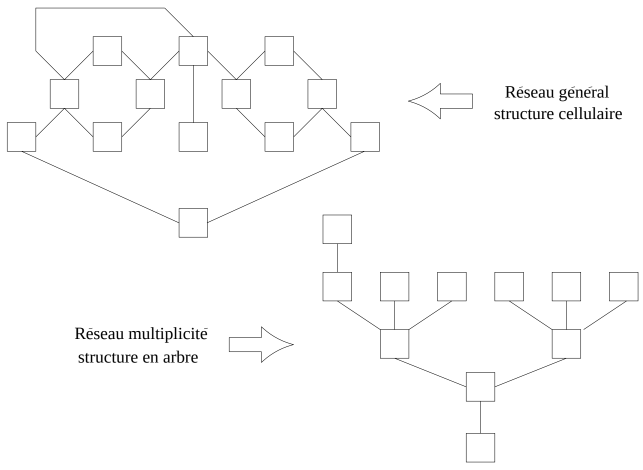


FIG. 3.4 - *Sur NA50, deux réseaux de transputers sont utilisés pour le transfert des données : un réseau de type cellulaire et un réseau en arbre.*

- La charge capacitive de chaque lien ne devient pas pénalisante lorsque l'on fait croitre la taille du réseau.
- La bande passante des transferts n'est pas saturée lorsque le nombre de transputers augmente. Au contraire, plus ce nombre est grand, plus la capacité de transfert globale est grande.

Ces trois points font du transputer un bon candidat face aux bus multiprocesseurs, en ce qui concerne le transfert des données. C'est la raison pour laquelle il a été retenu pour constituer le noeud élémentaire du dispositif de transferts des données.

Dans l'exprérience NA50 l'intégralité des dispositifs de transferts a été conçue au moyen transputer de type T800 et T400. Nous décrivons dans les sections suivantes les fonctionnalités d'un circuit T801.

En raison des dates butoires des premières prises de données (novembre 1994), l'utilisation des circuits de la génération T9000 [25], bien qu'a priori beaucoup plus efficace n'a pu être mise en oeuvre.

3.4 Description du transputer T801

Dans cette section, nous décrivons brièvement l'ensemble des éléments internes du transputers. Nous allons voir qu'un certain nombre d'entre eux peuvent fonctionner simultanément ce qui augmente les performances de la gestion multitâches. Le lecteur désireux d'approfondir les mécanismes propre à ce circuit pourra consulter les ouvrages [30], [32] et [33].

3.4.1 Bloc diagramme d'un transputer T801

Le T800 est un circuit développé par la société INMOS. Il intègre principalement (cf figure 3.5):

- un microprocesseur 32 bits.
- une unité de calcul en virgule flottante (FPU) de 64 bits.
- 4Ko de mémoire cache.
- un système de gestion multiprocessus (scheduler) microprogrammé.
- quatre pilotes de liens standard INMOS.

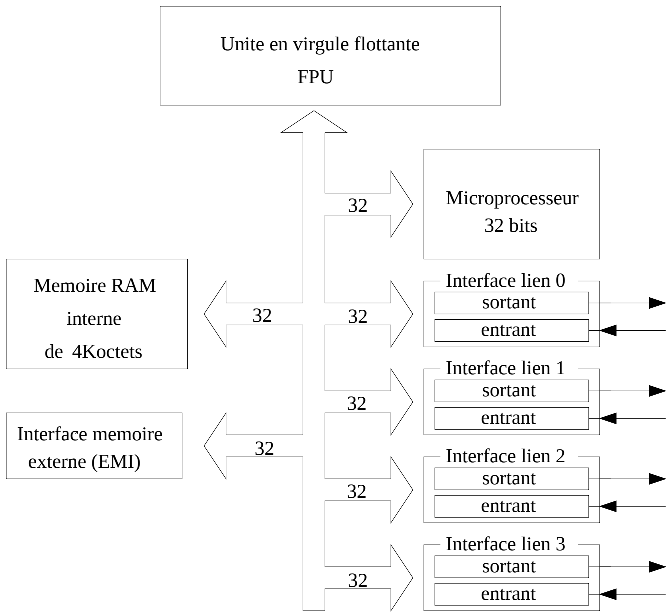


FIG. 3.5 - *Bloc diagramme simplifié d'un transputer T800.*

3.4.2 Le microprocesseur

Il s'agit de l'unité centrale du transputer (CPU). Une des propriétés remarquables de ce microprocesseur vient du fait qu'il intègre un scheduler microcodé [30] . Il s'agit d'une unité qui gère l'execution d'un nombre quelconque de tâches concurrentes grace à un partage du temps CPU. Toute la gestion multitâches ne demande donc pas d'un noyau logiciel. Outre cette unité, le CPU possède une unité arithmétique et logique qui lui permet d'effectuer toutes ses opérations de calcul.

L'unité centrale possède six registres qui sont utilisés pour l'exécution des séquences d'instructions :

- le pointeur d'espace de travail *WS*, qui pointe sur l'espace mémoire contenant les variables locales de la tâche courante.
- le pointeur d'instruction qui pointe sur la nouvelle instruction à exécuter.
- le registre d'opérande, utilisé pour former les opérandes d'instructions.
- les trois registres *A, B, C* qui forment la pile d'évaluation.

Les registres *A, B, C* sont les sources et destinations pour la plupart des opérations de l'unité arithmétique et logique (ALU) du CPU. L'ensemble des instructions d'un transputer est réalisé au moyen d'un code assembleur qui peut être soit direct soit indexé. Nous ne développerons pas ici les caractéristiques de ce langage bas niveau et renvoyons le lecteur aux ouvrages [30] et [31] pour plus de détails.

Ce microprocesseur 32 bits fonctionne à la fréquence de 30MHz. Il accède directement à la mémoire RAM interne de 4Koctets implantée dans le circuit. Elle lui permet de stocker les instructions et données du programme fréquemment utilisées, et optimise ainsi la vitesse d'exécution d'un processus :

- 25 MIPS (Millions d'Instuctions par Seconde) pour un T801.
- 30 MIPS pour un T805.

L'External Memory Interface (EMI) permet au microprocesseur d'accèder à un espace mémoire externe de 4Goctets. Ce module prend en charge tout le protocole de lecture/écriture en mémoire. Il est capable de soutenir un flux de 50 Moctets/s, s'il n'est pas pénalisé en temps par les blocs mémoires qu'il interface. Cette interface est exploitée pour augmenter l'espace de travail du processeur.

3.4.3 Le FPU

L'unité de calcul en virgule flottante de 64 bits est capable de fournir un flux d'opérations de 1.8Mflops à 25MHz. C'est cette unité qui permet d'effectuer des calculs mathématiques en virgule flottante. Elle possède la propriété de pouvoir travailler simultanément avec l'unité centrale. Elle complète l'unité arithmétique et logique (ALU). Toutes les communications entre cette unité et la mémoire sont supervisées par le microprocesseur.

3.4.4 Les interfaces liens

Comme le montre la figure 3.5, un transputer possède quatre interfaces qui permettent de créer des liens avec d'autres transputers. Un lien, par nature bidirectionnel, est formé d'une ligne entrante et d'une ligne sortante. Deux *sous-interfaces* (entrante et sortante) sont chargées de gérer les flux opposés propres à ces deux lignes. Entre deux transputers, un lien est établi au moyen d'un simple cable de deux fils reliant les interfaces des deux circuits (cf figure 3.6).

La communication par lien permet à deux circuits d'échanger des informations et d'assurer la synchronisation de leurs programmes.

Ces interfaces sont responsables de la communication avec l'extérieur du circuit. Elles accédent à la mémoire en mode DMA (Direct Memory Access). Ce type d'accès, géré indépendamment de l'unité centrale, permet d'exécuter simultanément des opérations de communication et des tâches purement CPU [23].

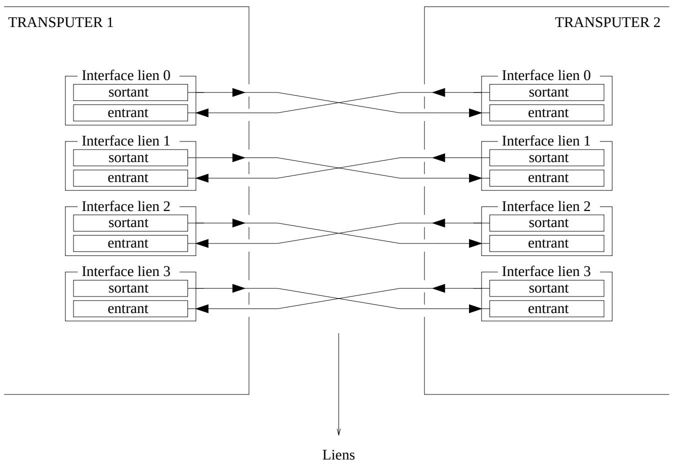


FIG. 3.6 - *Les liens entre transputers sont établis en interconnectant leurs interfaces liens. La connection est réalisée au moyen de simples cables à deux fils.*

La communication proprement dite est du type série et peut fonctionner suivant deux modes : le mode à 10 Mbits/s et le mode à 20 Mbits/s. Elle ne dépend pas des performances du processeur, ce qui permet de connecter différents types de transputers entre eux.

Chaque lien a la capacité de faire du transfert à 2.35Moctets/s sur courte distance. Il s'agit là de la valeur du flux bidirectionnel, en mode 20Mbit/s, sur une distance inférieure à 30 cm. Nous verrons par la suite que divers facteurs font chuter cette valeur lorsque la longueur des lignes de transfert augmente.

Sur NA50, c'est ce type de voies de communication qui nous permet de déplacer les données des différents détecteurs, depuis la mémoire tampon vers la mémoire finale.

3.5 Les processus

Le transputer est un système à temps partagé, qui constitue un support pour le modèle de concurrence et de communication OCCAM. Ce langage OCCAM [28], [29] sera brièvement décrit au paragraphe 3.6.4.

3.5.1 La gestion multitâches par le transputer

Un processus est une entité software, qui exécute des séquences d'instructions au même titre qu'un programme classique. Sur un transputer, il est possible de *juxtaposer* l'exécution de plusieur processus. La ressource CPU etant unique, il est clair qu'une exécution simultanée de n tâches est impossible : le circuit peut procéder à une exécution en temps partagé qui demande une gestion particulière du temps CPU. On parle d'exécutions concurrentes des tâches.

Cette gestion requiert un mécanisme permettant au microprocesseur de décider quel processus il va exécuter : c'est le noyau, qui permet aussi la création de ces processus. D'autre part, un second mécanisme doit allouer (automatiquement ou via l'utilisateur) le temps microprocesseur aux processus. Il s'agit de l'allocateur de temps (ou scheduler). Dans un système classique à temps partagé, le temps CPU doit être distribué en laps suffisamment courts pour permettre à chaque processus de disposer fréquemment de toute la ressource. Ces deux mécanismes présent au sein même de l'unité CPU sous forme de microcode, font du transputer un circuit multitâches. Le nombre de tâches que peut gérer un transputer n'est pas limité.

Un processus pris en charge par un transputer peut prendre deux états :

- **l'état actif :** le processus est en cours d'exécution (processus courant) ou en file d'attente pour être exécuté ultérieurement.
- **l'état inactif :** le processus est prêt pour émettre ou recevoir une information sur une voie de communication, ou bien désactivé pour un temps déterminé. Un processus désactivé est tel qu'il n'utilise pas le temps du microprocesseur.

La figure 3.7 représente la gestion concurrente de quatre processus **P1**, **P2**, **P3** et **P4** par un transputer. La file d'attente des processus actifs est une liste chaînée des espaces de travail associés à ces tâches. Deux registres pointent sur les processus en **Tête** et en **Queue** de liste ce qui complète la définition de la file.
Sur cette figure, **P1** est le processus courant (il utilise le CPU) tandis que **P2**, **P3** et **P4** sont en attente de traitement mais cependant actif. Les éléments de la liste pointent sur les instructions proprement dites de leurs processus. Tout processus courant dispose de l'unité centrale (CPU) pour exécuter ses instructions. Par conséquent, les cinq registres **A, B, C, WS, NextInstr et Opérande** lui sont consacrés.

Le processus courant (en tête de file) sera placé en queue lorsque le pointeur d'instruction rencontrera des opérations du type :

- boucle while
- boucle for
- saut (jump)
- émision sur canal
- attente sur canal

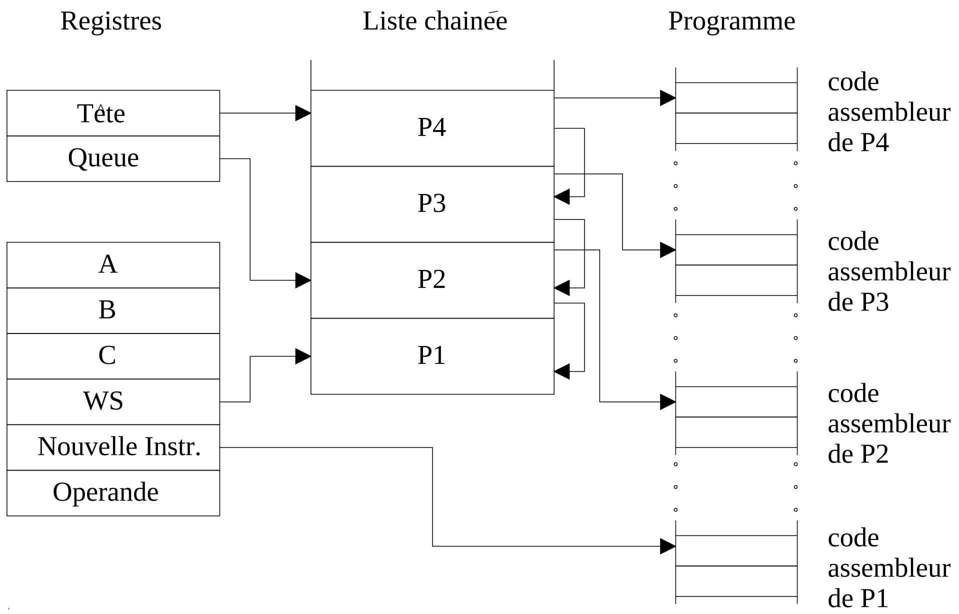


FIG. 3.7 - *Schéma d'une liste chainée de processus concurrents. Dans cet exemple, P1 est le processus courant, tandis que P2, P3 et P4 sont en attente de traitement.*

L'interruption du processus en vue d'une commutation ne se fait donc pas à intervalle de temps régulier, mais dépend des séquences d'instructions présentes dans le corps même du processus actif. Ce mécanisme est complètement transparent pour l'utilisateur.

Une telle gestion, multitâches, permet d'éviter des attentes trop longues dans un programme qui pénalisent le temps global d'exécution : c'est la technique du *pseudo-parallélisme*.

3.5.2 La communication entre processus, les canaux

Bien que pour bon nombre d'applications multitâches, certain processus soient isolés, il est souvent nécessaire d'échanger des informations entre processus concurrents : échanges de résultats, synchronisations, déclenchement. Sur le transputer, ce type de dialogues se fait par l'intermédiaire de canaux. Un canal est une entité logicielle qui permet d'établir une liaison entre deux processus. Elle doit être prédéfinie et permet d'émettre ou de recevoir des informations formatées. Selon que les deux processus sont sur un même transputer ou délocalisés, le canal est dit interne ou externe.

Un canal est implanté entre deux processus locaux par un simple mot en mémoire (canal interne), alors qu'il est rattaché aux registres des interfaces liens pour une communication entre deux transputers (canal externe). Dans les deux cas de figure, un canal

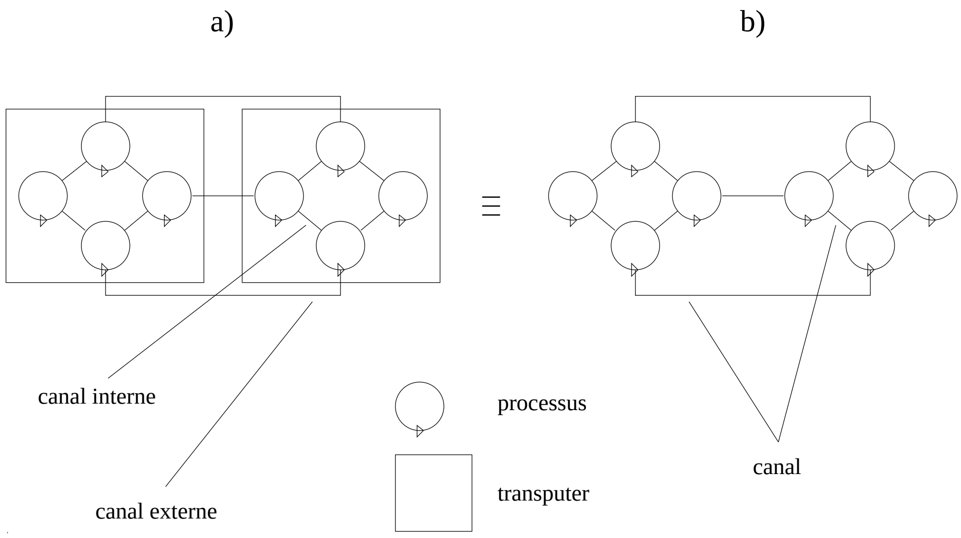


FIG. 3.8 - *L'appartenance des processus aux processeurs devient transparente, du point de vue des communications, tout se passe comme si l'on évoluait au milieu d'un réseau de processus.*

possède une adresse qui signe son appartenance au type interne ou externe. C'est donc en terme d'adresse que fonctionnent toutes les opérations d'entrées/sorties sur canal. De ce fait, du point de vue syntaxique, aucune différence n'apparaît entre une communication interne ou externe.
La figure 3.8 illustre ce point important : les 8 processus rattachés au réseau de deux transputers 3.8(a) peuvent échanger de l'information soit par des canaux internes, soit par des liens (canaux externes). Comme le montre la partie 3.8(b), du point de vue syntaxique il n'existe qu'un seul type de voie de communication : le canal.

Un échange d'informations entre deux processus prend effet lorsqu'à une émission sur l'extrémité d'un canal, correspond une réception à l'autre bout du canal : la communication est un point de rendez-vous pour deux séquences d'instructions (les processus qui dialoguent). Le premier processus qui arrive à ce point de synchronisation doit attendre que le second y parvienne ; le transfert de données est alors amorcé. Il restera bloqué si pour une raison quelconque personne ne se présente au rendez-vous : on dit que les dialogues sur les canaux sont du type bloquants.

Les deux opérations de communication fondamentales que peut exécuter un processus sont :

- la réception d'un message.
- l'émission d'un message.

Bien ces deux opérations soient rigoureusement identique pour des communications internes ou externes au transputer, les mécanismes qui les gouvernent sont différents pour chaque cas.

La communication interne

La figure 3.9 illustre les étapes d'une communication au travers d'un canal interne :

Avant d'activer une émission ou une reception, dans une première étape, le processus P1 arrivant au point de rendez-vous, charge les trois registres A, B, C de la pile d'évaluation avec : un pointeur sur le message (à émettre ou recevoir), l'adresse de la voie de communication et la taille en octet du message à transmettre.
Ensuite, l'adresse de son espace de travail est chargée dans le canal (étape 2). Cet espace contient lui même l'adresse du message à échanger. Le processus devient alors inactif : exclu de la liste chainée, il n'utilise plus de temps CPU.
Le second processus P2 devient courant (étape 3), et pointe sur le canal qui contient l'-adresse du message de son interlocuteur P1 : l'information est copiée, le premier processus P1 est réactivé (étape 3). La quatrième étape consiste à réinitialiser le canal.

Pour une communication interne, c'est le microprocesseur qui gère ces quatre étapes.

Dans le cas d'une communication entre deux transputers, il est necessaire d'associer un support physique au canal : c'est le lien, précédemment décrit, qui assure la connection entre processus distants. Le mécanisme de communication diffère alors du précédent.

La communication externe

Lorsque le pointeur d'instruction du microprocesseur rencontre une opération de transfert sur un canal externe (émission ou réception sur un lien), l'interface lien qui correspond au canal sollicité est déléguée pour effectuer le transfert. Au même moment, l'unité centrale désactive le processus (exclusion de la file d'ordonancement).
Pour accomplir l'opération de transfert, l'interface lien (entrante ou sortante) possède ses trois propres registres internes qui vont jouer le même rôle que les registres A, B et C du microprocesseur lors d'une communication interne.

La figure 3.10 nous montre le déroulement d'une communication sur canal externe qui permet à deux processus *distants* P1 et P2 de dialoguer. Pour cela, chaque transputer suit trois étapes :

*Etape*1 Lorsque le processus (P1 ou P2) exécute une opération de communication sur un canal externe, les 3 registres A, B et C de la pile d'évaluation du microprocesseur contiennent la taille du message, l'adresse du canal et l'adresse du message. Les trois registres propres à l'interface lien sont vides (figure 3.10.Etape1).

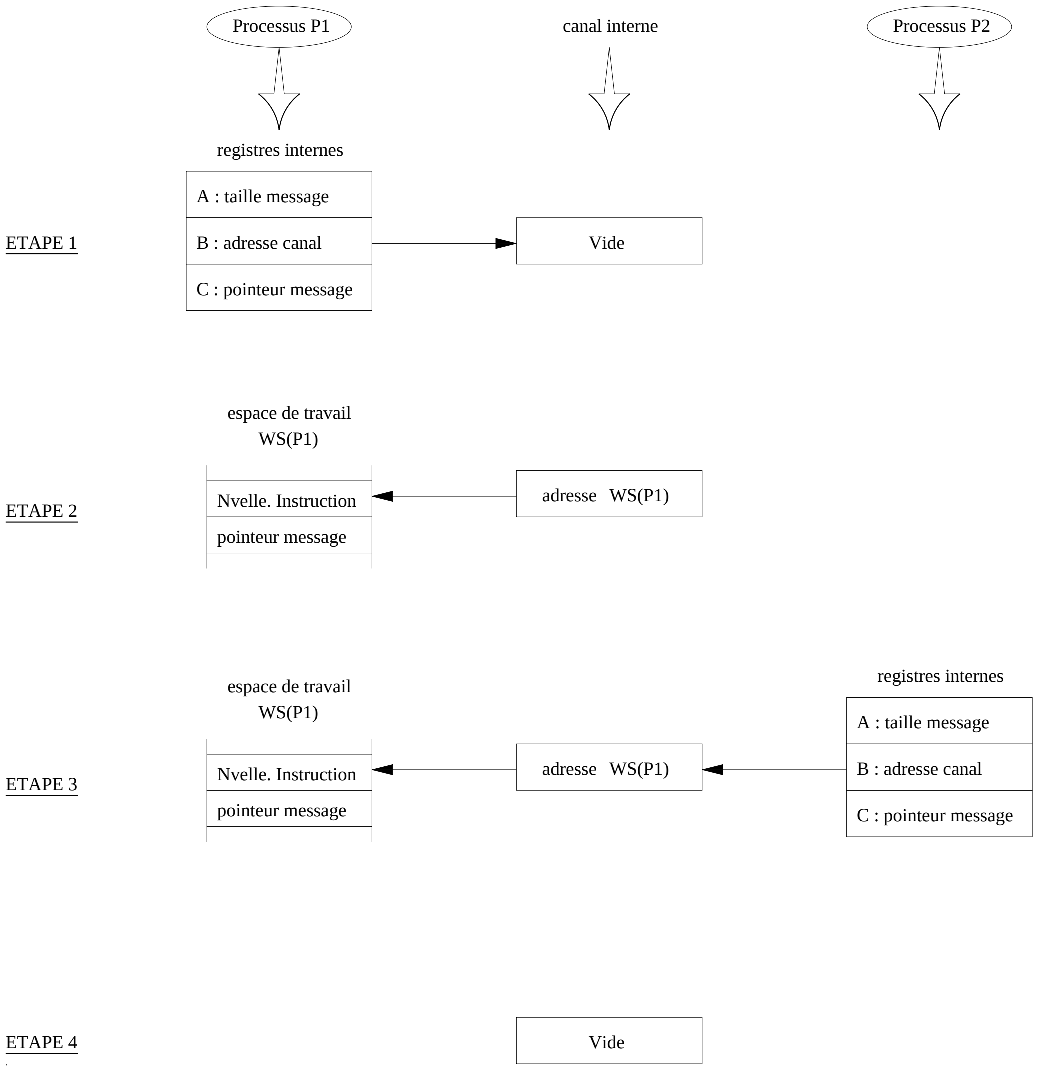


FIG. 3.9 - *Schéma des quatres étapes de communication entre deux processus P1 et P2 via un canal interne.*

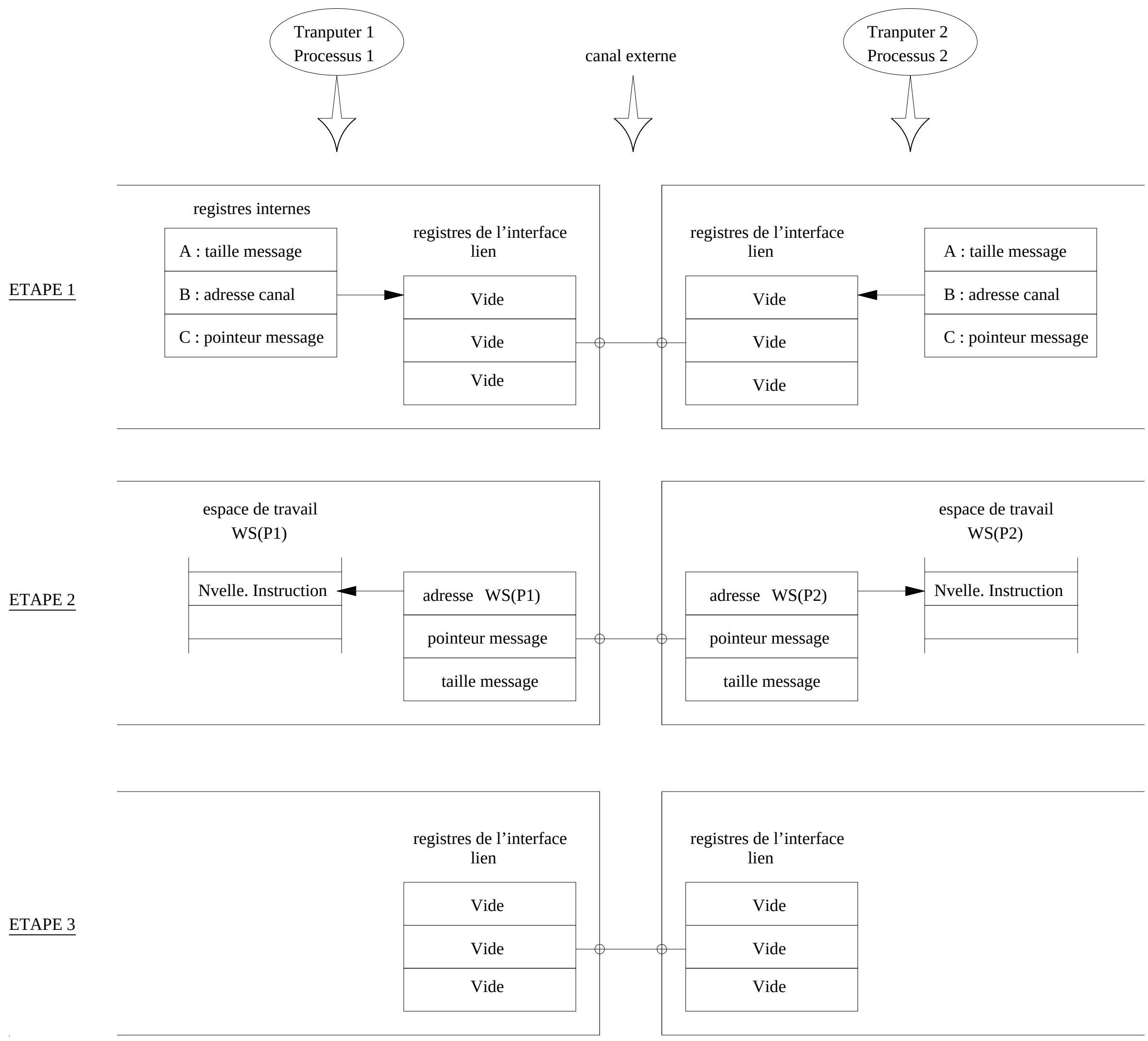


FIG. 3.10 - *Schéma des trois étapes de communication entre deux processus P1 et P2 via un canal externe.*

*Etape*2 Les trois registres de l'interface lien sont initialisés et font le lien avec l'espace de travail du processus WS(P1) et WS(P2). P1 et P2 sont alors exclus de leur file d'ordonnancement : c'est le début de l'opération de transfert, pilotée par l'interface lien et non par le microprocesseur (figure 3.10.Etape2)

*Etape*3 Une fois le transfert terminé, les processus P1 et P2 des deux tranputers sont réintégrés à la file d'ordonnancement, puis les registres d'interface sont réinitialisés (figure 3.10.Etape3)

De cette manière, les quatre interfaces lien peuvent travailler simultanément et assurer ainsi quatre dialogues externes dans un parallèlisme réel. La ressource CPU est alors disponible pour continuer l'exécution d'autres processus.

Le nombre de canaux internes par processus, illimité, permet d'aboutir à la création de véritables réseaux de processus au sein même d'un transputer. Ce point important, vient s'ajouter au fait qu'une communication se fait indifféremment sur un canal interne ou externe. Par suite, il est possible de simuler un réseau multiprocesseurs de taille quelconque sur un simple circuit transputer. Ceci permet de tester toute la logique de l'application et surtout de vérifier les bonnes synchronisations des tâches aux points de rendez-vous. Une fois l'application multitâches monotransputer terminée, le passage au réseau multitransputers se fait par une délocalisation des tâches vers les processeurs disponibles. Cette transformation ne demande pas de grands remaniements et permet d'assouplir considérablement la phase de développement.

Sur NA50, pour le réseau de multiplicité, cette facilité nous a permis de simuler des cartes transputers en fabrication ou en tests et donc de faire croître le réseau de transputers au fur et à mesure de leur disponibilité. La figure 3.11 illustre la croissance de ce réseau.

3.5.3 Le protocole de dialogue et structure de l'information

Le dialogue inter-transputers via les liens est basé sur un échange en **série** de l'information. Par opposition à une liaison parallèle, une liaison série n'utilise qu'une seule ligne pour transférer l'information : les digits se succèdent sur un seul fil. Sur un lien, l'unité de transfert est l'octet (8 bits).

Ces 8 bits de données sont envoyés sur la ligne sortante du circuit émetteur, précédés de deux bits d'entête : un bit de start niveau haut, suivi d'un bit niveau haut. Enfin, un bit de stop niveau bas termine cette information. Chaque entité émise possède donc une taille de 11 bits.
L'interface lien qui reçoit cette information, amorce un processus d'acquitement dès lors qu'elle a reçu les deux bits d'entête. Ceci se traduit par deux bits déposés sur la ligne : un bit de start niveau haut, suivi d'un bit de stop niveau bas (cf figure 3.12).
Comme nous l'avons vu au paragraphe 3.4.4 une interface lien est composée de deux sous interfaces indépendantes. Aussi, le processus d'acquitement s'exécute en même temps que la fin de la réception de l'octet.

Le transputer émetteur ne pourra renvoyer un nouvel octet, qu'après réception de ce signal d'acquitement. Cet acquitement systématique permet d'assurer la synchronisation des tâches implantées sur des processeurs distants.
Du fait du retard imposé par la propagation des signaux dans une ligne, l'attente d'acquitement avant toute réémission pèse sur les communications à grandes distances.

Ce protocole de dialogue par lien constitue la couche de plus bas niveau dans les échanges d'informations entre transputers. Nous verrons au chapitre 6, que nous avons

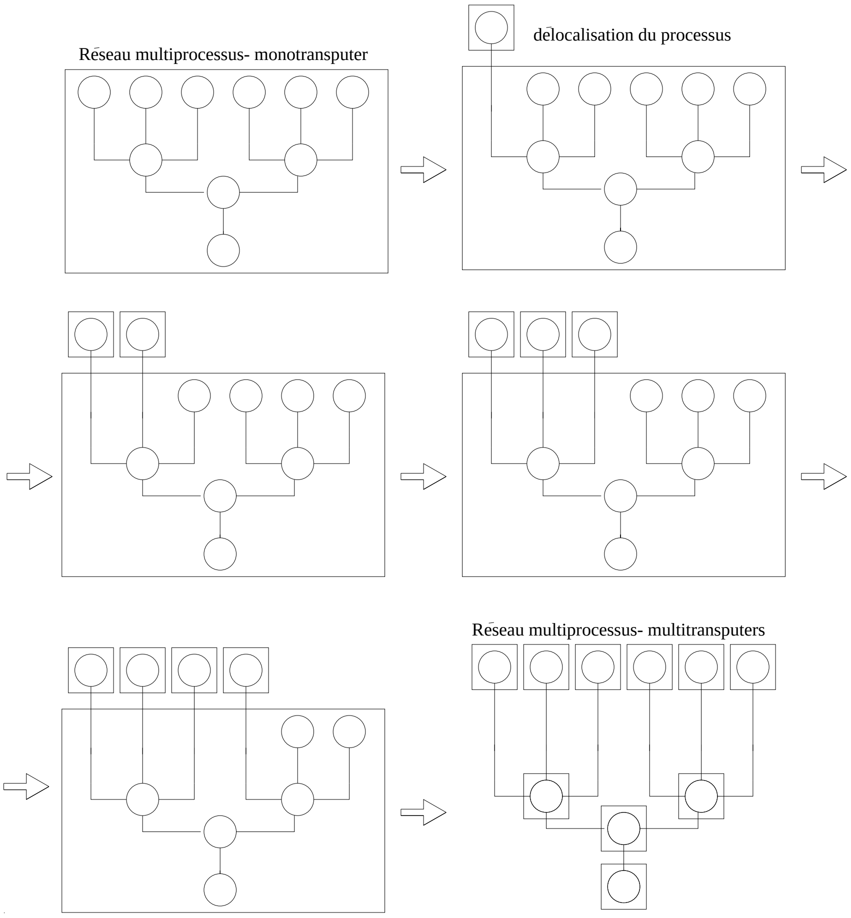


FIG. 3.11 - *Ce schéma illustre la croissance d'un réseau lors de sa phase de développement. Le premier stade intègre le réseau de processus sur un seul transputer. Au fur et à mesure de la disponibilité des cartes transputer et en fonction de l'état de l'application, on délocalise les tâches vers leur processeurs d'acceuil.*

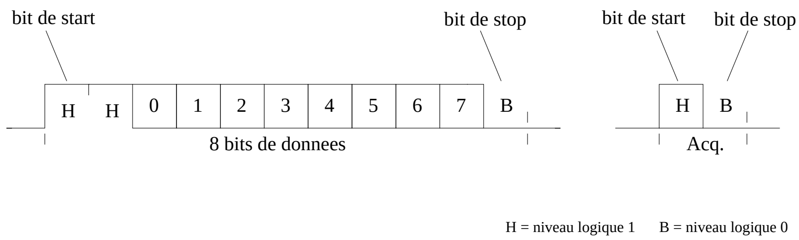


FIG. 3.12 - *Structure de l'information véhiculée sur les liens transputers. L'unité de transfert est l'octet. Chacun de ces mots est identifié en tête par deux bits niveau haut. Un bit niveau bas marque la fin du mot transféré. Le signal d'acquitement renvoyé par l'interlocuteur est, quant à lui, formé de deux bits contigus : un niveau haut suivi d'un niveau bas.*

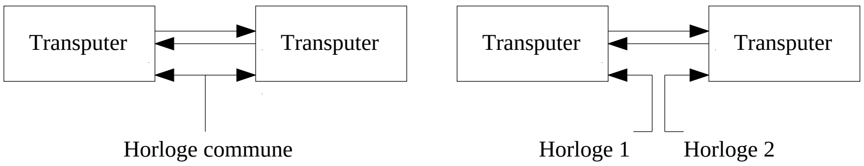


FIG. 3.13 - *Deux possibilités de cadencement pour un système à deux transputers.*

implantés un protocol de transfert par paquets, qui constitue une deuxième couche dans les échanges d'informations. Cette superposition permet d'accroître la souplesse et la fiabilité des échanges.

Afin d'évaluer la vitesse des échanges entre transputers, il est important de savoir que, pour une communication établie en mode 20Mbits/s, la largeur mesurée d'un bit transmis est de 50 ns. De plus, 275 ns s'écoulent entre la réception du premier bit de start et l'envoi de l'acquitement. Ce même laps de temps apparaît entre la réception d'un acquitement et la réémission d'un octet. A noter que ces valeurs sont doublées pour une utilisation des communications à 10Mbits/s.

Les signaux de données et d'acquitements sont supportés physiquement par deux lignes différentes. Il est important de noter qu'un octet envoyé sur un lien peut faire office d'acquitement, ce qui assure la bidirectionnalité des échanges.

Les quatre interfaces liens d'un circuit traduisent les bits d'informations par des niveaux logiques TTL. Ceci garantit une faible consommation pour le circuit, mais limite en contrepartie la longueur des liens. Au delà de 30 cm de ligne, on s'expose à des risques :

- d'atténuations de signaux,
- de diaphonie.

La communication par lien, insensible à la phase de l'horloge du système, rend possible le dialogue de composants synchronisés par des horloges différentes, à condition toutefois que les fréquences de transferts soient les même (cf figure 3.13).

3.6 Elaboration d'un réseau de processeurs

La réalisation d'un réseau de transputers passe avant tout par une phase de description de son architecture. Comme nous l'avons vu, celle-ci comporte deux points essentiels :

- l'aspect hardware.
- l'aspect software.

Lorsqu'il s'agit de construire un dispositif de transfert de données, l'objectif est d'établir des lignes de communication entre mémoires tampon et mémoires finales.

Une première approche consiste à évaluer le nombre de lignes nécessaire à transférer l'intégralité des données.

Pour cela, il faut prendre en compte :

- la quantité totale Q_{tot} des données à transférer,
- le temps maximum T_{max} accordé pour le transfert global,
- la bande passante BP_{ligne}, qui correspond à la capacité de transfert d'une ligne de communication.

En supposant que toute les lignes que l'on peut établir possèdent la même bande passante, le nombre minimum est donné par :

$$N_{lignes} = \frac{Q_{tot}}{T_{max} \times BP_{ligne}}$$

Cette relation suppose toutefois qu'il est possible de répartir équitablement la charge globale de transfert sur toutes les lignes.

Un fois cette valeur connue, de part et d'autre de ces lignes de transfert, il est nécessaire de construire des réseaux de processus capables de s'adapter aux flux entrant et sortant. Ces réseaux auront pour rôle d'émettre et de recevoir les données. Selon les éxigences et les contraintes, ces processus seront répartis sur un nombre plus ou moins grand de transputers.

Chaque processus est un programme particulier. Une fois les rôles de tous les processus défini (algorithmes individuels), il est nécessaire de décrire le réseau de processus dans son ensemble ainsi que le réseau de processeurs destiné à le supporter.

3.6.1 Description du réseau software, caractéristiques et connectique

Cette étape doit conduire à la description complète du réseau de processus du point de vue d'un observateur extérieur.
Elle doit pour cela inclure le descriptif de chacune des tâches du réseau :

- identité du processus,

- taille des zones mémoire : espace de travail, pile,
- positions des zones mémoires dans la mémoire physique,
- liste des canaux de communication vers les autres processus,
- interconnection des processus.

Cet ensemble de caractéristiques est contenu dans un fichier de configuration ".cfs". Il s'agit là d'une description purement topologique. Nous allons voir qu'une application spéciale (le configureur de réseau) fera référence à cette cartographie pour créer l'équivalence entre programmes compilés et processus pour générer le code exécutable.

3.6.2 Description du réseau hardware, caractéristiques et connectique

Cet phase menée conjointement à la précédente, décrit la topologie "physique" du réseau. L'ensemble des transputers présents dans l'application doit être listé. Le potentiel de chaque circuit (type de processeur et mémoire disponible) est précisé pour fixer les limites d'utilisation. De plus l'intégralité de la connectique hardware est établie, ce qui correspond à la "carte" du réseau de processeurs. Cette topologie est mise en correspondance avec celle des processus en répartissant l'ensemble des tâches sur les noeuds du réseau physique : c'est le placement des processus. Les canaux d'interconnection des processus distants, sont placés sur les liens des transputers d'acceuil : c'est le placement des canaux. Là encore, toute cette description est faite dans le fichier de configuration ".cfs".

3.6.3 le langage utilisé

L'utilisation de transputers caractérise un environnement essentiellement concurrent, c'est à dire que des actions particulières s'y déroulent simultanément, en divers endroits, avec ou sans interaction entre elles. Il est donc nécessaire, pour développer des applications concurrentes, d'utiliser un langage informatique qui facilite la description d'un tel environnement [23].
Le langage OCCAM [28], [29] permet d'accéder à la programmation concurrente. Il englobe pour cela les notions de simultanéité, de synchronisation et de compétition entre tâches. OCCAM est le résultat d'une mise au point interactive du circuit transputer et du langage lui même. C'est ce point là qui en fait un langage inhérent au transputer.

Tout programme écrit en OCCAM peut être construit par combinaisons de cinq processus élémentaires :

- Le processus d'affectation " := ". L'instruction *c:=d* affecte l'expression *d* à la variable *c*. Ce processus prend fin lorsque l'affectation est terminée.

- Le processus d'entrée sur canal repésenté par le symbole "? ". L'expression x?y affecte à la variable y la quantité émise à l'autre extrémité du canal x. Ce processus se termine lorsque l'affectation est réalisée.
- Le processus de sortie sur un canal repésenté par le symbole "! ". L'expression x!s transfert la quantité s vers le canal x. Ce processus est terminé lorsque le canal a émis.

Tout ces processus, peuvent être assemblés séquencielement (instruction SEQ) pour former des processus plus complexes. Les séquences d'instructions ainsi créées peuvent ensuite être exécutées séquencielement (instruction SEQ) ou parallèlement (instruction PAR). Dans le dernier cas, on construit des processus concurrents. Ces deux instructions *SEQ* et *PAR* délimitent ce que l'on appelle de façon générale des *constructions*.

D'autres langages tels que le Fortran ou l' ANSI C, se sont vus rajouter la base OCCAM, offrant ainsi à l'utilisateur un choix plus important quant à l'outil de programmation. Sur NA50, la majeure partie des programmes a été écrit en C parallèle[2]. Lorsque des transputers ne disposent que de peu de mémoire programme, l'utilisation directe d'OCCAM est avantageuse. En effet les programmes écrit en C, incluent systématiquement des librairies de bases qui fixent la taille minimum d'un code à 15ko. De plus, OCCAM, plus proche de l'assembleur transputer, conduit à des codes exécutables plus rapide qu'en C.

La gestion concurrente des processus en C se fait au moyen de fonctions regroupées en trois catégories :

- les fonctions de gestion des processus proprement dit. Ils s'agit d'un groupe de 26 fonctions qui permettent la création, l'exécution et le contrôle des processus. Les plus importantes d'entre-elles sont :
  - ProcAlloc, alloue de la mémoire de pile et initialise un processus.
  - ProcAlt, provoque la suspension d'un processus en attendant une entrée sur un canal de communication.
  - ProcPar, exécute deux processus ou plus en parallèle.
  - ProcRun, démarre un processus, avec le même niveau de priorité que le processus père.
  - ProcStop, entraîne l'arrêt du processus.
- les fonctions de gestion des communications inter-processus. Cette catégorie renferme 13 fonctions dédiées à la mise en place des communications entre processus locaux ou distants. Les trois fonctions de base sont :
  - ChanAlloc, alloue et initialise un canal.

2. ANSI C toolset. INMOS 72 TDS 225 00

  - ChanIn, récupère un message sur un canal.
  - ChanOut, émet un message sur un canal.

- les fonctions de gestion des sémaphores. Un sémaphore est une entité logicielle, qui permet d'éviter des conflits lorsque plusieurs processus accèdent à des ressources communes. Il s'agit d'un garde-fou. Quatre fonction permettent de gérer les sémaphores :

  - SemInit, initialise un sémaphore.
  - SemAlloc, alloue et initialise un sémaphore.
  - SemSignal, libère un sémaphore.
  - SemWait, prend un sémaphore.

Des combinaisons de toutes ces fonctions permettent de créer des processus concurrents susceptibles d'échanger entre eux des informations. L'ensemble des fonctions de communication est relative à des canaux logiciels. Ces canaux sont internes ou bien rattachés à des liens, et permettent ainsi à chaque processus de communiquer avec un processus connexe, qu'il soit sur le même transputer ou non.

3.6.4 Les outils nécessaire au développement des applications

Les outils utilisés pour le développement d'applications transputer sont supportés par divers type de machines : PC, SUN, VAX ou Machintosh. Nous verrons au paragraphe 3.7 le type de machine choisi sur NA50.
Ces outils se résument à 15 applications [34] listées dans le tableau 3.1.

Le cycle classique du développement d'une application multi-transputers, présenté sur la figure 3.14 nous montre les applications fréquemment utilisées.

L'édition, la compilation et l'édition des liens

Les trois premières phases de ce cycle correspondent au schéma classique suivi lors d'un développement informatique.

application	description
icc	compilateur ANSI-C
icconf	configureur
icollect	collecteur de code
icvlink	convertisseur de fichiers TCOFF
idebug	debugger
idump	examineur de mémoire
iemit	configureur de mémoire
ieprom	formateur de programme EPROM
ilibr	créateur de librairies
ilink	créateur de liens
ilist	listeur binaire
imakef	générateur de fichiers Makefile
iserver	serveur de réseau
isim	simulateur IMS T425
iskip	chargeur de programmes distants

TAB. 3.1 - *Liste des 15 applications utilisables pour développer un réseau de transputers.*

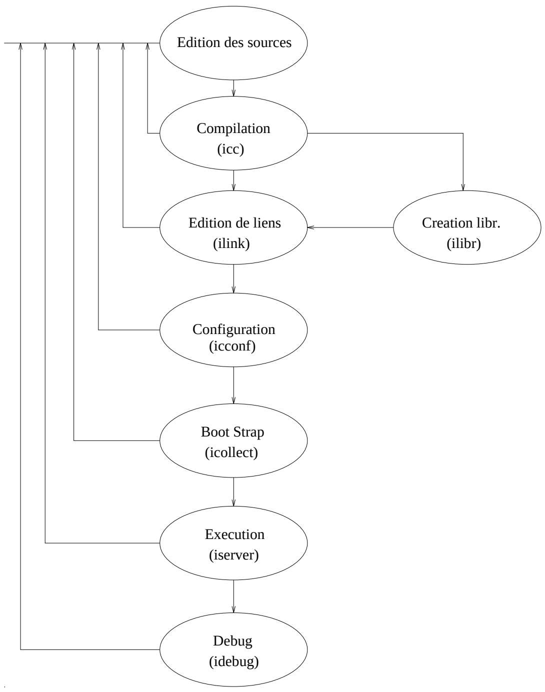


FIG. 3.14 - *Cycle classique du développement d'une application multi-transputers.*

La configuration

La configuration est le mécanisme grâce auquel toutes les entités logicielles développées dans les trois premières phases vont être "rattachées" à des processeurs physiques. En se basant sur les descriptions des fichiers de configuration ".cfs" et à partir des codes ".lku" (cf figure 3.15), le configureur *icconf* va établir des relations de correspondance entre processus et transputer, ainsi qu'entre canaux et liens transputers. Lorsque cette phase de configuration est terminée, on dispose d'un fichier binaire ".cfb" (cf ficgure 3.15), qui rassemble tous les codes exécutables de tous les processus de l'application multi-transputer.

Le "Boot" et l'exécution

Pour faire fonctionner ce type d'architecture multi-processeurs, une fois les codes développés (fichier ".cfb"), il est nécessaire de les charger sur les processeurs : c'est la phase de "Boot". Nous allons voir que pour accomplir cette phase il est nécessaire d'apporter des modifications au fichier ".cfb".

Le mode de chargement d'un code exécutable sur un transputer est un mécanisme très particulier. En effet, les transputers ne possèdent que des mémoires volatiles. Ainsi, lors de la mise sous tension du circuit, aucune initialisation ne s'active. C'est le rôle de la phase de *boot* qui, pour un transputer donné, se déroule en 4 parties successives :

- Un signal de *reset* doit être envoyé au transputer pour le placer dans un état determiné : c'est l'initialisation de bas niveau. Ce signal est envoyé sur une ligne dédiée et dure quelque μs,
- Par l'intermédiaire d'un lien transputer on place dans la mémoire du transputer un code assembleur appelé *Boot-Strap*. Il a pour rôle de continuer l'initialisation en cours et de charger un autre code appelé *Boot-Loader*,
- Toujours par le lien transputer, le *Boot-Strap* télécharge un second chargeur : *le Boot-Loader*. Celui-ci écrase le code du *Boot-Strap*, à l'exception des 2 premiers octets qui contrôlent son propre chargement,
- Une fois en mémoire, ce *Boot-Loader* termine les initialisations internes du transputer avant de télécharger un troisième code : le *Loader*. Il s'agit du chargeur le plus élaboré parmi les trois précédemment cités. Il a pour rôle de télécharger par paquet, l'intégralité du code exécutable de l'application transputer.

L'ensemble de ces mécanismes d'initialisation et de téléchargement sont détaillés dans l'ouvrage [32].

Afin de pouvoir charger un réseau, les séquences de codes exécutables de tous ces chargeurs sont rajoutés en tête des programmes de l'application développée. Le programme

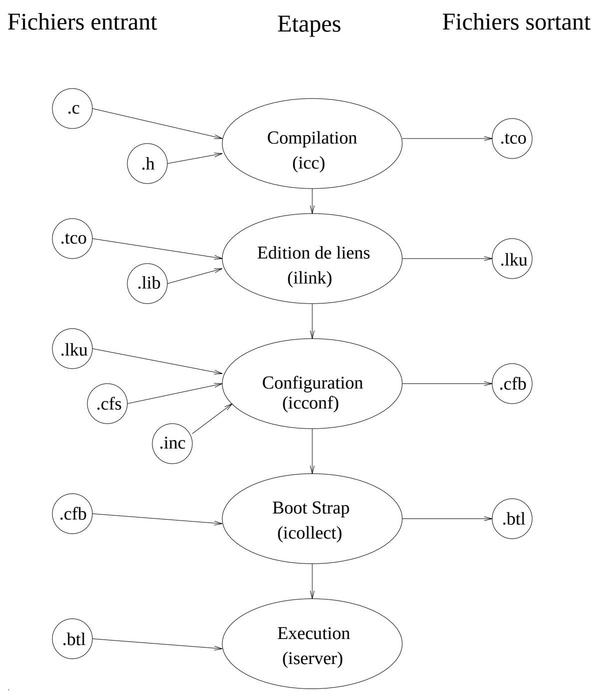


FIG. 3.15 - *Notations conventionnelles des fichiers utilisés lors du développement d'une application multi-transputers*

icollect permet de rajouter ces séquences, en tête du fichier binaire ".cfb". Il en résulte un fichier binaire ".btl", résidant sur la machine hôte (figure 3.14 et 3.15).

Pour répartir les différentes tâches (contenues dans le fichier ".btl") sur les transputers du réseau, ll faut faire appel à une application spéciale, qui a pour rôle d'amorcer le chargement du réseau et d'initialiser l'interface utilisateur-transputer. C'est le rôle du programme *iserver*. Une fois les programmes chargés sur les transputers du réseau, les processus répartis dans le réseau deviennent actifs : l'application multiprocesseur s'exécute.

3.7 La machine hôte

La machine hôte est une station de travail classique. Sur NA50, nous utilisons des SUN SPARC-5. Elles contiennent tous les logiciels décrits au paragraphe 3.6.4. Pour le développement et l'exécution des programmes, cette station doit pouvoir travailler de façon interactive avec un réseau de transputers.

Si la communication entre les transputers d'un réseau est immédiate grâce à l'utilisation des liens, pour dialoguer avec l'extérieur du réseau, il faut établir une interface

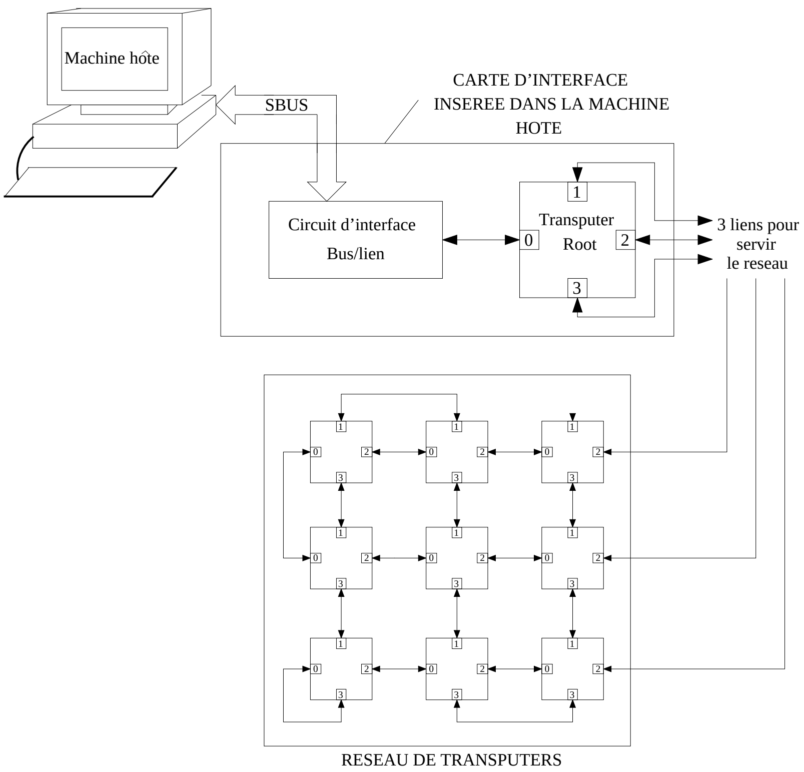


FIG. 3.16 - *Schéma général d'une interface machine hôte-transputers. La carte d'interface, implantée au sein même de la machine hôte, permet d'établir le dialogue entre l'utilisateur et le monde transputer, via un circuit d'interface et un transputer dédié : le "root". Trois liens restent disponibles pour établir une arborescence.*

"lien/machine-hôte". Cette liaison particulière jouera le rôle :

- De point d'entrée du réseau. En effet, c'est à partir de ce lien particulier que les programmes, résidant sur la machine hôte, sont injectés dans le réseau de transputers lors de la phase de démarrage. Une fois dans le réseau, les chargeurs de codes (voir paragraphe 3.6.4) distribuent les tâches sur les différents transputers.

- D'interface utilisateur. Lorsque l'application mutliprocesseurs s'exécute, par l'intermédiaire de ce lien, l'utilisateur peut recevoir des informations du réseau ou bien agir sur celui-ci en injectant des instructions.

L'interface proprement dite est un circuit de conversion qui permet de faire la liaison entre un lien série au standard INMOS et le bus parallèle de la machine hôte (ex : le circuit C012 de INMOS). Le lien transputer connecté à ce circuit d'interface provient d'un transputer communément appelé *Root*. Ces deux circuits sont implantés sur une carte qui vient s'insérer sur le bus de la machine hôte (le SBUS pour les stations SUN). Trois liens, sont donc disponibles sur le transputer *Root* pour servir le reste du réseau (cf figure 3.16).

Sur NA50, la carte d'interface insérée dans la machine hôte est une carte *TITAN-1S*. Les détails de cette carte seront présentés au cours du chapitre suivant.

3.8 Utilisation des transputers sur NA50

Actuellement, sur NA50, deux réseaux totalisant 24 transputers permettent de reconstruire, filtrer, contrôler et transférer, les données issues de tous les détecteurs du dispositif expérimental.
Dans ces deux réseaux, le transputer est utilisé pour transférer de façon simultanée des blocs d'informations physiquement séparés. Ceci exploite sa capacité à évoluer dans une architecture parallèle et à échanger rapidement de l'information. De plus, l'information étant manipulée par les processeurs, nous mettons à profit cette puissance de calcul pour faire de la reconstruction et du filtrage en-ligne des événements.

Ces réseaux de processeurs jouent donc le rôle d'interfaces *intelligentes* entre les mémoires tampon des détecteurs et les mémoires finales qui servent le système d'écriture sur bandes magnétiques.

Dans la partie qui va suivre, nous allons décrire rapidement le réseau de 13 transputers utilisé pour prendre en charge les informations issues de tous les détecteurs autre que la multiplicité : il s'agit du *réseau général*. Nous avons vu (voir paragraphe 3.2), qu'il doit être capable de transférer la moitié du volume total des données créé à chaque cycle SPS. Le réseau de multiplicité, fera l'obet des chapitres 4 et 5

3.9 Le réseau général

Hormis le détecteur de multiplicité, nous avons vu au paragraphe 3.2.1 que les données du dispositif expérimental proviennent de trois catégories de mémoires différentes :

- Les mémoires tampons du spectromètre à muons, implantées sur deux cartes *MEMRMH*. Elle sont accessibles par le *réseau général* de transputers via le bus VME et contiennent donc les données provenant des 8 chambres à fils et celles engendrées par les 4 hodoscopes. Nous utilisons un chassis VME pour ces deux cartes MEMRMH.
- Les mémoires des TDC et QDC formées de 27 modules VME. Qu'ils soient QDC ou TDC ils sont répartis dans deux chassis VME, et produisent environ 4Mo d'information à chaque cycle SPS.
- Les échelles VME. Elles produisent une faible quantité de données à chaque déversement : quelques Ko/burst. Cependant, ces échelles occupent une place importante en nombre de modules, puisque deux chassis VME leur sont réservés (maximum de 40 échelles).

La répartition de toutes ces informations : chambres, hodoscopes, calorimètre et autre sur une grand nombre de modules distincts répartis dans trois catégories amorce l'idée d'une gestion parallèle des transferts. Ceci nous à conduit à réaliser le réseau de 13 transputers représenté sur la figure 3.17.

L'ensemble des modules qui constituent ce réseau est localisé dans la salle de comptage hors du hall d'expérience. Les distances qui les séparent les uns des autres se limitent donc à quelque mètres et n'affectent pas les temps de transferts sur liens transputers. Nous verrons au chapitre 5 qu'il n'en est pas de même pour le réseau de multiplicté.

Il faut garder à l'esprit qu'une parallélisation des transferts doit amener à un gain de temps par la présence de plusieurs processus d'émissions actifs simultanément sur des voies de propagation distinctes (les liens). La multiplicité de ces liens augmente la bande passante totale des transferts et décorelle du même coup les échanges de données :
un transfert peu prendre du retard sur un lien sans pour autant pénaliser les flots de données associés aux autres lignes de communication parallèles.

Cependant, afin d'optimiser la bande passante globale du réseau, il est dans notre intérêt de répartir équitablement la charge globale à transférer sur l'ensemble des canaux disponibles. Pour cela, il est nécessaire d'énumérer : les transputers émetteurs de données, et les transputers dédiés à la réception.

Dans le cas qui nous concerne, nous avons six chassis VME distincts qui seront les sources d'information pour le réseau de transputers (voir figure 3.17) :

- Un chassis RMH-MWPC, pour les données des 8 chambres à fils.
- Un chassis RMH-Hodoscopes, pour les informations des hodoscopes du spectromètre.
- Un chassis ADC1 pour la première moitié des modules QDC et TDC.
- Un chassis ADC2 pour la seconde moitié des modules QDC et TDC.
- Un chassis SCA1 pour la première moitié des modules échelles.
- Un chassis SCA2 pour la seconde moitié des modules échelles.

Chacun de ces chassis est couplé avec une carte transputer B016, pour former ainsi une base de six transputers émetteurs. Ces cartes B016 que nous décrivons au chapitre 5 disposent d'une mémoire double port de 16Mo : transputer/VME. Il s'agit de modules, entièrement pilotés par transputer, qui intègrent toutes les fonctionnalités VME standards. Dans la configuration que nous avons choisie, une carte B016 doit pouvoir lire les n modules présents dans son chassis VME; le transputer de la B016 doit donc possèder les privilèges d'un maître VME. Le rôle du transputer est alors de rassembler les contenus

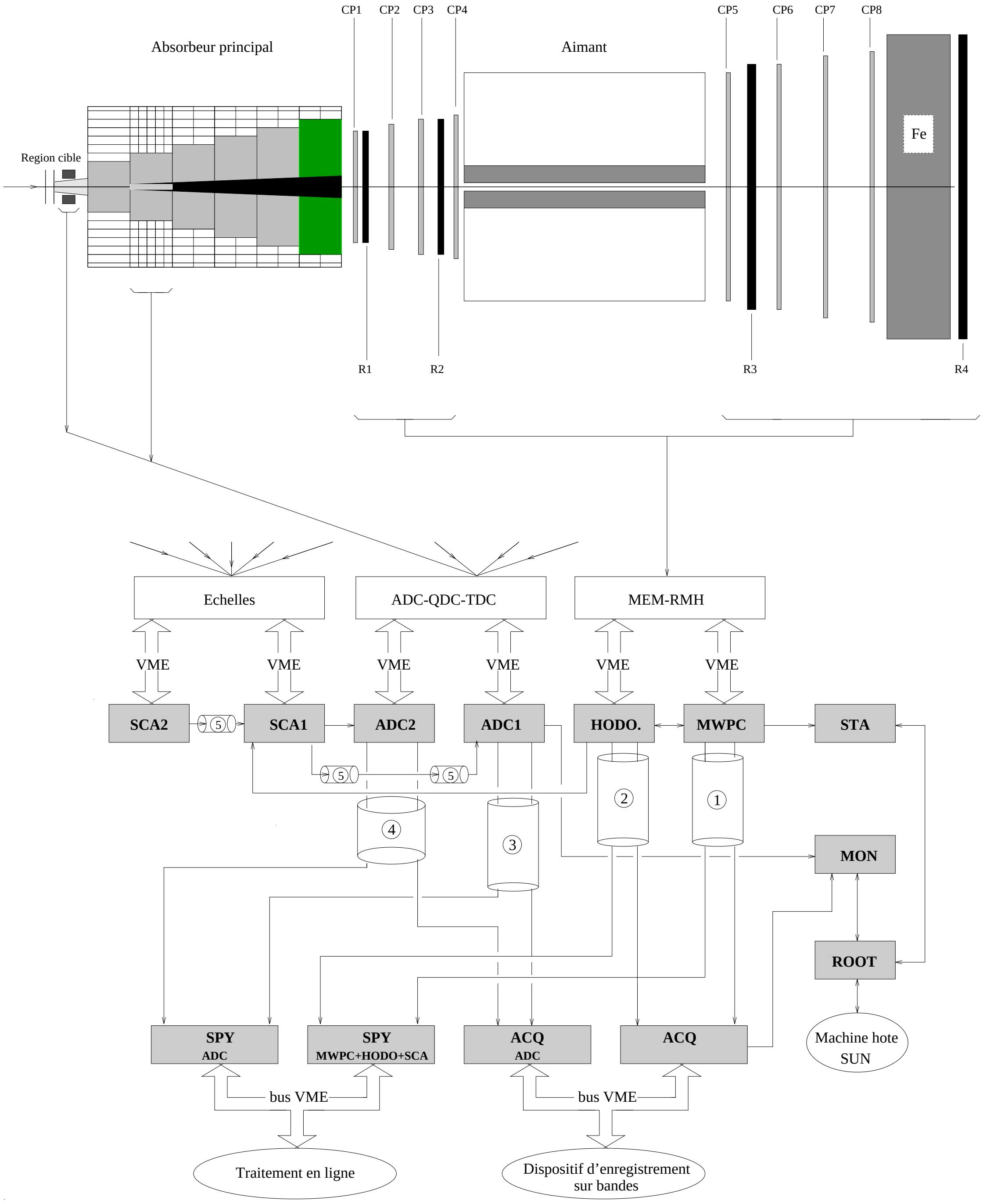


FIG. 3.17 - *Le réseau général de transputers. Les transputers sont représentés en grisé. Quatre lignes principales notées 1,2,3 et 4 évacuent de façon simultanée les données provenant des sources RMH-MWPC, RMH-Hodo, ADC1-Scaler et ADC2. Chacune de ces lignes est elle même composée de deux liens transputers qui travaillent en parallèle pour dupliquer l'information transmise vers les mémoires Acquisition et Espion. La ligne 5, transfert la faible quantité d'information des deux chassis échelles vers le module ADC1. Tout les transputers de ce réseau sont interconnectés par des liens.*

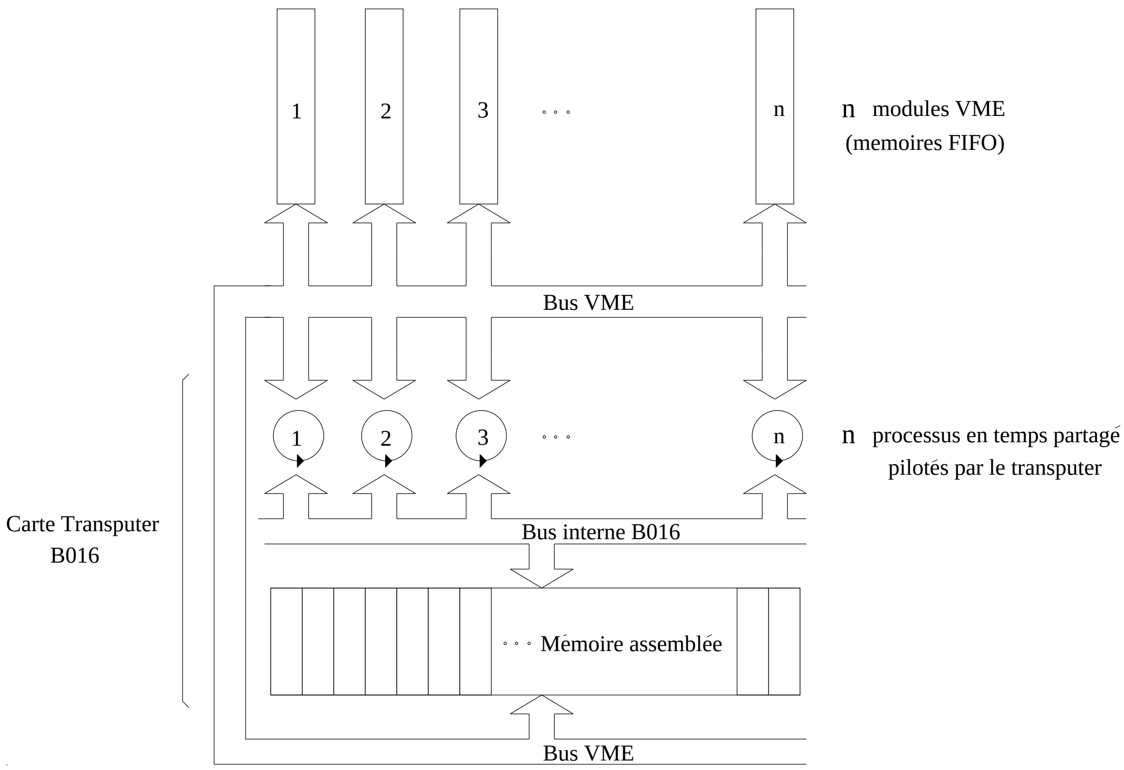


FIG. 3.18 - *Cette figure schématise l'implantation de n processus de lecture au sein du transputer des cartes B016. Ils sont chargés de lire les n modules présents dans le chassis VME. Les différents processus exécutés en temps partagé sur le transputer, accèdent en maître VME, aux modules QDC,TDC ou MEMRMH pour recopier l'information lue dans la mémoire tranputer, en vue d'être transférée vers les processeurs récepteurs.*

des mémoires FIFO des n modules, en un seul bloc mémoire, interne à la carte B016 (voir figure 3.18). Ainsi, une seule opération DMA permet de transférer cette mémoire assemblée vers les mémoires finales.

Comme nous le montre la figure 3.18, du point de vue software, le transputer exécute autant de processus *lecteur* qu'il y a de modules à lire dans son chassis (processus notés 1..n sur la figure). Ces n tâches évoluent dans un système à temps partagé et accèdent en lecture, via le bus VME, aux FIFO qui leur sont associés.

Les mémoires réceptrices sont également supportées par des cartes B016. Etant donnée la dualité mémoire *ACQUISITION*/mémoire *ESPION*, et au vue du nombre de liens disponibles par transputer, l'ensemble des mémoires réceptrices est formé de quatre cartes B016 : deux pour la mémoire *ACQUISITION* et deux pour la mémoire *ESPION* (cf. figure 3.17).

La communication entre les transputers émetteurs et les transputers récepteurs se fait par le biais de 4 voies parallèles schématisées par des cylindres sur la figure 3.17. Une

voie véhicule simultanément les même informations vers les mémoires *ACQUISITION* et *ESPION*, au moyen de deux liens transputers. L'ensemble du transfert repose donc sur 8 lignes d'évacuation qui travaillent en simultanéité pure. Ceci est possible grâce au propriétés du circuit décrites au paragraphe paragraphe 3.5.2.
Chacune de ces lignes est soumise à la même charge de transfert de 2Mo de données et possède une bande passante unidirectionnelle de 1.6Mo/s (mode bidirectionnel à 20Mbits/s). Cette valeur nous conduit à une bande passante de 12.8Mo/s pour l'ensemble du réseau de transfert.
Les blocs d'informations assemblés sur les cartes émettrices sont déplacés via les liens par un transfert unique (un seul appel DMA de chaque interface lien) en 2.5 s. Le protocole d'acquitement implanté sur les circuits transputers assure que le bon nombre d'octets est transféré. Cependant, pour garantir l'intégrité des transferts, un contrôle systématique par un *CheckSum*, est effectué sur les bits transférés : au départ et à l'arrivée des voies de communication. L'émission est réitérée en cas de différence.

Entre le système d'enregistrement sur bandes magnétiques et le réseau de transputer, les communications se limitent à des interruptions VME envoyées lorsque les blocs de données sont entièrement présents dans la mémoire finale *ACQUISITION*. Ces interruptions sont générées par le transputer de la B016ACQ.

Le dispositif de mémorisation sur cassettes accède aux données burst par burst. Ce qui importe donc pour ce dispositif, c'est le temps disponible entre deux surécritures de la mémoire finale *ACQUISITION*, pour enregistrer les 8Mo de données[3].
Pour cela, le réseau de transputers précédemment décrit, présente la propiété remarquable d'offrir au système d'acquisition, un burst apparant, d'une durée plus courte que celle du déversement réel (5s en prise de données Pb^{208}). Ceci provient du fait que toutes les opérations d'assemblage des blocs mémoires sont effectuées par les processeurs sources. Les transputers récepteurs, ainsi dégagés de ces tâches rendent la mémoire finale *ACQUISITION* indisponible pendant le seul temps réservé aux transferts, soit 2.5 s (cf figure 3.19). Aux yeux du système d'acquisition, les burst ne durent donc que 2.5 s, et les données sont disponibles en mémoire pendant 16.5 s.

3. il s'agit de 16Mo en réalité puisque l'on doit considérer le détecteur de multiplicité

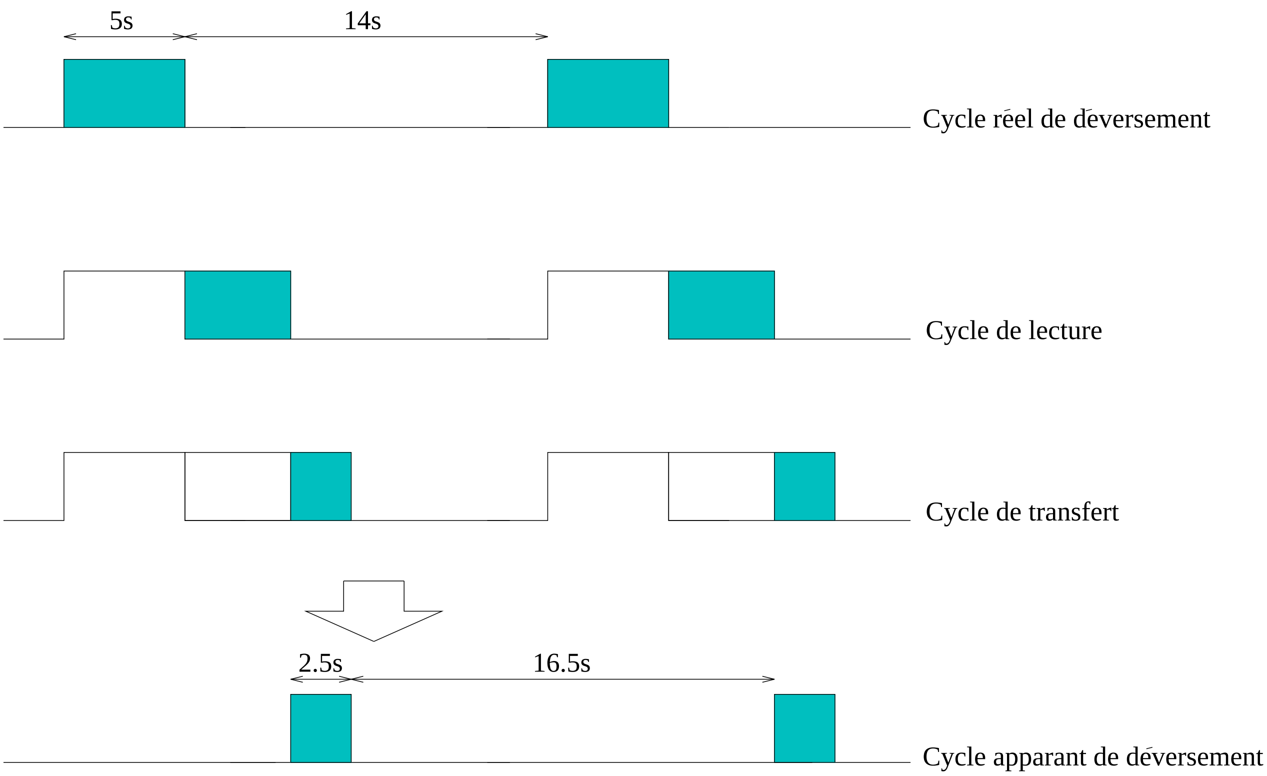


FIG. 3.19 - *L'assemblage des données par les processeurs sources permet d'offrir au dispositif d'acquisition sur bande un déversement apparant, d'une durée inférieure à la durée réelle.*

Chapitre 4

LECTURE DES DONNEES DU DETECTEUR DE MULTIPLICITE

4.1 Introduction

Nous avons vu au paragraphe 2.7.1 que l'ensemble du détecteur est composé de 13 824 bandes de silicium. Chacune de ces bandes, lors du passage d'une particule chargée (π^+,π^-,K^+,K^-...), génère une impulsion électrique. Ce signal analogique, par son amplitude et sa forme détermine l'état de la bande de silicium : bande traversée ou non par une particule chargée. Intégré sur les deux disques sensibles MD1 et MD2, ces informations indiquent l'état global du détecteur de multiplicité.
A partir de là, on définit *un événement multiplicité*. Pour cela, il faut reconsidérer le fait qu'un événement NA50 doit être signé par une paire $\mu^+\mu^-$ corrélée. La collision Pb-Pb responsable d'un tel dimuon, engendre une gerbe de particules chargées en aval du point d'interaction. Un événement multiplicité représente donc l'état du détecteur à l'instant t_0 où ces particules chargées traversent les bandes de silicium.

Dans ce chapitre, nous allons décrire le dispositif de lecture qui permet d'enregister les événements multiplicité. Nous allons voir en particulier que les signaux produits par le silicium sont immédiatement digitalisés après une préamplification, et que seule l'information digitale est mémorisée. Ce traitement est réalisé sur 64 canaux simultanément, par une chaîne de lecture formée de deux circuits en cascade : le FABRIC et le CDP. Ainsi, 216 chaînes de lecture parallèle assurent l'échantillonage des 13 824 voies du détecteur de multiplicité.
Nous verrons d'autre part que le caractère imprévisible des déclenchement $\mu^+\mu^-$ nous oblige à échantillonner périodiquement le détecteur de multiplicité. Pour finir, nous allons montrer comment, parmi tout les échantillons relevés, nous somme capables de retrouver et conserver celui qui correspond à l'état du détecteur à l'instant t_0.

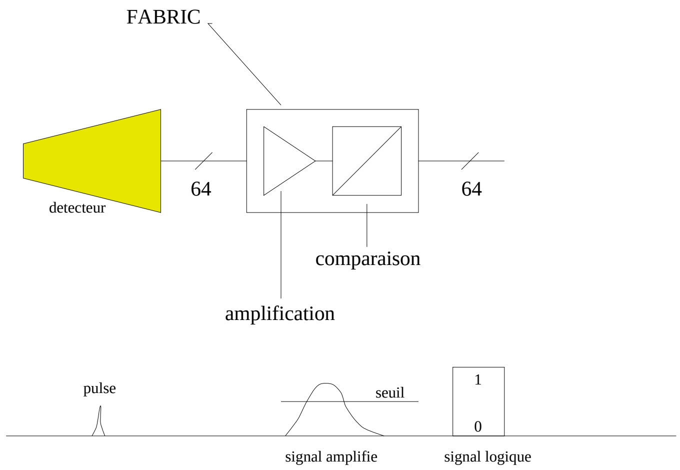


FIG. 4.1 - *Ce schéma nous montre les deux fonctions d'un circuit FABRIC [18]: l'amplification et la comparaison. Ce circuit, permet de digitaliser 64 voies simultanément et introduit un temps mort de 40ns [18].*

4.2 La digitalisation des signaux

Le détecteur de multiplicité doit compter le nombre de particules chargées qui le traversent à un instant donné. La digitalisation à pour but d'associer a chaque bande de silicium, un signal binaire qui signe deux états :

- l'état occupé, signé par la valeur logique 1 si une particule a traversé la bande de silicium.
- l'état non occupé, signé par la valeur logique 0 si aucune particule n'est passée à travers cette bande.

Pour construire cette information binaire, on utilise un circuit de type ASIC (Application Specifie Integrated Circuit) : le FABRIC [18] ou BAC (Bipolar Amplifier and Comparator). Comme le montre la figure 4.1, ce circuit intègre deux fonctions en cascade :

- L'amplification de 64 voies en parallèle.
- La discrimination ou comparaison des 64 signaux amplifiés.

Pour chacune de ses 64 entrées, ce circuit analogique, amplifie tous les pulses provenant des bandes de silicium, puis compare ce signal à un seuil fixé. Si le signal amplifié est plus grand que le seuil, le comparateur génère un niveau logique 1 (voir figure 4.1). Dans le cas contraire, un niveau logique 0 est créé. A la sortie du FABRIC, on dispose donc de 64 valeurs binaires qui représentent l'état de 64 bandes de silicium.

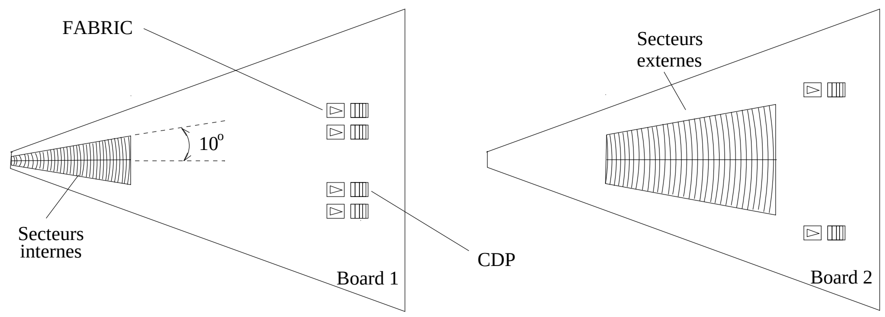


FIG. 4.2 - *Ce schéma nous montre l'implantation des circuits de lecture du détecteur de multiplicité, sur les cartes Board1 et Board2, supports des secteurs de silicium.*

Pour assurer la numérisation des 13 824 cannaux de silicium qui forment le détecteur, il faut donc 216 circuits FABRIC.

L'ensemble amplificateur/comparateur introduit un temps mort de 40 ns [18] : on ne pourra séparer deux impacts chargés s'ils se succèdent dans un intervalle de temps inférieur au temps mort.
La figure 4.2 nous montre que les circuits FABRIC sont implantés directement sur les cartes supports *"Board1"* et *"Board2"*. Un circuit occupe une surface de $4 \times 3.5\ mm^2$.

4.3 Evaluation du temps de latence d'un déclenchement

Un dimuon qui déclenche un événement est identifié par une série d'impacts en coïncidences dans les scintillateurs des hodoscopes R1, R2, R3 et R4, situés à 6 mètres et plus après le détecteur de multiplicité. Si les produits de la collision Pb-Pb traversent le compteur de multiplicité à l'instant t_0, ce n'est qu'à l'instant $t_0 + \delta t_{TRIG}$ que l'on sera informé (au niveau du détecteur de multiplicité) du passage du dimuon dans le spectromètre.

Le laps de temps δt_{TRIG} peut se décomposer en :

$$\delta t_{TRIG} = T_{vol} + T_{R1} + T_{trig} + T_{send} \tag{4.1}$$

où :

- $T_{vol} \simeq 20\ ns$ =temps de vol des muons depuis le détecteur de multiplicité jusqu'aux hodoscopes Ri.
- $T_{R1} \simeq 260\ ns$ = temps entre la sortie des signaux des PM des hodoscopes jusqu'aux discriminateurs associés.

- $T_{trig} \simeq 300\ ns$ = temps entre les sorties de ces discriminateurs jusqu'à la production du signal de déclenchement dans la salle de comptage.
- $T_{send} \simeq 356\ ns$ = temps estimé pour faire redescendre le signal de déclenchement depuis la salle de comptage jusqu'au MD (80 mètres), via une liaison optique.

Toute ces valeurs cumulées conduisent à une limite supérieure de 1000 ns pour δt_{TRIG} : **il s'agit du temps de latence du déclenchement.**

L'arrivée du signal de déclenchement et l'intervalle δt_{TRIG} sont nos seules références en temps. Pendant cet intervalle de temps $[t_0, t_0 + \delta t_{TRIG}]$, il faut donc être en mesure de mémoriser l'activité du détecteur de multiplicté avec la meilleur résolution temporelle. Nous utilisons pour cela un second circuit de type ASIC : le CDP (Clock Driven Pipeline) qui a pour rôle d'échantilloner les signaux binaires en sortie des circuits FABRIC.

4.4 L'échantillonage des signaux binaires

Le flot de particules chargées produit par une interaction Pb-Pb va provoquer une activité élevé des circuits amplificateurs/comparateurs FABRIC : on veut pouvoir mémoriser cette activité là (dans le cas où l'interaction a donnée lieu à une paire $\mu^+\mu^-$ dans le spectromètre).

La durée des signaux de sortie du comparateur (40 ns exigés pour un bon pouvoir de séparation en temps) et le caractère imprévisible des événements nous oblige à échantilloner en permanence les sorties des FABRIC. La largeur de 40 ns des signaux du comparateur impose une fréquence d'échantillonage f_e d'au moins 25MHz.
Cette opération est réalisée grâce à un buffer digital : le CDP (Clock Driven Pipeline) [19] qui peut échantilloner et mémoriser dans sa RAM interne 64 entrées binaires simultanément. La fréquence maximum d'échantillonage est de 50MHz.

La possibilité de voir de produire des interactions Pb-Pb, pendant le temps de latence $\delta t_{TRIG} = 1000\ ns$ du déclenchement, nous contraint à mémoriser l'état du détecteur pendant toute ce laps de temps.

Ceci nous donne une limite inférieure sur la profondeur de la mémoire RAM d'un CDP. Elle doit pouvoir contenir au moins N_{min} échantillons (64 bits de sortie du FABRIC), avec N_{min} défini par :

$$N_{min} = \delta t_{TRIG} \times f_e \tag{4.2}$$

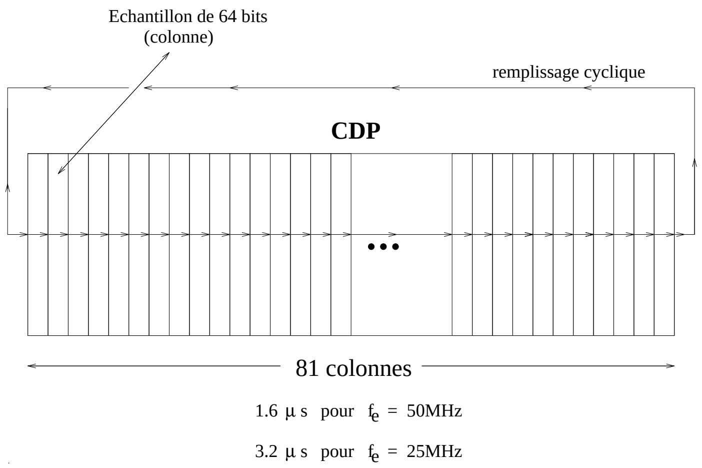


FIG. 4.3 - *Le CDP est un buffer circulaire, d'une profondeur de 81 mots de 64 bits. La fréquence de progression dans cette mémoire est imposé par une horloge externe et lui confère les profondeurs de* 1.6 μs *à 50MHz et* 3.2 μs *à 25MHz*

où,

δt_{TRIG} est le temps de latence du déclenchement
f_e est la fréquence d'échantillonnage de la sortie du FABRIC.

On obtient :

$N_{min} \simeq 25$ pour $f_e = 25\ MHz$, et $N_{min} \simeq 50$ pour $f_e = 50\ MHz$.

Nous utilisons ici des CDP qui ont la capacité de contenir 81 échantillons ($> N_{min}$). Nous entendons par échantillon un mot de 64 bits qui repésente les 64 voies de sortie du FABRIC. Par la suite, nous désignerons par colonne un tel échantillon de 64 bits.

La fréquence de remplissage (f_e) des mémoires RAM des CDP est contrôlée par une horloge extérieure au circuit, délivrée par la carte CCTD décrite au paragraphe 4.5. L'écriture, commandée par les fronts montants de cette horloge, est cyclique : lorsque les 80 colonnes sont pleines, le CDP pointe de nouveau sur sa colonne 0 pour réitirer le remplissage (figure 4.3).

Avec un tel dispositif, on conserve l'état du détecteur de multiplicité pendant 1.6 μs à 50MHz et 3.2 μs à 25MHz. Ces valeurs sont suffisantes pour contenir le temps de latence du déclenchement.

Comme le montre la figure 4.2 un circuit CDP est implanté derrière chaque FABRIC sur les cartes supports *Board1* et *Board2*. Couplés aux FABRIC, ils forment des chaînes

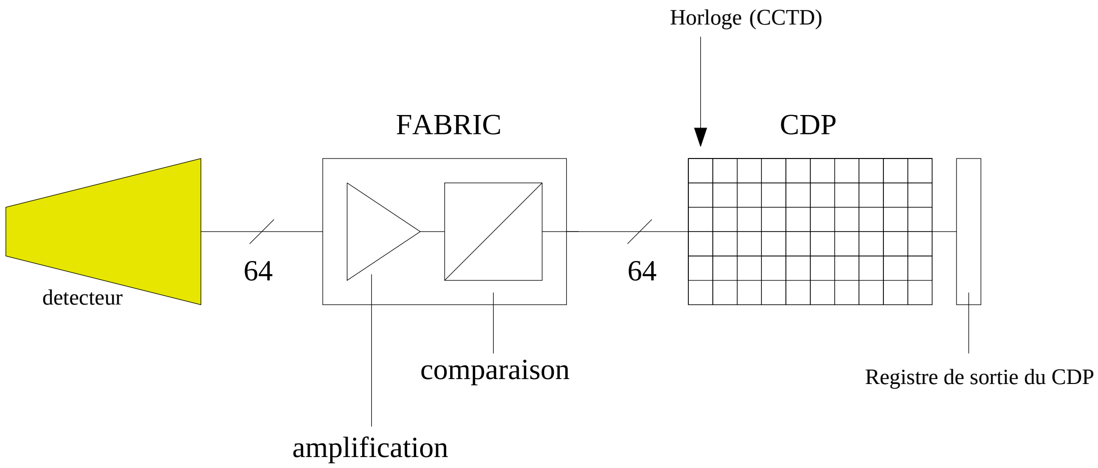


FIG. 4.4 - *Ce schéma nous montre le chaînage des deux circuits FABRIC et CDP qui constituent le dispositif de lecture pour 64 canaux de silicum. On remarque que le remplissage de la mémoire CDP est commandé par une horloge externe.*

de lecture identiques à celle représentée figure 4.4.

Pour chaque CDP, lorsqu'un signal de déclenchement $\mu^+\mu^-$ est généré, un seul mot de 64bits, sur les 81 contenus dans RAM, doit être conservé **définitivement** : celui qui correspond au passage des particules chargées dans le détecteur.
Sur un signal de déclenchement, il faut donc réaliser la séquence suivante :

1 *arréter* la progression de l'échantillonage dans la RAM interne du CDP,

2 *isoler* la colonne rechercher,

3 *transférer* cette colonne vers une mémoire tampon qui contiendra tous les événements multiplicité.

Pour cela, à chaque déclenchement, il faut envoyer des commandes au CDP.

Parmi les 16 commandes que reconnait un CDP [19], l'une d'entre elles (*CmdeRead*) permet de réaliser les deux premières séquences : arrêt de la progression du pointeur de colonne et copie de la colonne pointée dans le registre de sortie du CDP.
Dans la pratique, comme nous le montre la figure 4.5, les commandes sont envoyées au CDP par un groupe de circuits programmables implanté sur une carte BUSIF (décrite au chapitre suivant).
Puisque c'est la colonne pointée qui est copiée dans le registre de sortie, lorsque le signal de déclenchement arrive au détecteur, il faut attendre un temps supplémentaire pour laisser progresser le pointeur (progression en boucle) jusqu'à la colonne désirée. On envoie alors la commande *CmdeRead* qui extrait cette colonne de la mémoire RAM : **l'événement multiplicité est isolé.**

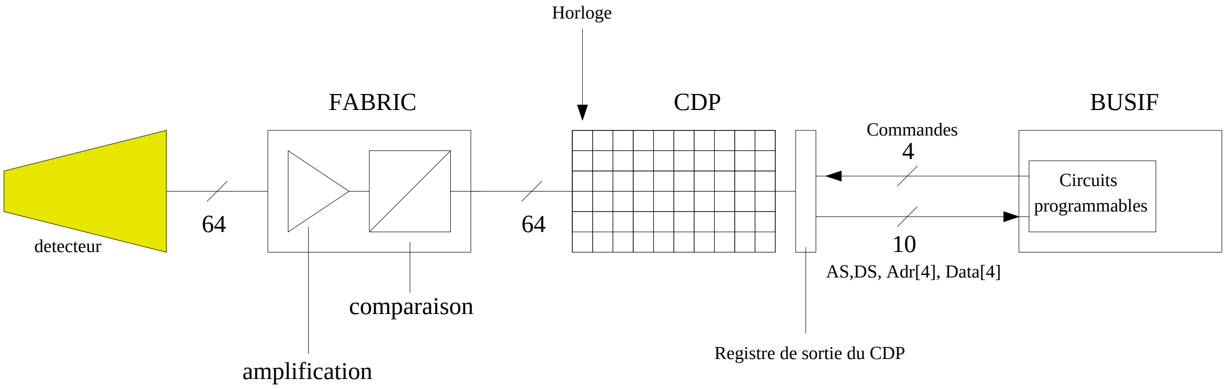


FIG. 4.5 - *Ce schéma nous montre les lignes de communication entre les CDP et les cartes BUSIF (décrite au chapitre suivant). Une commande codée sur 4 bits permet d'agir sur le comportement du CDP. La colonne sélectionnée sera transférée dans le registre de sortie du CDP, puis évacuée par les lignes de Data*

Notons que le registre de sortie du CDP fait 72 bits : 64 bits servent à contenir la colonne selectionnée, tandis que les huit autre contiennent le numéro de la colonne selectionné.

Le laps temps rajouté au signal de déclenchement est géré par une carte spéciale, la CCTD (Calibration, Clock and Trigger Distribution) chargée de distribuer les signaux d'horloge et de déclenchements.

4.5 La distribution des signaux par la CCTD

La carte CCTD (Calibration, Clock and Trigger Distribution), a été conçue par le groupe de l'INFN de Turin, et contrôle deux points crutiaux de la lecture : elle créé d'une part un signal d'horloge, de fréquence f_e qui gouverne l'incrémentation des pointeurs de colonnes pour les circuits de lecture CDP. Elle retarde d'autre part le signal de déclenchement des événements dimuons. C'est ce pulse retardé, qui servira à arrêter la progression cyclique des CDP sur les colonnes qui renferment les bons événements.

Grâce à ce module, il est possible de fixer :

- Une fréquence d'échantillonage f_e du détecteur dans une plage de 391kHz à 50 MHz
- Un retard r_c exprimé en terme de colonne allant de 0 à 127 colonnes.

La chronologie des retards imposés au signal de déclenchement est la suivante :

La CCTD reçoit le signal de déclenchement (fabriqué dans la salle de comptage) par une fibre optique à l'instant

$$t_{CCTD} = t_0 + \delta t_{TRIG}. \tag{4.3}$$

Les grandeurs t_0 et δt_{TRIG} on été définies dans le paragraphe S 4.3.

A cet instant t_{CCTD} la carte CCTD retarde le signal de déclenchement de :

$$r_c = f_e \times \delta t_{CCTD} \tag{4.4}$$

A la sortie de la CCTD, un nouveau retard δt_{BUSIF} est imposé par la distribution (cables), et la propagation (circuits programmables) des signaux au travers des cartes BUSIF (cartes décrites au chapitre suivant). Ce délai créé en aval de la CCTD est de l'ordre de 20 ns (une colonne à 50 MHz). Après ce dernier retard, le groupe de circuits programmables implantés sur la BUSIF envoie la commande qui permet d'arrêter la progression de l'échantillonage. Cette commande parvient au CDP au temps t_{TrigMD}.
Le schéma 4.6 illustre cette chronologie. Le schéma circulaire du haut montre le caractère cyclique de l'échantillonage, tandis que celui du bas nous montre la position de la colonne à isoler dans la mémoire RAM du CDP.

La valeur exacte de δt_{CCTD} est determinée par des séances de calibration que nous décrivons par la suite. Ce laps de temps est figé pour une période complète de prise de données. Pour les enregistrements de novembre 1995 la distribution devait considérer :

$$r_c = 43 \tag{4.5}$$

soit 1.72 μs à la fŕequence f_e = 25 MHz.

Il est important de noter que cette carte CCTD est piloté par un transputer T425, qui à pour rôle de programmer les valeurs de f_e et r_c. Nous verrons dans le chapitre suivant comment il s'intègre au réseau de transputer de la multiplicité et à quel moment il intervient lors d'une prise de donnée.

Pour résumer, la lecture des 13824 voies qui forment le détecteur de multiplicité est assurée par 216 couples de circuits FABRIC/CDP qui travaillent en parallèle. Le pouvoir de séparation de deux impacts est imposé par les FABRIC et vaut 40ns. Chaque FABRIC traite 64 canaux simultanément pour être échantilloné en sortie par un buffer circulaire digital, le CDP. Ce dernier à la capacité de garder en mémoire tout les échantillons de 64 bits pendant le temps de latence du trigger. La fréquence d'échantillonage est imposé

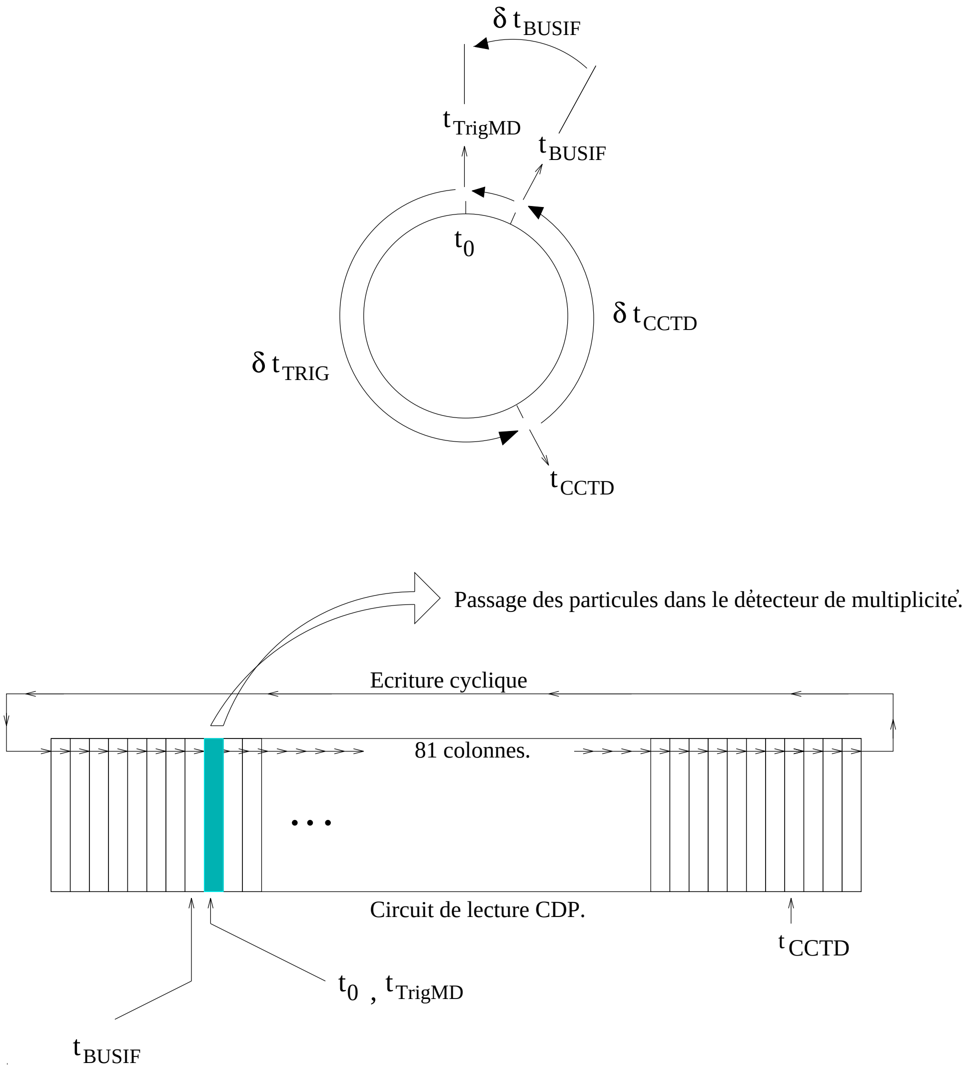


FIG. 4.6 - *Représentation des différents retards subis par le signal de déclenchement, depuis le passage des particules chargées au temps t_0 jusqu'à l'arrêt des pointeurs de colonnes des circuits de lectures CDP au temps t_{TrigMD}.*

au CDP par un organe extérieur : la carte CCTD. Lorsqu'un signal de déclenchement parvient au détecteur, un groupe de circuits programmables entreprend la récupération de la colonne correspondant à l'événement multiplicité corrélé. Nous allons voir entre autres, dans le chapitre suivant comment les circuits programmables des cartes BUSIF récupèrent cette colonne.

Chapitre 5

LE RESEAU DE TRANSPUTERS DE LA BRANCHE MULTIPLICITE

5.1 Introduction

La branche multiplicité, est constituée du détecteur, de son système de lecture associé et d'un dispositif de transfert de données. Nous avons vu, dans le chapitre précédent, comment fonctionne le système de lecture du détecteur basé sur 216 couples FABRIC/CDP. Dans ce chapitre nous allons décrire le réseau de transputers développé pour remonter les données du détecteur vers la salle de comptage. Nous allons voir qu'il est construit autour de 6 cartes BUSIF (BusInterface) chargées de lire et contrôler les 216 CDP. Nous verrons comment chacune de ces BUSIF rend les informations receuillies accessibles au transputer présent sur sa carte. Enfin, nous exposerons les mécanismes développés pour reconstruire et tranférer les données dans un temps minimum.

5.2 Le dispositif de transfert des données développé pour l'année 1995-1996

Toute information relative au passage d'une particule dans le détecteur de multiplicité sort de celui-ci sous forme de signal analogique pour être prise en charge successivement par trois dispositifs :

- un dispositif de lecture, qui numérise cette information,
- un dispositif de transfert qui véhicule cette information vers le lieu de stockage,
- un dispositif d'acquisition qui enregistre l'ensemble des informations récoltées sur bandes magnétiques.

Le sujet principal de ma thèse a été le développement du second point. J'ai réalisé une architecture capable de véhiculer l'ensemble des données du détecteur de multiplicité, depuis les circuits de lecture situés dans le hall de l'expérience, jusqu'à la salle de comptage, où un système d'acquisition prend en charge l'écriture de ces données.

Le cahier des charges était le suivant :

- transférer de façon cyclique 9 Moctets[1] de données sur 80 m de distance en 10 s maximum. La période du cycle étant celle des paquets d'ions déversés par le SPS sur la cible soit 20 s.
- mettre l'ensemble de ces données, d'une part à la disposition du système d'acquisition principal et d'autre part, accessible à tout utilisateur désireux d'utiliser en-ligne ces informations. Ces utilisateurs ne devant pas perturber la bande passante du dispositif d'acquisition.

Ceci m'a conduit à la réalisation d'un réseau de processeurs basé sur 11 cartes transputer de quatre types différents (cf figure 5.1) :

- Le type CCTD : CCTD.
- Le type BUSIF : Busif1, Busif2, Busif3, Busif4, Busif5, Busif6.
- Le type B016 : B016ACQ, B016SPY, B016MX.
- Le type Root : Root.

Nous allons dans la suite de ce chapitre décrire les éléments qui forment ce réseau.

5.3 Descriptions des différents éléments utilisés pour l'élaboration du réseau de transfert

5.3.1 La carte CCTD

Le rôle de cette carte a été décrit au chapitre précédent. Nous rappelons simplement qu'elle distribue les signaux d'horloge au CDP et qu'elle retarde les signaux de déclenchement.

1. volume créé pour 5000 événements par déversement

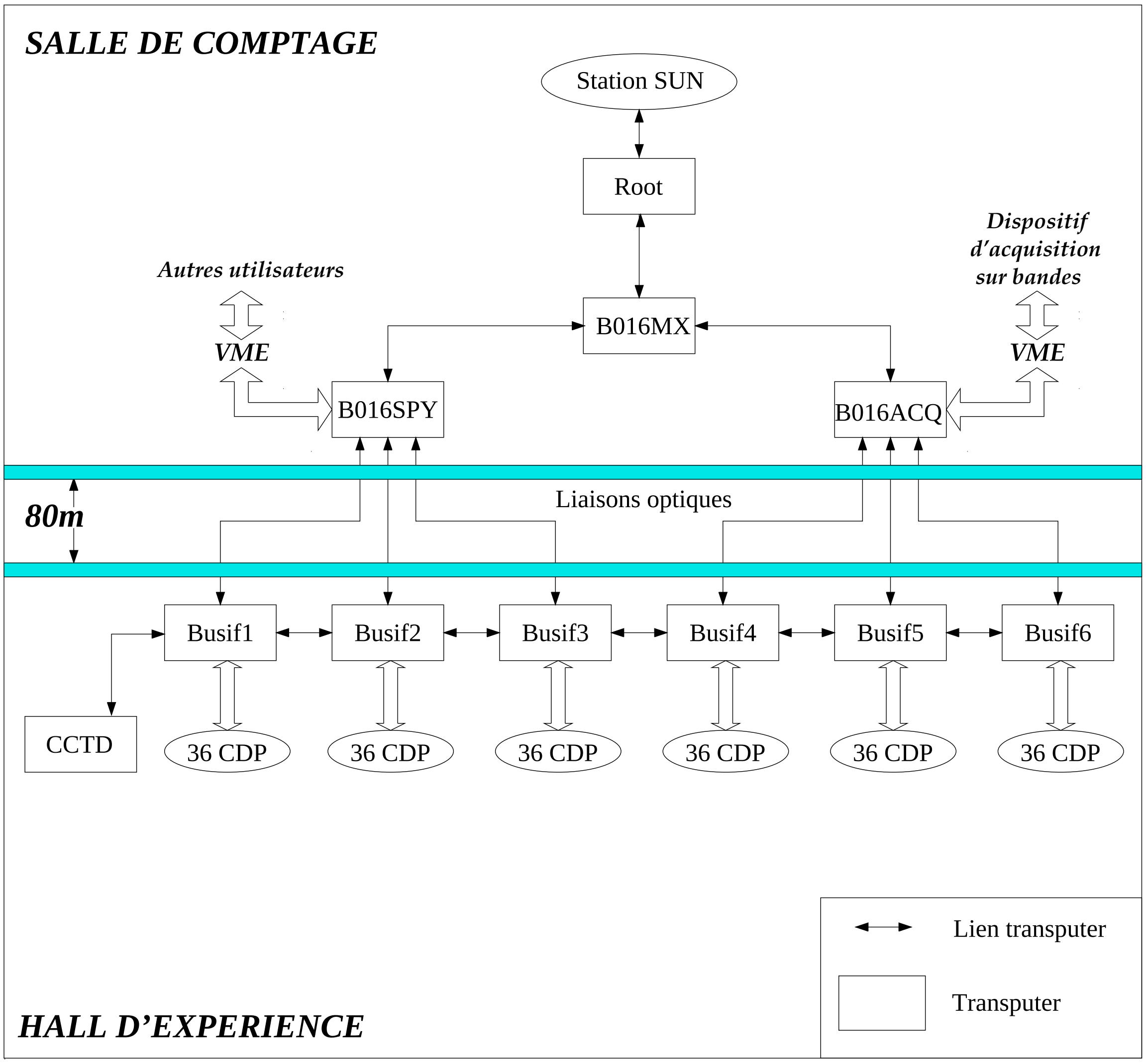


FIG. 5.1 - *Schéma global du réseau de transputers développé pour la multiplicité. Il s'agit d'un réseau en arbre, formé de 11 cartes transputer de 4 type différents (CCTD,BUSIF,B016,ROOT). Il a pour rôle de remonter les données du détecteur de multiplicté vers les deux cartes mémoires B016ACQ et B016SPY. Pour cela, 6 cartes BUSIF font l'interface entre les 216 circuits CDP et les B016.*

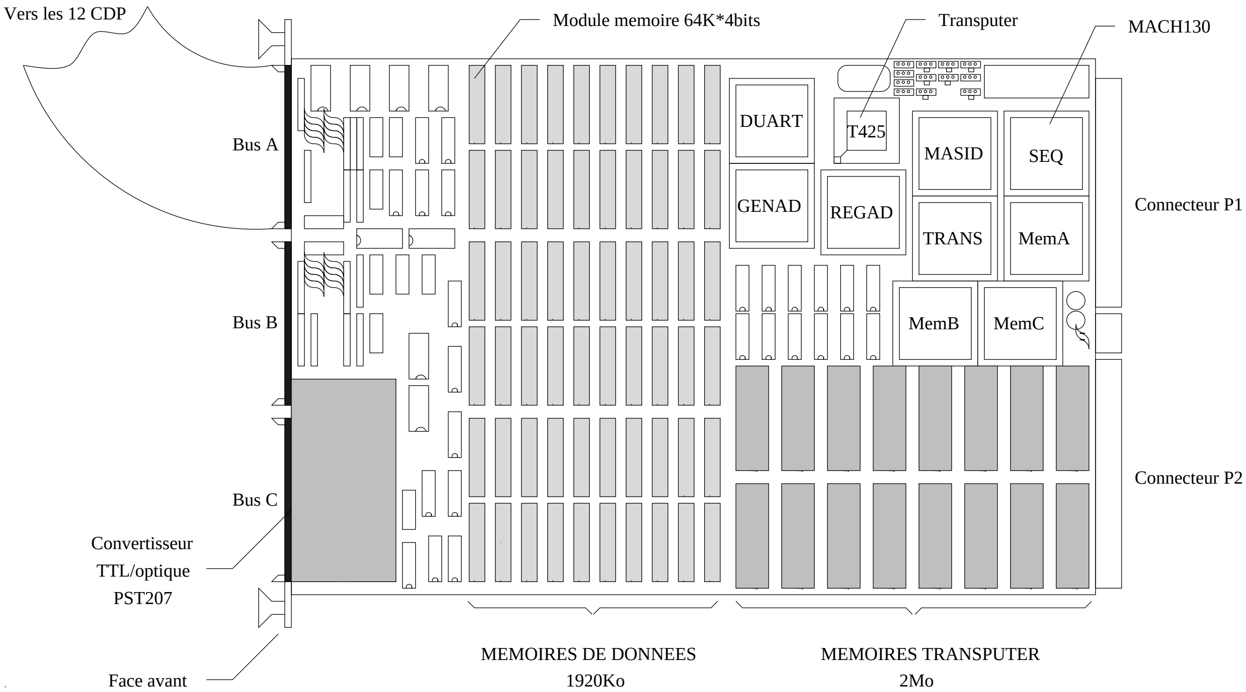


FIG. 5.2 - *Schéma de l'implantation des circuits de la carte BUSIF.*

5.3.2 La carte BUSIF

Cette carte BUSIF (BUS InterFace), développée par M.Forlen au L.A.P.P prend en charge la lecture des circuits digitaux du détecteur de multiplicité (CDP, cf chapitre 4) et rend accessible au monde transputer toutes les informations qui en découlent : la figure 5.1 nous montre les 6 cartes BUSIF qui font l'interface entre les circuits CDP et les mémoires finales BO16. Notons que chaque BUSIF à la charge de lire 36 circuits CDP.

La BUSIF est réalisée au format standard des cartes VXI Taille C, soit 340 $mm \times 233\ mm$.

Une vue de l'implantation des composants est représentée sur la figure 5.2, et permet de localiser les différents modules cités dans les descriptions suivantes.

Une carte BUSIF accède à $1/6^{eme}$ du détecteur complet. Chacune d'entre elles est construite autour de deux dispositifs essentiels :

- Un dispositif de lecture :

  à chaque déclenchement, il est chargé de lire 36 circuits digitaux (CDP) qui échantillonnent l'état du détecteur (une description de ces circuits est donnée au chapitre

4). Ce dispositif de lecture est constitué d'un groupe de 6 cicuits MACH130[2] et d'une mémoire de données de 1 920 Ko à entrées/sorties séparées. Les MACH130 sont des circuits reprogrammables de type PGA. Chacun d'entre eux renferme une structure logique programmée, synchrone ou asynchrone. Ce sont eux qui ont la charge (exécution et contrôle) de tout le protocole de lecture des CDP.
Le bloc mémoire es,t quant à lui, réalisé au moyen de 60 modules CY7C192 de 64 Kmots de 4bits. Les 4/5eme de cette mémoire sont utilisés pour stocker les informations binaires relatives aux impacts sur les bandes de silicium du détecteur. Les 20% restant sont dédiés au stockage de mots de contrôle des données. La capacité de 1 920 Ko permet de supporter un taux de déclenchement de 5 461 événements/bouffée soit plus de 1 000 événements/s. Pour la période de prise de données de novembre 1995, la valeur moyenne était de 500 déclenchements/s en faisceau de Pb à 160 Gev/nucléon.

- Un dispositif de transfert :

  constitué d'un transputer T425 doté de 2Moctets de mémoire pour l'exécution de ses programmes et de trois cicuits MACH130[3].
  Cet ensemble permet d'injecter les données issues du détecteur dans un réseau de transputers.
  Deux des ressources du transputer sont utilisées parallèlement :

  - la ressource CPU : elle conduit d'une part à la reconstruction des événements (Event-Building) et exécute d'autre part un algorithme de validation des données.
  - l'interface lien : sollicitée simultanément avec la précédente, elle permet d'évacuer le contenu des mémoires de données au flux maximum de 500 Koctets/s/lien. Un seul lien est utilisé par carte BUSIF pour acheminer l'information jusqu'à la salle de comptage, au travers de fibres optiques.

Ces deux dispositifs ont accès à une ressource commune : la mémoire de données de 1 920 Ko. Nous voyons sur le bloc diagramme 5.3 que le transputer voit cet espace comme un bloc mémoire unique (accessible en lecture uniquement), tandis que pour le dispositif de lecture, il apparaît divisé en trois zones mémoires distinctes *A,B* et *C* de 640 Ko chacune. Sur une BUSIF, la lecture des 36 CDP est faite par trois PGA (*MemA, MemB et MemC*) qui se répartissent la charge en 12 CDP chacun. Ils travaillent tous les trois simultanément et désservent les trois zones mémoires *A,B* et *C* par l'intermédiaire de trois bus séparés : les bus A, B et C (voir figure 5.3).

La figure 5.4 nous montre que les trois bus (A,B et C) sont constitués :

- d'un bus d'adresse sortant de 4bits. Il permet d'adresser un circuit CDP parmi douze.

2. nommés : *GENAD, MEMA, MEMB, MEMC, SEQ et MASID*
3. nommés : *REGAD, TRANS et MASID*

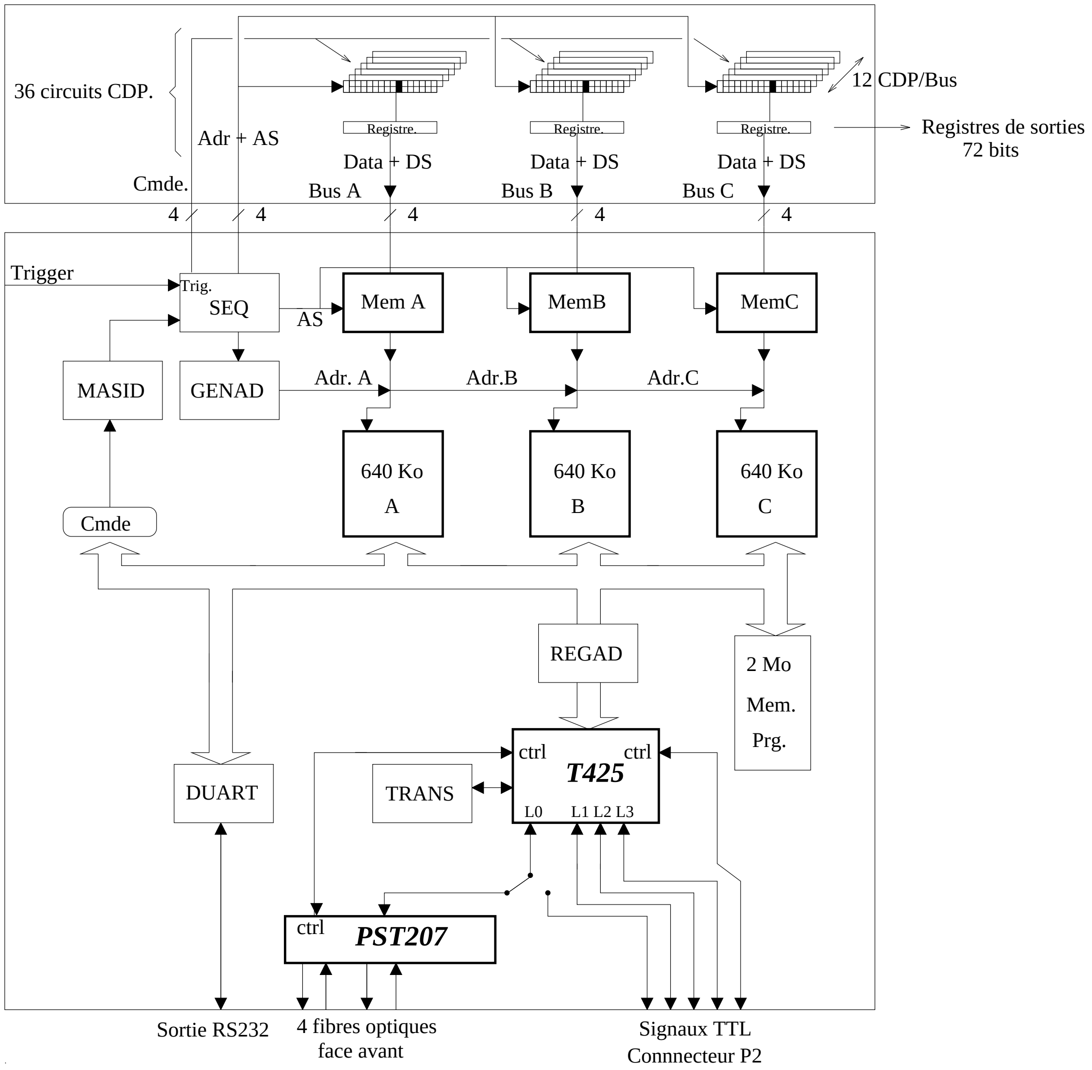


FIG. 5.3 - *Ce bloc diagramme de la carte BUSIF montre les interconnections des différents modules (circuits programmables et transputer) au sein même de la carte. Les modules les plus importants sont marqués en traits gras. Le rectangle de la partie supérieure représente les* 3×12 *circuits CDP accédés par la BUSIF.*

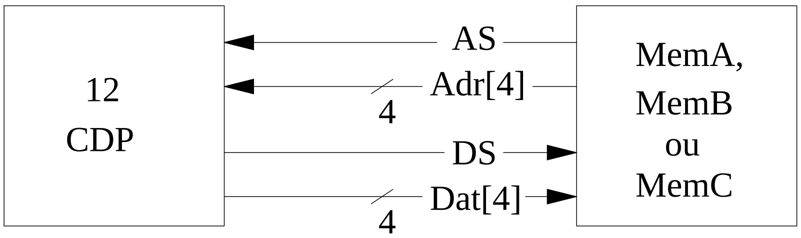


FIG. 5.4 - *Un bus qui relie un circuit programmable (MemA,MemB ou MemC) à ses 12 CDP est formé d'un bus d'adresse de 4 bits, d'un bus de données de 4 bits et de deux lignes de validation (AS et DC).*

- d'une ligne de validation d'adresse (adresse strobe : AS)
- d'un bus de données entrant de 4 bits.
- d'une ligne de validation des données (data strobe : DS)
- d'un bus sortant de 4 bits pour le mot de commande envoyé au CDP.

Les tâches affectées aux différents circuits programmables visibles sur la figure 5.3 sont les suivantes :

- Le circuit *GENAD* : impliqué dans le dispositif de lecture, ce circuit renferme une logique programmée qui permet de générer les adresses destinées à placer les mots de 64 bits dans les mémoires de 640 Ko (A, B et C). Il est utilisé en aval du séquenceur principal *SEQ*.
- Les 3 circuits *MemA, MemB* et *MemC* sont aussi affectés au dispositif de lecture. Ils sont cadencés par le séquenceur principal *SEQ*.

  Nous avons vu au chapitre 4 (paragraphe 4.4), que chaque registre de sortie d'un circuit CDP comporte 72 bits. A chaque déclenchement, 12 adresses (pour les 12 CDP) seront posées successivement sur le bus d'adresse, et à chacune d'entre elle seront associés 18 cycles de lecture pour vider le contenu du registre de sortie du CDP. Chaque cycle de lecture est effectué par un état du séquenceur principal. Les deux premiers sont dédiés à la lecture des 8 bits du numéro de colonne, tandis que les 16 derniers cycles permettent de lire les 64 bits de données. Chaque CDP qui voit s'achever ses 18 cycles de lectures, renvoit sa propre adresse (*Chip Id* sur 4 bits[4]).

  Après chaque phase de lecture d'un CDP, un bit de Time-Out (noté parfois TO) est forcé à la valeur 1 si les 18 cycles se sont déroulés correctement. En revanche, si cette phase ne peut se terminer dans un laps de temps contrôlé par la charge d'une capacité, une valeur 0 sera maintenue pour ce bit, signifiant un problème de lecture pour le circuit concerné (cf figure 5.5). Les 36 CDP associés à chaque BUSIF voient l'incrémentation de leurs pointeurs réactivée lorsque tous ont été lus. Ceci

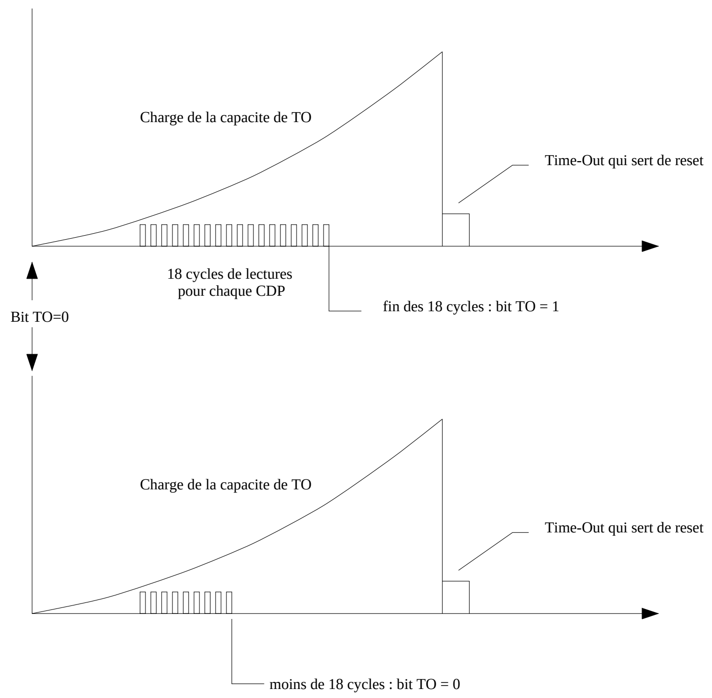


FIG. 5.5 - *Chronogrammes des deux types de phases de lecture d'un circuit CDP*

doit assurer une progression homogène des 36 numéros de colonnes pour chaque carte.

L'ensemble des 7 bits du numéro de colonne (pointeur de la RAM interne du CDP), des 4 bits du numéro de CDP et du bits de Time-out constitue un mot de contrôle de 12 bits appelé *TimeStamp*. Les trois séquenceurs *MemA, MemB* et *MemC* associent un mot *TimeStamp* à chaque donnée de 64 bits. Ces informations sont stockés dans trois zones mémoires disctinctes : it MemStampA, it MemStampB et it MemStampC. Nous verrons au chapitre 6 qu'une étude de ces informations permet d'isoler un grand nombre de pannes de lecture, et conduisent ainsi à la construction d'un critère de validation des données. De plus, une telle analyse en-ligne permet d'éviter un stockage brut de ces mots de 12 bits, qui représentent 18.75% de la quantité de données totale.

- le PAL *SEQ* est constitué d'une machine d'état. Celle ci est chargée de cadencer l'ensemble des opérations des PAL précédemment décrits. C'est le séquenceur principal. La fréquence de travail de cette machine d'état est imposée à 25 MHz par la carte transputer CCTD.

- le PAL *MASID* est reservé pour le décodage des mots de commande écrits par le transputer. Il permet au transputer de piloter le dispositif de lecture par l'intermédiaire de registres. En prise de donnée normale,*MASID* peut interprète deux valeurs du registre de commande : la valeur 10000000_{hex} qui signifit l'état de lecture

4. numéro entre 1 et 12

des CDP (*Acquisition*) et la valeur 00000000 qui signifit l'état de repos (*NoAcquisition*) pour le groupe de circuits programmables du dispositif de lecture. Le circuit *MASID* agit directement sur le séquenceur principal *SEQ*.

- le PAL *TRANS* arbitre les accès du transputer au bus de 32 bits, et contrôle son activité.

Cet assemblage de neuf circuits programmable de type MACH130, chargé de surperviser l'ensemble des opérations de cette carte, offre une souplesse de restructuration indispensable pour les mises au point. En effet, une grande partie des mises au point, consistent à corriger la logique de fonctionnement de ce groupe de PAL en changeant leur structures internes, via un programmateur et ce, sans affecter le routage de la carte.

Les liens du transputer T425 permettent finalement d'interfacer les trois blocs mémoires de 640 Ko avec d'autre transputers. Les 3 liens L1,L2 et L3 sortent en signaux TTL sur le connecteur standard arrière VME référencé P2. Ils permettent la connection avec trois autres processeurs du même type dans une limite de 30 à 40 cm de distance. Le lien L0 de ce processeur, grâce à un cavalier peut, quant à lui, être basculé soit vers l'interface optique PST207 soit sur ce connecteur P2 arrière. La liaison optique permet d'établir des dialogues sur des distances de l'ordre du kilomètre entre deux transputers. Elle est utilisée ici pour connecter les processeurs proches du détecteur avec ceux de la salle de comptage, distants de 80 mètres. On parlera de lien L0 utilisé "en optique" ou bien "en TTL", selon la configuration choisie. Le choix de L0 en TTL est particulièrement intéressant lors de tests en laboratoire, puisqu'il permet de s'affranchir de toutes les interfaces optiques, ciblant ainsi la recherche de pannes.

L'interface optique PST207 utilisée sur la BUSIF, l'est aussi sur une autre carte, c'est la raison pour laquelle il fait l'objet d'une description en tant que carte à part entière.

5.3.3 L'interface optique PST207

Il s'agit d'un module qui permet au transputer de communiquer sur de grandes distances. Il remplit simultanément les fonctions de canal émetteur et de canal récepteur. Son bloc diagramme est présenté sur la figure 5.6.
Ce module, au standard INMOS est produit par la société *PARATECH SOLUTIONS*. Pour cela, il utilise à l'émission, un convertisseur TTL/optique (LED), qui transforme le signal électrique en un signal lumineux avec une longueur d'onde de 820 nm (rouge visible). Les temps de conversions TTL/optique mesurés sont de 65 ns. Nous verrons par la suite que cette valeur contribue à faire chuter la bande passante sur lien optique. Les ondes lumineuses sont directement transmises au travers d'une fibre optique (diamètre de 50 μm). En réception, le PST207 convertit la lumière qui provient de la fibre optique en un signal électrique qui subira diverses étapes de remise en forme afin de produire le signal équivalent au standard lien INMOS. Ce module possède en plus deux convertisseurs chargés de véhiculer les signaux de contrôles du transputer. Les PST207 sont prévus pour soutenir une fréquence de dialogue de 20 MHz.

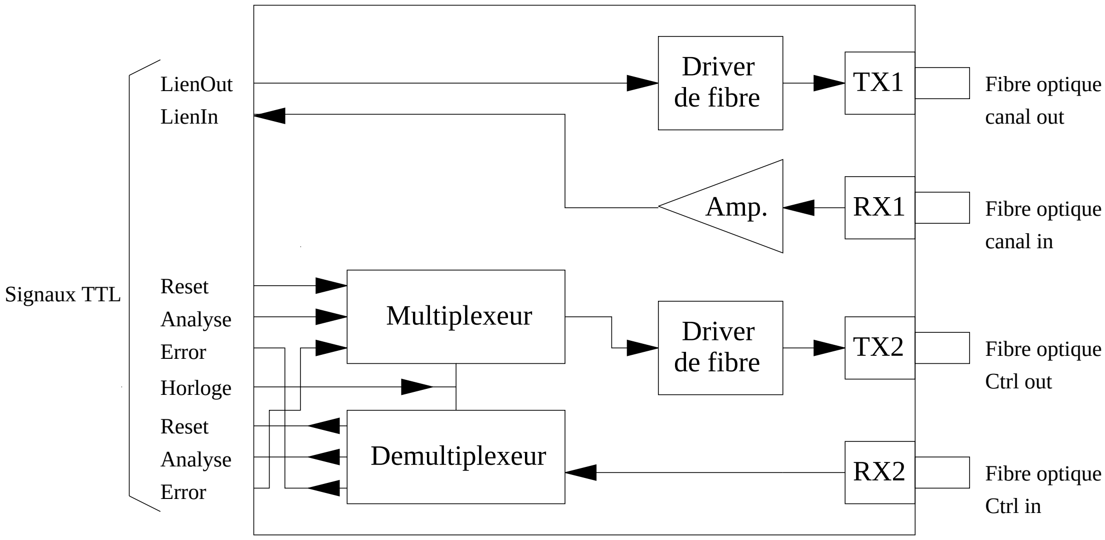


FIG. 5.6 - *Bloc diagramme du module convertisseur PST207.*

5.3.4 Les supports des canaux transputers

Nous avons vu au chapitre 3 que pour construire un réseau de transputer, il est nécessaire d'établir des liens entre les processeurs. Ces liaisons peuvent être matérialisées par des signaux "quelconques", pourvu qu'ils respectent le protocole de dialogue INMOS.

Sur NA50, on rencontre trois types de liaisons entre transputers :

- **La liaison TTL** : c'est la liaison qui transporte les signaux tels qu'ils sortent du transputer. Elle est faite au moyen de deux fils (ou pistes sur une carte), le *lien-In et le lien-Out*. Sa longueur ne peut pas excéder les 40 cm sous peine de voir se détériorer les signaux par atténuation. Ce type de liaison pose de gros problèmes de diaphonie, lorsque plusieurs canaux de liens sont transmis par l'intermédiaire de cables plats.

- **La liaison différentielle** : c'est la liaison que l'on utilise dans NA50 pour transporter les informations entre deux transputers. Elle utilise deux fils par canal (le signal positif et négatif) et permet d'éliminer par différence les parasites qui affectent de façon identique ces deux signaux. Avec ce type de propagation, les liens transputers ne peuvent dépasser des longueurs d'une dizaine de mètres. Quatre fils sont utilisés pour former un lien transputer différentiel.

- **La liaison optique** : c'est la liaison qui permet comme nous l'avons dit, d'établir des communications entre des processeurs très éloignés. Ce type de liaison présente en outre l'avantage d'être immunisé face aux perturbations électromagnétiques et radiofréquences. Elle est donc tout à fait adaptée aux environnements électriquement

bruyants comme les halls d'expériences de physique des particules. De plus l'atténuation d'un signal est proportionnelle à la longueur de la ligne pour une fibre optique tandis qu'elle croit avec le carré de celle-ci pour une liaison par cables ordinaires. Elle présente cependant le désavantage d'être coûteuse et fragile. Sur NA50, ce type de liaisons est utilisé pour matérialiser les connections entre les transputers situés à coté du détecteur et ceux de la salle de comptage. Deux fibres optiques sont utilisées pour former un lien. Un total de 12 fibres est nécessaire pour connecter les 6 cartes BUSIF aux deux modules B016 (cf fig 5.1).
Etant donné la vitesse de propagation dans la fibre optique (de l'ordre de 4 ns/m), cette ligne de 80 mètres introduit un retard de :

$$\delta t_{fibre} = 4 \times 80 = 320ns. \tag{5.1}$$

Il faut ajouter à cela les retards qu'impliquent les deux conversions optiques/TTL des modules PST207 soit :

$$\delta t_{lien-optique} = 320 + 2 \times 65 = 450ns. \tag{5.2}$$

Le protocole de dialogue utilisé par le transputer (cf section 3.4.7 : un acquitement bloquant par octet transmis) est donc tributaire de cette ligne à retard. On en déduit les bandes passantes attendues pour les deux modes de fonctionnement :

$$\phi_{20Mbits/s} = \frac{10^9}{2 \times (450 + 275) \times 1024} = 673Ko/s. \tag{5.3}$$

$$\phi_{10Mbits/s} = \frac{10^9}{2 \times (450 + 550) \times 1024} = 488Ko/s. \tag{5.4}$$

Il est clair, que pour assembler le réseau décrit figure 5.1 il est necessaire de réaliser des conversions de signaux pour rendre compatibles ces trois types de liaisons : des convertions TTL/optique et TTL/Différentielles. Le premier type d'adaptation est réalisé au sein même des modules PST207. Les conversions TTL/Différentielles sont faites par des cartes *TRANSIBELIENS* [21] réalisées par M.Forlen au L.A.P.P.

5.3.5 La carte *B016*

Il s'agit d'une carte transputer/VME [35], maître-esclave, utilisée sur NA50 comme interface entre les mondes transputer et VME. Comme on peut le voir sur la figure 5.1 ce sont ces cartes qui mettent les données dans les chassis VME *ACQ* et *SPY*.
Produit de la société INMOS, elle est construite au format VME 6U et présente entre autre, les caractéristiques suivantes (cf figure 5.7) :

- interfaçage avec le bus VME au moyen d'un circuit standard VIC068 implanté sur la carte,

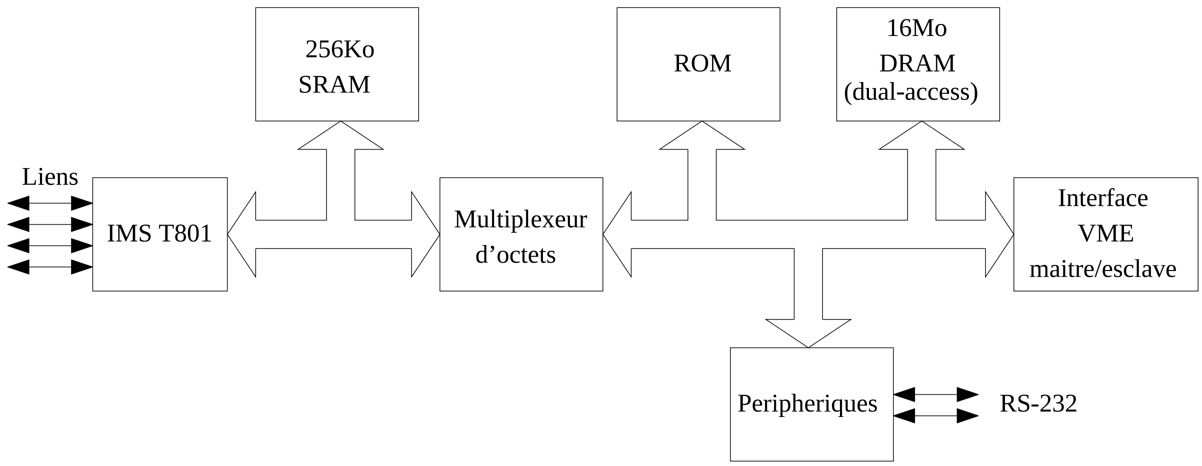


FIG. 5.7 - *Bloc diagramme de la carte INMOS B016.*

- un transputer IMS T801 permet de réaliser l'interface entre un réseau de transputer et un bus VME,
- un bloc de 256 Ko de mémoire RAM (la SRAM pour Static RAM), avec 80 ns de cycle, est dédié au transputer,
- 16 Mo de mémoire RAM double-accès (la DRAM pour Dynamic RAM) sont accessibles soit par le T801 soit par le bus VME. Elle peut être accédée en mode esclave par tout cycle VME : A32/24/16, D32/16/8, RMW, UAT, BLT,
- le transputer peut se comporter comme maître VME standard,
- tous les types et niveaux d'interruptions VME peuvent être générés,
- cette carte peut remplir les fonctions d'arbitreur VME,
- un multiplexeur d'octets programmable, permet de faire dialoguer le transputer avec des machines VME *big ou little endian*,
- deux ports de communication RS-232 utilisant l'interface DUART 2681. Ils permettent des dialogues à faible débit avec l'extérieur (typiquement 9600 bits/s). Ces ports sont utilisés sur NA50 pour l'affichage en "direct" d'échantillons de données, et pour échanger de façon non perturbative des informations entre le réseau de transputers multiplicité et l'autre réseau de 13 transputers.

Il est intéressant de noter que les quatre liens transputers sortent sur le connecteur P2 arrière de la carte. Chacune de ces cartes est donc couplée à une carte *TRANSIBELIEN* [21] afin de permettre au transputer T801 de dialoguer sur des liaisons différentielles.

Le transputer de la carte *B016SPY*, par l'intermédiare du bus VME se synchronise sur le cycle des déversements du faisceau SPS.

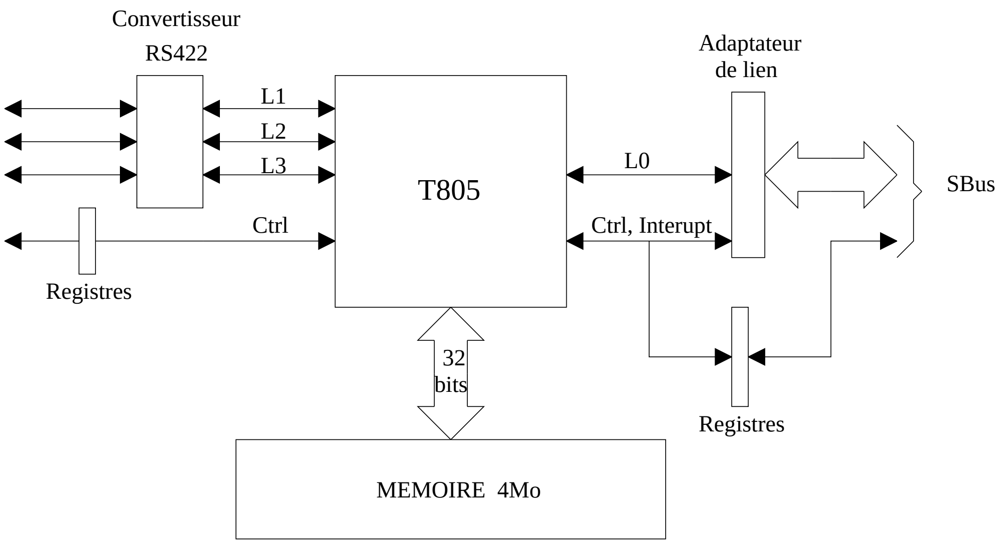


FIG. 5.8 - *Bloc diagramme de la carte Root (interface utilisateur) ALEX-TITAN-1S.*

Sur la figure 5.1, on peut noter la présence d'une troisième carte B016 : B016MX. Elle est utilisée pour échanger les contenus des deux mémoires *ACQ* et *SPY*. En effet, sans opérer à un tel processus d'échange, chacune de ces B016 contiendrait la moitié des données multiplicité. Nous verrons au cours du chapitre 6 que cet échange, et plus généralement le transfert complet, est géré par un protocole d'émission de petits paquets.

5.3.6 La carte Root et la machine hôte

Comme nous en avons parlé au chapitre 3 paragraphe 3.7 la carte Root a pour fonction de faire le lien entre la machine hôte et le réseau de transputer. Cette machine hôte qui joue le rôle d'interface utilisateur est une station SUN-SPARC5 qui renferme tous les logiciels permettant les développements d'architectures concurrentes : éditeurs, compilateurs, éditeurs de liens, collecteurs, configureurs et serveurs de réseau. La carte Root établit la connection entre le S-BUS de la SUN et le lien 0 du transputer communément appelé *root*. Il s'agit d'une carte ALEX-TITAN-1S [37], implantée dans la station SUN. Elle intègre un transputer T805 doté de 4 Mo de mémoire RAM. Produite par la société Alex-Informatique, son bloc diagramme est présenté sur la figure 5.8.

5.4 Conclusion

Nous venons de décrire les quatre types de cartes utilisées pour former le réseau de 11 transputers de la branche multiplicité. Nous avons vu que des fibres optiques permettent d'établir des liens de 80 m entre les 6 cartes BUSIF, situées dans le hall d'expérience et les cartes B016 de la salle de comptage. Dans le chapitre suivant, nous allons décrire l'application multitâches, implantée sur ce réseau de processeurs. Elle a pour fonction de filtrer, reconstruire et remonter les données contenues dans les mémoires des cartes BUSIF.

Chapitre 6

LE TRANSFERT DES DONNEES DE LA BRANCHE MULTIPLICITE

6.1 Introduction

Dans le chapitre précédent, nous avons décrit le réseau de transputer de la branche multiplicité, du point de vue hardware. Nous allons maintenant détailler le fonctionnement du logiciel multiprocessus, développé pour s'implanter sur ce réseau physique. Son objectif est d'exécuter, au rythme du cycle SPS, la reconstruction, la validation et le transfert des 9 Moctets d'information du détecteur de multiplicité, et ce en moins de 10 s.

Nous verrons qu'il intègre un algorithme de validation des données qui permet de détecter en-ligne des défauts dans les liaisons BUSIF's-CDP (voir chapitre 4). Cet algorithme permet d'autre part de repérer des problèmes de synchronisation entre circuits CDP.
Le second rôle de ce logiciel est de remonter les données validées du détecteur de multiplicité, jusqu'aux mémoires finales B016ACQ et B016SPY. Nous exposerons le mode de transfert par paquets, mis en place pour minimiser les pertes de temps lors d'une reprise sur erreur. Nous montrerons qu'il existe un fractionement optimum des blocs mémoire à transmettre, qui minimise le temps global de transfert.

Pour tous ces développements, j'ai choisi de construire une application, par transputer du réseau (écrite en C paralllèle). Nous allons voir qu'elles peuvent être, elles, même formées de plusieurs processus. Ainsi, selon les types de ressources utilisées par ces processus, il conviendra de procéder à des exécutions concurrentes ou non. Ces choix permettent de diminuer les temps d'exécution et ne peuvent se faire qu'en connaissant d'une part les caractéristiques des cicruits transputer (chapitre 3) et d'autre part les caractéristiques des éléments du réseau physique (chapitre 5).

La figure 6.1 nous montre un schéma d'ensemble des processus qui forment ce logiciel. Nous voyons qu'il renferme 35 processus (représentés par des cercles fléchés), répartis

sur les 11 transputers du réseau physique (voir chapitre 5 figure 5.1). Dans la suite de ce chapitre nous désignerons les processus par les numéros qui leur correspondent sur la figure 6.1.
La progression des paragraphes qui vont suivre est inspirée du déroulement temporel des opérations pendant un cycle SPS. Nous suivrons donc l'activité et le rôle des différentes tâches depuis la détection d'une fin de déversement des ions sur la cible, jusqu'à l'arrivée des données dans les mémoires finales *B016ACQ* et *B016SPY*.

6.2 Configuration des processus du réseau

Avant toute prise de données, il est impératif de configurer les processus du réseau de transputers. Cette première phase interactive, permet à l'utilisateur de décider des cartes BUSIF participant à la lecture du détecteur (topologie du réseau). Elle permet d'autre part de choisir une fréquence d'échantillonage f_e des CDP et un retard r_c imposé par la carte CCTD, au signal de déclenchement. Ces deux grandeurs ont été définies au chapitre 4, paragraphes 4.4 et 4.5.

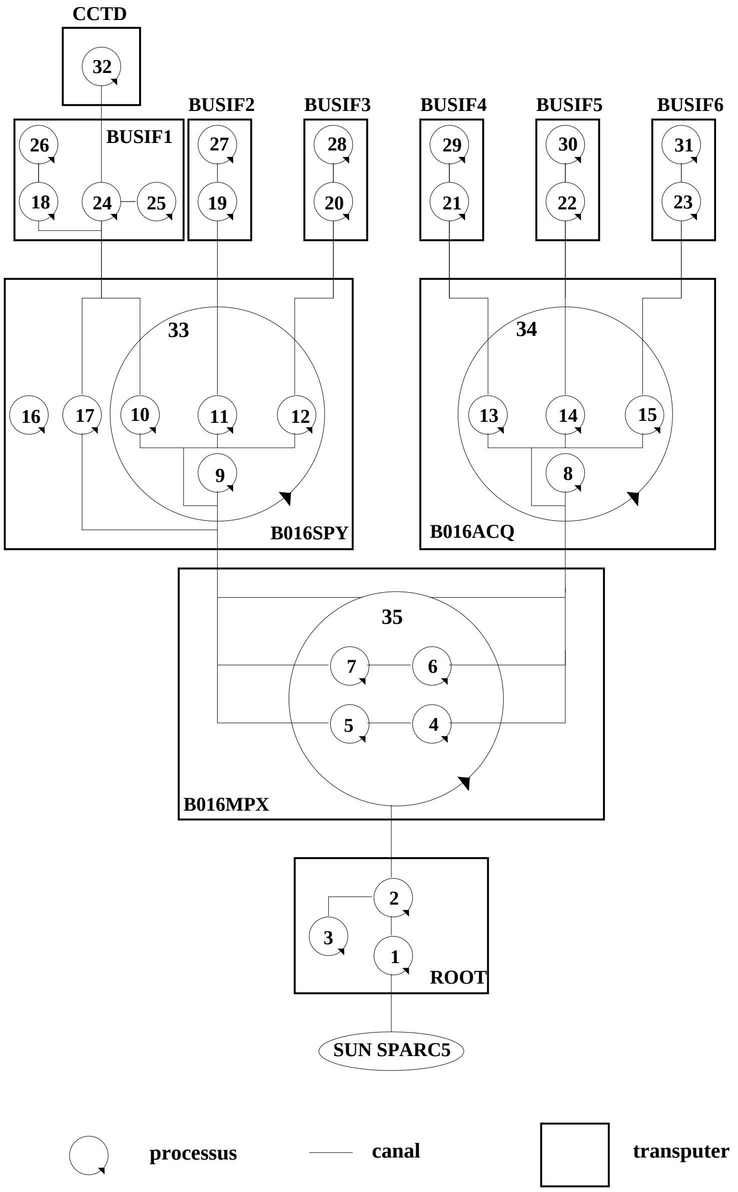
Pour cela, l'utilisateur fait des choix qui initialisent certaines variables dans les mémoires des transputers. Les choix sont effectués par l'intermédiaire de menus de configuration, proposés à l'utilisateur par les processus impliqués dans cette phase. En fonction des réponses de l'opérateur, le processus cible entreprend des actions de configurations localisées sur son processeur.
Les différentes tâches actives durant cette phase de configuration, ainsi que les menus qu'elles proposent, sont repésentées sur la figure 6.2.

Consulter un menu demande d'ouvrir une voie de communication entre le processus interface utilisateur (numéro 1) et le processus interlocuteur. Le choix d'une option, demande l'ouverture d'un dialogue entre ces processus.

Notons que pendant cette phase un processus peut être cible du dialogue ou bien intermédiaire si la tâche adressée par l'utilisateur est située en aval. Tout processus doit donc reconnaître si une information en transit lui est ou non destinée.

Pour cela, à chaque cible potentielle d'un dialogue, est associé un identificateur unique codé sur 32 bits. Lors de la phase de configuration l'information qui circule sur les canaux transputers est formée de trois entités successives :

- un code de 32 bits qui porte l'identité du processus à contacter,
- la taille du message à transmettre, codé sur 32 bits,
- le message.



LISTE DES PROCESSUS :

1 : interface utilisateur
2 : multiplexeur
3 : sauvegarde sur disque
4 : reception de B016ACQ
5 : emission vers B016SPY
6 : emission vers B06ACQ
7 : reception de B016SPY
8 : reception de B016SPY
9 : reception de B016ACQ
10 : reception de BUSIF1
11 : reception de BUSIF2
12 : reception de BUSIF3
13 : reception de BUSIF4
14 : reception de BUSIF5
15 : reception de BUSIF6
16 : mesure de periode SPS
17 : multiplexeur
18 : emetteur BUSIF1
19 : emetteur BUSIF2
20 : emetteur BUSIF3
21 : emetteur BUSIF4
22 : emetteur BUSIF5
23 : emetteur BUSIF6
24 : multiplexeur
25 : configuration BUSIF
26 : validation data BUSIF1
27 : validation data BUSIF2
28 : validation data BUSIF3
29 : validation data BUSIF4
30 : validation data BUSIF5
31 : validation data BUSIF6
32 : configuration CCTD
33 : gestion des receptions/emissions de B016SPY
34 : gestion des receptions/emissions de B016ACQ
35 : gestion des echanges entre B016ACQ, B016SPY et ROOT

FIG. 6.1 - *Schéma et implantation du réseau de 35 processus qui constituent le logiciel de transfert des données pour la branche multiplicité. Les noms des transputers sont à mettre en correspondance avec ceux du schéma 5.1 au chapitre 5. Ce logiciel va nous permettre de remonter, à chaque cycle SPS, les données du détecteur de multiplicité après les avoir reconstruites et validées. Il traite et transfert pour cela une masse de 8 Moctets d'information en moins de 10 s.*

Le code identificateur de 32 bits permet "d'ouvrir la voie" pour transmettre le message au processus cible.

Cette phase de configuration ne demande pas d'importants transferts de données, puisqu'on se limite à quelques centaines d'octets échangés.

Lorsque le dialogue est établi entre le processus numéro 1 (interface utilisateur) et la CCTD, l'utilisateur, par ses choix va affecter des registres internes à la carte. Les contenus de ces registres ne pourront être changés que par une nouvelle configuration ou bien une coupure d'alimentation de la carte.

La topologie choisie pour le réseau est transmise aux cartes *B016 ACQ,SPY* et *MPX*. En effet, ces cartes sont les clientes vis à vis des données des BUSIF. Par conséquent elles doivent *savoir* où s'adresser (cartes BUSIF accessibles) pour recevoir l'information détecteur.

Pour finir, tous les choix effectués par l'opérateur lors de cette phase de configuration sont sauvegardés par le processus numéro 3, dans des fichiers résidant sur la machine hôte. Cette sauvegarde permet de s'affranchir d'une nouvelle intervention d'un opérateur en cas de redémarrage du logiciel.

6.3 Détection d'une fin de déversement

Comme nous l'avons déjà dit au chapitre 3 (paragraphe 3.2.1), nous avons adopté sur NA50 une méthode de transfert par bouffée et non par événement, ceci afin de limiter les temps morts liés à la prise en charge de chaque événement (mémorisation, lecture et transfert). En ce qui concerne la branche multiplicité, pour les cartes BUSIF, un tel choix minimise les dialogues entre le dispositif de transfert (transputers) et le dispositif de lecture des CDP (circuits programmables) : les seuls échanges entre ces éléments sont des synchronisations en fin de déversement et en fin de transfert. Ceci accroît la fiabilité de l'ensemble.

Pendant toute la durée d'une bouffée de faisceau, le réseau de transputers est en attente. Ce n'est qu'en fin de déversement, que l'ensemble des processus qui prennent part à la reconstruction, au transfert et à la validation des données doit être activé (processus $4 \rightarrow 15$, $18 \rightarrow 23$, $26 \rightarrow 31$, 34 et 35). Il est donc indispensable de détecter cette transition du cycle SPS.
Ce changement d'état du faisceau, est caractérisé dans la salle de comptage par un signal NIM qui peut prendre deux états logiques:

- Un état haut qui indique l'absence de déversement d'ions.
- Un état bas maintenu pendant toute la durée d'une bouffée de faisceau.

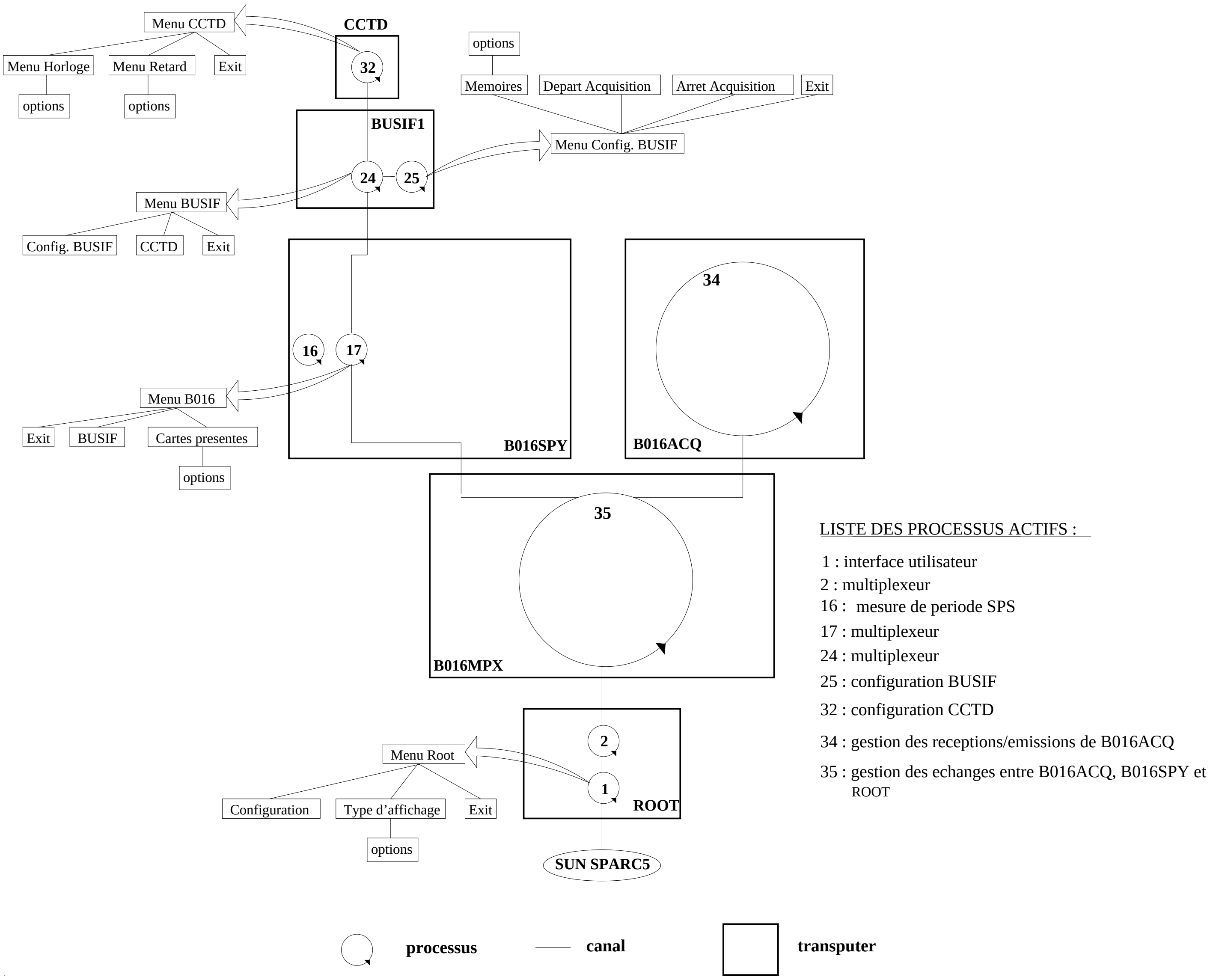


FIG. 6.2 - *Schéma des processus actifs pendant la phase de configuration du réseau. Nous voyons que chaque processus gère un menu qui permet à l'utilisateur d'initialiser des variables locales du transputer. Consulter un menu consiste à établir un dialogue entre le processus numéro 1 et le processus qui gère le menu.*

Un front montant de ce niveau nous indique donc le changement d'état recherché : la fin d'une bouffée d'ions sur la cible. Ce signal est envoyé sur une voie d'échelle (scaler VME V260 de CAEN). Ce module permet de compter le nombre de fronts montants appliqués sur la voie d'entrée. La valeur du compteur est un registre mémoire 16bits, accessible via le bus VME.

C'est le transputer de la carte *B016SPY* (visible au chapitre 5 figure 5.1) qui scrute, toutes les $\frac{1}{20}$ *s*, la valeur de ce registre en attendant l'incrément. Il s'agit là de la seule opération *Maître VME* effectuée par ce transputer. La férequence de lecture de ce registre par le transputer est faible pour ne pas perturber la bande passante du bus VME. De plus, une résolution de $\frac{1}{20}$ *s* pour détecter une fin de déversement est largement suffisante, puisque toute lecture des circuits CDP est suspendue depuis l'instant réel de la fin de burst (absence de trigger). Nous verrons par la suite que ce laps de temps est négligeable, en terme de retard de transfert, comparé au déroulement total qui approche les 10 s. Toutes ces opérations de scrutation sont faites par le processus 33 du tranputer B016SPY (voir figure 6.1).

Lorsque ce transputer à détecté un signal de fin de déversement, il doit "activer" les processus de reconstruction, de transfert et de validation cités plus haut. Pour cela, le processus 33 va émettre des codes sur les quatres liens du transputer. Ces codes vont se propager de proche en proche, indiquant ainsi aux processeurs connexes, la phase courante du cycle SPS. C'est sur la réception de ces codes que les processus 4 $\rightarrow$ 15, 18 $\rightarrow$ 23, 26 $\rightarrow$ 31, 34 et 35 vont entrer en action : le réseau est dans sa phase active.

Sur les cartes BUSIF, les processus 18 $\rightarrow$ 23, après réception des codes de fin de déversement, forcent le dispositif de lecture de la carte à passer en état *No Acquisition* (voir chapitre 5 paragraphe 5.3.2). Ceci permet au transputer de sortir de son activité réduite. En effet, nous avons vu que la priorité était accordée à la lecture des données pendant les bouffées de faisceau, l'activité transputer étant alors très réduite. Ce repos forcé du processeur se traduit notamment par un accès interdit aux mémoires de données lorsque le système est en état de lecture des CDP (24 $\mu s/evt$ pour les trois bus).

6.4 La reconstruction des événements

Les cartes BUSIF seront serveurs de données pour les B016 après avoir procédé à une reconstruction en ligne des événements mémorisés. Ceci est la première étape à franchir avant d'amorcer les transferts de données proprement dit.

D'une manière générale, reconstruire un événement, revient à rassembler dans une seule zone mémoire, toutes les données et informations qui concernent cet événement. Nous effectuons ici, une reconstruction au sein de chaque BUSIF. Ceci permet aux analyses en-ligne, pratiquées sur la B016SPY, de s'affranchir de ce travail gourmant en temps CPU.

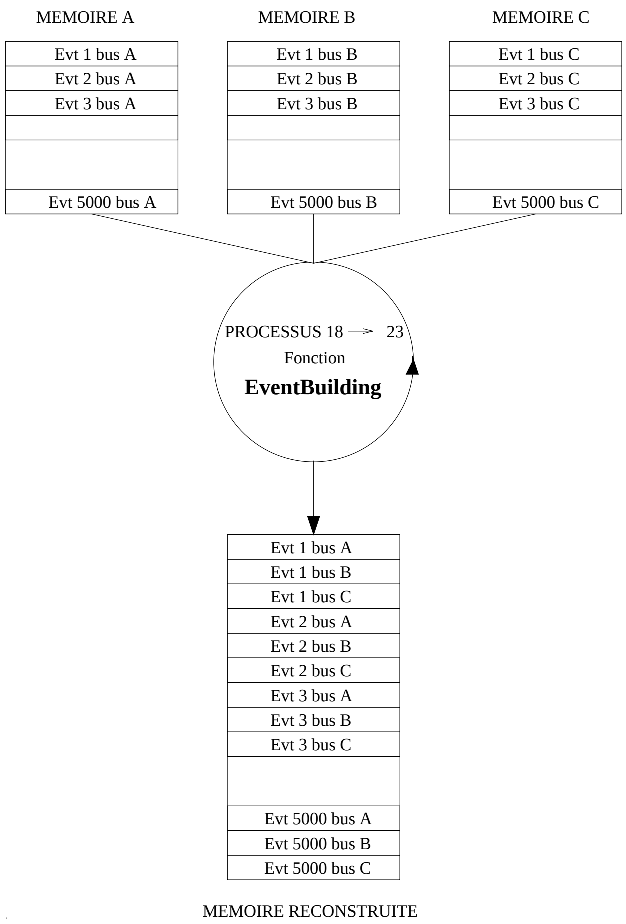


FIG. 6.3 - *Reconstruction des événements multiplicité au sein des cartes BUSIF. Nous voyons que le processus chargé de la reconstruction, rassemble les données des 3 blocs mémoires séparés en une seule zone de mémoires contigues : la mémoire reconsrtuite.*

6.4.1 Le principe de reconstruction

Nous savons que sur chaque BUSIF, les données du détecteur sont placées événement par événement dans trois blocs mémoire A, B et C, chacun étant associé à un bus de lecture (voir chapitre 5 paragraphe 5.3.2). Le transputer les accède en lecture par son bus 32 bits.

Une reconstruction oblige à créer une nouvelle zone mémoire qui contient une succession d'événements *ABC* (cf figure 6.3). Cette zone mémoire reconstruite est allouée par le transputer dans sa mémoire de code, elle occupe 1 600 Ko. Pour effectuer ce travail de réassemblage, j'ai créé une fonction *EventBuilding()* appelée sur les BUSIF par les processus 18 $\rightarrow$ 23. Elle parcourt systématiquement en lecture les trois blocs de données A, B et C pour créer la mémoire attendue.

Pour dupliquer les événements issus de *A,B et C* vers la mémoire reconstruite, on utilise la fonction optimisée *_memcpy()*. Cet routine, est une version du *memcpy()* standard directement traduite en assembleur transputer. Le gain de temps est de 30%.

La quantité de données manipulée pour chaque événement, pendant cette phase, est relative aux trois bus A, B et C soit :

$$Q_{evt} = 3\ bus \times 12\ CDP \times 64bits = 288\ octets/evt \tag{6.1}$$

Lorsque le CPU est entièrement dédié à cette tâche, le temps de reconstruction mesuré pour 3 000 événements est de 0.28 s. On en déduit le temps de reconstruction par événement :

$$t_{evt} = \frac{0.28\ s}{3000} = 93\mu s/evt \tag{6.2}$$

La valeur de 0.28 s a été mesurée avec la fonction *ProcTime()* qui possède une résolution de 64 μs. D'une manière générale, toutes les mesures de temps sont réalisée avec cette fonction standart.

Le temps total de 0.28 s nécessaire à la reconstruction de 3000 événements passera à 0.56 s avec une l'exécution concurrente d'un processus de condensation des mots de contrôle *TimeStamp*.

Une fois cette reconstruction achevée, les données sont prètes pour subir un transfert à destination des mémoires finales B016. Notons que, quelle que soit la "qualité" de ces données, elles seront toujours transférées intégralement.
Ces transferts, gouvernés par un protocole d'échanges de paquets de petites tailles, sollicitent principalement les interfaces liens qui accèdent à la mémoire en DMA. Comme nous l'avons dit au chapitre 3 (paragraphe 3.4.4), ces interfaces sont indépendantes de la ressource processeur, et peuvent fonctionner simultanément avec des tâches purement CPU. Nous exploitons cette particularité pour activer parallèlement aux transferts, un processus d'analyse des informations de contrôle *TimeStamp*. De cette manière, la plus rapide des deux opérations (transfert ou analyse *TimeStamp*) verra son temps d'execution masqué par la plus longue.

6.5 Analyse de l'information *TimeStamp*

L'information *TimeStamp* (12 bits) stockée conjointement avec les données dans la mémoire des BUSIF, renseigne sur le bon fonctionnement du dispositif de lecture (voir

chapitre 5, paragraphe 5.3.2).
Toute ces informations (environ 1/4 des données), prévue à l'origine pour les phases de mise au point, se sont avérées indispensables pour les périodes de prises de données standards. Une gestion "classique" de ces informations (transfert et enregistrement) imposait un allongement excessif du volume de données et du temps de transfert, incompatible avec le cahier des charges initial.

J'ai donc mis en place une condensation en-ligne de ces informations de contrôle, destinée à valider les blocs de 64 bits. Ce processus sera décrit au chapitre suivant. Il conduit à une réduction du volume *TimeStamp* d'un facteur 12 (on passe de 350 Ko/burst à 29 Ko/burst). Il est réalisé par les 6 processus 26 $\rightarrow$ 31 implantés sur les 6 cartes BUSIF (voir figure 6.1). Chacun d'entre eux utilise comme source les trois mémoires *TimeStampA*, *TimeStampB* et *TimeStampC* pour produire un bloc unique, reconstruit, qui renferme l'information réduite: bloc bon ou bloc mauvais (voir figure 6.4). Ces tâches n'affectent nullement les données et ne pratiquent aucun échange sur canal: il s'agit de processus purement CPU qui s'amorcent en même temps que la reconstrution des données, et continuent de s'exécuter parallélement au transfert des données. Sur chaque BUSIF, le temps d'éxécution du processus de condensation peut prendre de 5.5 à 7 s suivant la qualité des systèmes de lecture.
Conjointement à cette analyse en-ligne, des séquences d'instructions, implantées sur les transputers BUSIF, échantillonent les numéros de colonnes des CDP, avant et après la condensation. Ceci nous permet de construire des histogrammes, qui nous informent en-ligne sur l'état des synchronisations entre CDP et sur le taux de validation.

Lors des analyes hors-ligne on pourra utiliser les bits de validation (TimeStamp réduit) ainsi construit pour éliminer des données de 64 bits.
La démarche et les critères de validation qui conduisent au marquage d'un bloc de 64 bits font l'objet du chapitre 6.

6.6 Le transfert des données

Toute les descriptions de cette section, supposent que le réseau de transputers est dans sa phase de transfert de données.

Le rôle de ce réseau de transputers est avant tout de véhiculer les données, depuis les mémoires des cartes BUSIF, jusqu'aux cartes mémoires finales B016ACQ et B016SPY. Les BUSIF jouent le rôle de source ou serveur pour les cartes B016. La voie de communication entre BUSIF et B016 est matérialisée par un lien transputer, implanté sur fibres optiques. Trois de ces lignes permettent à chaque B016 de dialoguer directement avec 3 BUSIF. Il est cependant nécessaire d'interconnecter les cartes B016ACQ et B016SPY pour leur permettrent d'échanger leurs données. C'est le rôle de la B016MPX qui sert

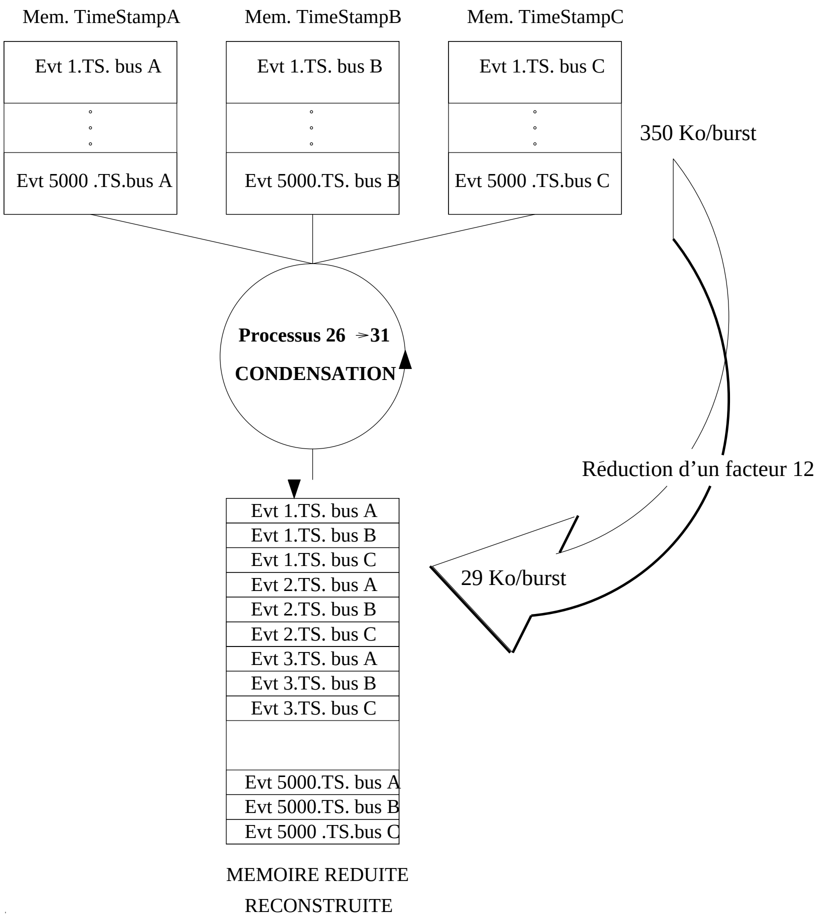


FIG. 6.4 - *Les processus* 26 → 31*, chargés de la réduction des informations de contrôle, accèdent aux trois mémoires MemStampA, B et C, pour produire un bloc réduit, reconstruit diminué d'un facteur 12. En final, ce bloc contient des informations binaires qui signent la qualité de chaque donnée de 64 bits. Ce traitement est fait parallélement à l'émission des données vers les mémoires finales. Le temps d'execution de ce processus peut prendre de 5.5 à 7 s suivant la qualité des systèmes de lecture.*

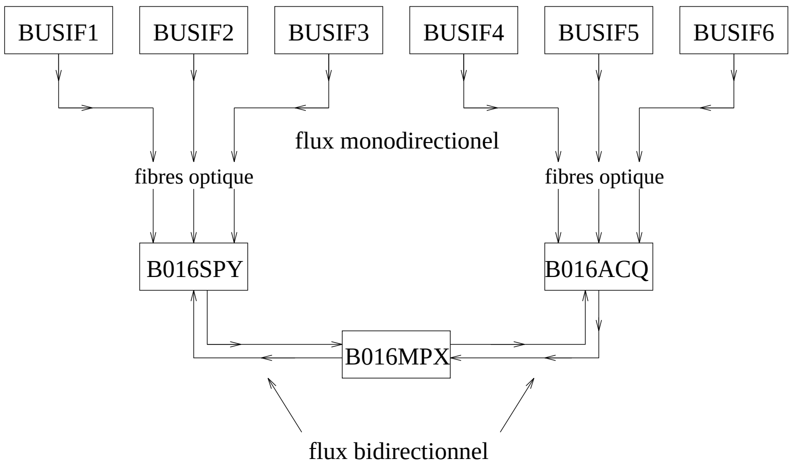


FIG. 6.5 - *Cette figure montre les cartes transputers impliquées dans le transfert des données. Chaque mémoire réceptrice B016ACQ ou B016SPY peut accéder à l'ensemble des 6 cartes BUSIF, via les liens optiques et la carte B016MPX.*

d'intermédiaire. Comme nous le montre la figure 6.5, Cette connectique permet aux cartes mémoires réceptrices B016 de recevoir les données de toutes les BUSIF.

Cette partie du réseau global présente les avantages suivants : toutes les BUSIF, du point de vue du transfert des données possèdent la même connectique : un lien optique émetteur relié à une B016. Cet atout permet d'implanter un code logiciel chargé du transfert, strictement identique pour les six processeurs de ces modules. D'autre part, la figure 6.5 nous montre aussi que les deux cartes réceptrices B016, apparaisent comme interchangeables dans le réseau : la gestion du transfert est donc la même pour les deux processeurs des B016.

Au niveau des transputers BUSIF, l'émission se fait par un processus unique P_i (processus 26 $\rightarrow$ 31 sur la figure 6.1), qui fait appel à l'interface lien 0. Une fois activée, cette interface accède à la mémoire de données[1] en mode DMA, pour transférer son contenu vers les B016, via la liaison optique. Pour la rception, chaque carte B016 recevant trois liens optiques de BUSIF, il apparait "naturel" d'associer un processus de réception R_i par voie d'entrée optique (processus 10 $\rightarrow$ 15 sur la figure 6.1). De cette manière, on établit 6 lignes de dialogue entre BUSIF's et B016 :

$$P_i \rightarrow R_i$$

Durant la phase de transfert, chaque donnée reçue par l'un de ces six processus R_i devra être réémisse vers l'autre B016 afin d'obtenir les même contenus pour les mémoires *ACQ* et *SPY*. La liaison entre ces deux cartes, faite par l'intermédiaire de *B016MPX*,

1. mémoire reconstruite par la fonction EventBuilding().

supporte donc un flux bidirectionnel qui permet cet échange d'information (cf fig. 6.5). Sur chaque B016, ACQ et SPY, il est donc nécessaire de dédier un 4ème processus, à la voie d'entrée de la liaison B016-B016MPX : on les notera R_{13} et R_{46} pour indiquer qu'ils reçoivent respectivement les données en provenance des $BUSIF1, 2, 3$ et $BUSIF4, 5, 6$. Enfin, la B016MPX doit gérer l'exécution de 4 processus, chacun associé à un canal d'entrée ou de sortie. Notés $EX1, EX2, RX1, RX2$, ils sont chargés d'émettre ou de recevoir les données, afin d'établir la liaison entre les deux autres B016.

La figure 6.6 illustre l'agencement de ces 18 processus. Sur ce schéma, les zones mémoires sont repésentées en grisé. A tout Ri est associé un bloc mémoire destiné à recevoir l'intégralité des données provenant de la $BUSIFi$.
Cette vue d'ensemble permet de constater, qu'en réalité, deux voies de transmitions fonctionnent sur le même principe, mais en sens inverse :

$$BUSIF_{1,2,3} \to B016SPY \to B016ACQ$$
$$B016SPY \leftarrow B016ACQ \leftarrow BUSIF_{4,5,6}$$

6.6.1 La parallélisation et le fractionnement des transferts

Nous avons ici à gérer le transfert des six blocs mémoires reconstruits sur les 6 BUSIF. Il ont comme particularité de décrire chacun $\frac{1}{6}$ du détecteur de multiplicité. Pour autant, le transfert d'un bloc est complètement décorrélé de celui des 5 autres : l'ordre chronologique de l'assemblage des blocs, au sein d'une B016, n'a aucune importance. Cette propriété appelle à une gestion la plus parallèle possible de ces six déplacements de mémoire. Cette parallélisation est immédiate pour les 4 processus récepteurs des B016. En effet, ces processus utilisent comme seules ressources les 4 interfaces liens du transputer. Or nous avons vu que ces interfaces modules possèdent la propriété de pouvoir travailler dans un parallélisme quasi parfait indépendemment de la ressource CPU. Il est donc dans notre intérêt de demander une exécution simultanée des ces tâches indépendantes. Cependant, le parallélisme n'étant pas parfait, le temps d'execution de 4 tâches, effectuant des transferts simultanément, s'écarte d'un facteur 1.25 [38] de la valeur théoriquement attendue.

Si l'on détaille maintenant le transfert complet d'un des six blocs de données, on s'aperçoit qu'il est lui même morcelé en 4 transferts successifs :

$$P_i \to R_i \to RX1 \to EX1 \to R13$$

Il s'agit là d'une structure en *pipe-line* qui demande à être utilisée de façon optimum. Utilisé pour propager l'information, elle doit être la plus fluide possible pour éviter les

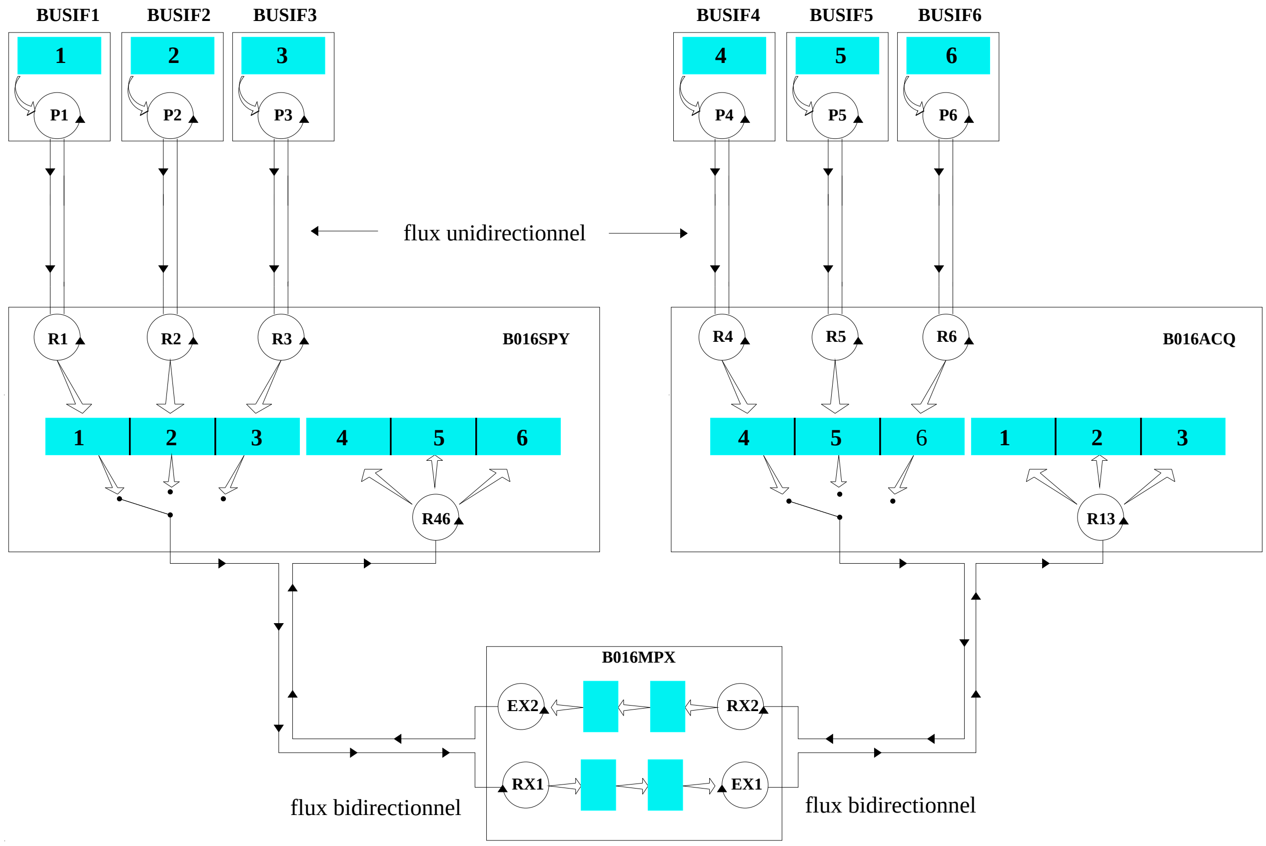


FIG. 6.6 - *Ce schéma représente le réseau des 18 processus impliqués dans le transfert des données. Les zones grisées représentent les blocs mémoires tranférés, tandis que les processus sont représentés par des cercles. En fin de déversement des ions sur la cible, la mémoire totale est répartie en six blocs physiquement séparés (les 6 cartes BUSIF). L'ensemble des processus de ce réseau a pour objectif de réunir ces mémoires en une zone unique, dupliquée sur les deux cartes B016ACQ et B016SPY. Pour cela, deux branches de processus, décorélées, travaillent en parallèle :* $BUSIF_{1,2,3} \rightarrow B016SPY \rightarrow B016ACQ$ *et* $BUSIF_{4,5,6} \rightarrow B016ACQ \rightarrow B016SPY$

points d'accumulation. Pour cela, à chaque instant, tous les éléments de sa chaîne doivent être actifs : ceci implique un fractionnement des blocs de données à transférer. Nous allons voir cependant qu'il existe une taille optimum des paquets, qui limite les points d'accumulation dans ce chaînage.
Posons simplement pour l'instant des blocs mémoires BUSIF initiaux de taille Q, fractionnés en N paquets de taille $Q_i = \frac{Q}{N}$. Le transfert total, pour une branche optique, se résumera donc à N mécanismes DMA émetteurs, activés par l'interface lien 0 de la BUSIF, tous synchronisés avec N mécanismes DMA récepteurs sur la B016. Pour la propagation au travers de la ligne $R_i \rightarrow RX1 \rightarrow EX1 \rightarrow R13$, $3 \times N$ processus DMA seront activés de part et d'autre puisqu'il s'agit du goulet d'étranglement des trois lignes optiques.

Le simple fait que les réceptions soient gérées par les interfaces liens du transputer, entraîne directement le placement des informations reçues en mémoire B016. D'autre part, tout paquet réceptioné est aussitôt injecté dans le *pipe-line* par les processus R_i, pour être redirigé sur la seconde mémoire finale, via la B016MPX.

A ce titre, deux points importants sont à signaler :

- lorqu'un processus R_i doit renvoyer un paquet vers la B016MPX, il accède à une ressoure unique (le canal de sortie), partagé par les trois processus R1,R2 et R3. Il est donc impératif, sous peine de bloquage, de ne pas accéder à cette ressource si elle est occupée. L'accès à ce canal commun est ici géré par un sémaphore. Il s'agit d'un drapeau, levé par le processus utilisateur de la ressource signifiant l'accès interdit à aux autre processus (voir figure 6.7).
- lorsqu'il pénètre dans le *pipe-line*, quel que soit sa provenance, le paquet utilise le même chemin. Il est donc indispensable de l'étiqueter pour qu'il puisse être placé dans la bonne zone mémoire par le processus récepteur final $R13$ ou $R46$. Chaque paquet possède donc un entête, qui spécifie son rattachement à une B$USIF_i$ et qui renferme sa taille en octet. Ce header est invariablement codé sur 32 bits. Sur une ligne optique, dédiée à une BUSIF, il n'est pas nécessaire de procéder à ces étiquetages.

Les quatres processus de la B016MPX, qui contrôlent les échanges de paquets traitent de façon identique toutes les informations reçues : réception d'un identificateur de paquet suivi de la réception du paquet. Là encore, la chronologie des réceptions n'est pas importante : la périodicité des réceptions des paquets d'une même source ($BUSIF_i$) n'est pas figée. En effet, si un dispositif idéal[2] tend à favoriser des réceptions cycliques, un retard sur une ligne $BUSIF_i \rightarrow B016$ augmentera la fréquence des transmissions sur les autres voies parallèles.

2. même débit sur toute les lignes identiques optiques

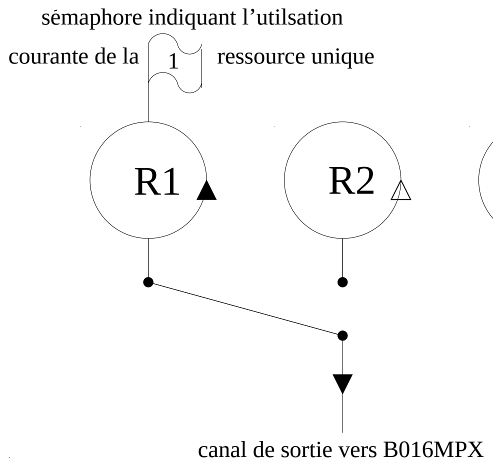


FIG. 6.7 - *Utilisé pour interdire les accès multiples à la ressource unique de sortie, le sémaphore joue ici le rôle de simple vérrou. Les processus désireux d'accéder au canal de sortie sont placés en file d'attente, puis désactivés jusqu'à la libération de cette ressource. Ceci évite des confilts bloquants.*

6.6.2 La gestion du transfert par paquets

Nous avons entrevu que le fractionnement des données en petits paquets permet d'utiliser au mieux la portion du transfert en *pipe-line*. Après amorçage des transferts il implique le travail simultané de tous les noeuds de la ligne. La taille d'un paquet va donc influer sur le temps de propagation entre deux processus et sur le temps d'occupation de ce même processus.
Ce découpage des transferts permet aussi de gagner du temps en cas d'erreur bloquante. Considérons la ligne de transfert $BUSIF_{1,2,3} \rightarrow B016SPY \rightarrow B016ACQ$, et supposons les blocs mémoires BUSIF initiaux de taille Q. Fractionnés en N paquets de taille $Q_i = \frac{Q}{N}$, ils vont être injectés dans les lignes parallèles. Pour cela, les processus P1, P2 et P3 vont itérer N émissions de paquets sur les liens 0. Pour recevoir ces paquets émis, sur la B016SPY, chaque processus de réception R1,R2 et R3 procède simultanément à N réceptions. Un échange de paquet, tributaire du protocole d'acquittement INMOS, se verra bloqué si des données sont altérées par la ligne optique. Pour de tels événements, on procède au rattrapage de l'erreur en demandant un nouveau transfert du paquet incriminé. C'est là le deuxième atout du transfert par paquet : en cas de reprise sur erreur, le temps perdu correspond au temps de réception d'un paquet (proportionnel à $\frac{1}{N}$). En effet, avec un protocole bloquant, l'erreur sur un transfert sera isolée si le temps maximum accordé à la communication est écoulé sans succès. Pour appréhender ces phénomèmes, les temps de transfert des paquets sont systématiquement mesurés au niveau des réceptions B016 par des *horloges* d'une résolution de 64 μs.

Comme nous le montre la figure 6.8, nous avons construit une couche supplémentaire dans les communications, qui vient se superposer à la couche de bas niveau (protocole décrit au chapitre 3, paragraphe 3.5.3). La couche 1, constituée par les communications

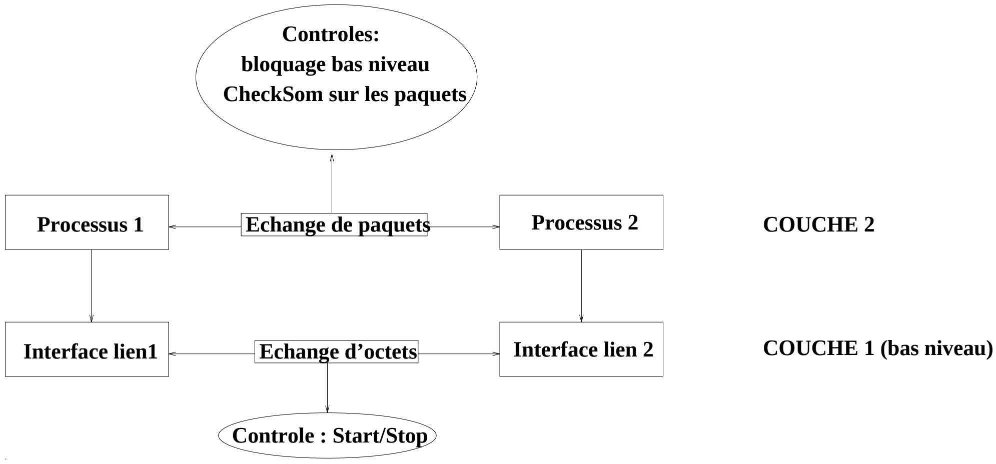


FIG. 6.8 - *Le transferts par paquet constitue une deuxième couche, supperposé à la couche de bas niveau.*

entre les interfaces liens, permet l'échange des octets. A ce niveau le contrôle des transferts est fait par un processus hardware de type *Start/Stop*. La couche 2, vient s'appuyer sur la couche bas niveau pour gèrer les échanges de données par paquets, entre processus. Dans ce cas, le contrôle des échanges inclue celui du niveau bas, mais est complété par l'implantation d'un processus de type *CheckSom* (décrit au paragraphe 6.6.3). La communication entre les deux couches se fait par des fonctions C parallèles qui sollicitent les interfaces liens.

Un fractionnement, trop important, bien qu'avantageux en cas d'erreur bloquante, entraîne cependant un accroissement du temps total de transfert. Ce phénomène est dû au poids croissant des itérations. En effet, à chaque nouveau paquet attendu, les processus récepteurs procèdent à une initialisation de leur interface lien. Cet action coûteuse en temps s'accompagne d'une intervention de l'ordonanceur de tâches qui désactive le processus. Cette dernière opération, commise à chaque paquet transmis impose un retard croissant avec le nombre de paquets.
Le temps total du transfert des données va donc varier en fonction de tous ces paramètres. Dans la section suivante, nous décrivons de façon simple l'allure de cette variation.

6.6.3 Contrôle de l'intégrité des données

Afin de s'assurer que pendant les transferts de paquets, les lignes de communication n'altèrent pas les données, nous avons mis en place une procédure dite de *ChecSom*. Il s'agit d'une opération simple, menée de paire avec chaque transfert de paquet, qui consiste à :

- sommer, avant le transfert, tous les octets qui forment un paquet,

- transférer la somme au processus cible,
- transférer le paquet,
- sommer, après réception, tous les octets qui forment le paquet reçu,
- réitérer le transfert si les sommes ne correspondent pas.

Il s'agit là d'une des plus simples opérations de vérification d'intégrité [36]. Cette détection d'erreurs par *ChecSom* donne lieu à une reprise moyenne de un paquet chaque semaine ($\simeq$ 50 Gbytes transférés/jour).

6.6.4 Temps de transfert total

Nous nous attachons dans un premier temps, à ne considérer qu'une branche de transfert :

$$BUSIF_{1,2,3} \rightarrow B016SPY \rightarrow B016ACQ$$

De plus, dans tout le développement qui va suivre, nous ne considérons pas le fait qu'il existe, parallèlement aux transferts, une analyse des *TimeStamp*.

Sur cette ligne, le transfert des données sera condidéré comme achevé, lorsque les trois blocs mémoires des BUSIF1, 2, 3 auront été dupliqués sur les *B016SPY* et *B016ACQ*. Afin d'évaluer le temps nécessaire à cette opération nous devons extraire du transfert global les différents trajets empruntés successivement par les paquets de données. Ces trajets sont les chemins qui joignent deux noeuds[3] du réseau de processus.
D'après la figure 6.6, un paquet i issu d'une source $BUSIF_i$ suit 4 trajets successifs :

- $P_i \rightarrow R_i = trajet\ 1$
- $R_i \rightarrow RX1 = trajet\ 2$
- $RX1 \rightarrow EX1 = trajet\ 3$
- $EX1 \rightarrow R13 = trajet\ 4$

Pour un paquet donné, le temps de passage d'un noeud au suivant, est fonction d'une part, du temps de transfert proprement dit (temps d'occupation de la ligne) et d'autre part, du temps nécessaire à l'amorçage de cette communication (initialisation de l'interface lien et délai d'ordonancement du processus).

Pour chaque paquet transmit, issu de la source k $BUSIF_k$, nous noterons par :

- T_{ik} le temps de transfert pur, pour le trajet i. Si toute les lignes optiques sont identiques, on a l'égalité $T_{ik} = T_{il} = T_i$ pour tout $k \neq l$. On conservera cette hypothèse.

3. Un noeud équivaut ici à un processus

- δt, le temps necessaire à la gestion d'un paquet.

Supposons dans un premier temps que tous les dialogues $P_i - R_i$, $RX_i - EX_i$ et $EX_i - R_{ij}$ peuvent être effectués dans une simultanéité parfaite, dans la mesure où il y a matière à dialoguer. Il ne reste plus qu'a considérer le fait que les échanges $R_i - RX_i$ doivent s'exécuter de façon séquentielle, du fait de l'unique ressource commune dont l'accès est géré par sémaphore.
Ces suppositions faites, pour la branche $BUSIF_{1,2,3} \rightarrow B016SPY \rightarrow B016ACQ$, nous pouvons construire le chronogramme des transferts de la figure 6.9.

Si le bloc initial, d'une taille globale Q demande un temps de transfert de T_{i0} pour parcourir le trajet i alors, d'une manière générale on peut dire qu'un paquet de taille $\frac{Q}{N}$ demandera un temps de transfert $T_i = \frac{T_{i0}}{N}$.

D'après le chronogramme 6.9, nous pouvons établir la relation :

$$T_{tot} = T_1 + T_2 + 2 \times \delta t + 3 \times N \times (T_3 + T_4 + \delta t) \tag{6.3}$$

soit

$$T_{tot} = \frac{T_{10}}{N} + \frac{T_{20}}{N} + 2 \times \delta t + 3 \times N \times (\frac{T_{30}}{N} + \frac{T_{40}}{N} + \delta t) \tag{6.4}$$

Le temps total nécessaire aux transferts doit donc suivre une loi du type :

$$T_{tot} = \frac{a}{N} + b \times N + c \tag{6.5}$$

avec,

$$a = T_{10} + T_{20}, \quad b = 3.\delta t \quad et \quad c = 2.\delta t + 3.(T_{30} + T_{40}) \tag{6.6}$$

Il renferme une contribution asymptotique $\frac{a}{N}$ qui caractérise l'ammorçage des pipelines. Cette contribution sera d'autant plus faible que le nombre de paquets sera grand. La composante linéaire $b \times N$ traduit le fait que la gestion du transfert par paquet telle que nous l'avons décrite devient vite pénalisante avec le nombre de paquets. En effet, chaque paquet transmis, demande une série d'opérations incontournables :

- Fabrication de l'identificateur de paquet (Id),
- Initialisation des interfaces liens pour le transfert de l'Id (à chaque extrémité du lien),
- Synchronisation des transputers en vue du transfert de l'Id,
- Transfert de l'Id,

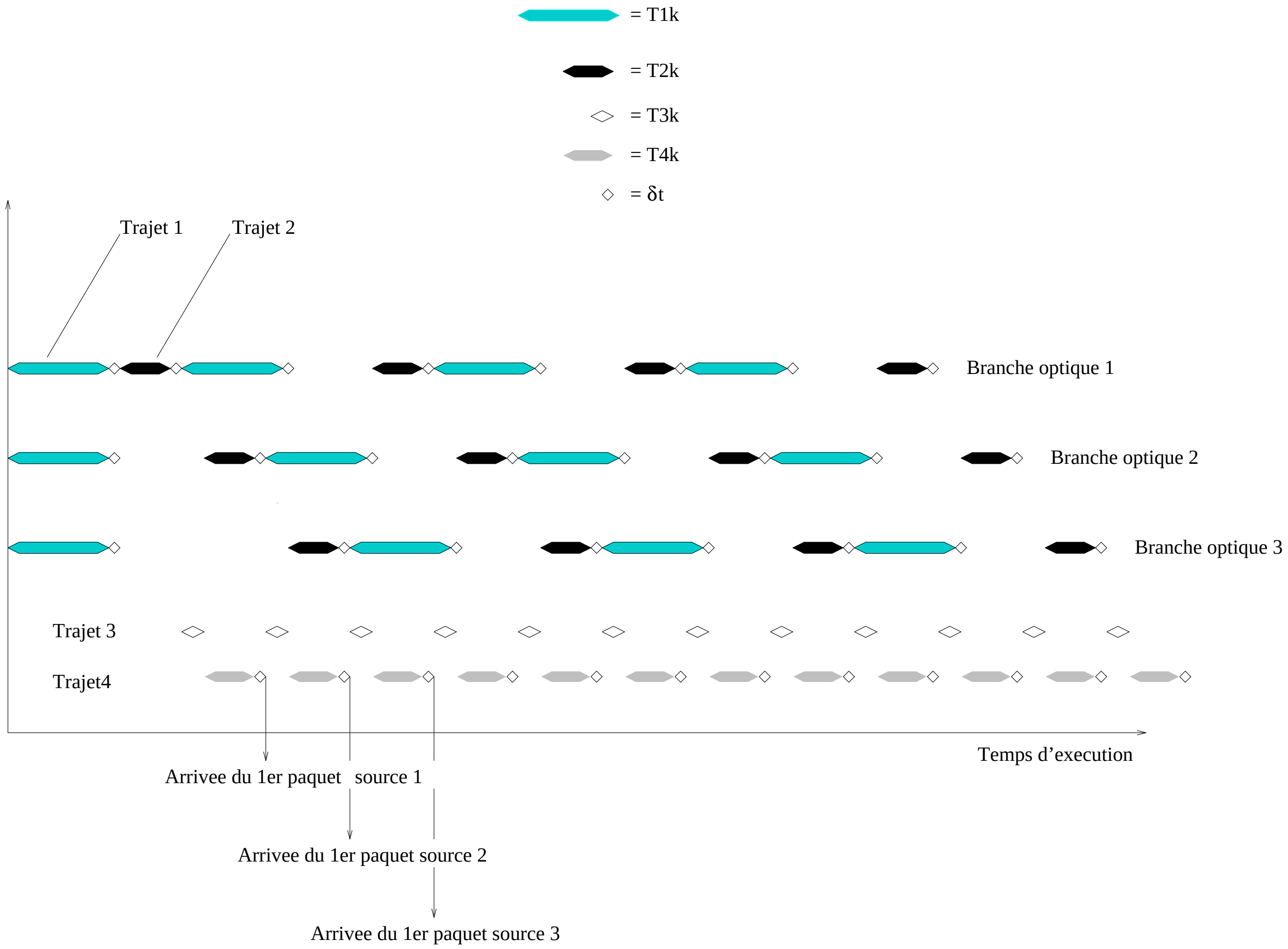


FIG. 6.9 - *Cette figure illustre la chronologie des transferts pour la branche* $BUSIF_{1,2,3} \rightarrow B016SPY \rightarrow B016ACQ$*. Il s'agit d'un exemple où les blocs initiaux ont été fractionnés en 4 paquets de tailles identiques. Les temps* T_{ik} *sont ceux nécessaires pour qu'un paquet issu d'une source* k *parcourt un trajet* i*. On constate que, hormis l'amorçage du pipe-line tous les temps de transfert sont masqués par ceux de la ligne* $RX1 \rightarrow EX1 \rightarrow R13$*.*

- Analyse de l'Id en réception,
- Procédure de vérification par *CheckSom*
- Temps d'ordonancement à chaque nouveau transfert de paquet.

Enfin, l'expression T_{tot} fait ressortir une terme constant c uniquement corrélé aux trajets T_{30} et T_{40}.

Pour une taille fixée des blocs de données initiaux, il doit exister un fractionement optimum N_{opt} qui minimise la fonction (6.5). Pour cette valeur N_{opt}, la taille optimum d'un paquet sera notée Q_{opt}.

6.6.5 Evaluation de la taille optimale Q_{opt} d'un paquet pour la configuration de novembre 1995

La taille optimale d'un paquet est celle qui minimisera la temps total donnée par l'expression (6.5) : $\frac{dT_{tot}}{dN} = 0$.

Or,

$$\frac{dT_{tot}}{dN} = -\frac{a}{N^2} + b \tag{6.7}$$

donc, $\frac{dT_{tot}}{dN} = 0$ nous conduit à :

$$N_{opt} = \sqrt{\frac{a}{b}} = \sqrt{\frac{T_{10} + T_{20}}{3.\delta t}} \tag{6.8}$$

Si Q est la taille totale d'un bloc à transférer, la taille optimale d'un paquet est donnée par :

$$Q_{opt} = \frac{Q}{N_{opt}} \tag{6.9}$$

De plus, si les débits sur les trajet 1 et 2 sont rescpectivement ϕ_1 et ϕ_2, nous pouvons réécrire (6.8) et (6.9) sous la forme finale :

$$N_{opt} = \sqrt{\frac{\phi_1 + \phi_2}{3.\delta t.\phi_1.\phi_2}} \times \sqrt{Q} \qquad et \qquad Q_{opt} = \sqrt{\frac{3.\delta t.\phi_1.\phi_2}{\phi_1 + \phi_2}} \times \sqrt{Q} \tag{6.10}$$

Il apparaît donc clairement, que pour une valeur fixée de δt, le fractionnement optimal varie en $\sqrt{Q}$. Cette propiété sera à prendre en compte pour la période de novembre 1995 où nous allons acquérir plus de 3000 événements par déversement.

Application à la configuration de novembre 1995

Durant cette période, nous avons fixé une limite supérieure de 3 000 événements à transférer.

En nous référant à l'équation (6.1) (paragraphe 6.4.1), ceci nous donne la taille Q de la zone mémoire reconstruite sur chaque BUSIF :

$$Q = 3000 \times 288 = 843\ Koctets \tag{6.11}$$

Comme nous l'avons dit au paragraphe 6.6, les deux branches

$$BUSIF_{1,2,3} \rightarrow B016SPY \rightarrow B016ACQ$$

$$B016SPY \leftarrow B016ACQ \leftarrow BUSIF_{4,5,6}$$

vont travailler simultanément. Ceci implique que pour chacune des trois B016, les 4 liens vont être solicités en même temps, ce qui nous oblige à apporter une correction de 25% sur les débits de tout les liens [38] :

$$T_{ik}^{4\ liens} = 1.25 \times T_{ik}^{1\ liens} \tag{6.12}$$

En considérant le flux sur une fibre $\phi_1 = 488\ Koctets/s$, nous en déduisons le temps de transfert pur (sur fibre optique) :

$$T_{10} = \frac{843}{488} \times 1.25 = 2.15\ s \tag{6.13}$$

De plus, les flux de transfert mesurés sur les liens B016ACQ-B016MPX ou B016SPY-B016MPX montre que l'on a un rapport de 1.9 entre les débits sur liens optiques et les débits sur liens normaux :

$$\phi_2 = 1.9 \times \phi_1 = 927\ Koctets/s \tag{6.14}$$

On en déduit donc :

$$T_{20} = T_{10}/1.9 \quad et \quad T_{40} = T_{10}/1.9 \tag{6.15}$$

En ce qui concerne le trajet 3, le temps T_{30} est celui d'une copie standard de buffer à buffer, et prend (comme nous l'avons vu pour les BUSIF) 93 μs/événement. Ceci est valable dans le cas où la ressource CPU est entièrement disponible. Dans notre cas, au sein du transputer B016MPX, deux transferts se font simultanément en sens opposés. Ceci double le temps nécessaire à la copie. Ainsi pour 3 000 événements nous avons :

$$T_{30} = 2 \times 93\ \mu s \times 3000 = 0.56\ s \tag{6.16}$$

Pour la gestion de chaque paquet, nous estimons, d'après la référence [38], que les délais d'ordonnancement et les durées d'initialsations des interfaces liens, portent la valeur de δt à 4×10^{-3} s.

Pour finir, afin d'estimer le temps total nécessaire à remonter les données, il faut rajouter au temps de transfert T_{tot} de l'expression (6.4), le temps de reconstruction des événement. Ce temps vaut 0.56 s pour 3000 événements.

Ces valeurs numériques nous permettent, à partir des équations (6.8) et (6.10) de calculer le nombre optimum de paquets et donc leur taille optimum pour minimiser le temps de transfert total (reconstruction + transfert pour 3 000 événements) :

$$N_{opt} \simeq 16 \ paquets \quad soit \quad Q_{opt} \simeq 56 \ Ko/paquet \tag{6.17}$$

Pour cette même configuration de novembre 1995, nous avons mesuré les temps de transfert en fonction du fractionnement N. La figure 6.10 nous montre la superposition de ces résultats (carrés) avec les valeurs calculées (pointillés). Les variations identiques de ces deux distributions nous confirment que le temps total de transfert est bien décrit par une loi du type :

$$T_{tot} = \frac{a}{N} + b \times N + c \tag{6.18}$$

D'autre part, on remarque que les désaccords entre les deux variations sont relatifs aux temps de transfert et non au fractionnement. Pour 3000 événements, on à donc bien un fractionnement optimal (expérimental) de l'ordre de 15 à 20 paquets, en accord avec la valeur calulée.

6.7 Chronogramme final des transferts

Au niveau d'un transputer BUSIF, pendant le déroulement de l'application, nous avons deux ressources solicitées par le microprocesseur : le CPU et l'interface lien 0. Nous pouvons, d'après les paragraphes précédent, et en considérant l'analyse des *TimeStamp*, construire le chronogramme simplifié 6.11. Au temp de "reconstruction + transfert" précédemment évalué, il faut rajouter des retards imposés par : le recouvrement d'une partie des transferts avec des activités CPU, la gestion des transferts des *TimesStamp* et des histogrammes. Toutes ces contributions décalent d'environ 2 s la courbe 6.10. Le temps minimum nécessaire à gérer toutes ces opérations est donc de 9.5 s.

Pour les périodes de prise de données avec un maximum de 3000 événements, nous avons donc choisie de fractionner les blocs de données initiaux en 16 paquets de tailles identiques, soit environ 53 Ko/paquet. Rappelons que ce fractionnement est effectué au sein de chaque carte BUSIF.

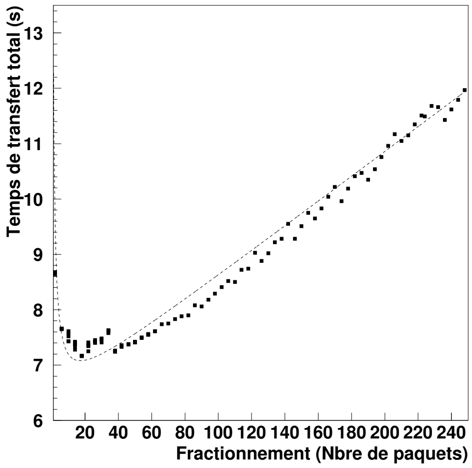


FIG. 6.10 - *La courbe théorique du temps de transfert total présente bien la même allure que les points expérimentaux (carrés). Le désaccord marqué pour les petites valeurs de N, ne déplace pas la région du temps minimum. Dans cette région le temps de transfert est de l'ordre de 7.5 s, pour un fractionement de 15 à 40 paquets. Ces mesures ne tiennent pas compte de léxecution concurrente de tâches CPU, dédiées à l'analyse des TimeStamp.*

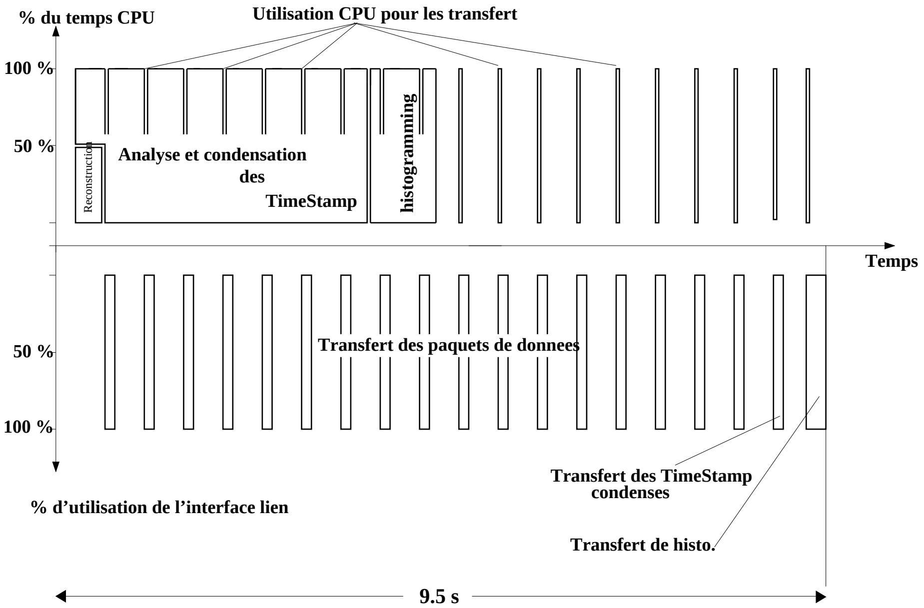


FIG. 6.11 - *Ce schéma illustre la "superposition" des executions dans le temps. En fin de déversement, on active la reconstruction des données, et la condensation des TimeStamp. Pendant toute la reconstruction, chaque tâche utilise 50 % du temps CPU. Ensuite pendant que la condensation continue, on active le transfert des données. Cette dernière tâche purement "lien", ne consomme du temps CPU que pour la gestion d'un paquet (δ t). Lorsque la condensation des TimeStamp est terminé, le transfert des données continue. Après qu'il ait pris fin, on transfert les TimeStamp condensés et les informations relatives aux histogrammes.*

6.8 Conclusion

Dans ce chapitre nous avons montré qu'un réseau de 35 processus forment le corps du logiciel multitâches mis en place pour reconstruire, valider et transférer les données du détecteur de multiplicité.
La reconstruction, pratiquée au niveau des processeurs des BUSIF, permet de diminuer les temps d'analyse en-ligne, pratiqués sur la cartes B016SPY.
La phase de transfert des données est supportée par un protocole d'échanges de paquets qui demande à être géré de manière optimum. La courbe 6.10 nous montre que le fractionnement idéal en 16 paquets de 53 Ko. L'activités concurrente des différente tĉhes, nous conduit à un temps de transfert total de 9.5 s pour toute les données du détecteur.
Dans le chapitre suivant, nous allons détailler les mécanismes du processus de condensation des *TimeStamp*, qui à pour objectif de valider les données.

Chapitre 7

LA VALIDATION DES DONNEES MULTIPLICITE

7.1 Introduction

7.1.1 Que signifie valider les données?

Après chaque fin de cycle SPS, le système de transfert (voir chapitre 4 et 5), basé sur les transputers, vide les mémoires de données des cartes BUSIF vers le système d'acquisition. Ces données sont sensées représenter l'état du détecteur à l'instant de la collision. Pour de multiples raisons, des portions de détecteur ou leurs chaînes de lecture associées, peuvent présenter des défaillances ponctuelles, voire généralisées, qui affectent ainsi l'intégrité des informations.
Valider une donnée présente en mémoire BUSIF, consiste donc à "affirmer" que l'information qu'elle contient représente fidèlement l'état du détecteur à l'instant de l'échantillonage, et qu'elle est de plus, corrélée en temps avec le déclenchement considéré: il s'agit d'un travail de réjection.

Les facteurs de réjections peuvent concerner une ou plusieur bandes de silicium isolée et sont généralement corrélés à:

- des bandes trops bruyantes (problèmes d'alimentation, de qualité du matériau),
- des bandes mortes donnant toujours le même signal (problème de bonding[1], d'alimentation),
- des défauts dans les circuits de lecture ou d'échantillonage, qui digitalisent l'information (parasites, réflection de signaux, pannes...).

1. Il s'agit des micro-connections qui permettent d'accéder aux entrées/sorties des circuits de lecture

S'ils n'ont pas été isolés lors de tests en laboratoire, les problèmes qui découlent des deux premiers points demandent une analyse fine de tous les canaux du détecteur. Ce type de scrutation sur les données, coûteux en temps CPU, ne peut être fait en-ligne (13 824 canaux/événement).
En revanche, les défauts liés aux dispositifs de lecture sont identifiables par un examen comportemental de ceux-ci. Chacun d'entre eux étant chargé de lire 64 canaux (voir chapitre 4), la recherche d'éventuelles anomalies, nécessite l'étude systématique des 216 chaînes de lecture, et peut être menée en-ligne. On ne recherche donc pas les défaut de lecture par l'analyse des données receuillies, mais par l'étude des mots de contrôle *TimeStamp* qui renseignent sur le déroulement de la lecture et sur la synchronisation des chaînes de lecture. Ces mots de contrôle de 16 bits sont produits de paires avec les données de 64 bits. Une réjection basée sur l'analyse de ces mots vise donc à valider des groupes de 64 bits. Nous rappelons, qu'en terme de surface de détection, 64 bits représentent un demi secteur pour la couronne interne et un secteur complet pour la couronne externe.

Parce que l'information de contrôle des lectures n'est pas stockée sur bande, il faut exploiter ces renseignements là où ils sont disponibles : au niveau des processeurs source des cartes BUSIF, sans quoi *l'historique* des lectures du détecteur est perdu : le filtrage doit donc être activé en-ligne.

Les sections suivantes développent le traitement de ces informations *TimeStamp* qui constituent la structure de ce filtrage.

7.2 Le traitement des informations *Time-stamp*

Comme nous l'avons vu au chapitre 5 (paragraphe 5.3.2), que ces mots de 16 bits renferment 12 bits d'information relatifs au numéro du circuit adressé (*ChiId*), au numéro de colonne accédée (*NumCol*) et à l'information de *TimeOut* (cf figure 7.1). Pour un circuit de lecture donné, ce code permet de valider la lecture du registre de sortie du CDP, ainsi que la cohérence des données qu'il propose. Il est donc impératif, lors de l'analyse des données, de prendre en considération ces *TimeStamp* : ils renseignent sur le bon déroulement de la lecture du détecteur.

Transférer systématiquement tous les mots de *TimeStamp* augmente d'environ 20% la charge du transfert. Ceci nous conduirait d'une part, à un temps de transfert plus élevé et surtout acroitrait de 10% la quantité de données à enregistrer sur bandes magétiques. En effet, nous avons vu que les données du détecteur de multiplicité représentent 50% du volume total, tous détecteurs confondus.
Afin d'éviter cette surcharge, j'ai mis en place un processus de validation en-ligne des données, basé sur une analyse systématique de l'information *TimeStamp* qui permet d'isoler efficacement les divers défauts de lecture. Les résultats de ce filtrage sont binaires et se résument à un bit par donnée de 64 bits : le mot de 64 bits et bon ou mauvais.

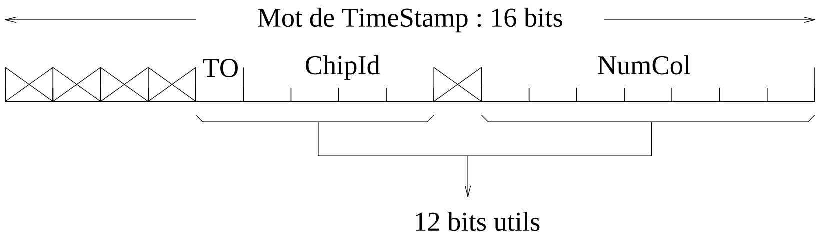


FIG. 7.1 - *Structure de l'information TimeStamp. Codée sur 11 bits, elle est incluse dans des mots de 16 bits. Chaque données de 64 bits est appariée avec son TimeStamp.*

L'élaboration de ce filtre à pu être faite à partir de données réelles, et notamment à partir des résultats obtenus lors de la prise de données de novembre 1995. Durant cette période, l'intégralité du détecteur n'étant pas présente[2], les mots de *TimeStamp* des CDP actifs ont été stockés à la place des données des CDP des couronnes externes. Ce biais a permis de mener à bien les études sur :

- le bit de *TimeOut*.
- l'identificateur de ciruit *ChipId*.
- les numéros de colonne *NumCol*.

7.2.1 Etude du bit de *TimeOut*

Le bit de *TimeOut* a été conçu, à l'origine, pour valider la lecture des données qui lui correspondent. Initialisé à zéro (signifiant un problème de lecture), il est forcé à la valeur 1 (pas de problème de lecture) lorsque les 18 cycles nécessaires à vider le contenu du registre de sortie du CDP se sont terminés correctement. L'information binaire qu'il fournit, ne nous renseigne en aucun cas sur la validité des bits de données.
La création de ce bit, par les circuits programmables des BUSIF, s'est avérée inefficace pour les prise de données de novembre 1995 et d'avril 1996. En effet, à de nombreuses reprises, des données annoncées comme issues d'une mauvaise lecture par ce bit pouvaient être récupérées par une analyse plus fine du mot de contrôle *TimeStamp*. L'étude de ce mauvais fonctionnement est prévu pour la prochaine prise de données proton en juillet 1996, et nécessite de nouvelles séquences logiques, intégrées aux circuits programmables des BUSIF.

7.2.2 Etude de l'identificateur de ciruit *ChipId*

Lorsqu'un CDP est adressé par une carte Busif, après avoir subit les 18 cycles de lecture sans anomalies bloquantes, il dépose sur les lignes de données son numéro d'identifica-

2. présence des couronnes internes soit : huits CDP actifs par bus 1, 2, 3, 4, 7, 8, 9 et 10

tion. Il s'agit du numéro logique associé au circuit[3]. Le terme "d'anomalie bloquante", caractérise un problème de lecture entraînant le bloquage du séquenceur (MemA, MemB ou MemC) dans l'un de ses 18 états. Leurs origines sont à rechercher au cas par cas et peuvent être dues à des pannes de circuits CDP, des parasites électriques (oscillations de signaux), des défauts sur la lignes de lecture (bus, accès mémoires).
Ce numéro ainsi produit, est lu par le bus de données de la BUSIF pour être ensuite intégré au mot *TimeStamp*. En cas de bloquage, la valeur attendue n'est jamais lue, et par conséquent l'information *ChipId* du *TimeStamp* n'est jamais surécrite en mémoire.

Si le 18ème cycle est acompli, la comparaison du numéro lu avec la valeur attendue permet d'une part de savoir que l'ensemble des cycles de lecture a été mené à terme et d'aure part de s'assurer qu'aucune perturbation ne détériore les signaux sur les lignes CDP-BUSIF.
En effet des perturbations sur les lignes CDP-BUSIF peuvent altérer les signaux émis par les CDP sans pour autant affecter la progression des cycles de lecture des BUSIF. Ce type de problèmes, bien que non bloquant, engendre cependant des *ChipId* différents entre l'émission et la réception : un circuit non adressé voire non existant semble répondre à la place du circuit en cours de traitement, on parle de circuit *fantôme*. Par simple comparaison avec la valeur logique attendue, ce type de problèmes est facilement identifiable.

Ces comportements "étranges" sont révélateurs d'une mauvaise communication entre la BUSIF et le CDP. Les 64 bits données, reçus de ce circuit, ont transité par la ligne perturbée : il ne faut pas les considérer.

Ceci nous conduit à établir la coupure *F1*:

F1 = rejet des 64 bits de données dont le *ChipId* n'a pas la valeur attendue.

Nous illustrons ces problèmes 7.2 par un cas pathologique rencontré en novembre 1995. Ces histogrammes représentent le nombre de réponses attendues par circuit (a), les réponses effectivement enregistrées (b) sans aucun filtrage appliqué et les circuits conservés après l'application de la coupure F1.

La différence entre les réponses obtenues sans la coupure F1, et les réponses attendues soit (a)-(b), est représentée sur la figure 7.3.

Il apparait clairement des dialogues avec des circuits inexistants (0 et 5), et des réponses excessives de la part du CDP 8. La réponse importante d'un circuit fantôme 5 est corrélée à la faible activité du circuit 7. Ce transfert peut s'expliquer par des perturbations faisant chuter le bit de poids faible de la valeur 7 à zéro :

3. de 1 à 12

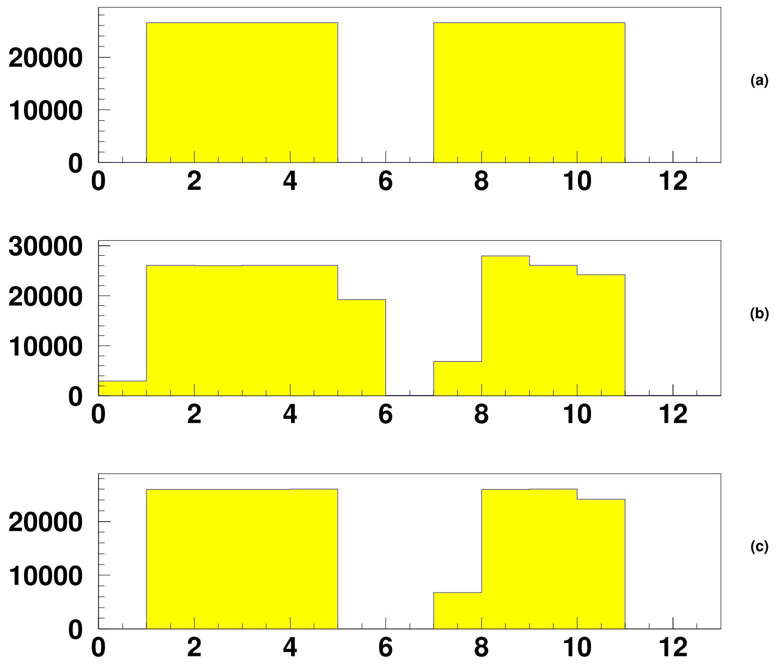


FIG. 7.2 - *Défauts rencontrés en novembre 1995 sur la busif3, bus A : les ChipId enregistrés (b) ne correspondent pas aux valeurs attendues (a). On constate l'apparition de valeurs fantômes : ChipId = 0 et 5, et une surcharge de la réponse du circuit numéro 8. La figure (c) montre les circuits conservés après la coupure F1.*

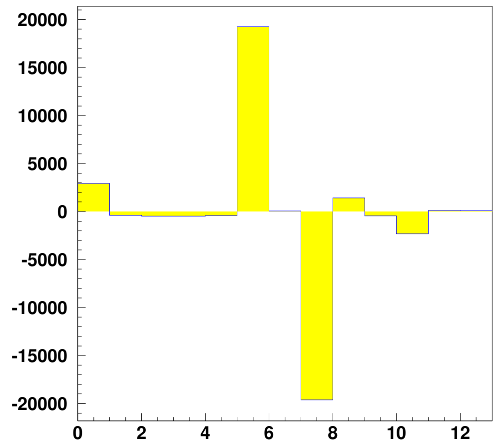


FIG. 7.3 - *La différence entre les réponses effectivement enregistrées et celles attendues montre la réponse de circuits fantômes 0 et 5.*

(dec)7 = (bin)0111 valeur émise par le CDP numéro 7.

(dec)5 = (bin)0110 valeur enregistrée par la BUSIF, après perturbation.

Cette perturbation est à imputer au CDP lui même ou au processus d'enregistrement sur la BUSIF.

Une fois la coupure appliquée, ne sont conservés que les circuits présentant un "bon" *ChipId* (cf figure 7.4).

Si une mauvaise valeur du *ChipId* est suffisante pour rejeter la donnée associée, une valeur cohérente ne l'est pas pour affirmer que l'ensemble des cycles de lecture a été mené à bien. En effet un circuit qui ne répond pas lorsqu'il est adressé (bloquage dans un des 18 cycles de lecture) ne donnera pas d'identificateur, par conséquent, son *ChipId* ne sera pas inscrit en mémoire. Or, durant la prise de données de novembre 1995, d'une bouffée de faisceau à la suivante, les mémoires des *TimeStamp* n'étaient pas réinitialisées. La surécriture de cet identificateur n'étant pas effective en cas d'anomalie bloquante, il est impossible d'affirmer si un *bon ChipId* provient du déversement courant, ou d'un déversement antérieur.

Pour les prises de données 1996, des modifications ont été apportées aux cartes BUSIF. Une remise à zéro de chaque mémoire d'événement (64 bits de données et 16 bits

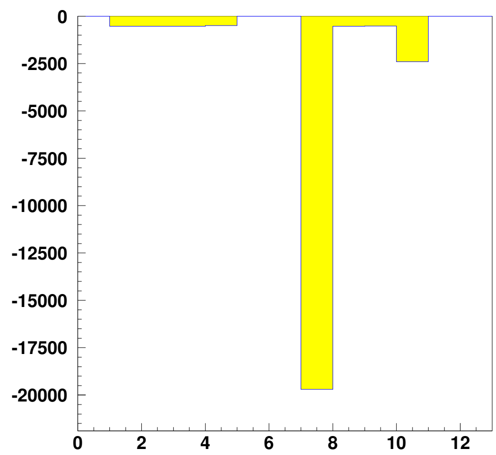


FIG. 7.4 - *Illustration de la différence entre : circuits conservés pour "bon" ChipId, et réponses attendues. Après application de la coupure F1 on constate une disparition totale de toute les réponses excessives.*

de TimeStamp) est opérée avant chaque nouveau déclenchement. Ceci devrait renforcer le rôle du *ChipId* en temps que "valideur" des lectures.

7.2.3 Etude des numéros de colonne

Rapellons que la colonne est l'unité de l'élément du buffer circulaire du CDP, sur lequel s'est arrêté le pointeur au moment du déclenchement. Elle est référencée par son numéro *NumCol* intégré au *TimeStamp*.
parmi les 18 cycles de lecture générés par les séquenceurs des BUSIF (MemA, MemB ou MemC), les deux premiers sont réservés à la lecture de ce numéro de colonne. Cette information est la première à être délivrée par les circuits CDP. Un numéro de colonne est donc créé quelle que soit l'évolution des 16 cycles de lecture à suivre.

Nous avons vu qu'un numéro de colonne est codé sur 7 bits, pour englober la plage de fonctionnement réelle : de la colonne 0 à la colonne 80 incluse. Ces bits de mémoire sont remplis par deux cycles de lecture consécutifs du bus de données de 4 bits. Il peuvent donc en cas d'anomalies décrire l'intervalle [0,127]. Une valeur abérante, superieure à 80, peut être due à un problème de lecture du bus de données : perturbation électrique ou bloquage dans l'un des deux premiers cycles de lecture à l'origine d'un numéro de colonne abérant.
Comme précédemment pour le *ChipId*, ces comportements "étranges" mettent à jour des

défauts de communication entre la BUSIF et le CDP. Les 64 bits données, reçus de ce circuit, proviennent de la même ligne perturbée : il ne faut pas les considérer.

Ceci nous conduit à établir la coupure *F2* :

F2 = rejet des données dont le numéro de colonne *NumCol* n'appartient pas à l'intervalle [0,80].

Une fois les deux coupures, *F1* et *F2* appliquées, nous disposons de données qui, proviennent de chaînes de lecture non perturbées. On peut donc supposer raisonablement que l'information présente dans les mémoires BUSIF est l'image de celle issue des colonnes des circuits de lecture CDP.

Cependant, pouvons nous affirmer que ces mots de 64 bits retenus, représentent l'état du détecteur à l'instant de la collision ? ou autrement dit, sommes nous certain qu'aucun problème n'a pu affecter la progression des pointeurs de colonnes au sein de chaque CDP ?

Il est impératif de répondre à ces questions pour chacun des 216 CDP, sinon nous serions amenés à considérer des mots de 64 bits non corrélés avec les déclenchements.

Lors de l'examen des *TimeStamp*, il faut donc voir le numéro de colonne *NumCol*, non plus comme une entité individuelle (coupure F2), mais comme faisant partie d'un ensemble dont les éléments progressent de manière synchronisée. C'est en comparant les numéros de colonnes au sein de ces ensembles et pour chaque événement, que l'on fait ressortir des défauts de synchronisation. Nous allons voir que ces comparaisons permettent d'affiner le filtrage avec deux nouvelles coupures *F3* et *F4*.

La taille des ensembles à étudier est de 12 CDP. Il s'agit du nombre de CDP pris en charge par un bus de lecture BUSIF (A, B ou C).

Un circuit CDP qui fonctionne correctement génére au cours des déversements des numéros des colonnes uniformément répartis entre les valeurs 0 et 80. En effet, pour deux déclenchements consécutifs, l'écart entre les numéros de colonnes associés est proportionnel à l'intervalle de temps séparant ces deux déclenchements (modulo 80) : il présente donc un caractère aléatoire.
Dans les analyses qui suivent, nous présentons sous forme d'histogrammes, la répartition de ces numéros de colonnes pour n événements. Les colonnes des CDP effectivement présents sont toutes placées en abscisse à la valeur $(CDP_n - 1) \times 100 + Col_n$, afin d'obtenir sur chaque histogramme le comportement d'un bus complet. La figure 7.5 illustre ce mode de représentation avec la répartion des colonnes sur 25 000 événements pour la BUSIF1 bus B. Cette représentation tient compte des coupures *F1* et *F2*.

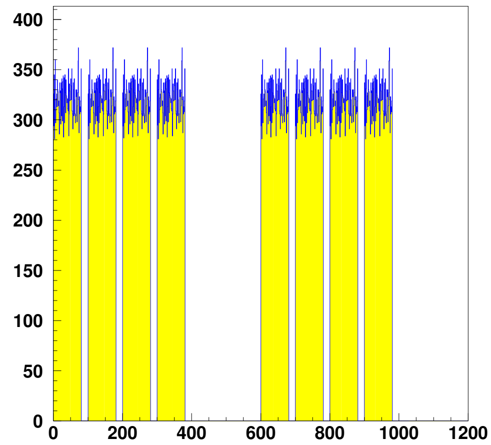


FIG. 7.5 - *Répartion des colonnes sur 25000 événements pour la BUSIF1 bus B, bande NF1292. En abscisse, de gauche à droite, sont représentés les numéros de colonnes des 8 CDP présents. Cette représentation qui tient compte des coupures F1 et F2 montre une répartition homogène des colonnes entre 0 et 80 pour tout les CDP présents.*

Pour un bus de lecture l'ensemble des 12 pointeurs de colonnes est remis à zéro (au niveau des CDP) à chaque nouvelle bouffée d'ions sur la cible. La BUSIF envoie pour cela le mot de commande :

$$CmdeRAZ. = 2_{hex} = 0010_{bin}$$

Au cours du déversement qui suit, entre deux déclenchements, la BUSIF maintient sur le bus de commande l'instruction de progression des pointeurs :

$$CmdeIncr. = A_{hex} = 1010_{bin}$$

Tous les CDP d'un bus reçoivent ce **même** mot de commande ; la progression de ces pointeurs doit donc être une progression d'ensemble.

Lorsqu'un déclenchement est signalé, la BUSIF arrête cette progression par l'envoi du mot :

$$CmdeRead. = 6_{hex} = 0110_{bin}$$

Les 12 numéros (huit pour la période de novembre 1995: de 1 à 4, et de 7 à 10) relevés à chaque nouveau déclenchement, doivent par conséquent être identiques. On ne peut tolérer une désynchronisation, ne serait-ce que d'une colonne (40 ns à la fréquence de 25 MHz). En effet, ceci conduirait à considérer des événements non corrélés avec le déclenchement.

De tels écarts par rapport au mouvement d'ensemble sont toutefois détectés et peuvent être dus à :

- un mauvais décodage du mot de commande envoyé par la BUSIF aux circuits de lectures CDP. Ceci peut provoquer une remise à zéro inopportune des pointeurs de colonnes,
- une fluctuation du signal d'horloge du CDP. Ce type d'anomalie faisant progresser le pointeur de colonnes de plusieurs unités,
- une défaillance d'un circuit CDP engendrant une colonne quelconque.

Ces points conduisent à former deux nouvelles coupures *F3* et *F4* :

F3 = rejet des données dont les colonnes sont bloquées.
F4 = rejet des données dont les colonnes n'appartiennent pas à un groupe de colonnes identiques. La taille minimum tolérée pour un groupe étant 3 CDP avec des colonnes identiques.

L'application de ces coupures permet d'isoler les problèmes de synchronisation au sein d'un bus de lecture. Elle demande une étude comparative des numéros de colonnes, au sein même de l'événement, la comparaison pouvant s'étendre aux événements antérieurs pour localiser des bloquages de pointeur.

Dans la plupart des cas, une colonne bloquée est dûe à un mauvais échantillonage du mot de commande envoyé par la BUSIF. Sur chaque coup d'horloge, les CDP échantillonent le bus de commande (4 bits). En acquisition normale, pendant le déversement des ions, l'état de ce bus est imposé par la carte BUSIF et vaut : soit $6_{hex} = 0110_{bin}$, soit $A_{hex} = 1010_{bin}$. Les 4 bits qui forment ces mots de commande étant sur des lignes séparées, une détérioration de la forme des bits de rangs 3 ou 4 entraîne une mauvaise interprétation de la commande par le CDP (cf fig. 7.6) :

$$0110_{bin} \text{ ou } 1010_{bin} \text{ seront interprétées comme } 0010_{bin}$$

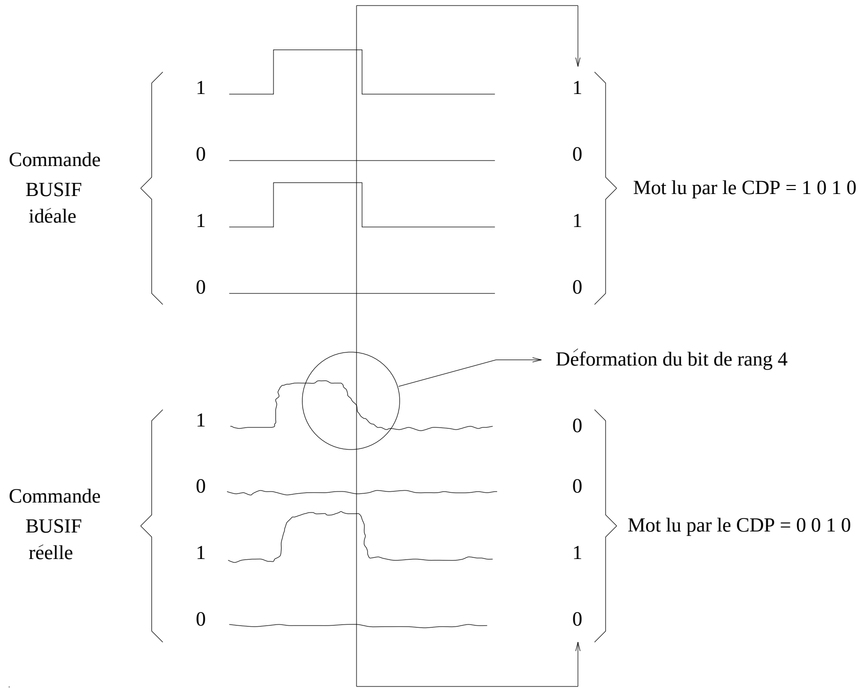


FIG. 7.6 - *Schéma montrant la mauvaise interprétation du mot de commande $A_{hex} = 1010_{bin}$. Le bit de rang 4 est déformé au moment de l'échantillonage par le CDP sur un coup d'horloge. Finalement, c'est le mot 0010_{bin} de remise à zéro qui sera échantilloné, provoquant une réinitialisation du pointeur de colonne : les données corrélées avec le déclenchement seront perdues par surécriture.*

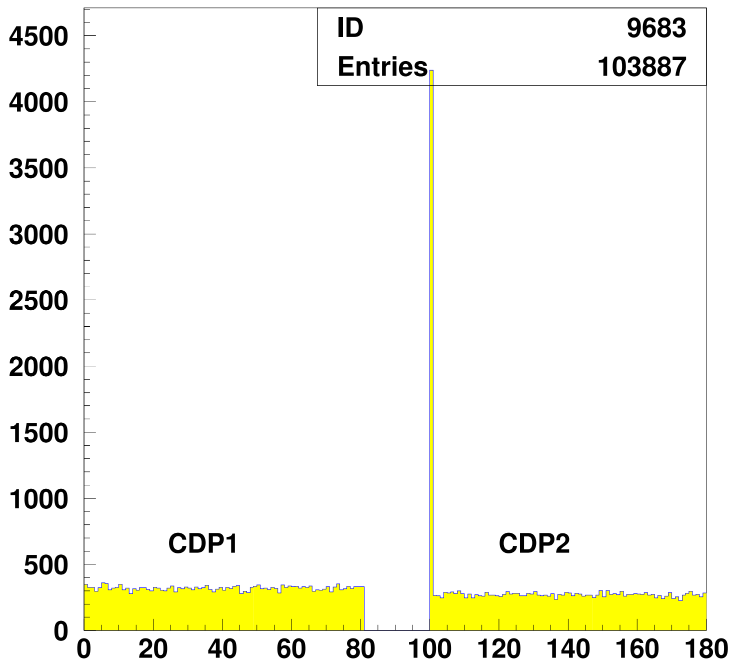


FIG. 7.7 - *Exemple de bloquage : remise à zéro systématique du pointeur de colonne du CDP 2, BUSIF6 C. Cette anomalie, due à un mauvais décodage des mots de commande par le CDP, entraîne la perte d'une partie des données de ce circuit par surécriture.*

Cette commande $2_{hex} = 0010_{bin}$ force une remise à zéro du pointeur de colonne, et conduit à un bloquage permanent sur la colonne 0 : **les données sont perdues par surécriture**.

La figure 7.7 illustre par un exemple, ce type de bloquage, rencontré en novembre 1995, sur le CDP 2 de la BUSIF6 bus C, bande NF1292. L'ensemble des données échantillonnées par ce CDP ont été surécrite dans les premières colonnes : elle ne sont donc pas valables.

Le repérage de colonnes bloquées demande une comparaison des colonnes de l'événement courant avec celles des événements précédents. Dans la pratique la comparaison est faite sur les deux précédents événements. Pour un CDP d'un événement n, la colonne Col_n est considérée bloquée si l'on à égalité des trois valeurs :

$$Col_n = Col_{n-1} = Col_{n-2}$$

En effet, la probabilité pour que trois déclenchements successifs se produisent sur une même colonne est proche de zéro. Cette méthode permet de repérer sytématiquement tout bloquage : **c'est la coupure F3**.

Les deux premiers événements ne sont cependant pas soumis à ce filtrage. Ils repésentent 2 pour 1 000 des données pour une prise de données classique.

La dernière coupure F4 a pour objectif d'isoler des défauts de synchronisation plus généraux, qui se traduisent par des progressions de colonnes hors d'un mouvement d'ensemble. Le problème ici est de définir ce mouvement d'ensemble.

Comme nous l'avons dit, nous somme amenés à considérer les CDP par groupe de 12 : il s'agit du nombre de circuits associés à un même bus A, B ou C : on notera E_{12} cet ensemble. Parmi ces 12 éléments, seront synchronisés les circuits faisant partie de sous-groupes contenant N_{sync} CDP de même colonne n ; ces sous ensembles seront notés E_n. Pour chaque événement, la coupure F4 vise à rejeter les sous-groupes E_n, dont la taille N_{sync} est inférieure à un seuil minimum N_{seuil}. Cette taille minimum N_{seuil}, est la limite, en dessous de laquelle le groupe de CDP associé, bien que synchronisé ne possède pas une taille suffisante pour apparaître comme représentatif de **la** syncrhonisation recherchée. Nous admettons pour l'instant la valeur de $N_{seuil} = 2$, la justification de ce choix étant faite par la suite.

Nous pouvons donc dire que les CDP classés par leur *TimeStamp* dans les sous-ensembles de type E_1 doivent être rejetés : **c'est la coupure F4**.

Pour isoler ces CDP déviants on procède à une double scrutation des numéros de colonnes dans l'ensemble E_{12}, qui permet, au premier passage, de construire les sous ensembles E_n. La seconde scrutation identifie les CDP des sous ensembles et procède à la validation F4 (voir figure 7.8).

Afin de se rendre compte de l'ampleur des désynchronisations au sein d'un bus de 12 CDP, nous procédons à la représentation suivante :

A chaque événement, on calcule la moyenne

$$Col_{moy} = \frac{\sum_{i=1}^{n} NumCol_i}{n}$$

des numéros de colonnes des n CDP ayant passés les coupure F1,F2 et F3. Ensuite, pour chacun des n CDP, on calcule les écarts indivuduels à cette moyenne en valeur algébrique :

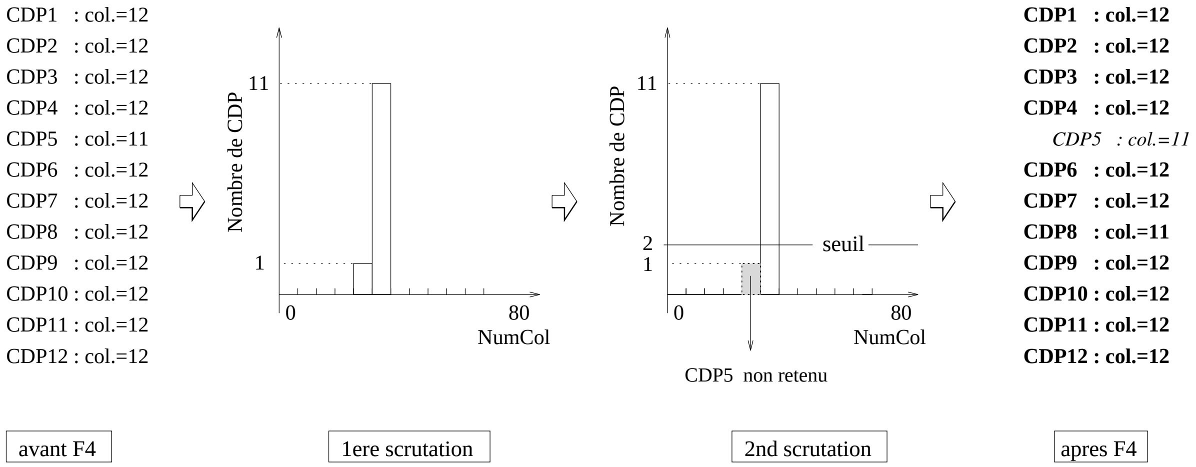


FIG. 7.8 - *La coupure F4, dans une première scrutation, classe les CDP par numéros de colonnes identiques, puis une seconde scrutation ne valide que les CDP appartenant à un groupe de 2 éléments ou plus. Dans cette illustration, le CDP 5 n'est pas validé.*

$$\delta_n = NumCol_n - Col_{moy}.$$

Cette quantité δ_n incrémentera le bin associé d'un histogramme qui s'étend donc de -80 à +80.

Intégré sur un grand nombre d'événements, ce mode de représentation en δ_n se traduira par un histogramme piqué en zéro pour une parfaite synchronisation des circuits de lecture.

La figure 7.9 illustre ce mode de représentation pour la BUSIF1 bus C. Il s'agit des écarts à la moyenne δ_n, intégrés sur 26 000 événements de NF1292. Le premier histogramme (a), qui correspond aux données brutes, montre une désynchronisation marquée des 8 CDP présents. Les filtrage F1 et F2 cumulés (b) éliminent les défauts de lecture (2% des données), mais pas les problèmes de synchronisation. En revanche, les coupures F3 et F4 rajoutés aux deux premières conduisent à un résultat correct : la distribution est piquée sur zéro, les circuits retenus présentent donc les même numéros de colonnes pour les 26 000 événements étudiés : pour ce bus, 30% des événements ont été marqués mauvais par la sélection.

7.2.4 Détermination de la valeur optimale de N_{seuil}

Condidérons un groupe de CDP synchronisés. Quelle est la taille limite d'un tel groupe, en dessous de laquelle on peut considérer raisonablement que tout ensemble de circuits

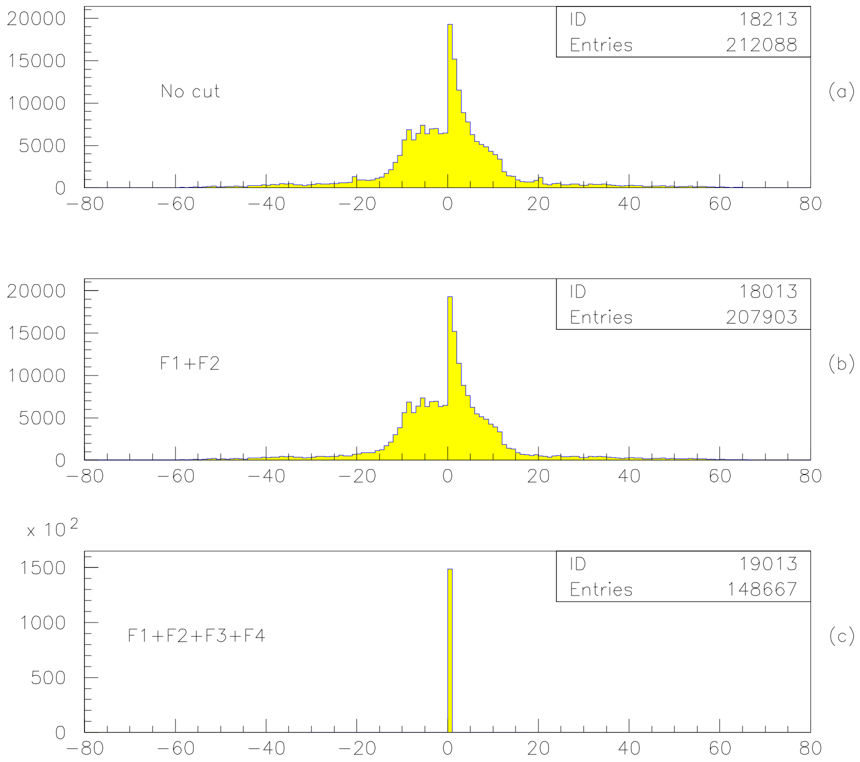


FIG. 7.9 - *Distributions des écarts à la moyenne δ_n, des colonnes des CDP de la BUSIF1C. La première ligne représente la distribution appliquée sur les données brutes, la seconde ligne est construite en aval des coupures F1 et F2, tandis que la troisième correspond au filtrage final F1, F2, F3 et F4. On constate que les coupure F1 et F2 suppriment quelques événements issus de lignes perturbées, mais ne corrige en aucun cas les défauts de synchronisation. Les coupures F3, F4 apportent un complément de filtrage qui éliminent tous les défauts de synchronisation.*

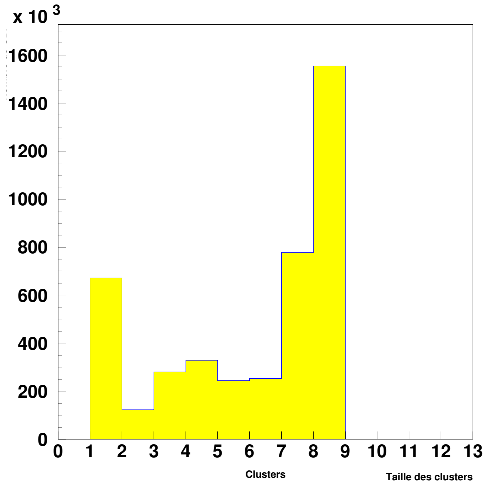


FIG. 7.10 - *Histogramme représentant les fréquences d'apparition des groupes synchronisés pour les 8 tailles possibles en novembre 1995. On constate que la distribution mesurée est loin de la distribution idéale piquée en* $N_{synch} = 8$

synchronisés est le résultat d'une coïncidence?

Pour répondre à cette question, nous nous plaçons tout d'abord, dans les conditions expérimentales de novembre 1995 afin d'observer la fréquence d'apparition des différents groupes synschronisés. Pour l'ensemble des 6 cartes BUSIF installées, nous observons la distribution de la figure 7.10, qui représente la fréquence d'apparition des différents clusters, pondérée par la taille de ces groupes. Pendant cette période 8 CDP étaient associés à chaque bus de lecture (A, B ou C).

On constate que la distribution mesurée est loin de la distribution idéale piquée en $N_{synch} = 8$. En particulier, on peut remarquer une forte contribution des CDP isolés : groupes de taille 1 qui représentent ici 18.9% de l'ensemble collecté.

Le filtrage de la coupure F4 sera d'autant plus pur, que le nombre de groupes synchronisés retenus par événement sera faible. Au vu du nombre de circuits présents par bus (huit), un seuil $N_{seuil} > 4$ entraîne une pureté de 100%. En effet, dans ce cas, il est impossible de former deux groupes E_n par événement. Pour ce qui est de l'efficacité du filtrage, il est clair que celle-ci va aller diminuant avec l'augmentation de N_{seuil}. Le choix de $N_{seuil} = 1$ revient à ne pas appliquer F4 et ne résoud donc pas les problèmes de synchronisation. Par conséquent, la valeur de N_{seuil} est à choisir parmi le triplet 2,3,4. Une impureté sera introduite par ce filtrage, chaque fois que pour un événement donné, on isolera plusieurs groupes synchronisés disjoints. Sur le mode de représentation choisi (voir figure 7.11), cela se traduit par des δ_n non nuls qui "polluent" les histogrammes piqués en zéro. On vérifie sur cette figure que la pureté augmente avec la valeur de N_{seuil}.

Le choix de ce seuil est donc le résultat d'un compromis entre pureté et efficacité. Le seul passage de $N_{seuil} = 2$ à $N_{seuil} = 3$ entraîne une perte d'efficacité de 3.4%. Rappelons que ce traitement est appliqué en-ligne. Il est donc préferable de primer l'efficacité à la pureté pour éviter de marquer trop d'événements comme étant mauvais.
Le seuil $N_{seuil} = 2$ est donc choisi. Il permet de repérer les éléments isolés qui représentent 19% des données et sont les principales manifestations de ces problèmes de désynchronisations.

Si l'on sait que le filtre F4, ainsi défini va éliminer, pour chaque événement, les CDP isolés ("groupe" de taille 1), il est intéressant de connaitre la proportion de bons circuits que l'on rejette dans cet ensemble.

Effectivement, si dans la plupart des cas, un CDP isolé est le résultat d'une perturbation quelconque engendrant de mauvaises données, on peut imaginer des cas de figures qui n'en font pas un mauvais circuit.
Prenons l'exemple d'un groupe de circuits synchronisés. Si à un instant du déversement, un des circuits du groupe subit une unique désynchronisation : il se voit isolé du reste des CDP par un décalage constant qui n'affectera pas les données qu'il propose pour la suite du burst. Classé parmi les circuits isolés, la coupure F4 ne fera par ressortir son caractère valide : ses données seront perdues. Il est donc important de quantifier la proportion de ces cas de figure.

Estimer cette quantité de circuits isolés mal jugés, revient en fait à évaluer la proportion de colonnes présentant un caractère aléatoire dans cette population. Sur les données issues de novembre 1995, pour tous les CDP connus comme étant rejetés par la coupure F4, nous imposerons des colonnes tirées aléatoirement dans la plage [0,80]. La différence entre l'enregistrement réel et cette simulation nous donnera donc la fraction de circuits peuplant cet ensemble, qui n'auraient pas dus être rejetés.

Pour résumer, la validation des données proposée, repose sur quatre coupures F1, F2, F3 et F4 appliquées en série. Chacune de ces coupures utilise comme base de selection l'information contenue dans les mots de *TimeStamp* :

- le *ChipId* pour F1.
- le *NumCol* individuel courant pour F2.
- les *NumCol* individuels et antérieurs pour F3.
- les *NumCol* d'ensemble pour F4.

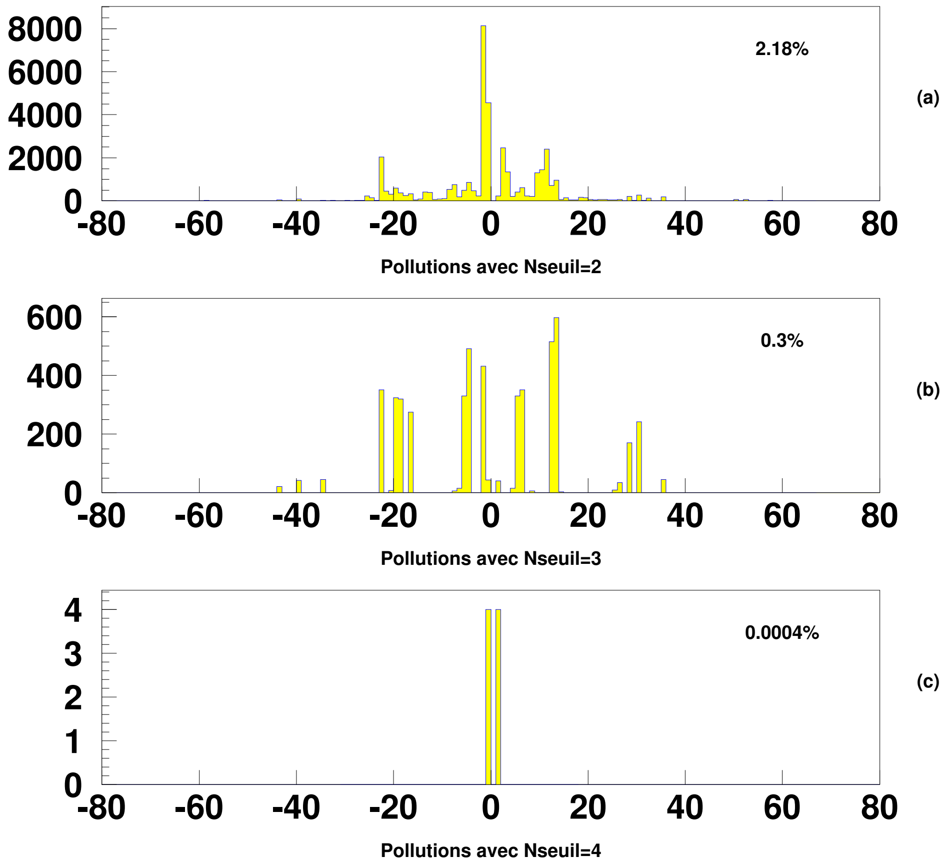


FIG. 7.11 - *Ces graphiques en δ_n illustrent les impuretés engendrées par la détection de plusieurs groupes synchronisés au sein de même événements. Ils représentent les résidus mals filtrés par la coupure F4. On constate que cette pollution diminue avec la valeur du seuil choisi. Le seuil 2 conduit à une pureté de* $97.8\% \pm 0.01\%$ *tandis que le seuil 3 donne une pureté de* $99.7\% \pm 0.004\%$. *La valeur* $N_{seuil} = 4$ *est la limite au delà de laquelle la pureté de F4 est égale à 100%.*

Ces quatre selections successives permettent d'éliminer les données issues de chaînes de lecture perturbées. Cette réalisation nous conduit à un critère de validation binaire des données. La réduction ainsi obtenue est d'un facteur 12. Toute cette étude a été réalisée off-line. Nous verrons par la suite quelles sont les contraintes imposées par une implantation en-ligne des ce filtrage.

La figure 7.12 illustre ce résumé par la progression du filtrage des données pour la BUSIF3A. Il s'agit d'un cas pathologique rencontré en novembre 1995 qui intègre tout les problèmes à la fois, et notamment des bloquages de colonnes et évolution de colonnes dans des zones préférentielles. L'évolution du filtrage se lit de gauche à droite : sans filtrage, avec F1+F2 et avec F1+F2+F3+F4.

La première ligne repésente les écarts δ_n des colonnes par rapport à la moyenne d'ensemble. La seconde illustre la distribution des numéros de colonnes pour tous les CDP présents, et la troisième les *ChipId* effectivement enregistrés.

Après F1+F2, 11% des données sont rejetées pour cause de *ChipId* incorrect ou numéro de colonne abérant. En aval de F1+F2+F3+F4, la répartition des colonnes est idéale. Au tolal, 28% des événements lus par le bus A de cette carte sont donc considérés comme mauvais.

7.2.5 Les secteurs affectés par le filtrage

Au cours de la prise de données de novembre 1995, seules les couronnes internes des détecteurs MD1 et MD2 étaient installées. Nous appelons configuration du détecteur de multiplicité, la façon dont les secteurs de détections sont répartis sur les bus A, B, C des cartes BUSIF. Trois configurations ont été utilisées successivement :

- la configuration 1 : runs 682 à 875. Tous les secteurs du disque MD1 interne, étaient cablés. Pour la couronne interne de MD2, seuls les secteurs 1, 2, 7, 8, 9, 10, 27, 28, 29, 30, 35 et 36 étaient cablés.
- la configuration 2 : runs 886 à 987. Tous les secteurs du disque MD1 interne, étaient cablés. Pour la couronne interne de MD2, seuls les secteurs 1, 2, 7, 8, 9, 10, 27, 28, 29, 30, 35 et 36 étaient cablés.
- la configuration 3 : runs 962 à 1059. Tous les secteurs du disque MD1 interne, étaient cablés. Pour la couronne interne de MD2, seuls les secteurs 1, 2, 7, 8, 9, 10, 15, 16, 17, 18, 23, 24, 25, 26, 29, 30, 35 et 36 étaient cablés.

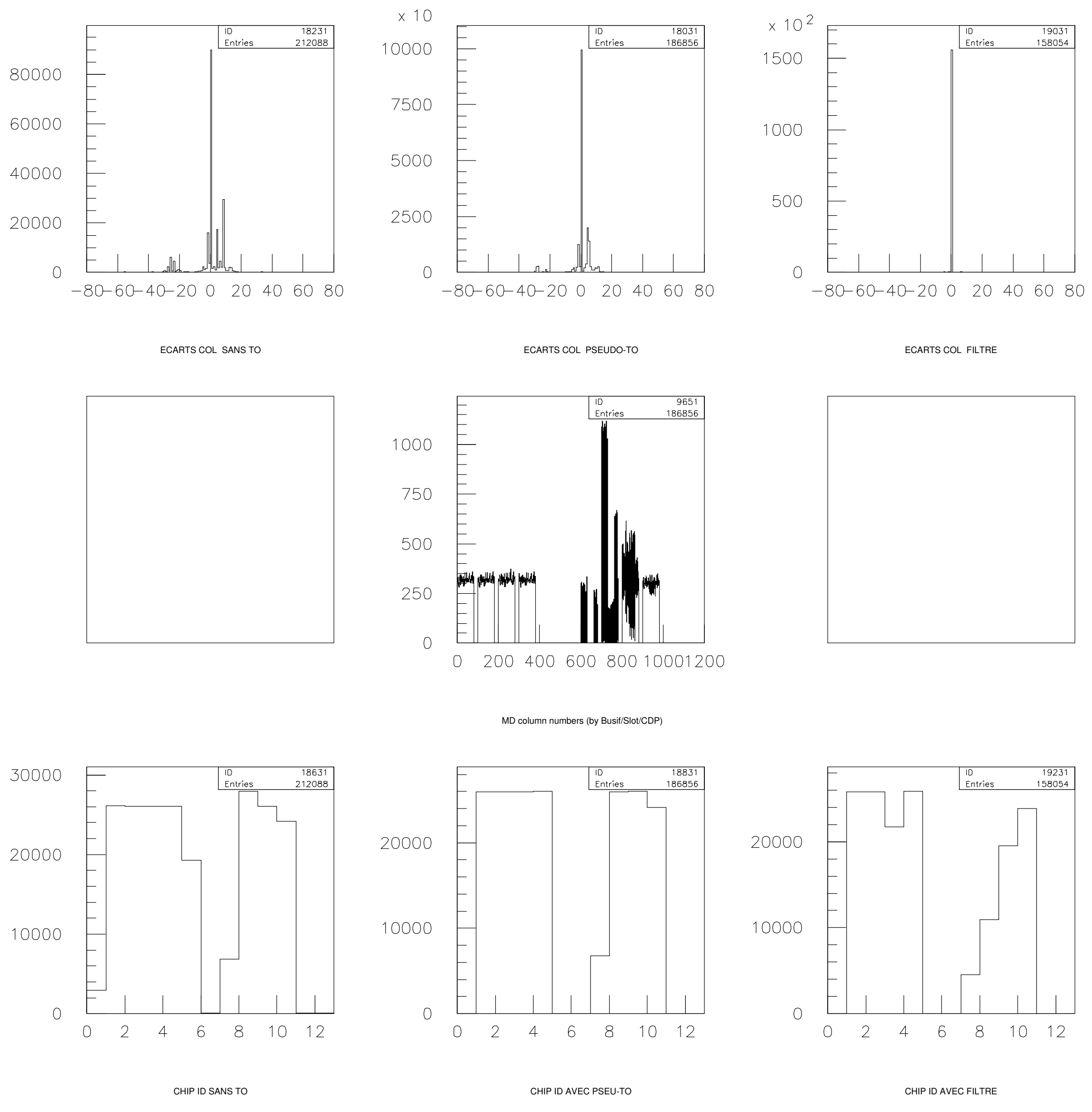


FIG. 7.12 - *Progression du filtrage des données TimeStamp de la busif 3A. En final 72% des données sont validées par le filtrage. Bande NF1292 novembre 1995, échantillon de 26000 événements.*

Notons, qu'un numéro de *run* est associé à une bande. Cependant, contrairement aux numéros de run, les numéros de bandes ne sont pas toujours consécutifs.

Pour ces trois périodes de novembre 1995, nous avons cherché à voir quelles étaient les parties du détecteur conservées après l'application du filtre à 4 coupures précédemment décrit. Les études étant les même pour les trois configurations, nous ne présentons dans les paragraphes suivants, que les résultats qui concernent la configuration 1.

7.2.6 Les secteurs affectés par le filtrage pour la configuration 1

Etude du disque MD1 :

Dans cette configuration 1, l'intégralité de la couronne interne de MD1 était cablée. Une installation idéale doit donc présenter une répartition uniforme des réponses des CDP sur les 36 secteurs.

Au lieu de cela, après filtrage, les secteurs retenus bons, présentent une distribution trouée, comme le montre la figure 7.13. On observe 5 trous plus ou moins marqués dans la surface sensible de MD1 :

1) un demi trou dans le secteur 4,

2) un rejet total des secteurs contigus de 13 à 14,

3) un rejet mitigé dans les secteurs contigus de 20 à 22,

4) un rejet total des secteurs contigus de 25 à 28,

5) un trou important dans le secteur 31.

Le trou numéro 1 :

Le secteur 4 est lu dans cette configuration1 par la BUSIF1, bus A. Le problème détecté par le filtrage est du à l'absence du CDP 9 sur ce bus. Cette absence se caractérise par un *ChipId* (voir figure 7.14) lu à la valeur 0 ou 1 plutôt que le 9 attendu. Ce circuit est donc rejeté par la coupure F1. L'autre moitié du secteur 4, gérée par le CDP 10[4] est cependant conservée, ce qui explique l'activité de 50% de ce secteur.

4. partie extérieur du secteur interne

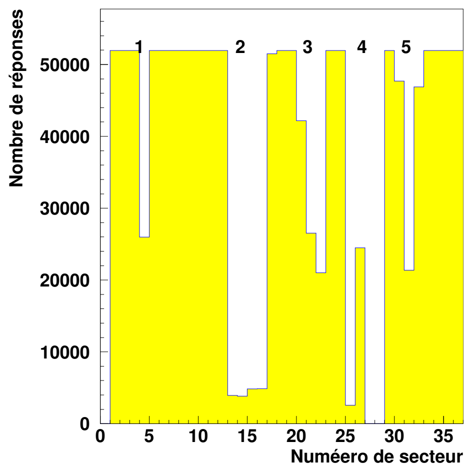


FIG. 7.13 - *Configuration 1, novembre 1995 : cet histogramme représente l'activité des 36 secteurs de la couronne interne du disque MD1 après le filtrage à quatre coupures. Les divers problèmes de synchronisation inter-CDP et les bloquages de colonnes conduisent à la réjection de 31% des canaux de détection sur l'ensemble de cette couronne* $MD1_{int}$.

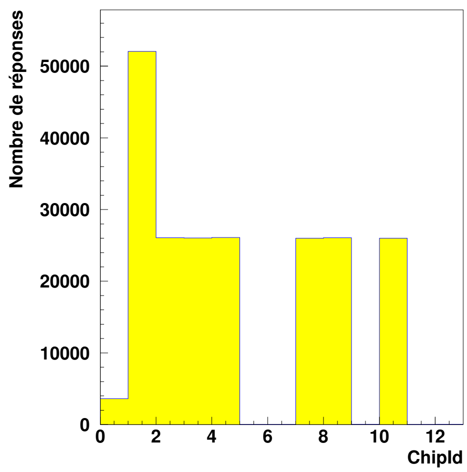


FIG. 7.14 - *ChipId effectivement lus par la BUSIF1A. L'absence du circuit 9 se caractérise par un mauvais ChipId lu. Les données associées à ce CDP sont repérées comme mauvaises par l'application de la coupure F1.*

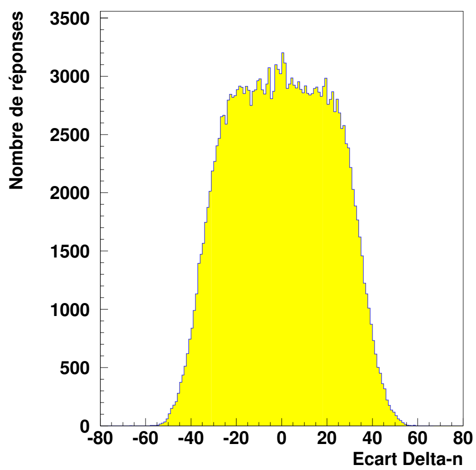


FIG. 7.15 - *Désynchronisation totale des 8 CDP de la BUSIF4B configuration1. Cette représentation tient compte des filtrages F1 et F2. Les données associées à ces 8 circuits seront intégralement rejetées par la coupure F4.*

Le trou numéro 2 :

Le rejet quasi total des secteurs contigus de 13 à 14 (trou numéro 2), s'interprète facilement en regardant le comportement du bus BUSIF4B. En effet la figure 7.15, représente la répartition des écarts à la colonne moyenne pour 26 000 événements. Son profil indique que ce groupe de secteurs, n'a pu être validé pour cause d'une désynchronisation totale des 8 CDP (voir figure 7.15). Il s'agit là d'un défaut dû à des fluctuations de signaux, sur les lignes d'horloge des CDP, entraînant des comportements totalement décorrélés des pointeurs de colonnes pour les 8 CDP présents. Les données associées sont rejetées par la coupure F4.

Le trou numéro 3 :

La perturbation des secteurs contigus de 20 à 22 est provoquée par une mauvaise synchronisation du CDP9 de la BUSIF4C d'une part (chute du secteur 20 dans 25% des lectures), et par des défauts du même type sur les 4 premiers CDP lus par la BUSIF 1C. Nous verrons que ce défaut est un cas particulier de désynchronisation, qui n'affecte pas les données enregistrées. Il provoque donc une rejection abusive de la part du filtre à quatre coupures.

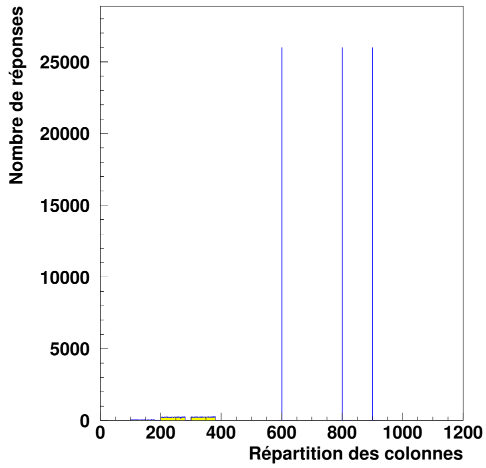


FIG. 7.16 - *Cette distribustion des colonnes pour la BUSIF3C, met en évidence les bloquages des CDP 7, 9 et 10 sur la BUSIF 3C configuration 1. Cette représentation tient compte des filtrages F1 et F2.*

Le trou numéro 4 :

Les secteurs 25 à 28 ont été pris en charge par la BUSIF3C. L'examen de ces quatre portions fait ressortir des problèmes de bloquage de colonnes pour les CDP 7, 9 et 10 tandis que les quatre premiers CDP présentent de mauvais *ChipId* et une mauvaise synchronisation. Ce bus est donc perturbé en lecture, par conséquent, les données accédées sont rejetées. Ici, l'ensemble des coupures est utilisé en cascade. La figure 7.16 représente la distribution des colonnes pour les 8 CDP concernés et met en évidence le bloquage de CDP 7, 9 et 10.

Le trou numéro 5 :

Le trou numéro 5 dans la disribution concerne les secteurs 29 à 32 gérés par la BUSIF 3A. Parmi ces 8 CDP, les circuits 7, 8 et 9 ont été les objets de pertubations importantes entraînant des évolutions de pointeurs dans des zones préférentielles. Ces défauts sont principalement les conséquences de remises à zéro non systématiques. La figure 7.17 montre clairement, par la répartition des numéros de colonnes, ces remises à zéro fréquentes.

L'ensemble de ces défauts entraîne une surface active globale, pour MD1 couronne interne, égale à 69% de sa valeur maximum.

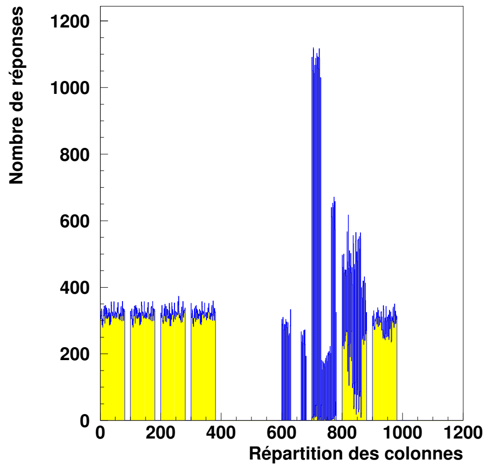


FIG. 7.17 - *Des remises à zéro fréquentes mais non systématiques perturbent les 4 derniers CDP responsables des secteurs 31 et 32. Les données sont perdues par surécriture. La coupure F4 permet de marquer ces données non valides. La lecture de ce bus est assurée par la BUSIF 3A.*

Etude du disque MD2 :

Comme indiqué précédemment, $\frac{1}{3}$ de la seconde couronne MD2 étaient cablée en novembre 1995 pour cette configuration 1. Sur ces 33%, les seuls secteurs retenus après filtrage sont affichés sur la figure 7.18. Ils représentent 54% de l'activité attendue, la plus grosse défaillance venant des quatre secteurs 27 à 30, complètement éteints après filtrage. En regardant dans le détail les enregistrements de la BUSIF 6C chargée de cette lecture, on s'aperçoit que les CDP 7, 8, 9 et 10 affectés aux secteurs 28, 27 sont considérés comme physiquement absents (mauvais ChipId), tandis que le pointeur de colonne du CDP 4 est constamment bloqué sur la colonne 0. Les 3 premiers circuits, quant à eux, présentent une désynchronisation complète et ne peuvent former un groupe à progression d'ensemble. Présentant tous les défauts possibles, les données de ces deux secteurs sont systématiquement marquées comme impropres à une analyse physique.

L'ensemble de ces défauts entraîne une surface active globale retenue, pour MD2 couronne interne, égale à 54% de la valeur optimale. Cette quantité correspond pour cette configuration 1 à 18% de l'activité maximum que l'on peut attendre de MD2in total.

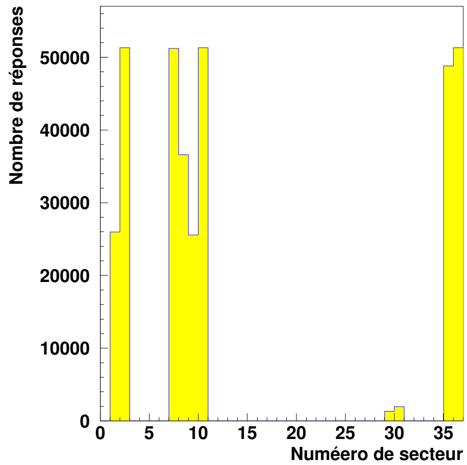


FIG. 7.18 - *Cet histogramme représente l'activité retenue de MD2in. Pour cette couronne seulement 54% des canaux installés ont été retenus après filtrage.*

Le filtre à quatre coupures est donc utilisé comme marqueur des données, ayant été délivrées par les chaînes de lecture. A ce titre, il exploite au maximum l'information *TimeStamp* lorsque celle-ci est disponible.
Cependant cette selection affecte-t-elle des données réellement valides ?

Pour répondre à cette question, il est necessaire de mener une autre étude permettant de porter un jugement sur la qualité des données, sans prendre en compte l'information *TimeStamp*.
Une analyse de la distribution des impacts des particules chargées sur les disques de silicium MD1 et MD2 permet d'isoler les secteurs à conserver. En effet, lors d'une interaction Pb-Pb, même s'il existe un plan privilégié pour les intéractions nucléons-nucléons, l'orientation en ϕ de ce plan est liée à un phénomène purement aléatoire : la localisation spaciale des ions. Il en résulte, que sur un nombre suffisamment grand de collisions, la répartission intégrée des impacts sur les bandes de silicium doit présenter un profil homogène en ϕ (voir figure 7.19).

La méthode à appliquer consiste donc en l'éxamen des distributions de multiplicité, secteur par secteur. La recherche de discontinuités, tant en ϕ qu'en r, permet de repérer les secteurs défaillant soit par leur mauvais fonctionnement soit par un dysfonctionnement de leur sytème de lecture. Une distribution de multiplicité pour un secteur donné a pour allure celle de la figure 7.20

Comme le montre les figures 7.21 et 7.22, les distributions de multiplicté, intégrées sur

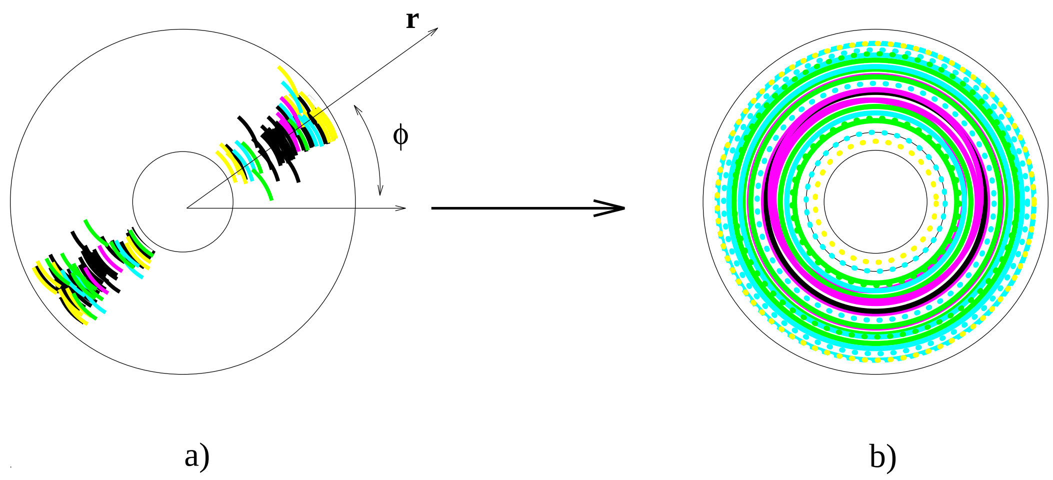


FIG. 7.19 - *Cette figure schématise la distribution en ϕ des impacts, provoqués par les particules secondaires chargées sur un disque de silicum : pour un événement (a) et intégré sur un grand nombre d'événements (b). Une couronne idéale doit présenter une distribution des impacts homogène en ϕ (b).*

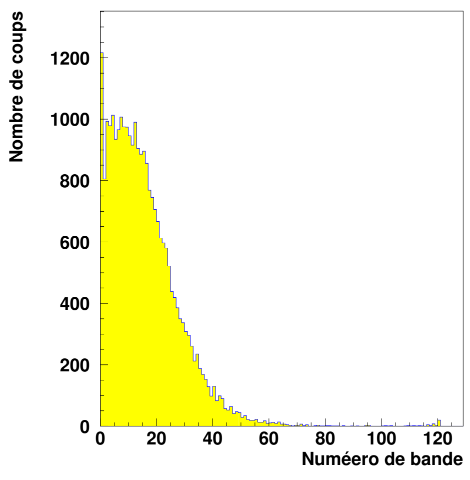


FIG. 7.20 - *Allure d'une distribution de multiplicité chargée pour un secteur. Il s'agit ici du secteur numéro 30, échantilloné sur 25000 événements, pour la configuration 1.*

les 36 secteurs, présentent des discontinuités plus ou moins marquées.

Les "trous" correspondent à des mauvais fonctionnements des circuits de lecture ou à des problèmes sur les éléments sensibles (défaillances ou absences). L'équipe de l'INFN de Turin à conduit une recherche systématique des discontinuités sur l'ensemble des 36 secteurs, pour la période de 1995. Les résultats obtenus, pour chaque configuration (1, 2 et 3), nous on donc permis de valider les secteurs du détecteur de multiplicité, par une méthode complétement décorrélée des informations *TimeStamp*.

La figure 7.23 montre les fréquences de validation des 36 secteurs par le filtrage à quatre coupures (a) par l'analyse de continuité des distributions (b) ainsi que la différence entre ces deux méthodes (c) = (a)-(b).

La différence entre les deux méthodes (fig. 7.23.c) nous montre que le filtrage à quatre coupures est moins restrictif que le filtrage par les distributions de multiplicité. Ceci est le point fondamental recherché, pour autoriser une implantation en-ligne : un filtre en-ligne ne doit pas rejeter de bonnes données.
Cependant, on peut tout de même noter, sur la figure 7.23.c, que pour certain secteurs, des données sont rejetées par le filtrage à 4 coupures, tandis qu'elle sont considérées comme bonnes si l'on examine les distributions de multiplicité enregistrées. Il s'agit des secteurs 20, 21, 22 pour le disque MD1 et du secteur 30 pour le disque MD2.
Or, nous avons vu que la rejection de données par les 4 sélections F1, F2, F3 et F4 signe des problèmes de lecture ou de synchronisation qui ne peuvent donner des informations cohérentes. Deux cas de figures peuvent expliquer cette contradiction :

1 Des défauts dans la distribution des signaux d'horloges par la carte CCTD, ont désynchronisé tout les circuits CDP, sans pour autant affecter l'enregistrement correcte des données. Ceci est possible si des fluctuations d'horloge, à l'origine de ces problèmes, se produisent en dehors du temps de latence de déclenchement δt (défini au chapitre 3).

2 Les premiers déversements sont enregistrés sans problèmes par les circuits CDP : les mémoires des cartes BUSIF contiennent donc des données cohérentes. Puis, l'apparition d'un problème (à déterminer) bloque fréquemment les circuits programmables des BUSIF dans les cycles de lecture des données CDP. Ce bloquage ne permet pas la surécriture des mémoires internes des BUSIF : elles contiennent donc toujours les données cohérentes des précédents déversements. L'analyse des distributions de multiplicité fera donc ressortir des données valables.

Si pour les secteurs incriminés, nous sommes dans les cas numéro 2, l'absence de surécriture doit aussi affecter les numéros de colonnes : pour un déversement numéro q, la colonne d'un événement n noté $Col\,_n^q$ doit être telle que :

$$Col\,_n^q = Col\,_n^{q-1} \tag{7.1}$$

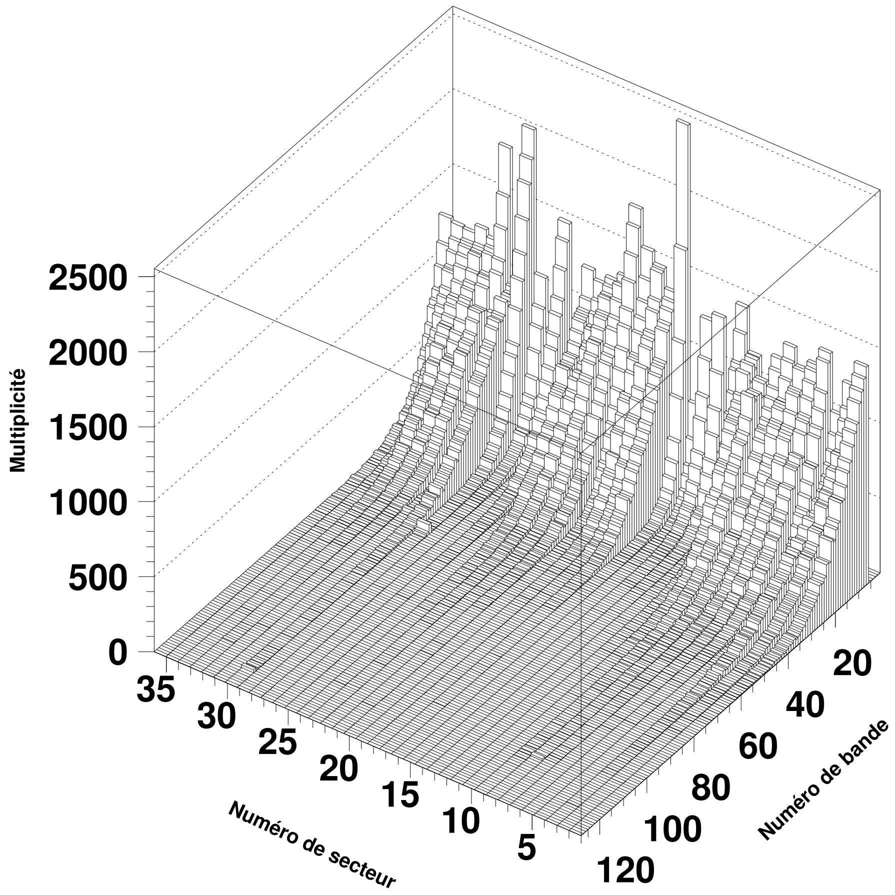


FIG. 7.21 - *Cette figure est relative au disque de détection $MD1_{interne}$, configuration 1. Elle nous montre la distribution de multiplicité chargée, pour les 128 bandes des secteurs internes et ce, intégré sur les 36 secteurs qui forment la couronne interne. On peut remarquer des discontinuités dans les distributions selon les valeurs de ϕ (numéro de secteur de 1 à 36).*

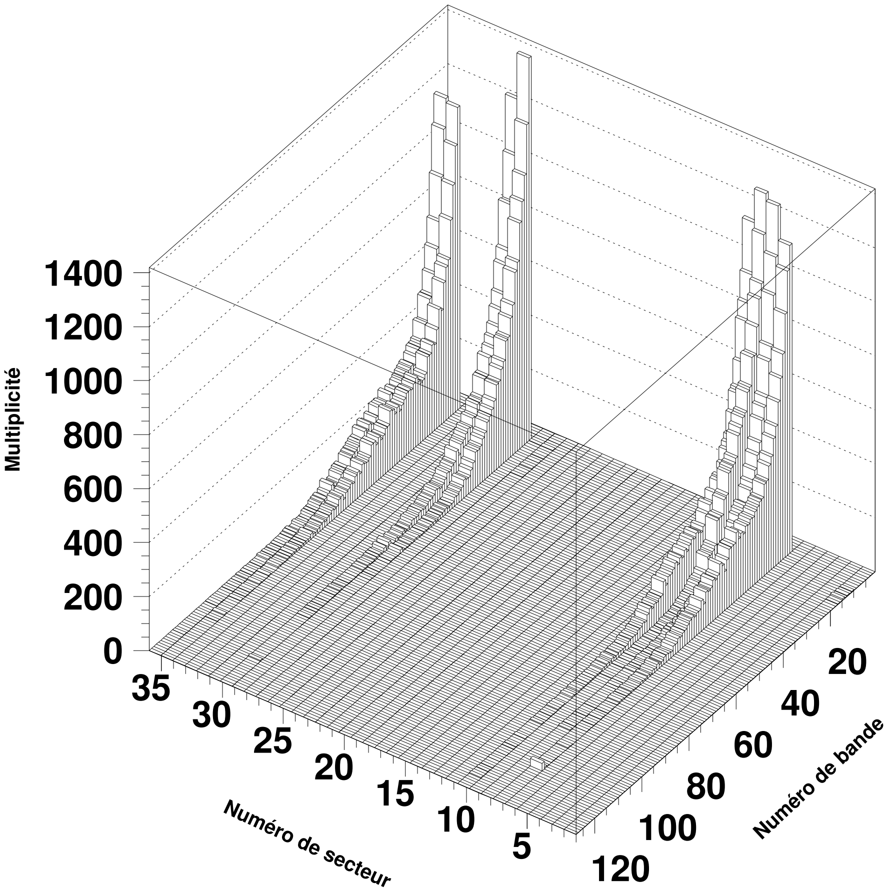


FIG. 7.22 - *Cette figure utilise le même mode de représentation que fig.7.21, mais concerne le disque* $MD2_{interne}$*, configuration 1. On constate qu'il est trés peu actif. Ceci vient principalement du fait que ce disque était en cours de réalisation pendant la période de novembre 1995, trés peu de secteurs étaient donc présents.*

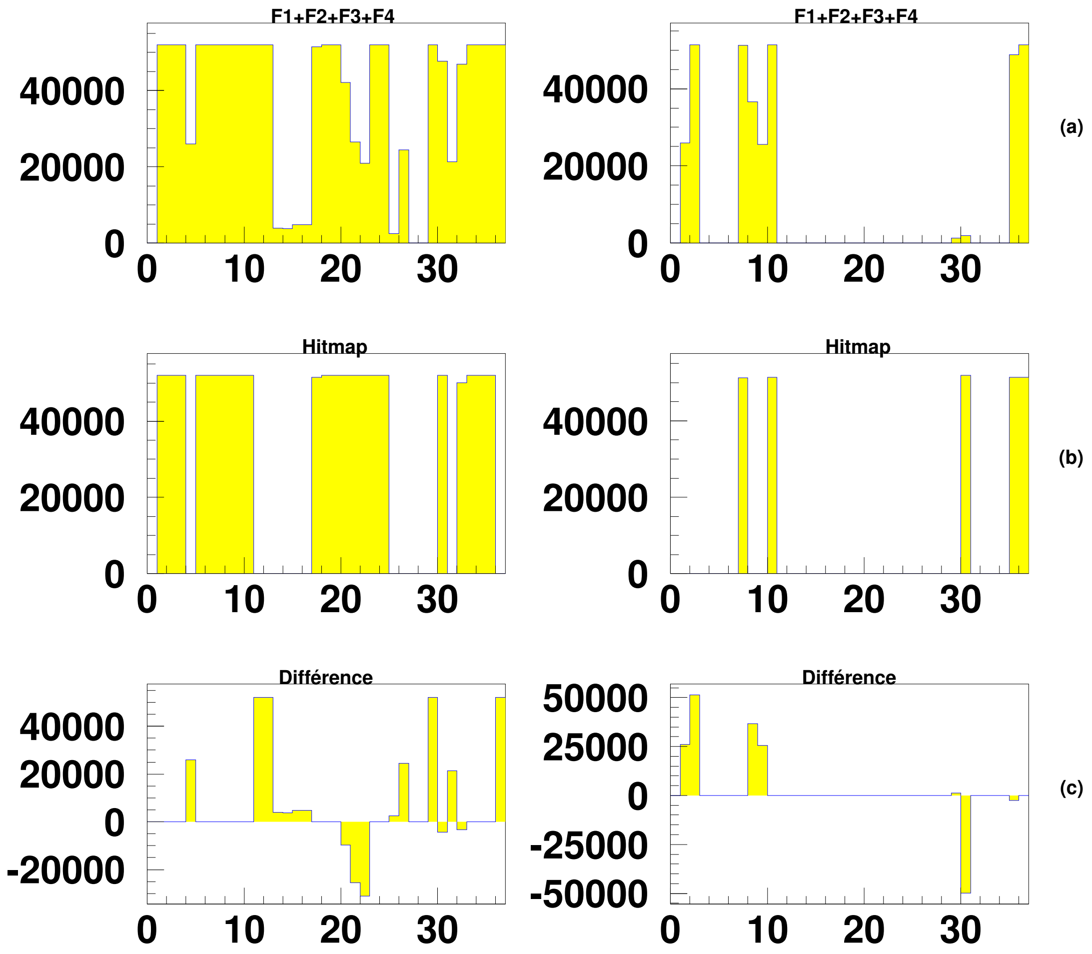


FIG. 7.23 - *Cette figure illustre la fréquence de validation des 36 secteurs, pour les disques MD1 (histo. de gauche) et MD2 (histo. de droite). La ligne (a) correspond aux résultats obtenus par le filtrage à 4 coupures, la ligne (b) par l'analyse des continuités de distribution obtenue par le groupe de l'INFN de Turin. La différence entre ces deux méthodes (c) nous montre que le filtre à 4 coupures a été plus sévère pour les secteurs 20, 21, 22 du disque MD1 et pour le secteur 30 de MD2.*

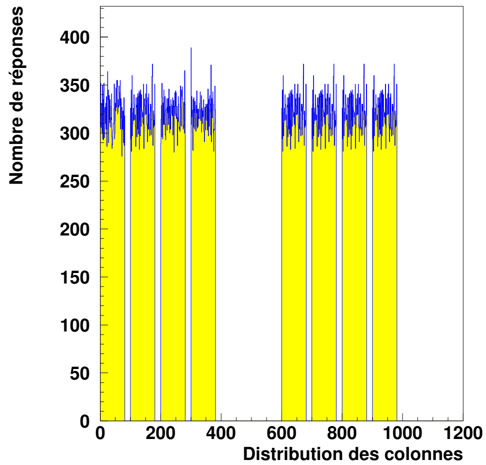


FIG. 7.24 - *Les CDP qui gèrent les secteurs 21, 22, 23 et 24 du disque MD1, présentent des distributions uniformes de colonnes. Ceci tend à montrer qu'il n'y a pas de bloquage des surécritures dans les mémoires des BUSIF.*

Comme le montre la figure 7.24 les CDP qui gèrent les secteurs incriminés présentent des distributions de colonnes uniformes qui tendent à rejeter ce cas de figure.

Pour conforter cette mesure, nous avons effectué, sur plusieurs déversements et pour tous les événements, les différences $Col\,{}^{q}_{n} - Col\,{}^{q-1}_{n}$, afin de vérifier si l'équation (7.1) pouvait s'appliquer. Les résultat pour les quatres secteurs 21, 22, 23 et 24 du disque MD1 sont présentés sur la figure 7.25. Il n'apparaît pas de persistances majeures de mêmes colonnes. Au contraire, les distributions des différences, de type triangulaire, montrent qu'entre deux déversements, les colonnes se comportent comme des variables aléatoires indépendantes.
Les désaccords ne proviennent donc pas de défauts du type "cas numéro 2". Le filtre à quatre coupures est donc innéfficace dans ce cas de figure : il marque des bonnes données comme étant mauvaises.

Les même phénomènes se retrouvent pour le secteur 30 du disque MD2.

Le filtre n'étant pas éfficace pour les chaînes de lecture de ces groupes de secteurs, il conviendra de les repérer avant les prises de données pour paramétrer le filtre.

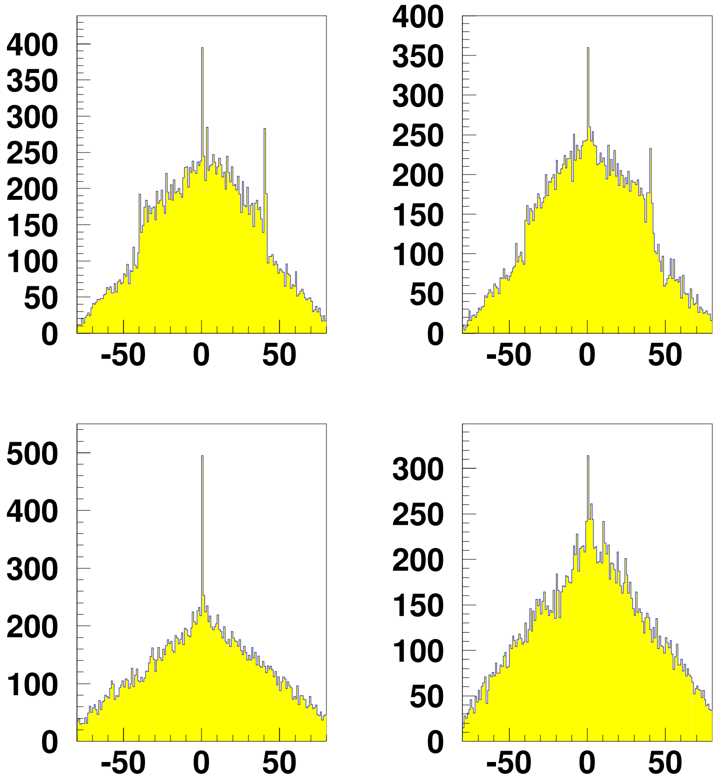


FIG. 7.25 - *Pour les secteurs 21, 22, 23 et 24 de MD1, les différences* $Col_{\,n}^{\,q} - Col_{\,n}^{\,q-1}$*, effectuées sur plusieurs déversements et pour tous les événements, ne montrent pas de persistances majeures de mêmes colonnes. Au contraire, les distributions des différences, de type triangulaires, montrent qu'entre deux déversements, les colonnes se comportent comme des variables aléatoires indépendantes.*

7.3 L'implantation en-ligne du filtre à 4 coupures

L'algoritme de ce filtre construit à partir des études précédentes est basé sur des opérations de comparaison simple. La figure 7.26. nous montre l'enchaînement logique des différentes étapes.

On retrouve en particulier les 4 points de marquage F1, F2, F3 et F4, disposés en cascade. Pour chaque déversement, le filtre est appliqué à chacun des événements des 3 bus A, B et C. Ceci repésente $3000 \times 3 = 9000$ itérations. Chaque itération compte environ 10 000 instructions processeurs. Le filtrage de 3000 événements demande donc quelques 90 millions d'instruction machine. Pour optimiser le temps d'exécution de ces séquences, nous avons dû :

- éliminer toute étape de calcul intermédiaire,
- remplacer le maximum d'opérations de comparaison par des opérations arithmétiques. Ceci demande l'utilisation de variables essentiellement booléennes.
- éliminer le maximum de boucles, en les remplaçant par les séquences d'instructions qu'elles réduisent.

Ces différents points conduisent à un code quelque peu hermétique, mais nous ont permis de diminuer le temps de filtrage d'environ 2 secondes.

D'autre part, comme nous l'avons vu, les transputers des BUSIF sont les seuls points du réseau qui manipule les informations *TimeStamp*. J'ai donc rajouté, dans le filtre, des séquences d'instructions qui échantillonnent les numéros de colonnes avant et aprés filtrage. A partir de ces relevés, des histogrammes en δn (voir chapitre paragraphe 7.2.3) sont construits puis remontés vers la station SUN. Une fois résidents sur le disque de cette machine hôte, nous utilisons une application graphique qui permet d'oserver on-line la qualité des synchronisations inter-CDP, et ce pour les 216 chaînes de lecture.

7.4 Conclusion

Toute cette étude nous à permis, à partir de données réelles, de concevoir un filtre basé sur 4 coupures en cascade : F1, F2, F3, F4. Son rôle est de valider les données par un marquage (et non une élimination). Il utilise comme base de travail les seuls informations *TimeStamp*. Ces informations n'étant disponibles qu'au niveau des transputers des BUSIF, le filtrage doit être fait en-ligne. Il apportera pour le dépouillement hors-ligne une information redondante sur la validité des mots de 64 bits.
Sur les transputers BUSIF, j'ai implanté conjointement à ce filtre, des processus d'histogramming qui permettent d'obtenir en temps réel les distributions en δn pour chaque groupe de 12 circuits CDP. Ce mode de représentation permet de se rendre compte, des divers problèmes de lecture et de synchronisations de CDP, pendant la prise de données.

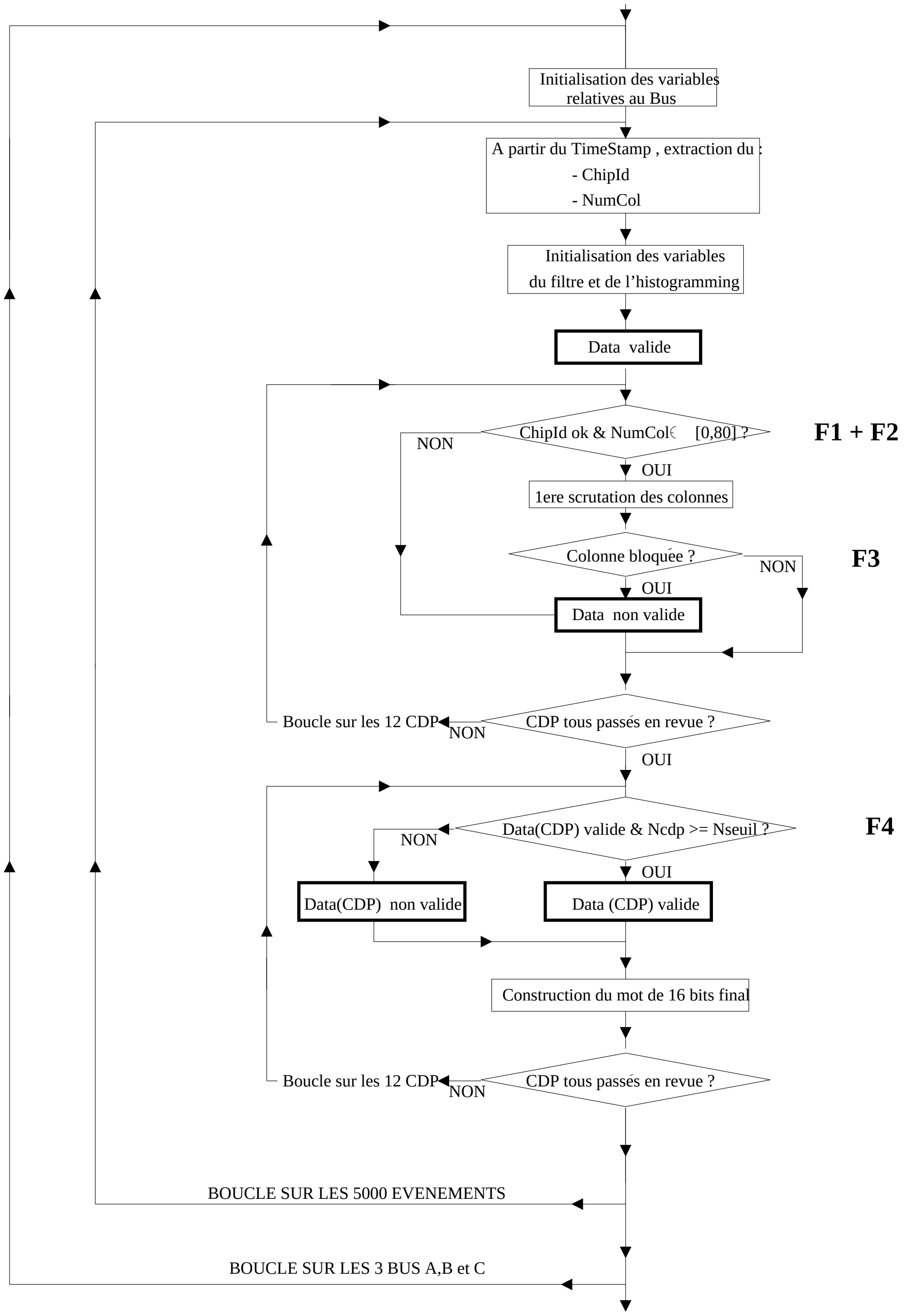


FIG. 7.26 - *Enchaînement logique des différentes étapes suivies par le filtre à 4 coupures. On retrouve les points de marquage F1, F2, F3 et F4, disposés en cascade. Cet algoritme, une fois optimisé et implanté sur transputer utilise quelque 90 millions d'instruction machines pour filtrer 3000 événements.*

Le suivi en direct du comportement des 216 chaines de lecture, permet des interventions rapides sur le détecteur de multiplicité, lorsqu'il est possible de descendre dans le hall d'expérience (quelque accès autorisé sur 30 jours de prise de données).

Le filtre à 4 coupures permet donc d'effectuer un marquage des données par une analyse des mots de contrôle *TimeStamp*. Ces informations n'étant disponibles qu'au niveau des transputers des BUSIF, le filtrage doit être fait en-ligne.

Bibliographie

[1] Ecole de Gif, 12-16 sept. 1988, "QCD sur réseau et température finie"

[2] Ecole de Joliot-Curie IN2P3, 1995.

[3] H. Statz, CERN-TH. 6216/91 (1991).

[4] F. Karsch, Nucl. Phys. A590(1995) 367c.

[5] D.B. Lichtenberg. Phys. Lett B193 (1987) 95.

[6] Z. Phys.C. Part. and Fields 41, 615 (1989).

[7] E. Eichten et *al.* Phys. Rev. D17 (1978) 3090.

[8] E. Eichten et *al.* Phys. Rev. D21 (1980) 203.

[9] F. Karsh et H. Satz, Z. Phys. C37 (1987), 617.

[10] H. Satz IX automn school on the QGP. Phys. Lisbonne, (1987) 243.

[11] J. D. Bjorken, Phys. Rev. Lett. D51 (1983) 140.

[12] T. Matsui and H. Satz, Phys. Lett. B 178 (1986) 416.

[13] J. D. Bjorken, Phys. Rev. Lett. D13 (1976) 1405.

[14] B. Alessandro et al., Nucl. Instr. Meth. A360 (1995) 189-192.

[15] A. Baldisseri, Thèse. Université de Savoie, LAPP Annecy (1990).

[16] C. Charlot, Thèse de l'Université Louis Pasteur, CRN Starsbourg (1988).

[17] B. Ronceux, Thèse. Université de Savoie, LAPP Annecy (1995).

[18] W. Dabrowsky et al., Nucl. Instr. Meth. A350 (1994) 548.

[19] J. DeWitt, IEEE Nuclear Science Symposium, San Francisco, 30 Oct - 6 Nov 1993.

[20] M. Forlen et G. Moynot Doc. L.A.P.P NA38, La MEMRMH mémoire double port RMH-VME, 1992

[21] M. Forlen, Doc. L.A.P.P NA50, La carte TRANSIBELIEN : conversions de liens TTL/Différentiel, 1994

[22] J.B. LINDSAY, Nuclear Instr. and Meth. 156 (1978) 329.

[23] E. Hirsh "Les transputers : Application à la programmation concurrente" ed. Eyrolles 1990.

[24] B. Humbert. Cours sur les bus VME et VXI.

[25] Thèse A. Masserot, Université de Savoie, LAPP, 1995.

[26] Thèse F. Staley, Université de Savoie, LAPP, 1989.

[27] R. Stoeckel, Communication et Unix. Ed. Armand Colin 1991.

[28] INMOS, OCCAM 2 Reference Manual, C.A.R Hoare, Series Editor 1988.

[29] D. Poutain et D. May, A tutorial introduction to OCCAM programming, ed. BSP.

[30] INMOS, The Transputer Databook SGS-THOMSON, 1991.

[31] INMOS, Transputer Instruction Set - A compiler Writer's Guide, 1991.

[32] INMOS, The Transputer Applications Notebook : 'Systems and Performance', SGS-THOMSON, 1991.

[33] INMOS, The Transputer Applications Notebook : 'Architecture and Software' SGS-THOMSON, 1991.

[34] INMOS, The Transputer development and iq systems databook SGS-THOMSON, 1991.

[35] INMOS, The IMS B016 Support Librairies, SGS-THOMSON, 1991.

[36] J.Clavier, G.Coffinet, Théorie et technique de la transmission de données, MASSON, 1984.

[37] ALEX-TITAN-1S Communications board - Technical Manual for SUN SBus.

[38] Analyse des divers parallélismes du transputer. Stéphane GAY Stéphane VIALLE *La lettre du transputer et des calulateurs distribués 15.*