



Apport d'un système modulaire de séquenceurs microprogrammables à l'acquisition et au filtrage des données en physique des particules

Jacques Lecoq

► To cite this version:

Jacques Lecoq. Apport d'un système modulaire de séquenceurs microprogrammables à l'acquisition et au filtrage des données en physique des particules. Physique [physics]. Université de Haute Alsace - Mulhouse, 1982. Français. NNT : . tel-00395840

HAL Id: tel-00395840

<https://theses.hal.science/tel-00395840>

Submitted on 16 Jun 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



C.R.N.

13 DEC. 1983

CRN/HE 83-01

THESE*présentée**pour obtenir le grade de***DOCTEUR ÈS-SCIENCES PHYSIQUES***par***Jacques LECOQ**

APPORT D'UN SYSTEME MODULAIRE DE SEQUENCEURS MICROPROGRAMMABLES
 AL'ACQUISITION ET AU FILTRAGE DES DONNEES EN PHYSIQUE DES PARTICULES



Institut National
 de Physique Nucléaire
 et de Physique
 des Particules

Université
 Louis Pasteur
 de Strasbourg

Thesis-1983-Lecoq

Centre de recherches nucléaires de Strasbourg

CM-P00068722



CERN LIBRARIES, GENEVA

CERN LIBRARIES, GENEVA

THESE

CRN/HE 83-01

presentee

**A LA FACULTE DES SCIENCES DE L'UNIVERSITE
DE HAUTE ALSACE DE MULHOUSE**

pour obtenir le grade de

DOCTEUR ES-SCIENCES PHYSIQUES

par

Jacques LECOQ

**APPORT D'UN SYSTEME MODULAIRE DE SEQUENCEURS MICROPROGRAMMABLES
AL'ACQUISITION ET AU FILTRAGE DES DONNEES EN PHYSIQUE DES PARTICULES**

Soutenue le 22 Septembre 1982 devant la Commission d'Examen

MM. G.METZGER

M.AUBRUN

J.BLAISING

J.M.MEYER

R.MORAND

C.VERKERK

Président

Examineurs

LISTE DES PROFESSEURS, MAITRES DE CONFÉRENCES DIRECTEURS ET MAITRES DE RECHERCHE CNRS ET INSERM

Président	Professeur	F.MARCOUX
Vice-Présidents	Professeur	C.GODBILLON
	Professeur	H.DURANTON
Présidents honoraires	Professeurs	G.OURISSON - P.KARLI
Secrétaire Général	Monsieur	G.KIEHL

U.E.R. DES SCIENCES MÉDICALES

U.E.R. des Sciences Médicales	:	Directeur	Marc DORNER
U.E.R. des Sciences Biomédicales	:	Directeur	Jean SCHWARTZ

Doyens honoraires : J.CALLOT - J.CLAVERTE - F.ISCH.

Professeurs honoraires : A.BASSET - J.BENOIT - J.CALLOT - J.CLAVERTE - E.FORSTER - G.GREINER - Ch.GROS - A.JUNG - P.MANDEL - H.METZGER - P.MULLER - A.ROHMER - E.SCHNEEGANS - J.SEROR - J.STAHL - J.VEDRINE - P.VINTEMBERGER - J.WARTER - G.WINCKLER.

Professeurs :

M.ADLOFF	Chirurgie générale	E.GROSSHANS	Clin.Dermatol.et Syphiligraphique	J.MEHL	Médecine du travail
N.APROSIO	Anatomie et Organogénèse	P.HABEREY	Physiologie	G.METHLIN	Physique biologique
CI.ARON	Histologie	J.HERAN	Méd.et Chir.Expérim.et Comparée	B.METZ	Physiologie appliquée
L.ASCH	Rhumatologie	L.HOLLENDER	Chir.Digestive et Générale	R.MINCK	Bactériol.Virol.Immunol.gén.
A.BATZSCHLAGER	Anatomie Pathologique	J.LIMBS	Pharmacologie	G.MORAND	Chirurgie thoracique
P.BEYER	Clin.de Pédiatrie et Puériculture	M.IMLER	Médecine Interne	F.OBERLING	Hématol.Maladies du sang
P.BLOCH	Radiologie	Fr.ISCH	Rééducation fonctionnelle	E.PHILIPPE (dét.)	Anatomie pathologique
R.BLOCH	Pharmacologie	L.ISRAEL	Psychiatrie d'Adultes	R.RENAUD	Gynécologie et Obstétrique
B.ROCKEL	Hépatologie, Gastro-entérologie	H.JAHN	Néphrologie	P.REVILLE	Endocrinol.Métabol.et Nutrit.
C.BOLLACK	Urologie	J.JUIF	Pédiatrie et Puériculture	Ph.REYS	Chir.Gén.Hôp.L.Pasteur Colmar
A.BRINI	Ophthalmologie	Th.KAMMERER	Clin.Psychiatrie	E.ROEGEL	Pneumologie Phthisiologie
A.BRONNER	Clin. Ophthalmologique	P.KARLI	Neurophysiologie	F.ROHMER	Clin.Neurologie
F.BUCHHEIT	Neurochirurgie	B.KELLER	Clin.Gynécol.et Obstétricale	J.V.RUCH	Biologie médicale
P.BUCK	Clin.Chirurgicale des Enfants	F.KEMPF	Radiologie	Y.RUMPLER	Embryol.et Morphol.expérim.
G.BURGHARD	Clin.et Prophylaxie de la Tuberc.	L.KEMPF	Orthopédie et Traumatologie	A.SACREZ	Cardiologie
P.CHAMBRON	Biochimie	T.T.KIEN	Parasitologie	G.SAVA	Chirurgie générale
M.CHAMPY	Physique biologique	R.KIENY	Pathol.et Clin.Sémiol.Chir.	G.SCHAFF	Physiologie
A.CHAUMONT	Stomatologie	A.KIRN	Virologie	E.SCHVINGT	Clin.Chir.Orthop.et Traum.Adul.
M.COLLARD	Médecine légale et Méd.Sociale	J.G.KORITKE	Anatomie Normale	J.SCHWARTZ	Pharmacol.et Méd.Expérim.
C.CONRAUX	Neurologie	M.KREMER	Parasitol.et Pathol.tropicale	A.SIBILLY	Clinique Chirurgicale A
P.DELLENBACH	Oto-Rhino-Laryngologie	D.KURTZ	Neurologie	L.SINGER	Psychiatrie d'Adultes
M.DORNER	Gynécol.et Obstétrique	G.LANG	Orthop.et Traumatologie	F.STEPHAN	Pathol.Gén.et Expérimentale
R.EBTINGER	Clin. Médicale B	J.LAVILLAUREIX	Hygiène	D.STORCK	Clinique médicale A
M.FABRE	Psychiatrie infantile	Y.LEGAL	Anatomie pathologique	G.VINCENDON	Biochimie
L.FINCKER	Histologie	J.M.LEVY	Pédiatrie, Génétique médicale	R.VOEGTLIN	Thérapeutique
R.GANDAR	Clin. Médicale B	J.M.MANTZ	Réanimation Médicale	A.WACKENHEIM	Radiologie
P.GAUTHIER-LAFAYE	Clin.Gynécol.et Obstétricale	F.MARCOUX	Médecine du travail	P.WARTER	Gastro-Entérol.et Hydrol.Thérap.
J.P.GERZARD	Anesthésiologie	J.MARESCAUX	Histologie	D.WILLARD	Pédiatrie, Génétique Médicale
J.GRENIER	Ophthalmologie	Ch.MARX	Physiologie	J.P.WITZ	Chirurgie thoracique
	Chirurgie générale	S.MAYER	Hématologie		

Professeur associé : G.VAN ALPHEN - Ophthalmologie.

Professeur conventionné : E.WEIL - Toxicologie industrielle.

Maitres de Conférences Agrégés :

S.BABIN	Orthopédie et Traumatologie	P.KEHR	Orthopédie et Traumatologie	J.C.OTTENI	Anesthésiologie
P.BAREISS	Médecine interne	R.KEILING	Cancérologie	G.PAULI	Pneumologie Phthisiologie
P.BOURJAT	Radiologie	J.KEMPF	Biochimie	J.RITTER	Gynécol.et Obstétrique
C.BRECHENMACHER	Cardiologie	G.KLOTZ	Oto-Rhino-Laryngologie	M.ROOS	Histologie
J.M.BROGARD	Médecine interne	F.KUNTZMANN	Médecine interne	P.SAUVAGE	Chirurgie infantile
A.CONSTANTINESCO	Physique biologique	J.M.LANG	Maladies du sang	J.P.SCHIEBER	Physiologie
J.P.DUPEYRON	Anesthésiologie	D.MAITROT	Neurochirurgie	G.SCHLAEDER	Gynécologie et Obstétrique
B.EISENMANN	Chirurgie cardio-vasculaire	J.L.MANDEL	Biochimie	J.L.SCHLIENGER	Médecine interne
J.FLAMBERT	Ophthalmologie	J.MARESCAUX	Chirurgie générale	H.SICK	Anatomie et Organogénèse
J.GEISERT	Pédiatrie, Génétique médicale	J.MARK	Biochimie	C.STOLL	Pédiatrie, Génétique médicale
P.GERLINGER	Embryologie	J.MESSER	Pédiatrie, Génétique médicale	J.D.TEMPE	Réanimation médicale
E.HID	Dermato-vénéréologie	C.MEYER	Chirurgie générale	J.TONGIO	Radiologie
D.JAECK	Chirurgie générale	H.MONTEIL	Bactériologie	J.P.WALTER	Neurologie
A.JAEGER	Réanimation médicale	G.OBERT	Virologie	J.M.WARTER	Stomatologie
M.JESEL	Rééducation fonctionnelle			A.WILK	

Directeur de Recherche : A.PETROVIC* - Physiologie — M.JACOB* - Biochimie

Maitres de Recherche :

M.GAUTHIERIE+	Thermologie biomédicale	G.REBEL+	Neurochimie	N.VIRMAUX-COLIN+	Neurochimie
G.GOMBOS+	Neurochimie	R.RECHENMANN*	Biophys.des Rayonnements	J.J.VOGT+	Thermophysologie
K.HAFFEN-STENGER+	Endocrinologie	M.SENSENBRENNER+	Neurochimie	A.WAKSMANN+	Neurochimie
G.LECLERC+	Chimie organique	J.VELLY+	Pharmacologie		

U.E.R. D'ODONTOLOGIE

Directeur **Robert FRANK**

Professeurs :

M.DOCQ	Dentisterie operatoire	P.KLEWANSKY	Parodontologie
R.FRANK	Biologie et Matière fondamentale	J.LITZLER	Prothèse

Professeurs de premier grade :

E.ALLEMANN	Dentisterie operatoire	R.HAAG	Pathol.et Therap.dentaires	P.NICOLAS	Pathol.et Therap.dentaires
M.BASTIAN	Prothèse	J.L.LACOSTE	Orthopedie dento-faciale	J.J.ROTH	Parodontologie
C.BOLENDER	Orthopedie dento-faciale	M.LANGER	Prothèse	A.SCHLIENGER	Prothèse
A.COMTE	Dentisterie operatoire	M.LEIZE	Prothèse	J.SOMMERMATER	Parodontologie

Professeurs de deuxième grade :

W.BACON	Orthopedie dento-faciale	B.KAESS	Pathol.et Therap.dentaires
P.CAHEN	Biologie et Matière fondamentale	H.TENENBAUM	Parodontologie
J.P.CHARLIER	Orthopedie dento-faciale		

U.E.R. DES SCIENCES PHARMACEUTIQUES

Directeur **Pierre MÉTAIS**

Doyens honoraires : P.DUQUENOIS — N.HASSELMANN — G.DIRHEIMER.

Professeurs honoraires : P.CORDIER — J.P.EBEL — G.GAZET du CHATELIER — P.JAEGER.

Professeurs :

R.ANTON	Pharmacognosie	J.C.KOFFEL	Pharmacie chimique	B.PESSON	Parasitologie
R.CARBIENER	Botanique	H.LAMI	Mathématiques	Ph.POINDRON	Virologie
G.DIRHEIMER	Toxicologie	Y.LANDRY	Pharmacologie	J.SCHREIBER	Chimie organique
G.FERRARD	Chimie biologique	C.LAPP	Chimie générale et minérale	A.STAHL	Biochimie pharmaceutique
A.GAIRARD	Physiologie	P.LAUGEL	Chimie analytique	A.STAMM	Pharmacie galénique
D.GERARD	Physique et Biophysique	G.LAUSTRIAT	Physique	J.C.STOCLET	Pharmacodynamie
M.HASSELMANN	Chim.Anal.et Bromatol.	A.LUGNIER	Toxicologie	D.VIDON	Bactériologie
C.HASSELMANN(dét.)	Chimie Analytique	J.MALGRAS	Immunologie	C.G.WERMUTH	Chimie organique
L.JUNG	Pharmacie chimique	C.MATHIS	Pharmacie galénique		
		P.METAIS	Biochimie		

Professeur associé : A.WEINREB - Pharm.biophys.

Professeur conventionné : B.ROTH-SCHECHTER - Pharmacodynamie

Maître de Recherche : I.N.S.E.R.M. : J.BIETH - Enzymologie

U.E.R. DE SCIENCES HUMAINES

U.E.R. de Géographie

Directeur **Pierre MICHEL**

U.E.R. des Sciences du Comportement et de l'Environnement

Directeur **Bruno WILL**

Professeurs honoraires : Et.JUILLARD — R.RAYNAL.

Professeurs :

J.M.AVENARD	Géographie	P.MICHEL	Géographie	R.SCHWAB	Géographie
L.LEGRAND	Sciences de l'Education	A.MOLES	Psychologie sociale	A.TABOURET-KELLER	Psychologie
A.LIEURY	Psychol. experim.	H.NONN	Géographie	M.TARDY	Psycho-pédagogie
P.LIMOZIN	Géographie	H.REYMOND	Géographie	J.TRICART	Géographie

Directeur de Recherche : C.N.R.S. : S.RIMBERT - Géographie

U.E.R. DES SCIENCES ÉCONOMIQUES

Directeur **Patrick COHENDET**

Doyens honoraires : P.CHAMLEY - J.P.FITOUSSI - R.DOS SANTOS FERREIRA - J.L.GAFFARD.

Professeurs :

Ph.ARTZNER	Mathématiques	P.CHAMLEY	Sciences Economiques	LAGERARD-VARET	Sciences Economiques
F.BILGER	Sciences Economiques	R.DOS SANTOS FERREIRA	Sciences Economiques	G.KOENIG	Sciences Economiques
A.CHABERT	Sciences Economiques	J.P.FITOUSSI (dét.)	Sciences Economiques	J.J.OBRECHT	Gestion

Chargés de Conférences : R.ERBES — A.LOSSER

U.E.R. DES SCIENCES EXACTES

Mathématiques
Sciences Physiques et Chimiques
Sciences de la Matière
Sciences de la Vie et de la Terre
Sciences du Comportement et de l'Environnement
Ecole d'Application des Hauts Polymères
Ecole Nationale Supérieure de Chimie
Observatoire
Physique du Globe
Ecole Nationale Supérieure de Physique

Directeur **Gérard SCHIFFMANN**
 Directeur **Henri BENOIT**
 Directeur **Jean José FRIED**
 Directeur **Geneviève LEBEURIER**
 Directeur **Bruno WILL**
 Directeur **Morand LAMBLA**
 Directeur **Marc DAIRE**
 Directeur **Alphonse FLORSCH**
 Directeur **Roland SCHLICH**
 Directeur **Gilbert SUTTER**

Doyens honoraires : P.LACROUTE - J.H.VIVIEN - G.MILLOT.

Professeurs honoraires : J.BRENET - Mme M.BRINI - J.BYE - H.CARTAN - C.CHABAUTY - A.CHRETIEN - J.DENY - Mlle A.GAGNIEU - Mlle S.GILLET - S.GORODETZKY - R.HOCART - P.JOLY - P.LACROUTE - R.LECOLAZET - G.LEMEE - P.L'HERITIER - A.LICHNEROWICZ - A.MAILLARD - G.MILLOT - L.NEEL - J.PARROD - A.ROCHE - R.ROHMER - J.P.ROTHE - L.SACKMANN - Ch.SADRON - H.SAUCIER - F.STUTINSKY - H.VILLAT - J.H.VIVIEN - Et.WOLFF.

Maître de conférence honoraire : R.WEIL.

Professeurs

J.P.ADLOFF
 R.ARMBRUSTER
 P.ARTZNER
 V.AVANNISSIAN
 G.BARBANÇON
 F.BECKER
 N.BEFORE
 G.BELLIARD
 C.BENEZRA
 H.BENOIT
 P.BENVENISTE
 D.BERNARD
 J.C.BERNIER
 J.BONNIN
 Y.BOULANGER
 J.F.BOUTOT
 J.BROSSAS
 R.BROUILLARD
 C.BURGGRAB
 H.BURNAGE
 M.CARA
 R.CERF
 P.CHARTIER
 P.CHEVALLIER
 A.CLAUSS
 A.COCHÉ
 M.DAIRE
 H.DANAN
 E.DANIEL
 M.DAUNE
 J.DEHAND
 A.DELUZARCHE
 J.DEMYUNCK
 J.F.DUFORD
 G.DUNOYER de
 SEGONZAC
 H.DURANTON
 J.P.EBEL
 J.P.EBERHART
 B.ERESMANN
 V.ERN
 J.FARAUT
 P.FEDERLIN
 P.FELTZ

Chimie nucléaire
 Physique
 Mathématiques
 Analyse supérieure
 Mathématiques
 Physique mathématique
 Biochimie
 Botanique
 Dermato-Chimie
 Physicochim.macrochimol.
 Physiologie végétale
 Méth.math.de la physique
 Chimie générale
 Géophysique interne
 Biochimie
 Mathématiques
 Chimie macromoléculaire
 Chimie org.et Chimie analyt
 Minéralogie
 Mécanique des Fluides
 Géophysique interne
 Physique générale
 Chimie
 Physique
 Chimie
 Physique nucléaire
 Chim.phys.indus.et Sc.des Matér
 Phys.Atom.et Phys.du Solide
 Physique expérimentale
 Biophysique
 Chimie générale
 Chimie
 Chimie organique
 Informat.fondam.et appl. (IUT)
 Géologie
 Botanique
 Biochimie
 Minéralogie
 Biochimie
 Physique
 Mathématiques
 Chimie
 Physiologie animale

X.FERNIQUE
 M.FINK
 J.G.FISCHER
 D.FOATA
 E.FOLLENIUS
 J.J.FRIED
 D.FROELICH
 A.FUCHS
 J.C.GALL
 A.GALLMANN
 F.GAUTIER
 R.GERARD
 G.GLAESER
 C.GODBILLON
 A.GOLTZENE

M.GOUNOT
 M.GROSMANN
 M.GROSS
 Y.GUEGUEN
 L.HIRTH
 C.JASCHEK
 G.JONARD
 J.P.JOUANOLOU
 F.JUNDT
 T.JUTEAU
 C.KEDINGER
 A.KIENEMANN
 R.KIRSCH
 F.LACROUTE
 J.C.LAFON
 A.M.LAMBERT
 M.LAMBLA
 G.LEBEURIER
 J.L.LEIBENGUTH
 J.LEITE-LOPES
 P.LEMOINE
 M.LEROY
 J.C.LIONS
 F.LOOR
 J.LUCAS
 D.MAGNAC
 J.MARTINET
 P.MIALHE
 A.MICHARD

Mathématiques
 Acoustique moléculaire
 Chimie
 Mathématiques
 Zoologie
 Mécanique des Fluides
 Chim.gén.Chim.physique
 Mécanique rationnelle
 Géologie
 Physique
 Physique
 Mathématiques
 Mathématiques
 Mathématiques
 Optique, phys.atom.et mol.
 phys.du sol.et cristallographie
 Botanique
 Physique
 Chimie physique
 Géophysique interne
 Microbiologie
 Astronomie
 Virologie
 Mathématiques
 Phys.nucl.et corp.et théor.phys
 Minéralogie
 Biochimie
 Chimie appl.et Génie chim
 Zoologie
 Biologie végétale
 Informatique appliquée
 Biologie
 Chimie générale
 Microbiologie
 Chim.appl.et Chim.des matér
 Phys.nucl.et corpusculaire
 Chimie
 Chimie
 Zoologie et Ecologie
 Immunologie
 Géologie
 Physique
 Mathématiques
 Physiologie animale
 Géologie

M.MIGNOTTE
 P.MIRABEL
 G.MONSONEGO
 B.MORIN
 P.NANOPOULOS
 J.OSBORN
 E.OSTERTAG
 G.OURISSON (dét.)
 J.M.PAULUS
 J.P.RAMIS
 G.REEB
 Ph.RICHARD
 J.J.RIEHL
 P.RIMMELIN
 Cl.ROBERT
 Ph.ROPARTZ
 J.ROUX
 J.P.SAUVAGE
 F.SCHALLER
 G.SCHIFFMANN
 A.SCHMITT
 P.SCHMITT
 J.P.SCHWING
 M.J.SCHWING
 J.C.SENS
 M.SIESKIND
 G.SOLLADIE
 J.SOMMER
 G.SUTTER
 Ch.TANIELIAN
 Y.TARDY
 J.TERRISSE
 J.J.THIEBOLD
 D.VIAUD
 R.VOLTZ
 J.H.WEIL
 G.WEILL
 R.WEISS
 P.L.WENDEL
 B.WILL
 C.WINTER
 C.WIPPLER
 J.WUCHER
 B.WURTZ

Informatique
 Chimie
 Physique théorique
 Mathématiques I
 Probabilités et statistiques
 Chimie
 Electr. et Electrotechnique
 Chimie
 Chimie générale
 Mathématiques générales
 Topologie
 Physiologie animale
 Chimie
 Chimie (IUT)
 Physique
 Psycho-Physiologie
 Botanique
 Chimie théor.et Chim.phys (IUT)
 Biologie générale
 Mathématiques
 Physique
 Psycho-physiol.et physiol.du comp
 Chimie
 Chimie physique
 Phys.nucl.et corp.et théorie phys
 Physique
 Chimie organique
 Chimie appliquée
 Physique électronique
 Chimie (IUT)
 Géologie
 Chimie
 Biologie animale
 Mathématiques
 Physique théorique
 Biochimie
 Physique
 Chimie
 Psycho-Physiologie
 Chimie
 Physicochim.des Hts Polymères
 Physique
 Biochimie

Professeur émérite : G.MILLOT - Géologie Paléontologie.

Professeur adjoint : J.SITTler - Géologie

P.BOUVEROT - Physiologie respiratoire - P.DEJOURS - Physiologie respiratoire.

Professeurs Associés

M.R.CHURCHILL
 B.J.COLE
 M.E.CONSTANTIN

Chimie
 Physique
 Chimie

E.CROUSE
 H.DUCAS
 C.PIERPONT

Biochimie
 Chimie
 Chimie

A.SIDOANS
 R.WEISS

Géologie
 Chimie

Professeurs conventionnés : P.BEY Chimie - P.BOUVEROT Physiol.respirat. - P.DEJOURS Physiol.respirat. - F.FARKAS Phys. - Y.NAKATANI Chimie

Astronomes adjoints : A.FLORSCH - A.FRESNEAU

Astronome adjoint associé : M.JASCHEK - Astronomie

Physiciens adjoints (Physique du Globe) : P.HOANG TRONG - Géophysique int. - R.MONTIGNY - Géophysique int. - E.PETERSCHMITT - Géophysique int.

Directeurs de Recherche C.N.R.S. :

P.ALBRECHT
 J.F.BIELLMANN
 P.BOUVEROT
 P.DEJOURS
 M.FRANCK-NEUMANN
 A.KNIPPER

Chimie
 Chimie
 Physiologie respiratoire
 Physiologie respiratoire
 Chimie organique
 Phys.nucl.et corpusculaire

A.KOVACS
 J.MARCHAL
 P.A.MEYER
 A.J.MEYER
 A.PORTE
 P.REMPF
 R.SCHLICH

Physicochim.macrochimol.
 Physicochim.macrochimol.
 Mathématiques
 Physique
 Biologie cellulaire
 Physicochim.macrochimol.
 Géophysique marine

A.SKOULIOS
 M.VAN REGENMORTEL
 R.VAROQUI
 A.VELLARD
 R.ZANA
 A.ZUKER

Physicochim.macrochimol.
 Virologie
 Physicochim.macrochimol.
 Chimie moléculaire
 Physicochim.macrochimol.
 Physique théorique

Maîtres de Recherche C.N.R.S. :

J.Ch.ABBE
 E.ASLANIDES
 F.BECK
 G.BECK
 J.P.BECK
 R.BERTINI
 M.BONHOMME
 H.BRAUN
 P.BRAUNSTEIN
 M.C.CADEVILLE
 H.CALLOT
 S.CANDAU
 M.CHAMPAGNE
 J.CHEVALLIER
 J.P.COFFIN
 A.CORET
 M.CROISSIAUX
 D.DISDIER
 J.DOUBINGER
 F.DURST
 S.EL KOMOSS
 B.FRANÇOIS
 E.FRANTZ
 J.M.FRIEDT
 B.FRITIG
 Y.GALLOT
 J.P.GERBER
 R.GIEGE
 Ph.GRAMAIN
 J.B.GRUN

Physicochim.des interactions et des interfaces
 Phys.nucl.et corpusculaire
 Phys.nucl.et corpusculaire
 Biochimie
 Physiologie
 Physique nucléaire
 Géologie
 Physique nucléaire
 Chimie
 Physique des Solides
 Chimie
 Physique
 Biophysique
 Physique nucléaire
 Phys.nucl.et corpusculaire
 Phys.nucl.et corpusculaire
 Géologie
 Physiologie végétale
 Physique
 Physicochim.macrochimol.
 Physicochim.macrochimol.
 Physicochim.macrochimol.
 Virologie
 Physicochim.macrochimol.
 Phys.nucl.et corpusculaire
 Biochimie
 Physicochim.macrochimol.
 Physique

J.HERZ
 J.HOFFMANN
 G.JENNER
 G.KAUFMANN
 G.KEITH
 J.P.KINTZINGER
 B.KOCH
 E.KOCHANSKI
 B.LANG
 J.LANG
 P.LAURENT
 A.LEJEUNE
 C.LERAY
 F.LEYENDECKER
 B.LOTZ
 B.LUU
 G.MAIRE
 A.MALAN
 E.MARCHAL
 R.MORAND
 D.MORAS
 Th.MULLER
 G.MUNSCHE
 M.AUCIEL-BLOCH
 A.NICOLAIEFF
 H.PAQUET
 M.PATY
 Cl.PICOT
 L.PINCK

Physicochim.macrochimol.
 Biologie animale
 Chimie
 Chimie
 Biochimie
 Chimie
 Physiologie
 Struct.et dynam.moléc.
 chimie de coordination
 Cristallographie
 Physicochim.macrochimol.
 Physiol.comparée des régul
 Phys.nucléaire théorique
 Physiol.comparée des régul.
 Chimie
 Physicochim.macrochimol.
 Chimie organique
 Chimie
 Physiologie respiratoire
 Physicochim.mol.et macromol
 Phys.nucl.et corpusculaire
 Chimie
 Phys.nucl.et corpusculaire
 Physique
 Physique des Solides
 Virologie végétale
 Géologie
 Phys.nucl.et corpusculaire
 Physicochim.macrochimol.
 Biologie cellulaire

P.POIX
 J.POUYET
 B.REES
 P.REMY
 J.RINGEISEN
 J.P.ROTH (s.Dir
 Inst.de rech.)
 R.SCHANTZ
 F.SCHEIBLING
 G.SCHUBERT
 N.SCHULZ
 C.SCHWAB
 R.SELTZ
 P.SIFFERT
 Cl.SITTler
 M.E.STOECKEL
 Cl.STRAZIELLE
 M.SUFFERT
 J.C.THIERRY
 K.TRAORE
 P.WAGNER
 G.WALTER
 Fr.WEBER
 J.P.WENIGER
 J.WITZ
 R.WOLFF
 J.P.ZIELINGER
 L.ZILLIOX

Chimie
 Biophysique
 Chimie
 Biochimie
 Physique
 Physicochim.macrochimol.
 Physiologie végétale
 Phys.nucl.et corpusculaire
 Chimie organique
 Phys.nucl.et corpusculaire
 Physique
 Phys.nucl.et corpusculaire
 Phys.nucl.et corpusculaire
 Géologie
 Biol.des interactions cellulaires
 Physicochim.macrochimol.
 Phys.nucl.et corpusculaire
 Chimie
 Physicochim.mol.et ionique
 Phys.nucl.et corpusculaire
 Phys.nucl.et corpusculaire
 Géologie
 Zoologie
 Biologie cellulaire
 Chimie
 Physique
 Mécanique des Fluides

<u>Président</u>	: JAEGLE Alain
<u>Vice-Présidents</u>	
<u>enseignants</u>	: KILLE Gérard - RIBSTEIN Max
<u>Secrétaire Général</u>	: SCHUB Bernard

PROFESSEURS, MAITRES DE RECHERCHE
RESPONSABLES DES DOCTORATS ES SCIENCES

Professeurs

WEY Raymond	Chimie	E.N.S.C.Mu
DONNET Jean-Baptiste	Chimie Physique	E.N.S.C.Mu
PERNY Guy	Physique Appliquée	I.U.T.
STREITH Jacques	Chimie	E.N.S.C.Mu
METZGER Gérard	Physique (Electronique)	I.S.E.A.
SCHMITT François	Physique (Electronique)	I.S.E.A.
GRESSER Julien	Electronique	I.S.E.A.
MANQUENOUILLE Robert	Physique Appliquée	I.U.T.
MEYER Jean-Marie	Electronique, Electrotechnique, Automatique	I.U.T.
STEIN Roland	Physique (Electronique)	I.U.T.
JAEGLE Alain	Physique	I.S.E.A.
LUTZ Robert	Mathématiques	I.S.E.A.
KALT André	Chimie Physique	I.S.E.A.
RAUCH Gérard	Mathématiques	I.S.E.A.
RIEDINGER Roland	Physique	I.S.E.A.
ROTH Jean-Pierre	Mathématiques	I.S.E.A.
SCHULTZ Jacques	Chimie	I.S.E.A.
SPEHNER Jean-Claude	Informatique Appliquée	I.S.E.A.
CHEZEAU Jean-Michel	Chimie Physique	I.S.E.A.
FOUASSIER Jean-Pierre	Chimie	E.N.S.C.Mu
WOLFF Claude	Mécanique	E.N.S.I.T.
HAMMANN Théophile	Physique	I.S.E.A.
ROHMER Michel	Chimie	E.N.S.C.Mu
LE DRED Ronan	Chimie	I.S.E.A.
HANGAN Théodore	Mathématiques	I.S.E.A.
FRANCON Jean	Mathématiques	I.S.E.A.
DERRIEN Jacques	Physique	I.S.E.A.
SCHULTZ Guy	Physique (Electronique)	I.U.T.
PRADO Gilles	Thermodynamique, Energétique (Combustion)	I.S.E.A.
SIGLI Danièle	Mécanique	E.N.S.I.T.
FLEURY Jean-Pierre	Chimie Organique	E.N.S.C.Mu
HATTERER André	Chimie Physique	E.N.S.C.Mu
KIRSCHNER Auguste	Génie Mécanique	E.N.S.I.T.
RIESS Gérard	Chimie Macromoléculaire	E.N.S.C.Mu

Directeurs et Maîtres de recherche

LAHAYE Jacques	Physico-Chimie des interactions et des interfaces	C.N.R.S.
PAPIRER Eugène	Physico-Chimie des polymères et des molécules biologiques	C.N.R.S.
SIFFERT Bernard	Physico-Chimie des interactions et des interfaces	C.N.R.S.
DECKER Christian	Physico-Chimie des polymères	C.N.R.S.
EHRBURGER Pierre	Physico-Chimie des interactions et des interfaces	C.N.R.S.
KESSLER Henri	Chimie Minérale	C.N.R.S.

Personnalités scientifiques

FREYTAG René	Chimie Macromoléculaire	E.N.S.C.Mu
REBE Jean-François	Sciences Economiques	I.U.T.
SCHUTZ Richard Adrien	Chimie Macromoléculaire	E.N.S.I.T.M.

PROFESSEURS RESPONSABLES DES DOCTORATS ES LETTRES

THILL Andrée	Latin	F.L.S.H.
GIDEL Henry	Français	F.L.S.H.
MOUREAU François	Littérature française	F.L.S.H.
BAUDEMONT Suzanne	Anglais	F.L.S.H.
EICHENLAUB René	Allemand	F.L.S.H.
WACKERMANN Gabriel	Géographie	F.L.S.H.
CLAUDON Francis	Français	F.L.S.H.
GLASER Rosette	Anglais	F.L.S.H.
GUYAUX André	Français	F.L.S.H.

E.N.S.C.Mu	: Ecole Nationale Supérieure de Chimie de Mulhouse
E.N.S.I.T.M.	: Ecole Nationale Supérieure des Industries Textiles Mulhouse
C.N.R.S.	: Centre National de la Recherche Scientifique
I.S.E.A.	: Institut des Sciences Exactes et Appliquées
I.U.T.	: Institut Universitaire de Technologie
F.L.S.H.	: Faculté des Lettres et Sciences Humaines
C.R.T.M.	: Centre de Recherches Textiles de Mulhouse

TABLE DES MATIERES

	PAGES
INTRODUCTION	1
CHAPITRE I	3
I.1. Définition des problèmes informatiques rencontrés dans une expérience de physique des particules de hautes énergies	
I.2. Principaux aspects de l'acquisition de données	6
a. Gains en temps actif	8
b. gains en sélectivité	9
c. gains en contrôle	10
d. formattage	10
e. détection des erreurs matérielles	11
f. gains en temps de calcul différé	
g. souplesse d'adaptation.	
I.3. Contexte matériel des acquisitions de données	12
a. Les processeurs de lecture	13
b. Les ordinateurs de contrôle	
c. Filtrage d'événements et gestion mémoire	14
d. Conclusion	15
CHAPITRE II : RECHERCHE D'UNE STRUCTURE DE BASE-CONCEPTION DU NOYAU GESPRO	
II.1. Principe de base des processeurs microprogrammables	16
II.2. Organisation des mémoires de commande	18
a. Format	
b. Chaînage des micro-instructions	19
c. Opérateurs spécifiques et interfaces	21
II.3. Choix des structures et des composants	
a. Choix des composants	22
b. Etude de la structure du noyau GESPRO	23
i. Longueur du mot mémoire centrale	24
ii. Longueur du mot de la mémoire de commande	

II.4. Apport de la microprogrammation	26
II.5. Incidence de la microprogrammation sur l'utilisation.	29
CHAPITRE III. DESCRIPTION DU NOYAU GESPRO	30
III.1. Résumé des tâches spécifiques	30
a. Gestion DMA	
b. Gestion mémoire	31
c. Communication	35
d. Traitement des données	36
III.2. Description du matériel GESPRO	37
a. Unité centrale	
b. Description des différents champs	40
c. Unité de contrôle de la mémoire de commande	43
d. La mémoire centrale	46
e. Le module d'interruption et les périphériques standards.	
III.3. Le jeu d'instructions GESPRO	48
a. Choix du jeu d'instructions - partie banalisée	
b. Instructions spécialisées système	49
c. Instructions spécifiques à chaque expérience	50
III.4. Développement des outils logiciels associés à GESPRO	51
a. Logiciel propre à GESPRO	51
b. Logiciel croisé sur NORD	52
c. Logiciel interactif NORD-GESPRO	54
d. Logiciel d'acquisition GESPRO	55
CHAPITRE IV . DESCRIPTION DU NOYAU GESPRO	56
IV.1. Expérience WA2-1975-1979	
A. Description de l'expérience et de l'appareillage	
a) Le compteur DISC	
b) Les chambres faisceaux	
c) L'aimant	58
d) Le télescope avant	
e) Le télescope arrière	
f) Le mur de verre au plomb	
g) Le compteur Cerenkov	
h) Les chambres GAMMA	
i) Le compteur à neutrons	

B. Le déclenchement de l'expérience	59
C. Organisation du système de lecture	
a) Description	
b) Conséquences	61
D. Philosophie du système d'acquisition	
E. Configuration GESPRO dans sa version WA2	63
a) Le module d'entrée-sortie programmée	66
b) Le module d'interruption	67
c) Le shifteur câblé	69
d) Le contrôleur mémoire à huit canaux	
F. Mise en oeuvre de GESPRO	71
a) Acquisition	
b) Détection d'erreurs matérielles	72
c) Formattage de l'événement	74
d) Les rejets	
G. Résultats	77
a) Taux de microprogrammation	
b) Gain en temps mort	
c) Gain en efficacité	79
d) Gain en histogrammation	
e) Gain en temps d'analyse	
f) Gain en statistique d'erreurs	
H. Les faiblesses du système Hypéron	80
IV.2. Expérience NA10	82
A. Description de l'appareillage	82
1. La cible	
2. L'absorbeur	
3. Les hodoscopes	
4. L'aimant	
5. Les chambres proportionnelles	
B. 1. Le déclenchement	
2. La logique associée au déclenchement	84
C. Organisation du système de lecture	87
D. 1. Les simulations	92
2. Résultats des simulations	93

a) résultats généraux	
b) résultats spécifiques à la conception des calculateurs	
3. Conceptions de calculateurs	
E. Description détaillée des mémoires d'événements "EVB"	94
1. Description générale	
2. Les interfaces "GESPRO"	96
3. Séquences et dialogues	
F. Description de la configuration de GESPRO en version NA10	98
1. Généralités	
2. Les opérateurs câblés spécifiques à l'expérience NA10	100
a-1) Fonctionnement	
a-2) Réalisation	102
b) L'opérateur arithmétique rapide	103
b-1) Fonctionnement	
b-2) réalisation et interface	108
G. Mise en oeuvre des ensembles EVB-GESPROS	110
a. L'acquisition	
b. Déclenchement logiciel	111
c. Le filtrage	113
d. Les contrôles en ligne	116
H. Performance de l'ensemble EVB-GESPRO	117
a. Performances	
b. Résultats expérimentaux	119
c. Déclenchements spécialisés	
CONCLUSION	122
<u>ANNEXES</u>	125
Annexe au chapitre II	125
Micro-code de fonction (I3002) (document INTEL)	131
Micro-code de sauts (I3001) (document INTEL)	132
Jeu d'inscriptions GESPRO	133
Résumé d'une prise de donnée de l'expérience NA10.	139
REFERENCES	149
BIBLIOGRAPHIE	151
REMERCIEMENTS	153

Légende des Figures

- I.1. Evolution des paramètres des expériences de 1974 à 1982
- I.2. Principales étapes d'une acquisition de données
- II.1. Structure de base d'un ordinateur microprogrammé
- II.2. Format général d'une microinstruction GESPRO
- III.1. Structure pipe line
- III.2. Structure pipe line en F1-F0
- III.3. Structure WA2
- III.4. Structure idéale
- III.5. Architecture simplifiée de l'unité centrale
- III.6. Horloge déclenchée
- III.7. Format de la micro-instruction GESPRO
- III.8. Arbre de description du code opération.
- IV.1. Appareillage de l'expérience WA2
- IV.2. Organisation du système de lecture
- IV.3. Système informatique WA2
- IV.4. Configuration GESPRO Version WA2
- IV.5. Module de base des interruptions
- IV.6. Les ordres CAMAC du contrôleur mémoire
- IV.7. Table des erreurs trouvées par GESPRO
- IV.8. Résultats d'une prise de données Omega 1978
- IV.9. Traitement des convertisseurs temps-numérique (T.D.C)
- IV.10. Chronogramme de l'acquisition WA2
- IV.11. Gain en histogrammation
- IV.12. Appareillage NA10
- IV.13. Mise en temps des données de tous les capteurs, mémorisation et emplacement d'un calculateur.
- IV.14. 1.-IV.14.2. Système de déclenchement - PT encodés et PT regroupés
- IV.15. Variation du nombre de déclenchements en fonction de l'intensité et histogramme du nombre total de mots par événement sans filtrage.

- IV.16. Ensemble du système de lecture
- IV.17. Déclenchement de second niveau
- IV.18. Tableau des signaux de l'EVB
- IV.19. Diagramme des états du système
- IV.20. Chronogramme des opérations sur le bus interne
- IV.21. Principe de détermination du PT
- IV.22. Format et capacités arithmétiques
- IV.23. Multiplieur entier et flottant
- IV.24. et IV.25 : Conversions
- IV.26. Réduction au même exposant
- IV.27. Tables de chaînage du calcul de masse
- IV.28. Histogramme des temps d'exécution du déclenchement logiciel
- IV.29. Temps d'exécution des différentes coupures à partir de STA
- IV.30. Histogrammes montrant la précision de la masse calculée en ligne.
- IV.31. Histogrammes de la masse calculée en ligne pour des événements biaisés et non biaisés.

I N T R O D U C T I O N

On assiste actuellement à un développement sans précédent des applications des microprocesseurs, qui font irruption dans les domaines de la vie courante. Ce travail, dont les premières idées ont été lancées en 1974, concerne l'application des microprocesseurs à l'acquisition et au filtrage des événements mesurés dans les expériences de physique des hautes énergies. Il s'est traduit en 1976 par l'introduction du processeur GESPRO dans le système d'acquisition de l'expérience WA2 au SPS du CERN à Genève [1]. C'était alors le premier microprocesseur en tranches [2] en opération sur une expérience en cours de prise de données.

La fiabilité et la versalité obtenue nous ont amené à concevoir une deuxième génération dont la construction a débuté en 1979 pour l'expérience NA10. Actuellement, cette expérience est en cours de prise de données et le système que nous avons mis au point contribue efficacement à ses performances [3].

La première partie de ce mémoire décrit l'évolution des applications des microprocesseurs aux systèmes de filtrage et d'acquisition mis en oeuvre dans les expériences de physique des hautes énergies. Elle souligne les paramètres importants dans la partie automatique-informatique des expériences et précise les tâches demandées aux systèmes de mise en ligne.

La deuxième partie est consacrée à la recherche d'une structure matérielle et logicielle efficace, économique, souple et modulaire, capable de s'adapter à plusieurs expériences et d'évoluer en cours d'expérience.

La troisième partie décrit le noyau du processeur microprogrammable GESPRO. Elle précise les performances obtenues et montre l'évolution du système vers une architecture multi-processeurs à intelligence répartie, utilisant des modules standards. Elle souligne l'importance des outils de développement et de

tests nécessaires à la réalisation, la mise au point, la maintenance et l'utilisation des processeurs. Plus précisément il s'agit des programmes de test, de l'assembleur interactif permettant l'utilisation d'instructions évolutives, du compilateur de micro-code qui prend en charge le délicat problème de l'optimisation de l'implantation des microinstructions en mémoire de commande.

La dernière partie décrit les applications de cette étude aux expériences WA2 et NA10 réalisées auprès du SPS du CERN à Genève. En version WA2, GESPRO est dédié CAMAC multi-source (philosophie Elliot System Crate) et placé en frontal d'un miniordinateur NORD-10. Grâce à un temps d'exécution dix fois plus faible qu'un miniordinateur, il est en mesure de réaliser de façon performante l'acquisition et le filtrage des données. Ainsi l'unité centrale NORD peut assurer un contrôle plus efficace de l'expérience et le volume des données acquises est réduit.

La version NA10 comporte quatre processeurs GESPRO enrichis d'opérateurs câblés spécialisés associés à des mémoires externes rapides placées devant un couple NORD 100- NORD 500. Ce système fait partie intégrante du système de déclenchement et contribue directement à la sensibilité de l'expérience. Sa versalité a permis à l'expérience faire évoluer ses critères de sélection au fur et à mesure que la connaissance des phénomènes étudiés s'affinait.

A présent, plus de 15 modèles de microprocesseurs sont utilisés sur plus de 70 projets européen en physique des hautes énergies [4]. Ils sont souvent utilisés pour les contrôles de processus, les mesures et les tests, plus rarement dans le domaine du filtrage rapide en ligne ou dans le domaine du filtrage différé.

C H A P I T R E I

I.1. DEFINITION DES PROBLEMES INFORMATIQUES RENCONTRES DANS UNE EXPERIENCE DE PHYSIQUE DES HAUTES ENERGIES.

Les techniques visuelles (émulsions, chambres de Wilson, chambres à bulles) mises en oeuvre en physique des hautes énergies ont marqué les grandes étapes des progrès de nos connaissances dans ce domaine. La qualité et la quantité des informations recueillies sur une simple photographie ont permis l'analyse et la compréhension de phénomènes extrêmement complexes, mais rencontrés suffisamment fréquemment pour permettre ce type d'acquisition (sections efficaces $\geq$ microbarn).

Les techniques électroniques permettent de rechercher des phénomènes beaucoup plus rares (sections efficaces $\approx$ picobarn). La quantité et la qualité des informations recueillies par ces techniques étaient longtemps limitées par les performances et le coût des détecteurs. Ces facteurs approchent à présent ceux que l'on obtenaient par les techniques visuelles, tout en ayant des temps de réponse beaucoup plus courts, ce qui permet de rechercher des phénomènes de plus en plus rares.

En général une expérience électronique se caractérise par la mise en oeuvre de deux familles de capteurs qui diffèrent par leurs temps de réponse :

- Les premiers, définissant essentiellement la sensibilité de l'expérience. Ils ont un faible temps de réponse et permettent de réaliser un premier filtrage très rapide à l'aide d'une logique câblée. (Quelques centaines de nanosecondes sont suffisantes pour décider de conserver ou de rejeter un candidat).
- Les seconds visualisent les détails de l'événement grâce à leur précision et à l'abondance des informations qu'ils délivrent (chambres proportionnelles, chambres à dérives, calorimètres).

L'évolution des dix dernières années montre que le nombre et la complexité des capteurs électroniques mis en oeuvre sur une expérience croît très rapidement. La figure 1 montre cette évolution en faisant apparaître, au niveau volume mémoire équivalent capteur, une augmentation d'un facteur proche de 10 entre 1974 et 1981. Une évolution du même ordre est dès à présent prévisible pour l'expérimentation préparée en vue du futur accélérateur LEP. On conçoit qu'une telle évolution soit directement conditionnée par l'informatique :

En effet, celle-ci intervient de façon vitale à tous les niveaux de l'expérience :

- contrôle automatique des détecteurs
- sélection des événements
- acquisition et transfert de l'information mesurée sur bande magnétique
- exploitation hors faisceau en temps différé des informations stockées sur bandes magnétiques.

Les contrôles automatiques, l'acquisition et l'analyse hors faisceau sont des tâches qui relèvent des applications classiques de l'informatique industrielle, c'est à dire compatibles avec celles rencontrées dans d'autres domaines.

Il n'en est pas de même de la sélection des événements où la plupart des fonctions demandées sont spécifiques au domaine ou même à l'expérience considérée et où les temps d'exécution se doivent d'être aussi courts que possible. En faisant intervenir l'informatique au niveau de la sélection des événements, on apporte la souplesse du logiciel au point critique de l'expérience, ce qui permet, outre le fait de sélectionner les événements intéressants :

- d'adapter les flux de données aux capacités de transfert des systèmes d'acquisition
- de réduire le nombre de données à analyser et ainsi de ménager l'expansion des centres de calcul.

Par amélioration du rapport signal à bruit on obtient une double optimisation :

- optimisation du temps accélérateur
- optimisation du temps centre de calcul

Ceci conduit à un meilleur contrôle de l'expérience, aussi bien en ligne qu'en différé.

	HYPERON PS 1974	HYPERON SPS 1977 WA2	DIMUON SPS 1981 NA10
Déclenchement	Télescope faisceaux (1 plan) Cerenkov (DISC)	Télescope faisceaux 3 plans X,Y 1 plan U, V Cerenkov (DISC)	Quatre hodoscopes de six sextants chacun (32 compteurs à scintillation par sextant)
Données	Télescope avant (4 plans) Télescope arrière (1 plan X,Y) Chambres à dards	Télescope Avant(4 plans X,Y,U,V) Télescope arrière (4 plans X,Y,U,V) Cerenkov secondaire	Quatre chambres proportionnelles avant (3 plans de fils) Quatre chambres proportionnelles arrières (3 plans de fils)
Volume mémoire réparti au ni- veau détecteur	3600 bits	6000 bits	$2,2 \cdot 10^4$ bits
fréquence d'ac- tivation maxi- male	2 à 3 burst	200/burst	1000/burst
Logique de sé- lection (dé- clenchement) Sélection (filtrage)	câblée	câblée microprocesseur spécialisé	câblée combinatoire, programma- ble et microprogrammable multi-microprocesseurs spécialisés
puissance informatique	un mini-ordinateur (30 0000 op/s)	deux mini-ordinateurs un microprocesseur 1.400 000 op/s	- multi-mini-ordinateurs - multi-microprocesseur - opérateurs spécialisés $> 10^7$ op/s
mémoire	rapide 12K mots lente 256 K mots	rapide 120 K mots lente 5 M mots	ultra-rapide : 8 K mots rapide : 0,5 M mots lente : 60 M mots

Figure I.1. : Evolution des paramètres de quelques expériences de 1974 à 1982

I.2. APPORT DES PROCESSEURS DE FILTRAGE

La figure 2 montre les principales étapes de l'acquisition et du traitement des données.

L'information détectée dans les différents capteurs est stockée sur des lignes à retard (câbles) jusqu'à ce que la décision d'acquisition soit prise par la logique de sélection attachée aux capteurs. Cette décision ou "déclenchement" provoque la mémorisation des informations dans des registres et initialise le processus de transfert de ces registres vers l'ordinateur de contrôle qui se charge de la mise sur bande magnétique. Une liaison appropriée permet d'envoyer des lots d'événements vers le centre de calcul en vue d'une analyse complète et immédiate. En effet, compte tenu de toutes les charges de l'expérience d'une part, et des temps de calcul nécessaires à l'analyse d'autre part, l'ordinateur local n'a guère le temps que d'histogrammer les informations brutes.

Sachant qu'une expérience moyenne se traduit par l'accumulation de plusieurs dizaines de millions d'événements sur bande magnétique, on mesure l'importance du rôle de la logique de sélection et du processeur de filtrage. Ils déterminent le volume des données acquises. A ce titre c'est d'eux que dépendent et la consommation de temps faisceau et la charge de tous les processeurs situés en aval. Lorsque le processeur de filtrage agit au niveau de la logique de sélection (déclenchement logiciel), son temps de traitement étant nettement inférieur au temps de lecture, son action se traduit non seulement par une réduction des données mais encore par un gain en temps actif, c'est à dire en sensibilité de l'expérience. Ceci est vrai au niveau déclenchement, où ce gain se traduit par une meilleure utilisation du faisceau, mais aussi au niveau des ordinateurs de contrôle qui disposent ainsi de plus de temps pour le traitement de chaque événement.

Lorsque le processeur de filtrage agit au niveau du transfert (fig.2). Le temps de lecture intervient plus directement au niveau du temps mort induit par le filtrage. Cependant, suivant les rejets réalisés un tel système peut encore produire des gains en temps actif. Par ailleurs, il peut aussi gérer l'acquisition et modifier la présentation des données pour les rendre plus digestes aux processeurs situés en aval et ainsi alléger leur tâche.

Au cours des dix dernières années, les performances et les coûts des ordinateurs et des capteurs ont bien évolués. Cette évolution n'a pas éliminé les processeurs de filtrage et de formattage en incorporant leurs fonctions dans de nouveaux ordinateurs plus puissants. Bien au contraire, les compteurs

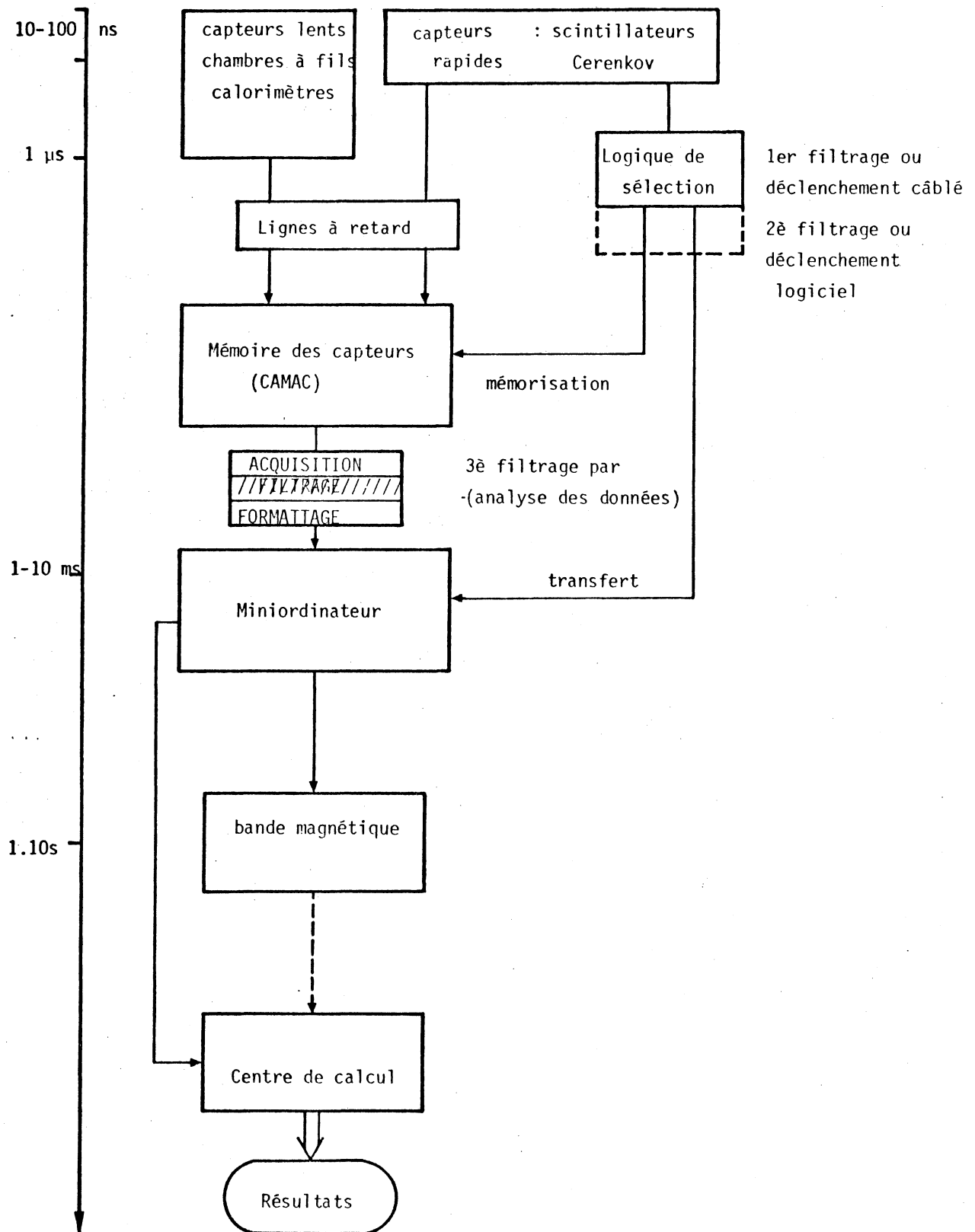


Fig.I.2 Principales étapes d'une acquisition de données

devenant eux-mêmes plus complexes, les processeurs de filtrage et de formattage sont devenus les interfaces nécessaires entre capteurs et ordinateurs.

a. Gains en temps actif

Il ne peut y avoir de gain dans ce sens que si le temps nécessaire à la prise de décision est inférieur au temps nécessaire au transfert de l'événement. Cette exigence limite considérablement la complexité des traitements envisageables.

- Dans le cas de l'expérience WA2, la "mémoire capteurs" était réalisée en standard CAMAC (2 μ sec de temps d'accès). Le processeur d'acquisition (200 ns de temps de cycle) était localisé entre CAMAC et un miniordinateur NOR 10 (Structure "pipe-line"), c'est à dire à un noeud du système de transfert des informations (fig.2). Ce processeur rapide gérait l'acquisition de façon beaucoup plus efficace que le miniordinateur, définissait le format de l'événement et réalisait un filtrage à partir d'algorithmes simples. Grâce à la vitesse d'exécution du processeur la lecture et le traitement d'un événement par le processeur étaient aussi rapides que son acquisition directe par le miniordinateur. Les rejets obtenus se traduisaient alors directement par un gain en temps actif provenant des lectures interrompues.
- Dans le cas de l'expérience NA10, la "mémoire capteurs" est réalisée dans le standard RMH (200 ns de temps d'accès) et transférée vers la mémoire d'un couple NORD 100-500 via CAMAC (2 μ sec de cycle). Ce transfert se réalise à travers une mémoire d'événement rapide remplie par RMH et vidée vers CAMAC. Le processeur de filtrage couplé à cette mémoire dispose alors d'un temps de traitement gratuit correspondant à la différence des temps de lecture CAMAC et RMH. Par ailleurs, le même processeur dispose d'un accès privilégié à un résumé d'informations. Ceci lui permet de réaliser des rejets en cours de lecture RMH.

Ces deux applications montrent qu'il est possible d'introduire un processeur de filtrage et de réaliser un traitement élémentaire sans pour autant augmenter le temps mort du système. Bien au contraire, tout rejet réalisé à ce niveau se traduit par une réduction du temps de lecture, c'est à dire du temps mort. Le temps de traitement moyen disponible dépend alors fortement de la réjection du système. Dans le cas de NA10 le temps de traitement a été optimisé

en réalisant un système comportant un ensemble de 4 couples "mémoire/processeur" capables de travailler en parallèle. Le temps de traitement ne constitue plus une limitation en système puisqu'il peut être augmenté à volonté en augmentant le nombre de couples "mémoire-processeur".

b. Gains en sélectivité

Lorsque le temps actif approche de 95 % l'amélioration de la sélectivité conduit à une réduction du volume des données à traiter. Le gain en sélectivité va de pair avec le préjugé que le physicien est prêt à introduire dans l'expérience. Il y a donc un compromis à établir entre sélectivité et préjugé. Tant que le temps actif est faible, l'amélioration de la sélectivité conduit aussi à une augmentation de la sensibilité de l'expérience. On est alors prêt à augmenter la sélectivité pour gagner en sensibilité.

- Dans le cas de l'expérience WA2, le déclenchement rapide était réalisé à partir du compteur Cerenkov DISC qui identifiait les quelques centaines d'hypérons présents dans un faisceau de $10^6 \pi^-$. Les désintégrations leptoniques étaient identifiées en exigeant :
 - 3 particules dans l'état final (amplitude du signal d'un compteur)
 - 1 proton dans l'état final (compteur)
 - 1 électron dans l'état final (ensemble de 100 blocs de verre au plomb)

Grâce à ce dispositif, le taux de déclenchement était réduit à quelques dizaines d'événements par cycle. Ce taux, parfaitement adapté aux capacités du système d'acquisition, a néanmoins produit plusieurs millions d'événements dont seuls quelques milliers ont été exploités pour la physique. La sélectivité aurait pu être augmentée, mais elle a été limitée à l'essentiel afin de limiter le préjugé introduit dans les données. On s'est donc limité à rejeter les événements inexploitable.

- Dans le cas de l'expérience NA10, il n'a pas été possible de réaliser un déclenchement aussi actif à partir de la logique câblée sur les capteurs. La sélection est obtenue après traitement de 714 compteurs dans des sous-ensembles de 192×2 , $32 \times 6 \times 32$, puis 24×24 combinaisons logiques rapides. Dans un faisceau de $1,5 \cdot 10^9 \pi^-$, cela conduit encore à plus de 600 déclenchements par déversement, alors que l'on a atteint la limite des possibilités en logique rapide. Un microprocesseur disposant de l'ensemble

des informations mesurées devient alors d'un intérêt considérable pour la réduction du temps mort. Le système utilisé sur NA10 fournit un facteur de réjection supérieur à 6, limitant ainsi le taux d'acquisition à une quarantaine d'événements par déversement. Suivant les conditions expérimentales, les événements recherchés représentent alors plus de 50 % des événements enregistrés. Dans cette application le filtrage a été plus sélectif parce qu'il était justifié par les gains en temps actif.

c. Gains en contrôle

Nous avons vu que l'emploi d'un processeur de filtrage permettait d'améliorer le rapport signal sur bruit. Il est évident que si le déclenchement de 1er niveau présente un rapport signal sur bruit très faible, l'histogrammation des événements ne représente que l'histogrammation du bruit de fond. L'amélioration du rapport signal sur bruit se traduit donc par une nette augmentation de la qualité du contrôle.

Comme cette amélioration du rapport signal sur bruit se traduit aussi par une réduction du nombre d'événements on allège d'autant la tâche d'histogrammation qui est généralement la tâche la plus lourde du miniordinateur. Elle autorise alors la multiplication des contrôles.

d. Formattage

Les données issues des processeurs de lecture sont généralement compactées pour ne comporter que les informations significatives [5], [6], [7]. De ce fait les événements, de longueur non définie, doivent être lus mot à mot. Le processeur chargé du filtrage réalise le reformattage de l'information par création d'une tête d'événement permettant aux processeurs situés en aval de trouver rapidement les informations qui leurs sont nécessaires sans parcourir la totalité de l'événement.

Les performances des systèmes de lecture ont beaucoup évolué, et leurs temps d'accès sont passés de 4.5 μ s/mot à 100 ns/mot [5], [6], [7], de 1974 à 1982. Lorsque ces données sont transférées à l'ordinateur via CAMAC, ces performances ne sont vraiment utilisables que si un système réducteur de données est placé entre ces systèmes de lecture et les ordinateurs de contrôle.

e. Détection des erreurs matérielles

Lorsque l'ordinateur de contrôle n'a pas le temps de réaliser l'histogrammation sur l'ensemble des événements acquis, il procède par échantillonnage. Les erreurs matérielles rares deviennent alors difficiles à mettre en évidence. Le processeur de filtrage et d'acquisition réalise cette surveillance sur l'ensemble des événements. Par ailleurs, étant plus performant il est capable de récupérer rapidement ces erreurs et de réarmer l'acquisition. Ainsi un compacteur lisant normalement une mémoire à la cadence de 2 μ sec par mot peut être bloqué par une erreur matérielle. Le processeur peut détecter le blocage après 100 μ sec et réarmer le système pour le préparer à l'événement suivant. Cette opération est réalisée en 100 μ sec par le processeur WA2 alors que l'ordinateur de contrôle ne le réalisait qu'en plusieurs ms ou dizaines de ms.

f. Gains en temps de calcul différé :

Toute réjection non préjudiciable à la physique se traduit directement par un gain sur le temps de calcul en différé.

Néanmoins ce gain de temps ne porte généralement que sur la première partie du calcul, c'est à dire la production de bande magnétique résumée, c'est à dire débarrassée des événements non restructuribles.

Cette partie du calcul ne concerne que 20 % environ du temps d'exécution global ce qui fait que dans ce cas, une réduction d'un facteur 10 ne conduit qu'à 18 % de gain sur le temps de calcul en différé.

Ceci n'est plus vrai si l'examen de tous les événements permet une restructuration de l'information allégeant la tâche des gros calculateurs. Ce peut être le cas d'une simple restructuration comme pour l'expérience WA2, ce peut être le cas aussi pour le résultat d'un traitement plus complexe comme le calcul de la masse des particules (exp. NA10), ou tout autre traitement (cf. en particulier la réf.8 où un gain de 40 a pu être obtenu par rapport à une CDC 7600).

g. Souplesse d'adaptation

Bien des critères de filtrage n'apparaissent qu'au fur et à mesure que les conditions expérimentales se précisent, et que les prises de données permettent une analyse réelle des différents algorithmes.

Il est évidemment bien plus simple de faire évoluer le logiciel d'un processeur de filtrage que de modifier les détecteurs du déclenchement ou la logique rapide qui leur est associée.

De plus ce type de déclenchement de second niveau peut être rendu plus efficace par la multiplication du nombre des processeurs, permettant ainsi l'exécution simultanée de plusieurs tâches, voir le traitement simultané de plusieurs événements (cf. ch.IV, exp. NA10).

Lorsque la modularité de la structure le permet cette évolution est extrêmement simple et permet la résolution d'algorithmes de plus en plus complexes.

I.3. CONTEXTE MATERIEL DES ACQUISITIONS DE DONNEES

Etant données leurs dimensions, la réalisation des expériences de physique des hautes énergies se fait dans un contexte de plusieurs laboratoires souvent éloignés les uns des autres et de nationalités différentes. Ceci est à l'origine d'une diversité des systèmes de lecture et des ordinateurs et créer des difficultés d'interconnection et d'utilisation des différents sous-ensembles.

En plus de cette diversité, les réalisations matérielles s'effectuent dans un contexte technique difficile, caractérisé par :

- Un milieu parasité du fait de la grande concentration de matériel
- Une évolution rapide du cahier de charges avec néanmoins des impératifs en temps de fabrication
- L'obligation d'utiliser et de concevoir des systèmes modulaires permettant de fractionner le travail et d'obtenir des ensembles évolutifs et réutilisables.
- Enfin, localisés entre les processeurs de lecture et les processeurs de contrôle, les processeurs de filtrages sont tributaires des deux.

L'ampleur du travail à réaliser, ainsi que les performances du système sont donc directement corrélés aux efforts mis en oeuvre pour la standardisation du système de lecture. Dans le cas de WA2, beaucoup de temps était consacré à la gestion de plusieurs compacteurs compatibles avec le standard CAMAC mais nécessitant néanmoins des traitements particuliers. Ceci a été grandement simplifié pour NA10 où toutes les données étaient lues à partir du même "compacteur". Même si cela alourdit les procédures, un processeur d'acquisition et de filtrage est néanmoins suffisamment versatile pour les prendre en compte de façon plus efficace que l'ordinateur standard.

a. Les processeurs de lecture

Ces processeurs ont subi une évolution considérable car leur temps de lecture est à l'origine du temps mort de l'expérience. Les grandes étapes de leur développement ont été marquées par :

- le standard CAMAC (1971)
- l'apparition de compacteurs réduisant le nombre des informations (1975)
- l'utilisation de la technologie ECL pour les compacteurs (1979)
- l'apparition d'un nouveau standard améliorant toutes les performances :
le FASTBUS dont les premières réalisations sont actuellement utilisables

Malgré toutes ces nouveautés, le standard CAMAC reste utilisé systématiquement pour l'interconnection ordinateur de contrôle-acquisition-filtrage des données. En effet, il est le seul à pouvoir être utilisé quel que soit le type d'ordinateur et ses performances sont compatibles avec les temps d'accès de ces derniers, lorsqu'il s'agit d'entrée-sorties.

Souvent plusieurs générations de compacteurs cohabitent dans la même expérience (WA2). Ceci revient à alourdir considérablement la tâche du programme d'acquisition lorsqu'il est réalisé à partir d'un processeur unique. Ce n'est plus le cas lorsque l'ordinateur de contrôle gère une mémoire tampon remplie en parallèle par un ou plusieurs compacteurs intelligents (processeurs) associés chacun à un type de capteur (NA10). Ainsi la diversité est introduite au niveau du compacteur qui est modulaire alors que l'ordinateur central est cantonné dans la gestion de la mémoire tampon.

b. Les ordinateurs de contrôle

Les tâches confiées à ces processeurs, ainsi que les performances qui leur sont demandées rendent toute indiquée l'utilisation d'ordinateurs banalisés du commerce. On obtient ainsi les avantages suivants :

- facilité d'utilisation (langage évolué)
- système fourni par le constructeur
- compatibilité avec les autres machines de la collaboration (Fortran)
- maintenance constructeur
- grande capacité mémoire (nécessaire car devant contenir le système, les programmes utilisateurs, les histogrammes et les événements en cours de transfert sur bande magnétique). Quel que soit son fabricant, ce processeur est pourvu d'un accès CAMAC le rendant compatible avec

tous les autres processeurs. Nous avons eu la chance de travailler pendant huit ans avec le même constructeur, ce qui nous a permis de rentabiliser au mieux les investissements réalisés dans le logiciel.

c. Filtrage d'événements et Gestion mémoire

Dernière réduction de donnée avant transfert à l'ordinateur de contrôle, la tâche de filtrage est étroitement liée avec celle de Gestion mémoire, puisqu'elle bouleverse l'arrangement des événements en mémoire d'accumulation. La mémoire d'accumulation d'événement doit être accessible au processeur de filtrage et au processeur de contrôle. L'efficacité du filtrage est conditionnée par son temps d'exécution (le filtrage ne doit pas ralentir le transfert sur bande). L'accès processeur de filtrage-mémoire est donc plus critique que l'accès processeur de contrôle-mémoire.

Comme la mémoire doit être très rapide pour supporter les flux d'acquisition avant filtrage, il s'en suit que l'on est amené à dissocier les tâches de filtrage de celles de contrôle.

- Utilisation de la mémoire de l'ordinateur banalisé

Ce procédé a deux avantages principaux :

- Il n'y a pas d'interface ordinateur-mémoire à réaliser, elle existe -
- La programmation du contrôle en ligne et de la mise sur bande est simple.

Par contre il y a beaucoup d'inconvénients :

- temps d'accès lent, encore ralenti par le vol de cycle souvent inévitable
- problèmes de conversion d'adresses dûs à la pagination et à la mémoire virtuelle
- gestion mémoire peu efficace et difficile à réaliser de l'extérieur dans le cas de processeurs de filtrages spécialisés
- en multi-miniordinateurs : lourdeur des messages, liaisons difficiles à réaliser.

- Utilisation d'une mémoire externe spécialisée

Ce procédé n'a qu'un inconvénient : une telle mémoire n'existe pas en tant que telle. Ses avantages sont nombreux, on peut considérer notamment :

- un temps d'accès très rapide (à la limite de la technologie du moment)
- gestion mémoire prévue de l'extérieur en multi-utilisateur
- interface avec les processeurs d'acquisition facile à réaliser
- adressage absolu autoincrémenté.

d. Conclusion

Les tâches de contrôle en ligne, d'histogrammation de tests et de mise sur bande magnétique ou d'analyse hors faisceau sont du domaine des machines banalisées. Certaines tâches peuvent être confiées à des ordinateurs banalisés au prix d'une baisse de performances.

Une étude des flux et des capacités de transfert permet de déterminer à priori l'utilité des processeurs spéciaux.

Du fait de leur temps d'exécution et de leur spécificité, les tâches de déclenchement et d'acquisition doivent être confiées à des processeurs spécialisés, câblés ou microprogrammés. Les tâches de filtrage, de gestion mémoire et de gestion DMA sont du domaine des processeurs spécialisés programmables et microprogrammables pour accroître les performances.

C'est donc cette voie de recherche qui a été suivie depuis 1975, en tenant compte de deux facteurs essentiels : l'amortissement multi-expérience et la fiabilité. L'amortissement multi-expérience est un impératif lié à l'effort de développement demandé à un tel système, effort difficile à renouveler à 3 ans d'intervalle. La fiabilité, essentielle en regard des prix de revient du temps d'expérience ne peut être obtenue qu'au prix d'une certaine modularité permettant les tests sérieux des différents éléments indépendamment les uns des autres.

Pour ces raisons notre choix s'est porté sur la conception d'une unité de base performante et bien adaptée à l'acquisition de données (cf. chapitre II), complétée d'interfaces et d'opérateurs câblés spécifiques à chaque expérience. L'ensemble est entièrement microprogrammable mais pourvu d'un jeu d'instructions classiques permettant son utilisation en mode programmé.

C H A P I T R E I I

RECHERCHE D'UNE STRUCTURE DE BASE - CONCEPTION DU NOYAU GESPRO

Les résultats des analyses précédentes ont dégagé l'utilité d'un système microprogrammé. La structure de base la mieux adaptée sera obtenue par optimisation successive d'une structure classique avec priorité à la vitesse d'exécution. Cette approche pré-expérience est suivie d'une optimisation spécifique à l'expérience considérée. Un noyau de base a donc été dessiné compte tenu de ces impératifs : l'adaptation d'opérateur câblé rapides doit être simple et efficace, la souplesse et la modularité obtenue permettent une connection standard et efficace vers le haut (processeur de niveaux supérieurs) et vers le bas (processeur de niveaux inférieurs ou acquisition).

Cette conception aboutit à une machine réalisable uniquement en laboratoire et incompatible avec toute réalisation industrielle. Les problèmes de système de développement, de maintenance, de test et d'exploitation sont donc étudiés et résolus localement.

1. PRINCIPE DE BASE DES PROCESSEURS MICROPROGRAMMABLES

Ce principe, vieux de plus de trente ans, est illustré figure II.1. On remarque sur cette figure le remplacement du séquenceur câblé par un ensemble mémoire - organe de commande programmable.

Cette formule nous convient particulièrement de par les avantages qui lui sont spécifiques :

- Un module de base générale. Peut être spécialisé et adapté par logiciel donc rentabilisé
- L'étude du matériel est simplifiée et mieux structurée
- L'amortissement du matériel est bien supérieur : l'adaptation à l'expérience est réalisée par programme. Dans ces conditions une bonne étude peut être utilisable toute une génération technologique soit entre cinq et sept ans.

Les optimisations sont réalisées en deux plans :

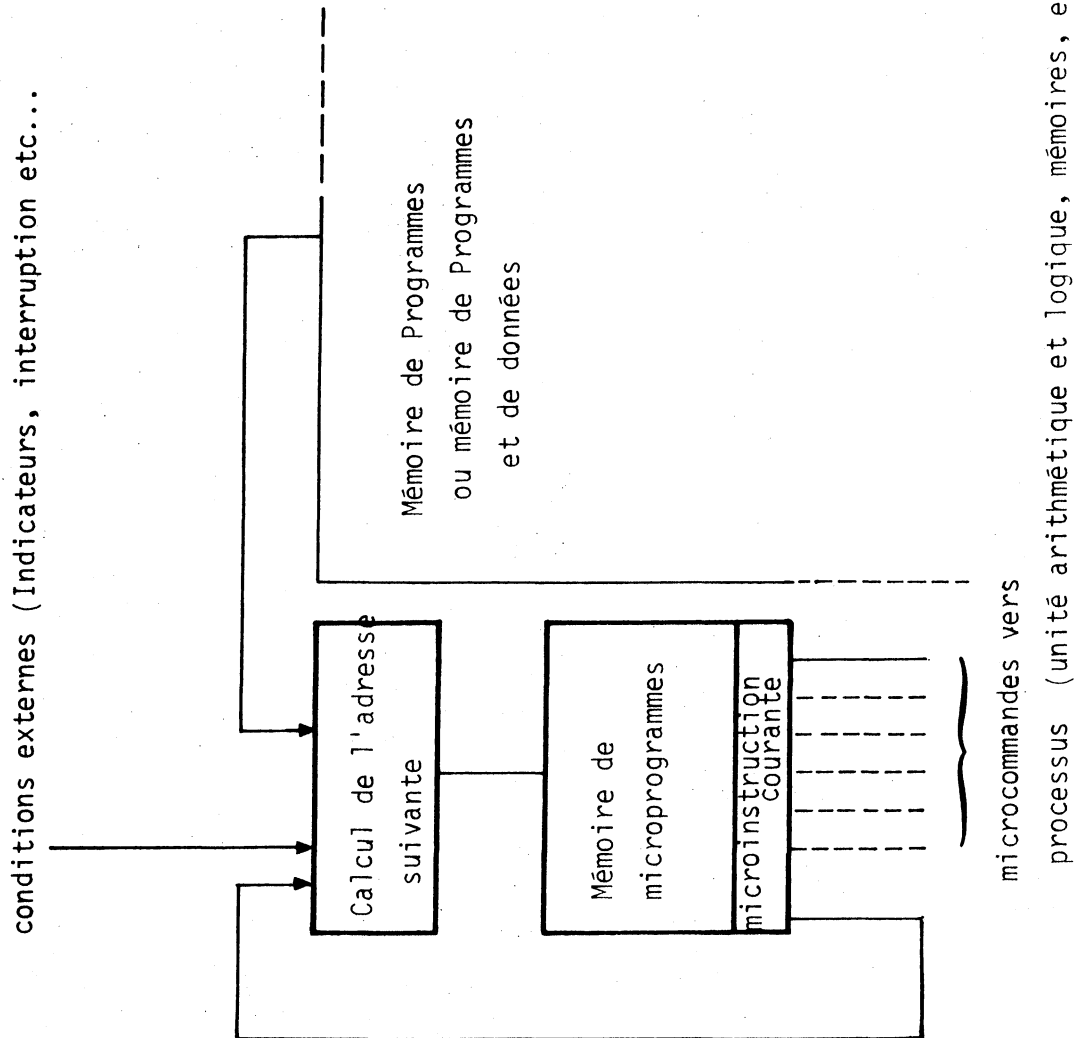


Fig.II.1. Structure de base d'un ordinateur microprogrammé

sur le plan structure et sur le plan technologique
sur le plan structure la priorité vitesse d'exécution est indirectement prise en compte par la recherche d'une structure codée et dense capable de réduire le nombre de composants et surtout la taille des mémoires. Cette recherche inclue la définition des formats (longueur des mots mémoire de contrôle et de programme)

Sur le plan technologique l'unité centrale et les mémoires sont choisies en fonction de leur temps d'exécution.

2. ORGANISATION DES MEMOIRES DE COMMANDE

Cette organisation se définit en deux étapes différentes

La première concerne le format du mot mémoire ou format de microinstruction

La deuxième concerne le type d'adressage de la mémoire ou mode d'accès et de chaînage des microcommandes.

a. Format

Une microinstruction contient l'ensemble des informations nécessaires à l'exécution d'un cycle machine complet. Cette description codée des actions peut être contenue en un seul mot (microprogrammation horizontale) ou en plusieurs (microprogrammation verticale).

De même elle peut être codée de façon maximale (2^n actions pour n bits) ou minimale : (n actions sur n bits).

Le premier critère définit le nombre d'accès à la mémoire par cycle, le second détermine le nombre d'actions simultanées possibles.

L'évolution technologique de 1972 à 1975 ayant permis une grande intégration donc un temps d'exécution court, l'intérêt de la microprogrammation vertical ne concerne plus que l'économie de mémoire. Pour une machine à priorité vitesse, ce critère ne se discute donc plus par contre le deuxième critère détermine et la taille de la mémoire et la vitesse d'exécution.

La division du mot mémoire en champs fonctionnels indépendants permet d'optimiser l'encodage sans perte de vitesse d'exécution.

Chaque champ se comporte comme une instruction et la microprogrammation se comporte comme la programmation simultanée de plusieurs machines.

Parmi ces champs, trois sont difficiles à concevoir ; Il s'agit des champs de synchronisation temporelles, de conditions de saut et de recherche de l'adresse suivante.

2.b Chainage des micro-instructions et adressage de la mémoire de commande

Le champ des fonctions des synchronisations temporelles a pour but l'optimisation du temps d'exécution du cycle courant.

Il peut être évité par l'utilisation systématique du mode synchrone mais dans ce cas impose à tous les processeurs la vitesse du plus lent. En mode asynchrone chaque opérateur active indique son temps d'exécution et chaque cycle est optimisé.

L'ensemble de ces liaisons sont gérées par ce champ qui peut être partiellement codé dans la mesure où certaines actions sont exclusives (utilisation de ressource commune).

Plusieurs unités de temps d'exécution voisins peuvent également utiliser le même code.

Le champ de condition : Ce champ contient les codes utilisés pour effectuer les différents débranchement conditionnels. Lorsque l'on veut donner priorité à la vitesse d'exécution, les actions de ce champ sont simultanées à celles des autres. En conséquence, ses conditions ne s'appliquent pas aux résultats arithmétiques ou logiques de la micro-instruction courante, mais à ceux des micro-instructions précédentes. C'est à cette seule condition que le temps d'exécution du saut conditionnel peut être masqué par celui de l'unité centrale. Il s'ensuit une difficulté supplémentaire de programmation, surtout s'il est nécessaire d'utiliser ce type de micro-instructions plusieurs fois de suite.

Le champ de recherche de l'adresse suivante s'applique à un processeur spécial : l'unité de contrôle de la mémoire de microprogramme et à ce titre est peu différent des autres.

Cependant son implication avec la structure matériel et logiciel de la machine est beaucoup plus grande. Indirectement il détermine la taille de la mémoire de commande (nombre de mots et nombre de bits/mot). Le degré de complexité de recherche dépend du type d'adressage utilisé.

L'adressage peut être incrémental : Il est alors implicite sauf en cas de débranchement. Ce cas est généralement traité en microprogrammation verticale, par une microinstruction de saut utilisant les autres champs pour définir l'adresse. L'unité de contrôle est dans ce cas réduite à un simple compteur.

Ce type d'adressage, calqué sur celui utilisé en programmation, convient très mal à une mémoire de commande, en effet :

- tous les microprogrammes démarrent au même endroit (recherche de l'instruction en mémoire centrale)
- ils ont tous une séquence identique plus ou moins longue suivant le type d'instruction.

Le mode d'adressage le plus efficace au point de vue fonctionnel doit être fonction de l'endroit d'où l'on vient et du type de l'instruction en cours.

En conséquence, l'adressage incrémental, s'il est simple à utiliser, est le moins efficace.

L'adressage peut être explicite :

Chaque microinstruction contient l'adresse de la suivante

Le nombre de bits utilisé est grand et grandit avec la taille de mémoire (extension difficile)

Les sauts sont rigides et le lien avec le code opération de l'instruction a disparu. Le champ condition doit de plus être étendu puisqu'il représente la seule possibilité de débranchement.

C'est donc un type d'adressage peu efficace, mais qui est très agréable à utiliser de par l'absence de codage.

Pour réduire le nombre de bits nécessaires, l'adressage relatif peut être utilisé avec succès, d'autant que ce type d'adressage favorise un parcours arborescent de la mémoire de commande.

L'adressage peut être exclusivement relatif et conditionné au code opération de l'instruction courante. Les avantages de la structure sont nombreux :

- Le nombre de bits nécessaires est réduit (une partie du code opération est utilisé à sa place)
- Le champ condition est réduit aux seuls traitements des indicateurs
- Le code opération de l'instruction sert de titre consulté en permanence par l'unité de contrôle pour retrouver son chemin dans la mémoire de microprogramme.

C'est le mode d'adressage le plus efficace, aussi bien en rapport performance/nombre de bits qu'en vitesse d'exécution. C'est aussi le mode d'adressage le plus difficile à utiliser. Les débranchements en arbre permettent l'utilisation systématique de zones communes à plusieurs micro-instructions ce qui minimise la taille de la mémoire.

On approche le cas idéal ou l'écriture d'un nouveau microprogramme ne rajoute aucune microinstruction supplémentaire mais indique simplement l'ordre dans lequel doivent être exécutés les microinstructions déjà présentes en mémoire.

Ce type d'enchaînement peut avoir un intérêt différent lorsque, en acquisition de données, chaque donnée peut donner lieu à un traitement parmi beaucoup d'autres. On établit alors une bijection entre structure de la donnée et type de traitement et on utilise cette méthode d'enchaînement sur le code de la donnée. Tous les tests de recherche du traitement se font alors en parallèle sans temps d'exécution supplémentaire au traitement. Ce procédé est particulièrement efficace dans le cas de traitement court où le temps d'exécution est critique.

La donnée considérée est vue comme une instruction et l'unité de commande de la mémoire de micro-programme permet de se débrancher immédiatement vers le traitement correspondant. Cet algorithme de recherche est exécuté en parallèle avec les autres actions et ne pénalise en rien le temps d'exécution.

Il est intéressant de comparer les temps d'exécution que peut obtenir un mini-ordinateur rapide pour le même résultat. La méthode la plus rapide consiste à utiliser une table contenant autant d'adresses de saut que de valeur possible de la donnée. Il faut lire la donnée, la transférer dans un registre d'indexe et exécuter un débranchement indirect à la table indexée sur la donnée. Ceci nécessite un minimum de 3 ou 4 microsecondes. Or il est fréquent dans les traitements de filtrage d'obtenir par microprogrammation des temps d'exécution de cet ordre de grandeur (en 4,5 microsecondes on exécute 30 micro-instructions). Dans ces conditions, si l'on ne peut utiliser **un type** de chaînage simultané à l'unité centrale, on **perdra** plus de temps pour trouver le traitement que pour l'utiliser, ce qui est à répéter **pour** chaque nouvelle donnée.

2.C. Opérateurs spécifiques et interfaces

Les différents processeurs d'acquisition et de traitement sont implémentés sur l'expérience grâce à des interfaces et, pour des raisons d'efficacité de résolution temporelle, sont munis d'opérateurs câblés, spécifiques à l'expérience considérée.

De la structure matériel d'interconnection de ces éléments dépend l'efficacité de l'ensemble. Les trois structures de bases sont :

- la structure unibus : Périphérique et mémoire se partagent le même bus par multiplexage. C'est la solution la plus simple, la plus économique et la moins efficace en temps d'exécution puisque on ne peut accéder qu'à un seul opérateur à la fois.
- Structure à deux busses : L'un des busses est connecté à la mémoire et l'autre aux "entrée-sorties". C'est la structure la plus répandue qui offre un compromis acceptable pour une machine banalisée si les deux busses peuvent être utilisés simultanément.

- Structure multibus : l'unité centrale est munie de plusieurs busses et l'occupation de l'un d'eux n'empêche pas l'unité centrale de lancer une opération sur un autre. De plus dans certains cas, certaines opérations peuvent être exécutées directement de bus à bus, ou encore deux opérations simultanées différentes donnent deux résultats sur deux busses différents.

Cette structure peut être avantageusement dotée de registres tampons sur chaque bus ce qui permet d'utiliser le bus pendant le temps d'accès d'un opérateur.

Cette solution est la plus efficace et la moins répandue car elle est beaucoup plus onéreuse et plus difficile à mettre en oeuvre.

Au stade de la conception ces types de structure doivent être examinés avec soin, en tenant compte que :

- Une certaine modularité est indispensable tant pour réduire la difficulté de conception que pour permettre un amortissement multi-expérience par une évolution suffisante
- Des possibilités d'extension doivent être prévues et ce surtout au niveau des commandes microprogrammées d'activation d'opérateurs câblés nouveaux.

Enfin les limites technologiques des composants et de certains montages doivent être bien connues avant simulation et choix définitif des structures.

C'est notamment le cas des allocateurs de ressource commune (contrôleur des mémoires de données inévitablement partagées par exemple).

3.a. Choix des structures et des composants (1975-1978).

Les premiers microprocesseurs offraient des performances tellement faibles qu'il était impensable de les utiliser dans des domaines où la vitesse d'exécution est le critère principal.

1975 montre l'apparition du premier processeur en tranche. Ce processeur, l'Intel 300Q, bipolaire, annonce des temps d'exécution 20 fois plus rapide que les minis classiques de l'époque (150 ns pour une addition de 16 bits). Cependant une étude approfondie montre qu'on ne profite de ces temps d'exécution que si l'on utilise les circuits en microprogrammation horizontale.

Le gain obtenu dans ces conditions varie entre 10 et 100 suivant la structure (nombre de bits, nombre d'opérateurs câblés, nombre de busses).

Cette unité de base, nouvelle et unique en 1975, possède tous les éléments principaux découlant des impératifs de fonctionnement dans notre domaine (acquisition et traitement avec priorité à la vitesse).

- Ce circuit impose une architecture multibus dont l'un (le bus de masque) est utilisable simultanément avec les autres
- Les dimensions de mémoires ne sont pas limitées, ni en nombre de mots, ni en nombre de bits par mot.
- L'implémentation d'opérateurs câblés spécifiques est simple et n'impose pas le multiplexage des interconnexions mémoires.
L'horloge n'est pas phasée et aucune fréquence n'est imposée :
un fonctionnement entièrement asynchrone et contrôlable de la périphérie n'offre pas de difficulté.
- L'unité de commande de la mémoire de microprogramme est optimisée en nombre de bit par mot et les modes d'adressages, bien que complexes, permettent la réutilisation systématique des cases précédemment utilisées, avec marquage du chemin par mémorisation d'une partie du code opération utilisable par le logiciel même après référence mémoire. Enfin le jeu des fonctions des éléments centraux est très riche en fonctions logiques et permet généralement deux opérations simultanées par action du bus de masque.

Un premier prototype, réalisé en 1975 de façon modulaire avec ces éléments fournit les bases de travail nécessaires aux simulations relatives à l'expérience "Hypérons" (1977-1980).

Les autres microprocesseurs en tranches apparaissent peu après (1976) : l'AMD 2900 et le MOTOROLA 10800.

Entre 1976 et 1979 des études comparatives ont été menées afin de vérifier périodiquement l'efficacité relative des éléments I 3000 par rapport aux éléments concurrents.

Ces études, résumées en ANNEXE, ainsi que des considérations d'amortissement multi-expérience, n'ont pas abouti à l'abandon des I3000 jusqu'en 1981-1982, date d'apparition de nouveaux circuits (iapx 432 d'Intel notamment) dont l'impact dans le domaine des hautes énergies n'est pas encore bien défini.

3.b. Etude de la structure du noyau GESPRO

Rappelons que cette étude donnait priorité à la vitesse d'exécution. Le choix des composants étant fait, l'étude de structure se résume à :

- la longueur des mots mémoires (instruction donnée et micro-instruction)
- le nombre et les affectations des buses
- l'horloge et la synchronisation sur processus externe
- l'adaptabilité aux opérations spécifiques (évolutibilité)
- l'interruptibilité

b.1. Longueur du mot mémoire centrale

Cette mémoire contient instructions, adresses et données. Les données sont à 16 bits. Les adresses peuvent l'être également, du moins en mémoire centrale.

Priorité à la vitesse nous a conduit à mettre dans un même mot donné et opération à effectuer et adresse et code opération. Le code opération est défini sur 8 bits, ce qui nous amène à un mot de 24 bits au moins.

D'autre part les calculs intermédiaires nécessitent souvent plus de 16 bits mais dans notre cas très rarement 32 bits.

24 bits est un compromis efficace, qui s'avère à l'expérience satisfaisant.

Enfin la première version, conçue pour l'expérience HYPERON, est dédiée CAMAC, lequel est défini sur 24 bits.

b.2. Longueur du mot de la mémoire de commande

- Définition des champs

2 champs constructeur (CPE-MCU) et un troisième de masque : le K bus

1 champ supplémentaire regroupe tous les sous-champs des processeurs parallèles.

Le champ MCU nécessite 12 bits (7 bits MCU - 4 bits de condition, 1 bit de chargement).

Le champ de masque pourrait être à 24 bits, mais a été réduit à 17 en tenant compte du fait qu'un mot mémoire contient en principe un code opération sur 8 bits et une donnée ou une adresse sur 16 bits. Avec 17 bits agissant bit à bit pour les 16 premiers il est possible de masquer les données ou les adresses bits à bits et tout ou rien du code opération.

- Le CPE demande 7 bits soit 24 bits pour les champs CPE et K

- Les champs concernant les opérations parallèles ont été définis petit à petit par utilisation simultanée de simulations et d'approximation successives.

2 bits ont été réservés à la capacité mémoire de commande (4 plans

maximum) 4 bits ont été réservés à l'attribution du bus d'entrée I

3 bits à la mémoire centrale

1 bit de commande du mode d'adressage de la mémoire centrale
(ce bit est en fait un bit de condition)

2 bits de modificateur d'instruction permettant d'indiquer le numéro du registre courant de trois sources différentes

Ceci porte à 12 bits la dimension de ce champ spécial avec, à l'origine plusieurs codes libres, en particulier sur le bus I. Le format, représenté fig.2 est donc de 48 bits, ce qui pour une machine microprogrammable spécialisée et extensible est relativement peu.

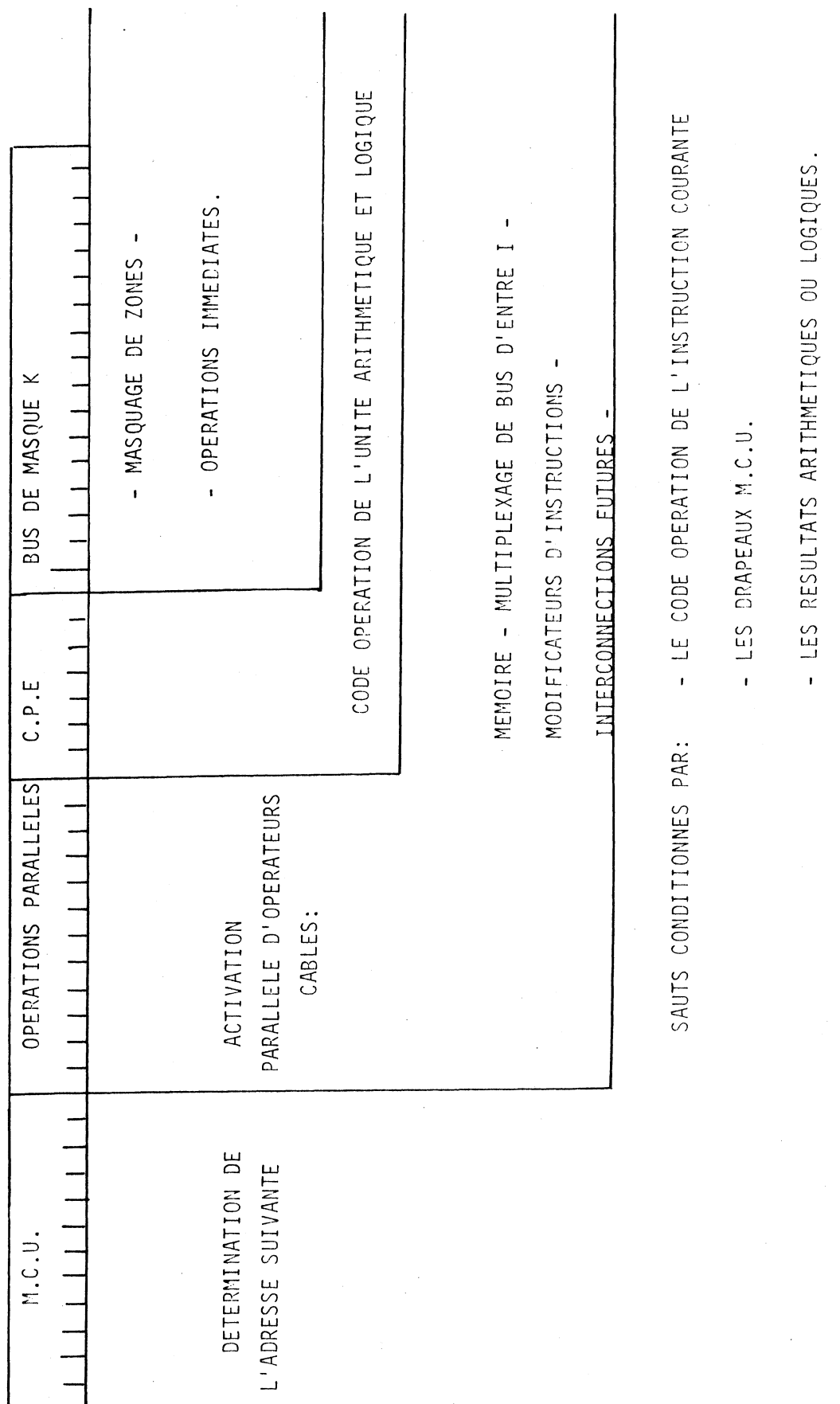


Fig.II.2 : Format général d'une microinstruction (48 Bits)

4 . APPORT DE LA MICROPROGRAMMATION

Le gain apporté par la microprogrammation à la conception d'une machine spécialisée se situe à plusieurs niveaux.

Le temps de conception du matériel est notablement réduit car la spécialité du processeur et ses performances se situent au niveau logiciel. Il s'ensuit un matériel banalisé plus efficace et comportant moins de risque d'erreur. L'architecture elle-même bénéficie du concept ; en étant plus général, elle devient évolutive et ~~peut~~ être conçue pour, par prévision de microcommandes indépendantes laissées inutilisées à la conception.

L'utilistation d'une telle machine en acquisition est de même beaucoup plus efficace : les entrées-sorties de petits blocs de données, entièrement microprogrammés sont souvent plus rapides qu'en D.M.A. D'autre part les entrées sorties prioritaires qui doivent être très rapides peuvent être connectées directement aux microcommandes en attente. Ces derniers peuvent alors s'exécuter à la vitesse de la machine et font en quelque sorte partie intégrante de l'unité centrale.

Par exemple une unité de calcul ultra-rapide, développée quatre ans après la conception du noyau GESPRO a pu être connectée directement et effectuée en 200 ns une multiplication flottante en simultanéité vraie avec les autres processeurs internes (cycle mémoire centrale, débranchement, masque etc...).

La microprogrammation permet de tirer le meilleur parti d'une machine en traitement, par écriture d'un jeu d'instruction spécialisé, incluant les traitements dans les instructions. Ces traitements peuvent évidemment être exécutés en parallèle avec des D.M.A. avec possibilité de synchronisation microprogrammée des deux processus.

Enfin l'interruptibilité est plus facile à traiter : comme elle est dépendante de l'expérience, elle peut être conçue sous forme d'un opérateur placé sous commandes microprogrammées. Le nombre de niveaux et le nombre de registres visibles au logiciel peuvent être variables et n'engage que le logiciel.

Ainsi pour trois expériences différentes trois types d'unités d'interruptions ont été réalisées : l'une à 24 niveaux (8 matériels, 16 logiciels)
l'autre à 16 niveaux (8 matériels, 8 logiciels)
enfin la dernière à 8 niveaux (matériels ou logiciels).

Ces trois unités utilisent le même principe et les mêmes microprogrammes seuls change l'occupation mémoire centrale utilisée pour les sauvegardes des contextes.

L'interruptibilité peut dans certains cas être impérative en temps et demander l'exécution ultra-rapide d'une tâche. On peut dans ce cas soit négliger toute sauvegarde soit l'entreprendre après l'exécution de la tâche qui n'utilise dans ce cas que les registres internes (propres à la microprogrammation). Pour les trois unités précitées, la prise en compte complète d'une interruption est de 3,5 μ s.

Une application (expérience NA10) a nécessité un temps de réponse beaucoup plus court, de l'ordre de grandeur de processus câblés. L'écriture d'un microprogrammable activé directement a permis sans modification de la structure matériel de démarrer l'exécution de la tâche moins de 200 ns après l'interruption.

Deux applications permettent de chiffrer le gain obtenu grâce à la conception banalisée de GESPRO.

La première concerne la mise en ligne d'une unité arithmétique supplémentaire à exécution simultanée avec le CPU.

Cette unité a été nécessaire sur les expériences NA10 et UA2.

Dans le cas d'un système classique, l'unité est connectée via un interface sur le bus d'entrée-sortie. En admettant qu'un tel interface soit développé pour un NORD 10 (100), et qu'il utilise d'autant d'IOX différents qu'il n'est nécessaire pour optimiser le temps d'exécution. Le temps nécessaire pour multiplier deux flottants cherchés en mémoire et mettre le résultat en mémoire par un programme écrit en assembleur, devrait comprendre les phases suivantes :

- chercher en mémoire les deux mots du premier opérande et les charger dans l'opérateur câblé (4 instructions)
- même algorithme pour le deuxième opérande (4 instructions) avec lancement simultané de l'opération (4 instructions)
- lecture du résultat (deux mots) et sauvegarde en mémoire (4 instructions)

L'ordre de grandeur du temps d'exécution d'un tel programme est d'environ 25 μ sec.

Cette même unité, câblée sur GESPRO, est activable directement par micro-programme. Les fonctions sont réduites de 16 à 8 car un flottant de 24 bits tient en un mot machine. Ces fonctions sont définies par 3 bits du champ "opérateur" de la microinstruction qui avaient, lors de la conception été laissé libre en prévision d'opérateurs câblés rapides. Les cinq busse de la machine sont donc laissées libres en particulier les busse d'adresse et de donnée mémoire n'interferrent pas. Le microprogramme exécutant cette même multiplication ne nécessite que 7 microinstructions soit de 1050 ns à 1300 ns suivant la nature de l'instruction suivante, et d'autres opérations peuvent être éventuellement exécutées simultanément.

On obtient ici un gain d'un facteur 20 en utilisant le même matériel. Le gain serait d'environ 10 si l'on utilisait une machine à 24 ou 32 bits pour faire le même travail.

Le second exemple concerne le temps de lancement d'une DMA CAMAC. Ce temps est d'environ 200 μ s en utilisant un mini-ordinateur programmé en assembleur (NORD 10).

La programmation de ces mêmes modules par GESPRO permet de descendre ce temps à 25 μ s environ et 15 μ s par une instruction spécialisée.

Si la DMA est conçue autour de GESPRO, en utilisant les deux busse de sorties et le busse d'entrée réservé aux périphériques, ce temps peut être descendu à 5 μ s (cas du projet d'expérience UA2). Le gain ici est donc d'environ 40. De plus il permet d'utiliser GESPRO de façon efficace en filtrage de donnée avec une simple structure en pipe line. En effet le temps d'exécution permet de lancer plusieurs DMA par événements sans pénalisation du temps mort, même en déclenchement saturé. Enfin la modularité, si précieuse dans le domaine, se trouve grandement amélioré :

d'une part une partie des tâches sont gérées uniquement par logiciel ce qui permet de banaliser le matériel de modules qui seraient spécialisées s'ils étaient réalisés sous forme câblée.

D'autre part la conception est plus générale et l'introduction de nouvelle tâche se réalise sous forme de module qui n'affecte pas la structure, un peu comme des interfaces qui sont réalisés pour un calculateur.

5 INCIDENCE DE LA MICROPROGRAMMATION SUR L'UTILISATION

La programmation d'une machine microprogrammée est rigoureusement identique à celle d'une machine câblée. Néanmoins lorsque la machine est microprogrammable et destinée à être microprogrammée fréquemment, l'ensemble du système de développement doit être prévu pour. En particulier l'ensemble assembleur, éditeur de liens et chargeur doit pouvoir opérer sur jeux d'instructions quelconque et si possible prévoir une syntaxe conversationnelle utilisable pour modifier les jeux d'instructions.

D'autre part, en utilisation, la microprogrammation pure est peu envisageable : d'une part sa complexité ralentit l'écriture des algorithmes et d'autre part l'interruptibilité est nécessaire pour la synchronisation de ces processeurs avec l'environnement.

Une méthode mixte est donc utilisée.

Elle consiste à disposer toujours d'un jeu d'instructions complet de telle sorte que l'on puisse utiliser la machine en mode programme.

D'autre part la plus grande partie de la mémoire de microprogramme est laissée libre pour l'écriture des instructions spécialisées à l'expérience en cours.

On peut ainsi écrire le système et les enchaînements des tâches en assembleur et réserver la microprogrammation pour les algorithmes de filtrage et les boucles qui demandent des temps d'exécution plus courts.

C H A P I T R E III

DESCRIPTION DU NOYAU GESPRO

1 . RESUME DES TACHES SPECIFIQUES

Les mini-ordinateurs utilisés pour le contrôle en ligne sont de plus en plus performants sur le chapitre du nombre d'opérations arithmétiques exécutées par seconde. Ils sont également très performants sur les plans capacités d'adressage et facilité d'utilisation. De ce fait ils sont moins performants en acquisition temps réel.

Ces difficultés se résument, pour l'utilisateur en trois problèmes : les lancements de DMA, les gestions mémoires et les communications.

Les lancements de DMA sont trop lents (temps d'exécution de nombreux niveaux de programmation, mémoire virtuelle, pagination). Les gestions mémoires sont lourdes et lentes (très grande capacité mémoire, pagination).

Les communications avec l'extérieur sont très lentes (mauvais temps de réponse aux interruptions). Par contre ces tâches sont faciles à réaliser au moyen de processeurs spécialisés rapide et de taille plus modeste . A ces trois tâches s'ajoutent évidemment les traitements proprement dits.

1. a. Gestion DMA :

Le grand nombre de données à lire impose une lecture rapide par DMA ou par microprogramme (ou les deux simultanément). Ces données sont issues de systèmes de lectures généralement centralisés sur des compacteurs permettant de ne lire que les données significatives. (Réf.5,6,7).

L'idéal est de tout centraliser sur le même type de système de lecture (cf. expérience NA10, chapitre IV) ce qui permet de n'utiliser qu'une DMA par événement, ou même, qu'une seule DMA par déversement.

Ces systèmes de lecture n'ont pas tous le même mode de fonctionnement :

- . en mode "compacteur" un seul module est à lire, et les réponses (Q et X) à chaque ordre décrivent l'une des quatre possibilités : lecture valide, donnée non prête, fin de lecture, panne du module.
- . sans compacteurs par contre il faut changer d'adresse à chaque lecture et s'arrêter lorsqu'une position vide est adressée. Sur l'expérience Hypéron (WA2), par exemple

les TDC (convertisseurs temps-numérique), les ADC (convertisseur (analogique-digitaux), les MWPC (fils de chambres proportionnelles) et un bloc de données fixes (registre, échelles....), étaient lus à quatre adresses différentes par quatre DMA différentes. On comprend alors l'importance de l'efficacité des systèmes de lancement des DMA. Or, la taille des machines utilisées pour le contrôle en ligne, l'histogrammation et la mise sur bande magnétique est devenue suffisamment importante pour que les systèmes de mémoire virtuelle et de pagination deviennent nécessaires avec pour conséquence un très mauvais temps de réponse aux ordres de lancement de DMA et aux interruptions. Les ordres de grandeur atteignent ici le facteur 100 et plus entre un VAX et un micro-ordinateur spécialisé par exemple.

1.b. Gestion mémoire

La notion de rejet d'événements avant enregistrement implique celle de gestion mémoire. Il faut en effet organiser la mémoire pour pouvoir éliminer rapidement et efficacement les mauvais événements après lecture et avant enregistrement. Cette mémorisation temporaire peut être réalisée sous plusieurs formes. Nous en distinguerons ici trois principales étant entendu que nous ne nous intéressons qu'à la mémoire visible des processeurs de filtrage.

- structure Pipe-line (fig. III.1)

Le processus de filtrage est inséré entre les processeurs précédents (acquisition) et les minis (contrôle en ligne et enregistrement). Un événement est transféré par DMA dans la mémoire du processeur de filtrage, il est traité, et s'il n'est pas rejeté il est transféré dans la mémoire du mini. Ce procédé, le plus simple, n'est efficace que dans la mesure où le processeur de filtrage

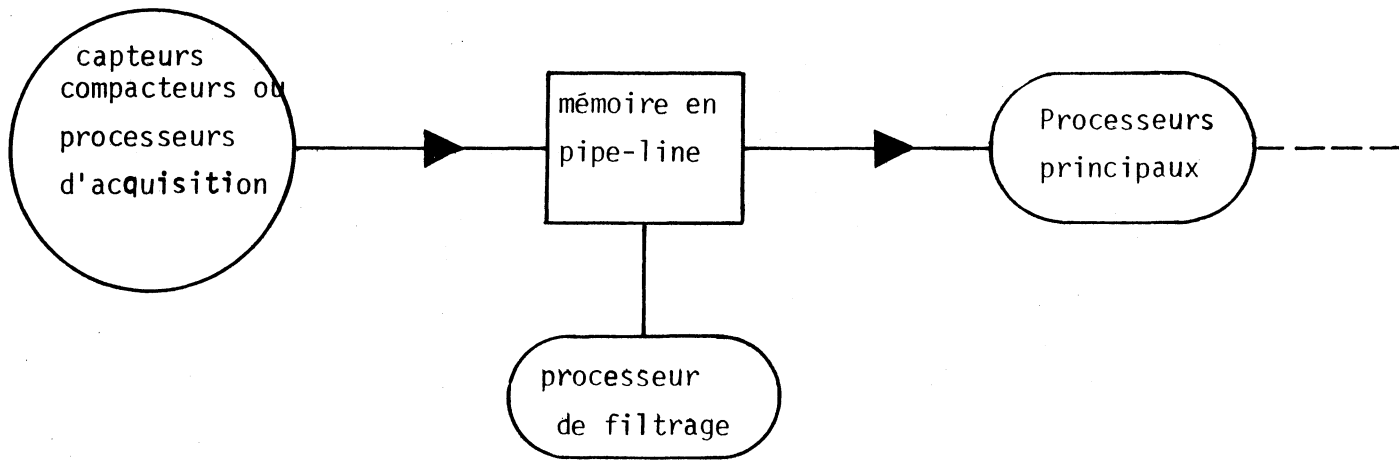


Fig.III.1 : Structure pipe line

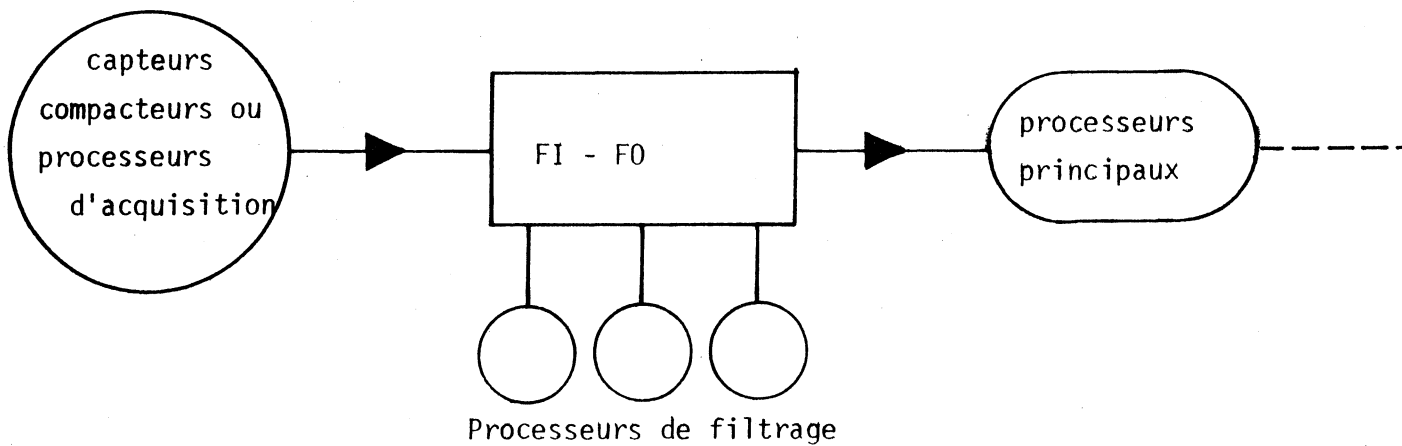


Fig.III.2 : Structure pipe line en F1-F0

fait aussi office de processeur d'acquisition et est capable de stocker 3 événements : l'un en acquisition, l'autre en traitement et le troisième en transfert vers les processeurs de rang supérieur. Ce type d'architecture a été utilisé pour le projet d'expérience UA2. Il peut être amélioré en utilisant plusieurs couples mémoires-processeurs en FI-FO (fig.2). Ce qui apporte l'avantage de régulariser la distribution des événements dans le temps. De plus il n'est plus nécessaire de stocker 3 événements par machine.

- structure utilisée sur WA2 (fig III ,3)

Une DMA lancée par le processeur de filtrage envoie l'événement vers la mémoire du processeur principal et vers la sienne. La gestion des mémoires est assurée par le processeur de filtrage. Si l'événement n'est pas rejeté il n'y a rien à faire et l'événement suivant est envoyé à la suite par incrémentation du compteur d'adresse DMA. Si l'événement est rejeté, l'adresse DMA est modifiée par le processeur de filtrage et l'événement suivant surécrit l'événement précédent. Les deux transferts ont lieu en DMA, pendant les transferts l'unité centrale est libre et on peut envisager d'acquérir une autre partie des données par micro-programme en exécutant le traitement en vol et en utilisant le système Pipe-line décrit précédemment. (Réf.9)

- structure avec mémoire commune à tous les processeurs (fig.III,4)

C'est la structure idéale :

- elle élimine tous les transferts de processeur à processeur
- tous les processeurs peuvent être banalisés
- les extensions sont faciles
- la dimension du FI-FO peut être portée à la taille maximale de la mémoire.

Malgré tous ces avantages cette structure n'est pas utilisée. En effet, pour chaque machine il est difficile d'utiliser autre chose que la mémoire prévue par son constructeur. Ceci est d'autant plus vrai que la machine est plus grosse (problème de système d'exploitation, d'adaptation en temps, de maintenance constructeur, de mémoire virtuelle et de pagination...).

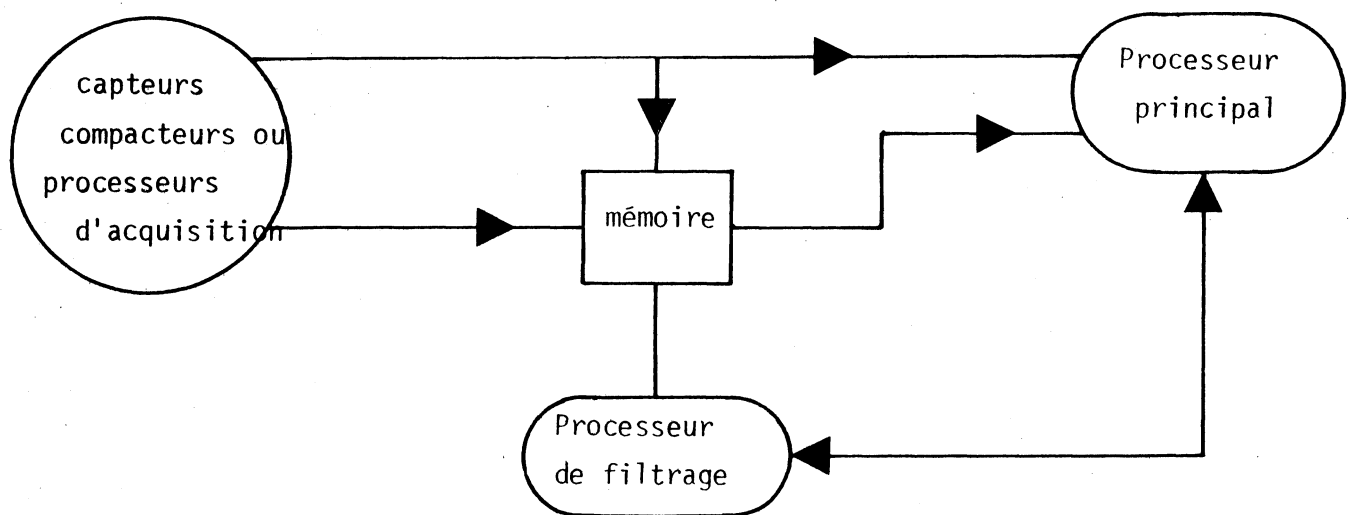


Fig.III.3 : Structure WA2

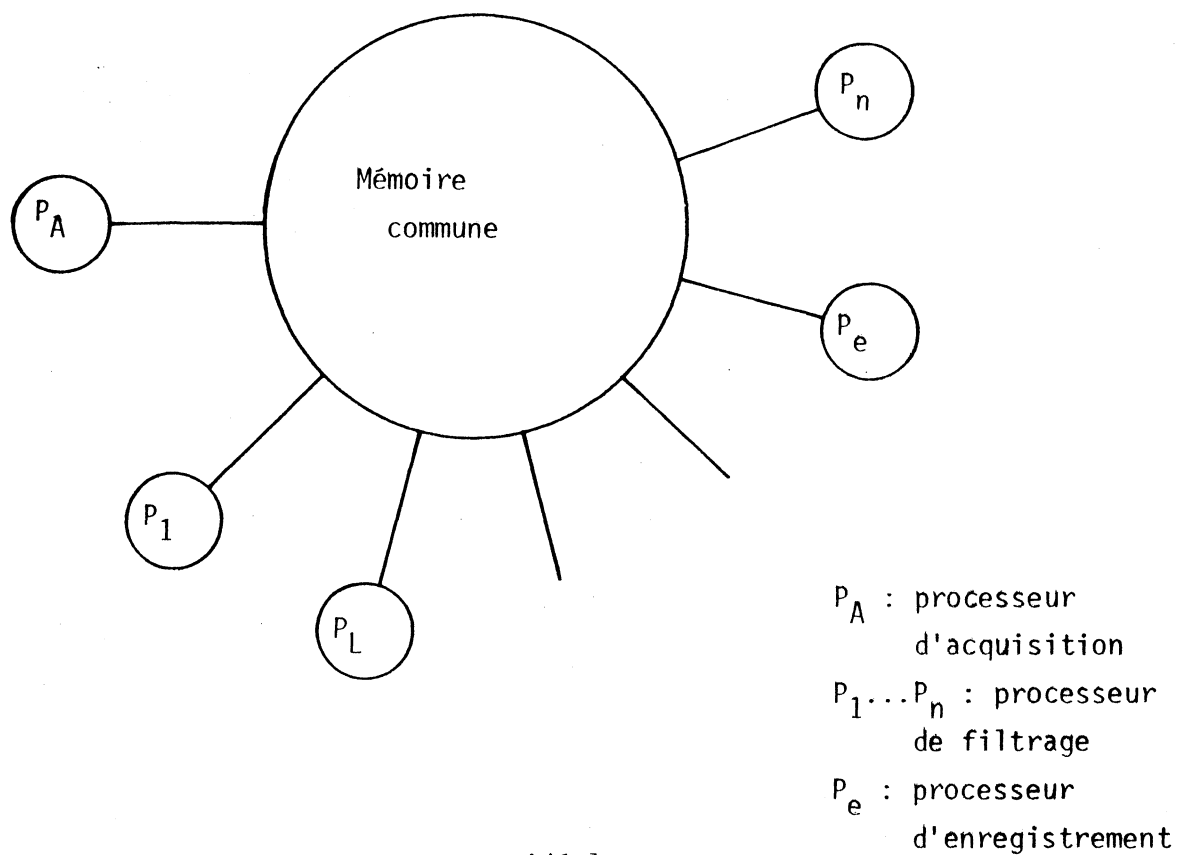


Fig.III.4 : Structure idéale

Cependant on peut approcher cette structure en employant une mémoire indépendante, rapide et à accès multiples qui s'ajoute aux mémoires propres aux différentes machines.

Cette solution comporte beaucoup d'avantages :

- chaque processeur est indépendant et peut utiliser le système "constructeur" sans trop de modifications
- les accès à la mémoire rapide peuvent être standard (CAMAC par exemple) lorsque la vitesse d'exécution n'est pas critique. Ils peuvent être directs lorsqu'ils doivent être rapides (processeurs de filtrage, de gestion mémoire ou d'acquisition)
- un contrôleur est chargé spécifiquement de la gestion et rend l'ensemble complètement transparent aux processeurs de contrôle et d'enregistrement. Ceux-ci ne voient qu'un périphérique intelligent qui fournit les événements triés.

Cette solution, utilisée sur l'expérience NA10 avec 4 mémoires d'événement et 4 processeurs GESPRO sera décrite en détail dans le chapitre IV.

La conception de GESPRO a été réalisée de telle sorte que ces trois solutions ont été possibles sans avoir à modifier le noyau. Ceci a été obtenu en intégrant au noyau un contrôleur mémoire simplifié et rapide, fonctionnant en vol de cycle. Ainsi l'accès externe a été multiplexé en 8 accès sur WA2, utilisé tel quel sur NA10 et en multi DMA sur le projet UA2.

c. Communication

Les problèmes de communication sont de deux types : Les communications interprocesseurs et les communications avec l'extérieur ; (expérience essentielle-ment). Dans les deux cas il faut éviter les pertes de temps, ce qui nécessite, au moins pour les communications avec l'expérience un système d'interruption efficace à plusieurs niveaux. L'interruption prioritaire de déclenchement doit quelquefois être très rapide (cf. NA10).

Les communications entre processeurs peuvent éventuellement utiliser une ou plusieurs boîtes aux lettres (partie de mémoire de donnée généralement), ou de modules de communication, ou des interruptions spécialisées.

d) Traitement des données

But principal de l'ensemble, les traitements sont spécifiques à chaque expérience et très souvent mal définis au début du projet. Les algorithmes définitifs apparaissent donc en cours d'expérience. Malgré cette diversité seuls deux types de traitement ont une influence réelle sur la structure matérielle de la chaîne d'acquisition. On distingue :

- Les traitements effectués avant ou pendant la lecture des capteurs
- Les traitements effectués sur données acquises (après lecture)

Le premier type fait suite au déclenchement et est susceptible d'empêcher l'acquisition proprement dite. Le deuxième utilise les données comme base de décision et n'empêche que l'enregistrement.

Il est évident que le premier type ne peut utiliser les données. Il utilise seulement un résumé réduit, rapidement utilisable. Ce résumé peut nécessiter la conception d'opérateurs câblés spécialisés que l'on devra connecter sur le processeur de la façon la plus efficace possible. (C'est ici l'endroit où la vitesse d'exécution est la plus critique).

Les tâches qui ont été énumérées permettent de définir, outre les résultats énoncés au chapitre précédent, les critères généraux nécessaires à la conception d'un processeur de filtrage.

- La priorité est donnée à la vitesse d'exécution :

Il faut donc une unité centrale rapide, asynchronisable, microprogrammable.

- La mémoire de microprogramme, très rapide, est réalisée en mémoire vive pour permettre la mise au point d'algorithme en cours d'expérience.
- Un bon compromis programmation-microprogrammation doit être facilement déterminé pour chaque expérience.
- Une petite mémoire rapide, utilisée pour les programmes doit être extensible jusqu'à pouvoir contenir en plus au moins 3 événements pour les cas d'utilisation en simple structure pipe line.
- Une unité d'interruption performante est nécessaire.
- Des opérateurs câblés spécifiques à chaque expérience doivent pouvoir être facilement connectés sans nuire à la rapidité de l'ensemble.

2. Description du matériel GESPRO

a. L'unité centrale elle-même

Comme décrite au chapitre II, elle est réalisée autour des éléments Intel de la famille I 3000. Ces circuits sont assemblés pour former une machine à 24 bits, microprogrammable sur des mots de 48 bits, à 150 ns de temps de cycle.

Les fonctions de l'unité arithmétique et logique ont été décrits au chapitre II. Le jeu d'instruction correspondant est donné en ANNEXE.

Des registres tampons à un ou deux étages, assignés à tous les périphériques, sont disposés sur chaque liaison périphérique - Unité centrale, afin de minimiser les temps d'accès par entrelacement. La liaison avec la mémoire de microprogramme passe également par un registre tampon, afin d'exécuter la micro-instruction courante pendant le temps d'accès à la suivante.

La figure 5 montre l'architecture simplifiée de la partie unité centrale. On remarque (cf. chapitre II) les 24 bits sur tous les buses (opération + donnée ou opération + adresse en un mot).

Et le bus de masque à 17 bits.

Pour accélérer les temps d'exécution de décalage multiple un opérateur de décalage a été connecté sur le bus I. Soit par paquet de 8 bits soit de 1 à 24 bits suivant les besoins de l'expérience (cf. chapitre IV, expérience WA2 et UA2)

Toutes les interconnexions étant indépendantes il est possible d'utiliser les temps d'accès mémoire ou périphérie pour exécuter d'autres opérations sur

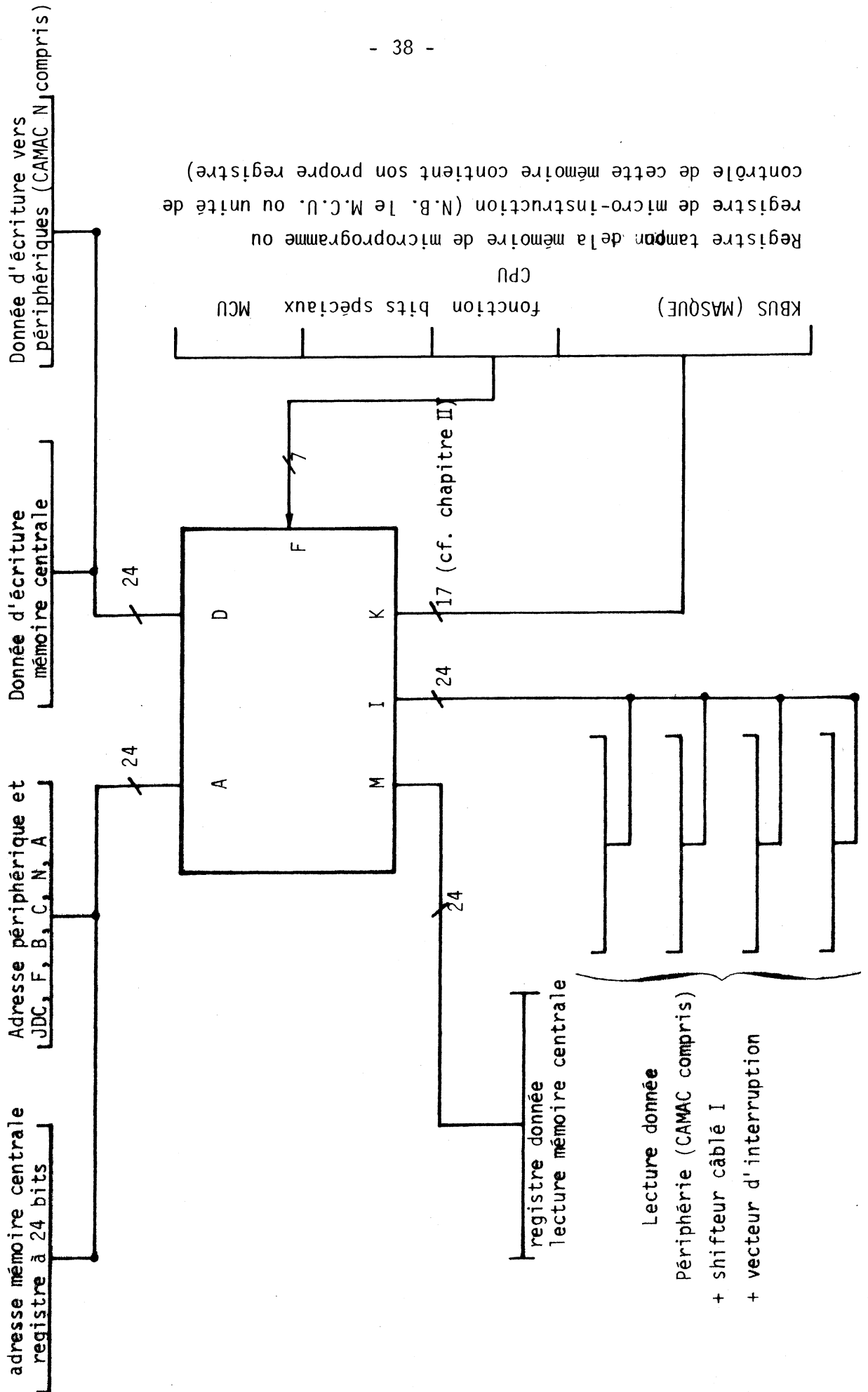
les autres buses, tout en utilisant l'unité centrale et le bus de masque en entrée, en parallèle avec le champ de saut à la micro-instruction suivante.

La machine se comporte alors en processus asynchrone, séquencée de la périphérie par un jeu de bits spécialisés (3^e champ de la microinstruction). Le dispositif est simple grâce à l'horloge monophasé déclenchée utilisée : chaque périphérique émet, lorsqu'il est activé, un signal d'occupation pendant toute la durée de l'opération concernée.

L'unité centrale, grâce à un bit spécial peut tenir compte ou non de cette information d'état. Lorsqu'elle en tient compte, le cycle courant est rallongé au début du cycle de la durée d'occupation restante.

Lorsque plusieurs périphériques sont activés simultanément, l'unité centrale sera synchronisée ~~sur~~ le plus lent (fig.III,6).

Fig.III.5 : Architecture simplifiée de la partie Unité Centrale



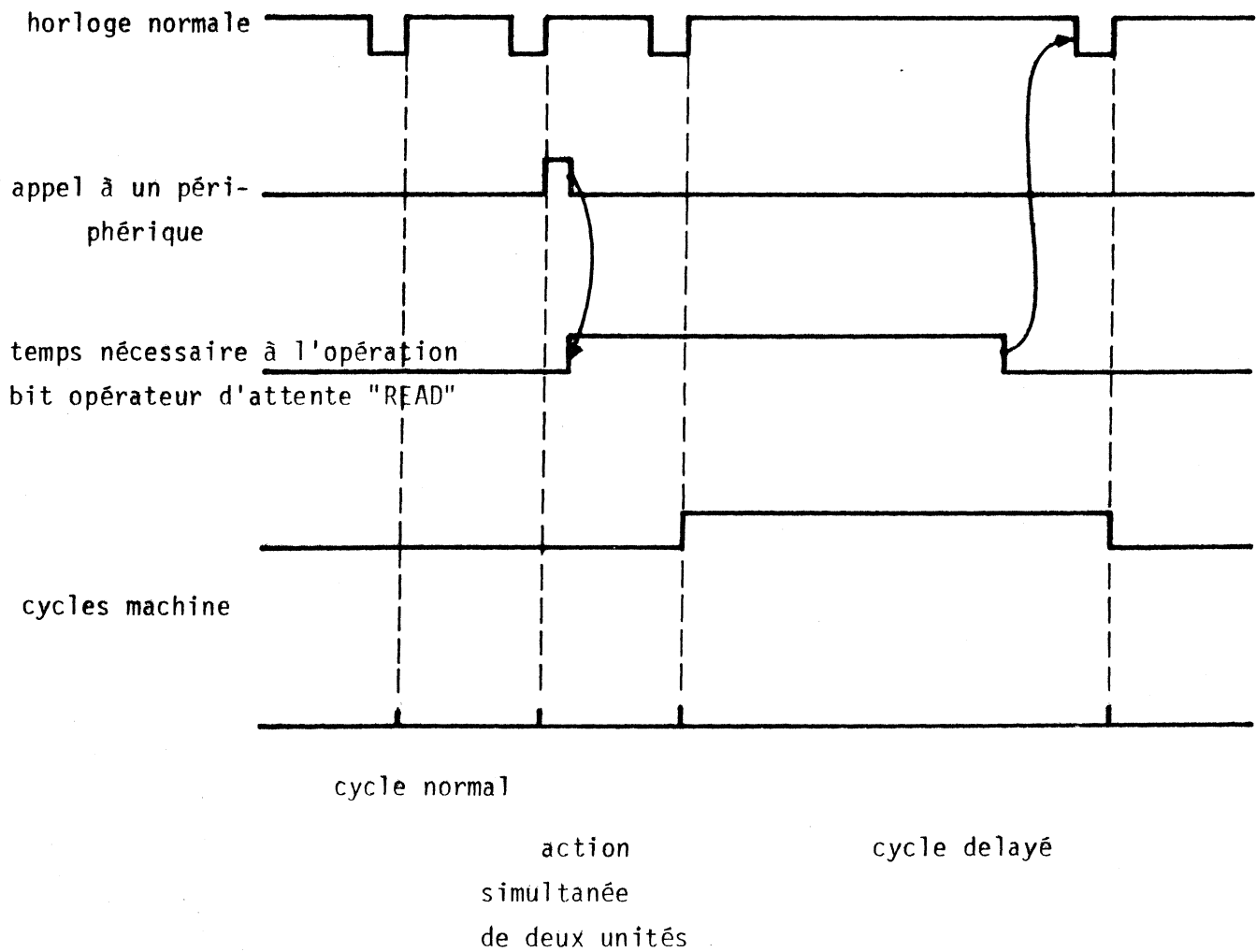
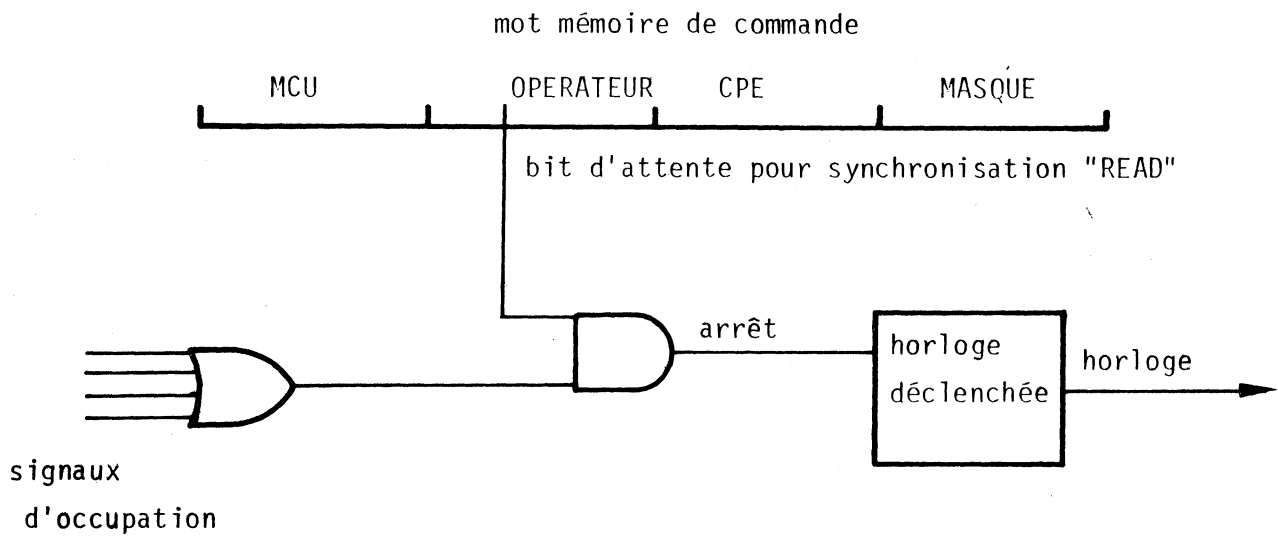


Figure III.6.

b. Description des différents champs

La figure 7 montre le format général d'une microinstruction. Les bits moins significatifs sont occupés par le bus de masque K déjà décrit (17 bits). Les bits 17 à 23 par le code fonctions de l'unité arithmétique et logique (Annexe 1), conforme aux normes du constructeur. Les bits 24 à 35 sont spécifiques à GESPRO, et ont été ajoutés pour accroître les performances de la machine par activation parallèle d'opérateurs câblés.

En effet certaines tâches, réalisables par microprogramme sont facilement accomplies par un opérateur câble spécifique avec, comme avantage supplémentaire un temps d'exécution négligeable et un taux d'occupation moindre en mémoire de commande.

Ce champ spécifique est reporté ci-dessous :

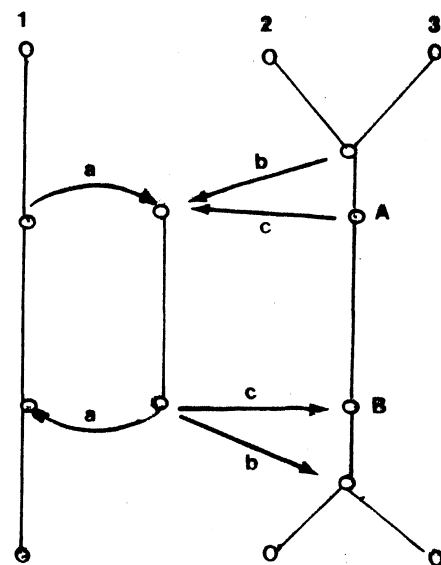
Req	Read	W	C2	C1	C0	MP1	MPO	EN	MP	MM	IT
-----	------	---	----	----	----	-----	-----	----	----	----	----

Le premier bit IT autorise un débranchement prioritaire non visible à l'unité de contrôle, ce qui permet un retour au microprogramme initial par un saut conditionné au code opération de l'instruction.

Ceci permet en particulier, au cours d'une zone (commun ou non à plusieurs instructions) d'utiliser ou non une déviation commune.

Exemple :

Les instructions 1, 2 et 3 ont la même variante et les instructions 2 et 3 une zone commune (par exemple trois instructions à référence mémoire nécessitent, en cas d'adressage indirect une recherche de plus en mémoire centrale, action identique pour les trois). La figure ci-contre montre les trois types de déviations possibles :



- a) Remplacement d'une zone par une autre (instruction 1)
- b) Remplacement d'une zone commune par une autre zone commune
- c) Adjonction d'une zone supplémentaire.

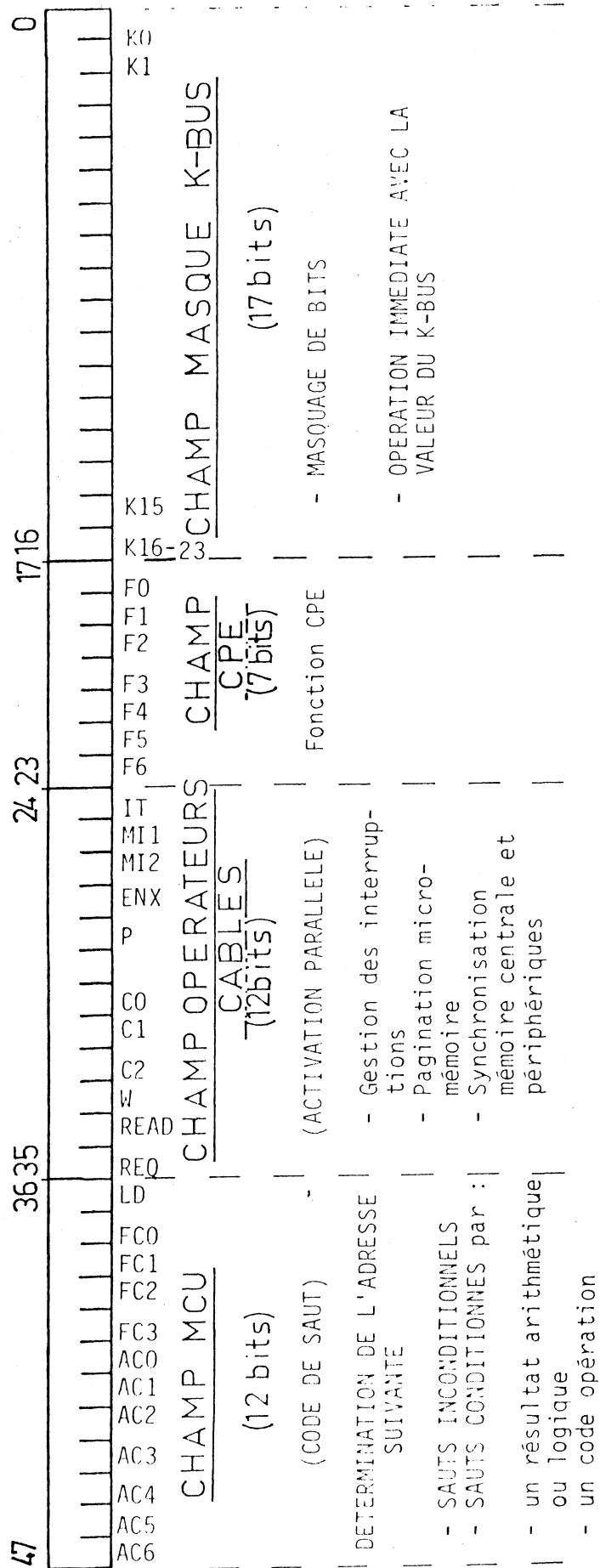


Fig.III.7. FORMAT DES MICROINSTRUCTIONS GESPRO

A noter un point important :

Les trajets C) et b) peuvent être conditionnés au code autrement dit on peut avec la même microinstruction passer par c) pour l'instruction 2 et passer par b pour l'instruction 3, sans perdre le bénéfice de la zone commune AB utilisée dans le cas général.

Outre le mode d'adressage, le choix du registre d'index et le débordement de l'unité arithmétique rapide ont été connectés à ce débranchement.

Les deux bits suivants (MP et MM) multiplexent les deux derniers bits de la microinstruction courante. Ils ont été implantés pour réduire le nombre de cases utilisées en mémoire de microprogramme, en multiplexant le numéro du registre concerné par l'opération courante.

En effet, plusieurs instructions peuvent être identiques ou presque, au numéro de registre près (c'est le cas notamment des instructions à référence registre). Dans ce cas, puisque le numéro de registre fait partie du champ "fonction CPE" issu directement de la mémoire de microprogramme, il est nécessaire d'écrire autant de micro-instructions que de registres concernés.

Or il est agréable de pouvoir commander ce numéro de registre directement de la mémoire centrale par le code opération de l'instruction. Deux cas peuvent se présenter : le numéro de registre est utilisé avant référence mémoire, et dans ce cas il existe sur le bus "mémoire centrale M", après tampon, ou bien ce contenu a été détruit par un nouvel appel mémoire et dans ce cas il est récupérable en sortie de l'unité de commande de la mémoire de microprogramme. D'où deux bits de commande de multiplexage, nommés "modificateurs" d'instruction.

Les bits MP_0 et MP_1 sont simplement deux bits d'adresse supplémentaires permettant d'étendre l'espace de travail de l'unité de contrôle de la mémoire de microprogramme à 2K mots de 48 bits.

Les bits EN et C_0 , C_1 , C_2 règlent l'utilisation du bus I réservé à la périphérie.

Ce bus peut être alloué soit aux périphériques dits "standards" (prévu sur le noyau), soit à des périphériques d'extension. Les huit possibilités fournies par les trois bits $C_0C_1C_2$ sont réservées aux périphériques standards si EN = 0 (interruption, shifteur, CAMAC) et à un périphérique non standard à huit fonctions maximum si EN = 1 (unité arithmétique rapide sur les versions NA10, UA2).

Le bit "W" est le bit d'écriture, utilisé par les opérations mémoire et ou les opérations périphériques non standards.

Le bit "Read" est le bit de synchronisation, utilisé, comme w pour les références mémoire et ou pour les entrées-sorties non standards. Le bit "Req" indique que la mémoire centrale est concernée (cf. fig. 6)

Tous ces bits peuvent, au moins théoriquement, être tous activités indépendamment (simultanément par exemple). Seul le sous-champ $C_0C_1C_2$ a été codé puisqu'il concerne l'utilisation du même bus.

Si "Req" et "EN" sont présents simultanément, les deux opérations seront simultanées mais seront toutes deux de lecture ($W = 0$) ou toutes deux d'écriture ($W = 1$).

Enfin Req Read Write présents simultanément indique que l'on attend la fin d'une opération précédente (si il y a lieu) pour enchaîner une nouvelle opération à la plus grande vitesse possible (avec écriture en mémoire) exemple : incrémentation d'une case mémoire centrale :

Adresse → MAR Req

M + 1 → AC Read, Req, W

(Seul le champ CPE et les trois bits de synchronisation ont été représentés)

c . Unité de contrôle de la mémoire de commande

Les bits 36 à 47 du mot mémoire de microprogramme sont réservés à l'unité de contrôle de la mémoire de microprogramme (M.C.U.). Ils sont utilisés strictement comme indiqué par le constructeur et comportent deux sous champs : code opération "saut" et code opération "drapeau", à 7 et 4 bits, et un bit de chargement utilisable soit à l'initialisation soit pour faire correspondre directement à un code opération une adresse de la mémoire de commande.

Le jeu d'instructions de cette unité est porté en ANNEXE. On remarque essentiellement, comme décrit au chapitre II, que la quasi totalité des sauts sont relatifs, soit au code opération en provenance de la mémoire centrale, soit à l'adresse de la microinstruction courante (soit à ces deux bases réunies).

L'adresse de la micro-instruction est de plus désignée sous forme d'intersection ligne colonne ce qui permet de réduire le nombre de bits de commande.

L'exécution d'un microprogramme comporte quatre types d'étapes :

- 1) La recherche du code opération en mémoire centrale avec remise à jour du compteur ordinal (étape commune à tous les microprogrammes)
- 2) choix d'une classe d'instruction parmi 16 selon l'état des quatre bits les plus significatifs du code opération.

L'exécution du saut (JPX) correspondant engendre automatiquement la mémorisation des quatre autres bits du code, afin de conserver l'itinéraire de l'instruction en cas de référence mémoire.

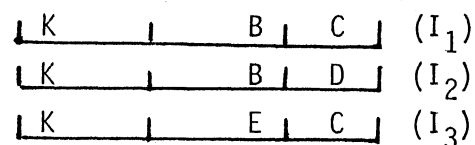
Cette étape est commune à toutes les instructions d'une même classe.

- 3) A chaque classe peut correspondre quatre branches secondaires, par le déchiffrement des deux bits les plus significatifs des 4 bits restant
- 4) chacune de ces branches peut mener à quatre nouvelles branches selon la valeur des deux derniers bits.

Il est possible de remplacer 3) par un test direct sur les quatre bits. Cette valeur détermine alors un chemin parmi 16.

Les regroupements de zone communes sont possibles.

Exemple :



Les trois instructions I_1 I_2 I_3 appartiennent à la même classe. Leur trajet peut être défini par leurs codes partiels K, B, C, D, E.

Soit

K	$\xrightarrow{\text{JPX}}$	k	B	$\xrightarrow{\text{JLL}}$	b	C	$\xrightarrow{\text{JRL}}$	c
D	$\xrightarrow{\text{JAL}}$	d	E	$\xrightarrow{\text{JLL}}$	e			

I_1	suivra le trajet	k, b, c
I_2	" "	k, b, d
I_3	" "	k, e, c

I_1 et I_2 ont leurs deux premiers trajets communs

I_3 et I_1 se séparent au 2ème segment mais se retrouvent en c.

Sur cet exemple 5 segments ont été écrits au lieu de 9 d'où un gain important sur l'encombrement en mémoire de commande. De plus le gain augmente avec le nombre de micro-instructions écrites : une instruction d'une classe différente peut aussi réutiliser b et c ou l'un de ces deux segments.

La figure III,8 résume l'arbre des sauts obtenu par cette méthode. Enfin les résultats obtenus sur toute une expérience (WA2 ; 1377, 1380) ont permis de montrer :

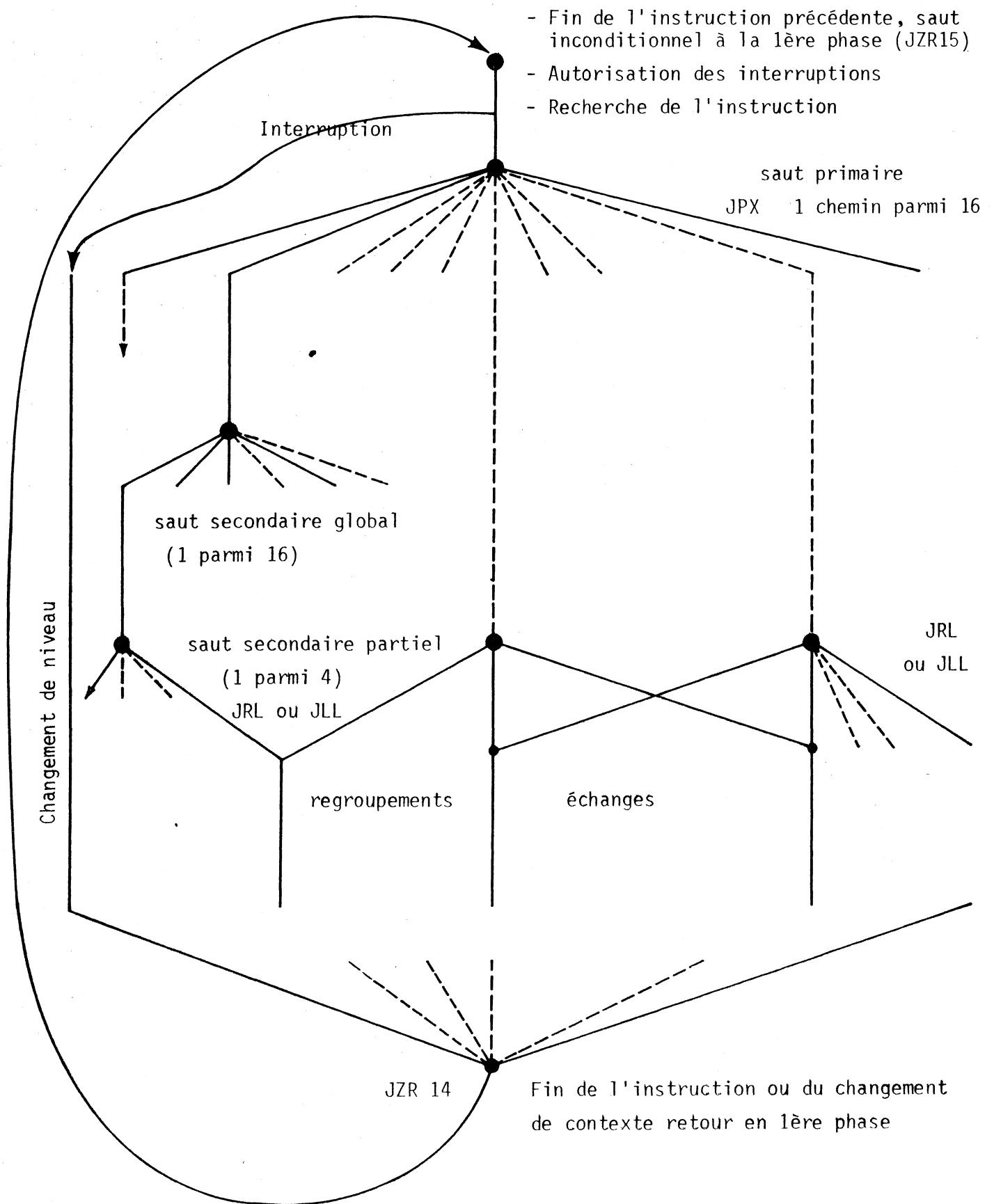


Fig.III.8 : Arbre de description du code opération

- que le gain obtenu était d'environ 3 (chaque case du jeu d'instruction a été en moyenne réutilisée trois fois)
- que la difficulté de microprogrammation supplémentaire était du même ordre de grandeur que tous les autres champs réunis (ceci double le temps d'écriture des microprogrammes)
- La méthode de répartition par classe a été appliquée "à la main", selon des caractéristiques macroscopiques des instructions (référence registre, référence mémoire, adressage immédiat etc...). Cette méthode se prête très mal à une interprétation entièrement automatisée des microprogrammes. En conséquence une conception légèrement différente a été adoptée pour l'écriture d'un compilateur-chargeur de micro-code (Réf.10). Cette dernière consiste à considérer chaque micro-instruction comme un segment et n'écrire une nouvelle micro instruction que si la réutilisation d'une ancienne est impossible. (cf. § IV).

d - La mémoire centrale

Utilisée pour les programmes comme pour les données, elle est extensible jusqu'à 32 k mots de 24 bits en adressage direct et 2^{24} mots en adressage indirect ou indexé.

Le principal but étant la coexistence de la fonction et de l'adresse ou de la donnée dans le même mot machine, il est clair que l'on cherchera à n'utiliser autant que possible que l'adressage direct, et donc de ne pas dépasser 32 k mots de mémoire centrale, ce qui par ailleurs s'est révélé largement suffisant étant donné que toute la partie logiciel de développement a été croisée sur mini ordinateur (cf. § IV).

Cette mémoire est à deux accès principaux en vol de cycle. L'un des accès est réservé à GESPRO, l'autre à CAMAC pour le chargement la maintenance, le développement et dans les cas les plus simples, l'acquisition (cf. expérience WA2, chapitre IV). Ceci permet de ne pas trop ralentir la machine tout en lui laissant des possibilités d'extension en multiaccès en greffant sur l'un ou l'autre des accès une plage mémoire multiaccès (cf. expérience NA10, ch.IV).

e. Le module d'interruption et les périphériques standards.

Ce module permet les changements de contextes rapides nécessités par l'environnement (communication avec l'expérience essentiellement).

Il fait partie des périphériques "standards" et utilise les bus D et I. Son efficacité provient d'une part de la non utilisation du bus A (le registre d'adresse mémoire n'est pas touché), d'autre part de la conception du matériel, conçu pour soulager le logiciel au maximum (Réf. 11) et enfin de la non utilisation du bus M : les données nécessaires au module (niveau courant, niveau futur, passent par I).

Prévu à l'origine pour 24 niveaux dont un à 24 sous niveaux. Ce module a été réalisé en 3 versions :

l'une à 24 niveaux pour WA2, quatre machines à 8 niveaux pour NA10 et enfin une machine à 16 niveaux pour le projet UA2.

La sauvegarde des contextes s'exécute en mémoire centrale où 8 mots par niveaux ont été préservés. Jusqu'à présent cinq mots seulement sont utilisés puisque seuls quatre registres et le compteur ordinal sont sauvés. Les quatre registres sont les quatre accumulateurs visibles au logiciel, ils peuvent, dans certains cas, être les quatre utilisés comme indexes.

Le temps de passage d'un contexte à l'autre est d'environ 3,5 μ s, ce qui est performant pour une machine ne possédant qu'un seul jeu de registres. Lorsque ce temps est prohibitif (cf. exp. NA10), une interruption prioritaire et ultra rapide est utilisable. Vue du logiciel comme le niveau zéro, elle est masquable et capable de lancer une tâche en 150 ns quelque soit l'instruction courante. Cependant :

- Le contexte précédent n'est pas sauvegardé
- Le microprogramme interrompu est irrécupérable

En conséquence cette interruption est utilisée en guise de déclenchement, lorsqu'aucune autre tâche n'est activée. Sur l'exp. NA10, Elle permet de ne pas allonger le temps mort et de ne pas trop ralentir le contrôleur câblé chargé de la gestion mémoire (cf. exp. NA10).

Un autre périphérique standard est composé d'une unité de décalage. Selon les expériences cette unité est très simple (Rotation d'un octet à droite en lisant la mémoire sur le bus I en place du bus M, avec masquage possible), ou beaucoup plus complexe, composé d'un registre à décalage combinatoire complet à 24 bits.

En résumé notre noyau se compose :

- d'une unité centrale à 24 bits
- d'une mémoire de microprogramme d'au moins 1 k mot de 48 bits en mémoire vive avec un accès CAMAC pour le chargement
- d'une mémoire centrale d'au moins 4 à 8 k mots de 24 bits (jusqu'à 32K mots)
- d'une unité d'interruption d'au moins 8 niveaux.

Compte tenu du nombre de circuits utilisés, cette unité de base nécessite une carte d'environ 50 x 30 cm. Ces cartes ont toutes été réalisées en wrapping à partir de fichiers descriptifs.

3. Le jeu d'instructions de GESPRO

Il se divise en trois parties :

La première concerne les instructions banalisées nécessaires à l'écriture de modules en mode programmé. Dans ce mode de fonctionnement GESPRO se comporte comme un ordinateur classique (temps d'exécution relativement lent).

La deuxième partie comprend les instructions spécialisées aux systèmes d'acquisition et de traitement. Ces instructions facilitent l'écriture des programmes tout en diminuant leur temps d'exécution. Enfin la dernière partie est spécifique à l'expérience en cours. Elle est écrite en grande partie pendant l'expérience, et est pratiquement inutilisable ensuite.

Ceci nous a conduit à utiliser une mémoire de microprogramme entièrement en mémoire vive (remise à jour constante) et extensible jusqu'à 2K mots de 48 bits (la troisième partie du jeu d'instructions peut être la plus volumineuse).

a. Choix du jeu d'instructions - partie banalisée

Cette partie du jeu d'instruction est donnée en ANNEXE. Elle est caractérisée par :

- une grande simplicité d'utilisation
- un nombre d'instructions relativement faible
- une puissance de traitement relativement importante par rapport aux mini-ordinateurs classiques, due à la structure multibus d'une part, à la longueur du mot d'autre part.

Quatre registres sont visibles au logiciel (contre 11 en micro-programmation). Ils peuvent servir d'accumulateur ou d'index.

La longueur du mot à 24 bits nous donne ici un triple avantage :

- structure à "opération-donnée ou opération-adresse" en un seul mot machine
- mode d'adressage réduit puisque tout l'espace de travail peut être adressé directement

Chargement en mode immédiat court sur un seul mot pour les données ou en deux mots sur 24 bits pour des commandes ou instructions.

- adresse CAMAC (IDC, F, B, C, N, A) sur un seul mot.

En conséquence le nombre d'instructions est réduit (seul les modes d'adressage direct, indirect et indexé sont nécessaires) et le temps d'exécution est amélioré.

La structure de sous-programmes est accessible par utilisation de sauts avec marquage, ce qui évite la gestion d'une pile et facilite le déverminage des programmes. Chaque sous programme débute donc par une case vide destinée à recevoir l'adresse de retour lors de son activation et se termine par un saut indirect à cette dernière.

Enfin les différents contextes des niveaux d'interruption sont rangés en mémoire à des adresses fixes ce qui permet de travailler facilement sur les différents contextes(chargement et modification) et évite à nouveau l'emploi d'une pile.

b. Instructions spécialisées système

Il s'agit essentiellement de la mise sous forme d'instructions d'un certain nombre de programmes utilisé fréquemment.

Le but est toujours de diminuer le temps d'exécution, en substituant la notion d'instructions évoluées à celle de sous-programme.

Le gain obtenu varie entre 3 et 10 ou plus suivant la longueur et la nature de la tâche.

Le tableau ci-dessous montre les résultats obtenus par quelques exemples :

Instruction	Description	Gain
CPZRO ADR CONF (2 mots)	Regarde si les bits qui sont à zéro dans CONF le sont aussi à l'adresse ADR. Saut d'une case dans ce cas.	3
CPONE ADR CONF (2 mots)	Regarde si les bits qui sont à un dans CONF le sont aussi à l'adresse ADR sont d'une case dans ce cas	3
STM ADR DATA (2 mots)	Initialise un tableau dont le nombre de mots est contenu dans le registre A à la valeur DATA. ADR est l'adresse du tableau	15
MOVE SADR DADR NBMOT (3 mots)	recopie complète d'un tableau depuis SADR vers DADR. Le tableau initial et le tableau final peuvent avoir une partie commune : le choix du sens de parcours est pris en charge par l'instruction.	10

Ce type d'instruction permet d'obtenir un compromis entre le temps d'exécution d'un programme et celui d'un microprogramme tout en étant plus facile à écrire qu'un programme assembleur classique.

c. Instructions spécifiques à chaque expérience

Il s'agit essentiellement d'algorithme d'acquisition ou de traitement. C'est la partie la plus importante et la plus efficace du jeu d'instructions, et c'est aussi la plus difficile à écrire car les algorithmes ne sont bien définis qu'en cours d'expérience où les conditions de travail ne sont pas toujours propices à l'écriture de microprogramme complexe. De plus l'environnement (interface Expérience, interface ordinateur hôte, système d'acquisition) ne peut pas toujours être dupliqué ou simulé en laboratoire. Pour toutes ces raisons cette partie du jeu d'instructions suit toute une procédure de test et de vérification "off-line" en cours d'expérience avant d'être définitivement activée. Le gain obtenu grâce à cette microprogrammation varie entre 10 et plus de 50 suivant les tâches.

Sur l'expérience WA2 (1976-1979) la lecture, et la vérification des données issue des TDC (convertisseur temps-numérique) aurait pris plus de 100 μ s/mot sur un ordinateur NORD10. Ce temps a été réduit à 7 μ s/mot en écrivant l'ensemble du traitement sur une seule instruction à débranchement multiple (Réf.3).

Sur cette même expérience une gestion DMA CAMAC, nécessitant plus de 200 μ s à un miniordinateur a pu être exécuté en 15 μ s par microprogrammation complète des instructions de démarrage et d'arrêt de la DMA. Enfin sur l'expérience NA10 le calcul complet de la masse des dimuons produites est exécuté en ligne en utilisant la formule exacte. (Réf.12-13-14).

$$(1) \quad M^2 = 2 \cdot P_1 P_2 (1 - \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 \cdot \cos(\varphi_1 - \varphi_2))$$

sur des lignes trigonométriques tabulées. Le calcul est effectué en flottant sur 24 bits grâce à un opérateur câblé combinatoire. L'ensemble prend 12 μ s alors que le même calcul, exécuté par un NORD 10, prend 120 μ s ce qui amène à un gain de plus de 100 μ s par calcul (cf. expérience NA10, chapitre IV).

4. Développement des outils logiciels associés à GESPRO

Etant donné les particularités de cette machine, et son format non standard, la totalité des outils logiciels était à créer de toute pièce. Nous avons utilisés le plus souvent possible les facilités de l'ordinateur hôte, un NORD10 sous un système SINTRAN.

Le logiciel GESPRO, autre que l'écriture du jeu d'instructions peut être divisé en quatre parties :

- Le logiciel propre à GESPRO. Il s'agit de micro-programme ou de programme écrit en langage GESPRO, destiné à la mise en route d'unité nouvelle, à la maintenance et aux tests de fonctionnement
- Le logiciel croisé sur NORD : écrit en langage NORD il se compose d'un ensemble d'outils d'aide à la programmation de GESPRO.
- Le logiciel interactif GESPRO ↔ NORD permet de contrôler le bon fonctionnement des parties communes aux deux machines, d'exécuter des programmes de tests évolués à la vitesse GESPRO en ayant les facilités de dialogue NORD et dans certains cas de tester les périphériques par programme à la vitesse d'une DMA.
- Le logiciel d'acquisition et de traitement, spécifique à chaque expérience.

L'ensemble de ces programmes, quelque soit leur langage d'écriture est écrit sur NORD et stocké sur mémoire de masse NORD.

a. Logiciel propre à GESPRO

Ce logiciel comprend tous les micro-programmes de mise en route, de test et de maintenance de la machine et de ses périphériques.

Les plus simples comprennent uniquement deux ou trois microinstructions et n'activent que quelques champs du mot mémoire de commande : Ils permettent de vérifier en laboratoire le comportement de l'unité centrale.

Puis viennent des programmes de test de plus en plus sophistiqués, mettant en oeuvre de plus en plus d'actions simultanées. Ces microprogrammes ont été écrits petit à petit au cours des différentes évolutions de la machine.

Les moyens de communication homme-machine évoluent également pour permettre des diagnostics plus évolués et plus complets, et passent de l'oscilloscope et de l'analyseur logique à un véritable dialogue NORD-GESPRO, utilisant généralement la mémoire GESPRO comme boîte aux lettres. Tous ces tests supposent une mémoire de microprogramme déverminée : ceci est obtenu par l'accès NORD permettant des tests pratiquement complets.

On peut ainsi contrôler successivement :

L'unité de contrôle de la mémoire de microprogramme et cette dernière.

Les busses de sorties de l'unité centrale A et D (donnée et adresse). Le fonctionnement des opérations arithématique et logique et des retenues.

On vérifie ensuite les busses d'entrée : Le bus de masque K puis le bus M (mémoire centrale) puis le bus I (périphérie, interruption).

Enfin un ensemble de programmes permet de tester chaque élément dans des conditions plus sévères que son utilisation normale. En particulier l'unité centrale et la mémoire centrale ont été l'objet de tests particulièrement sévères (Réf. 15 et 16). Ces tests permettent de vérifier que toute la mémoire est accessible, que toutes les configurations y sont correctement représentées. Ils permettent ensuite de s'assurer qu'une action ne dépend pas des précédentes ni des adresses où elles ont eu lieu.

L'ensemble de ces algorithmes, décrits dans les références 15 et 16, a été microprogrammé entièrement ce qui permet d'une part de rendre les tests beaucoup plus sévères et d'autre part d'obtenir un gain d'un facteur 20 environ en temps d'exécution par rapport au même programme écrit en assembleur sur un miniordinateur classique. Ceci n'est pas négligeable lorsque l'on sait qu'un seul test peut durer plus de 2 minutes et demi en microprogramme soit près d'une heure sur une machine classique, programmée en assembleur.

b. Le logiciel croisé sur NORD

Ce logiciel supporte tous les programmes d'aide à la microprogrammation et à la programmation de GESPRO.

Il comporte tout d'abord les programmes de test et de maintenance les plus simples. Généralement écrit en basique ou en FORTRAN ses programmes se contentent des vérifications sommaires nécessaires pour pouvoir utiliser le logiciel décrit en a.

Ils consistent en tests de la mémoire de microprogramme, de la mémoire centrale, des périphériques (CAMAC, multiplieurs.....).

Dans certains cas (cf. expérience NA10) ces tests ne sont possibles que grâce à une interface spécifique. C'est le cas lorsque l'unité à tester n'est, en fonctionnement normal, visible que par GESPRO et que la dimension de cette unité impose des tests spécifiques. Ainsi les tests de deux périphériques dessinés pour l'expérience NA10 ont imposé le dessin de deux interfaces supplémentaires (encodage d'une valeur maximale, multiplication rapide, cf. chapitre IV).

Ce logiciel comprend également un moniteur, permettant d'effectuer toutes les opérations simples sur GESPRO par l'intermédiaire des interfaces NORD :

- lire et écrire en mémoire de microprogramme
- lire et écrire en mémoire centrale
- décrire l'état de la machine
- démarrer-arrêter GESPRO
- charger un microprogramme (ou un jeu instructions) depuis le disque NORD
- charger un programme
- lancer un test mémoire.

Enfin le logiciel comprend les modules nécessaires à l'écriture des programmes et des microprogrammes en langage symbolique (Références 10 et 17)

Programmation symbolique

La programmation d'une machine telle que GESPRO, tout en étant aussi simple que celle d'une quelconque machine en assembleur, ajoute néanmoins le concept de multiplicité des jeux d'instructions.

Ce concept est la base du travail réalisé pour permettre la programmation de GESPRO sur un jeu d'instruction quelconque et la mise à jour de l'un quelconque des jeux d'instructions. En conséquence (Réf.17) il s'agit non seulement d'un macroassembleur interactif mais aussi d'un éditeur de jeux d'instructions permettant la mise à jour de ceux-ci dans une syntaxe aisée, grâce aux fonctions classiques à tous les éditeurs : adjonction, suppression, modification, impression des instructions, création, effacement recopie du fichier (jeux d'instruction) facilité d'édition du texte (syntaxe compatible avec l'éditeur NORD). Une syntaxe appropriée, permettant une description aisée de toutes les instructions a été recherchée. Cette syntaxe ne permet pas la description des instructions à adressage relatif, compte tenu de la particularité de GESPRO de pouvoir toujours adresser directement la totalité de son espace de travail (mémoire inférieure à 32K mots pour 15 bits d'adresse par mot de 24 bits à référence mémoire).

Tout ces programmes ont dû être optimisés en temps d'exécution pour minimiser les temps d'attente rendus plus importants par l'utilisation de l'assemblage croisé et utilisation du "file system" NORD et par l'utilisation d'une machine 16 bits assemblant les instructions d'une machine de 24.

Pour ces raisons ces programmes ont été en grande partie écrits en assembleur NORD.

Microprogrammation en langage symbolique

La difficulté de microprogrammation provient d'une part de la description exacte des différents champs, et d'autre part de l'implantation des microinstructions en mémoire de commande compte tenu du mode d'adressage particulier du M.C.U. permettant la réutilisation de cases préalablement utilisées pour d'autres instructions (cf. chapitre II).

Cette deuxième partie, la plus complexe, est responsable du temps de microprogrammation important lorsque l'on écrit en langage machine.

Afin de résoudre ce problème automatiquement par compilation croisée, un algorithme d'implantation en mémoire de commande instruction par instruction a été mis au point, après écriture d'un émulateur de M.C.U.

L'ensemble de ces travaux (Référence 10) a abouti à l'écriture d'un compilateur de microcode et du décompilateur associé.

c. Logiciel interactif NORD-GESPRO

Il comprend essentiellement les programmes de test et de maintenance les plus sévères et permet des diagnostics plus explicites et plus complets.

Les unités utilisables par plusieurs sources peuvent être testées en simultanéité ou vol de cycle par NORD et par GESPRO. C'est le cas de la mémoire centrale, qui dans certaines configurations peut être accessible par plusieurs accès différents (GESPRO, NORD, mode programmé, NORD DMA...).

C'est également le cas de certains périphériques spécifiques à une expérience (contrôleur mémoire sur WA2, "event buffer" sur NA10....).

D'autre part tout ce qui concerne les tests GESPRO les plus longs utilise NORD comme moyen de communication avec l'extérieur et la mémoire centrale GESPRO comme boîte aux lettres. On obtient ainsi un système souple et d'utilisation simple permettant des algorithmes de tests complexes, notamment pour :

- le vol de cycle en mémoire centrale
- les périphériques rapides (multiplieurs, combinatoire.....)
- l'unité et les microprogrammes d'interruption.....
- les périphériques spécifiques à chaque expérience (cf. chapitre IV)

d. Le logiciel d'acquisition GESPRO

Ce logiciel est propre à chaque expérience pour tenir compte de la structure de chaque chaîne d'acquisition, de chaque type de traitement et également du flux déterminant le temps d'exécution utilisable pour maintenir un temps mort raisonnable.

Ces logiciels sont décrits en détails dans le chapitre IV.

C H A P I T R E I V

APPLICATION AUX EXPERIENCES WA2 ET NA10 DU SPS DU CERN

I. EXPERIENCE WA2 - 1975-1979.

Collaboration Bristol, Genève, Heidelberg, Orsay, Rutherford et Strasbourg [1] .

1. A. Description de l'expérience et de l'appareillage (18)

Cette expérience a consisté à étudier les désintégrations leptoniques des hypérons chargés. Le faisceau d'hypérons est produit par l'interaction d'un faisceau de protons extrait à 200 GeV du SPS avec une cible en oxyde de beryllium. Il a été conçu pour produire des hypérons chargés négativement ou positivement. La machine fonctionne par cycles de 8,4 secondes avec une durée d'extraction de 1,5 seconde. L'intensité sur la cible hypéron est de $4 \cdot 10^{10}$ proton par cycle, ce qui produisait un faisceau de 10^6 à $1,5 \cdot 10^6$ particules chargées négativement.

L'appareillage utilisé pour cette expérience comprend (fig.IV.1) :

a. Le compteur DISC :

C'est un compteur Cerenkov spécial qui définit le déclenchement sur l'hypéron incident.

b. Les chambres faisceaux qui sont de petites chambres proportionnelles placées devant et derrière le DISC. Elles servent à mesurer la direction de l'hypéron signé par le DISC.

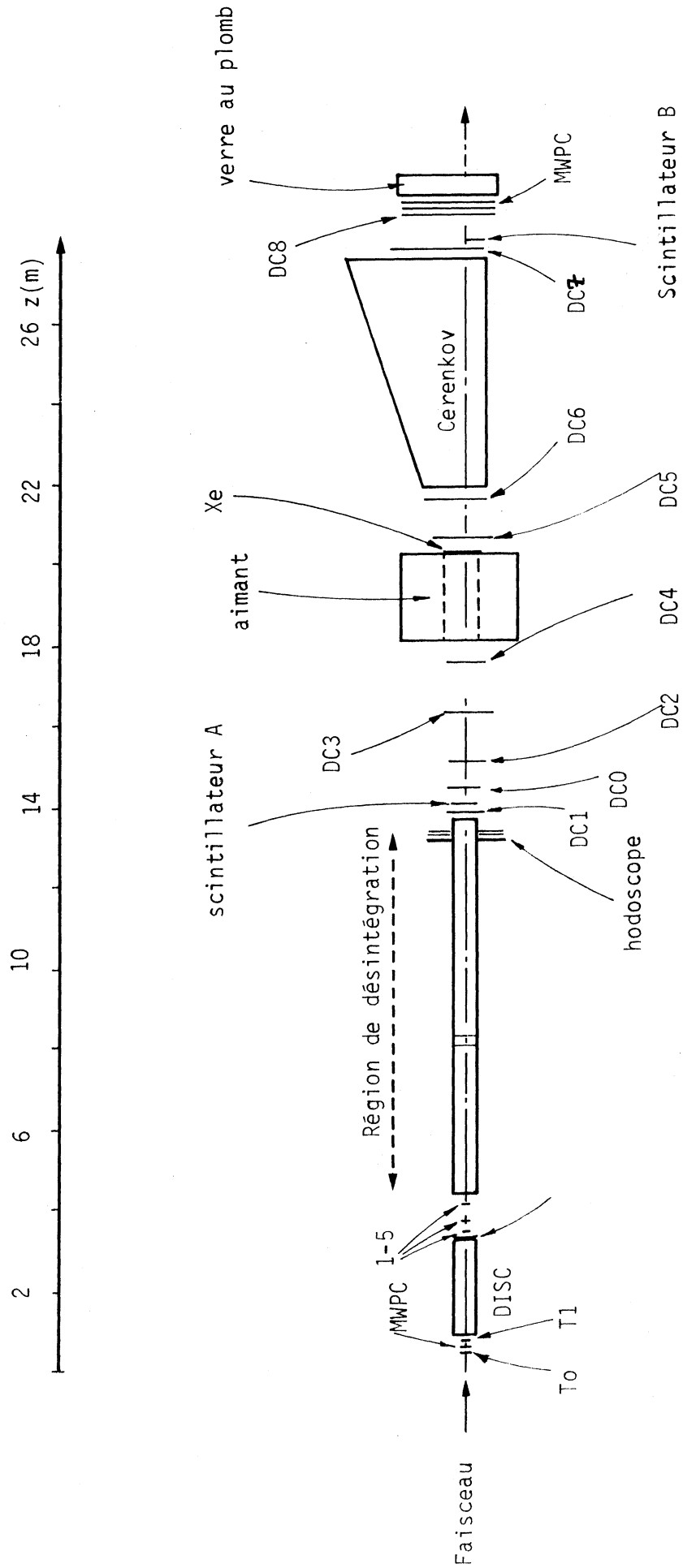


Fig.IV.1 : Appareillage de l'expérience WA2

c. L'aimant : Il dévie les particules chargées provenant de la désintégration des hyperons et permet de déterminer leur charge et leur quantité de mouvement. Long de 2,4 m il fournit un champ magnétique de 2,3 Tm.

d. Le télescope avant (fig.1, DC1 à DC4)

Il est constitué de 4 chambres à dérive réalisées à Strasbourg et sert à la mesure des angles et des positions des traces avant leur passage dans l'aimant. Ces chambres sont situées à 1,2 m les unes des autres. Elles mesurent trois coordonnées X, U, V. Chaque coordonnée est déterminée à partir de la mesure des temps de dérive sur deux plans de fils décalés de 1 cm. Ceci permet de résoudre l'ambiguïté gauche-droite. L'espace de dérive est de 5 mm ce qui, avec une vitesse de dérive de 50 μ /ns et une résolution en temps de 4 ns, correspond à une résolution dans l'espace inférieure à 200 μ sur la coordonnée spatiale.

e. Le télescope arrière (fig.1, DC5 à DC8)

Il est constitué de 4 chambres à dérive réalisées à Heidelberg et sert à la mesure des angles et des positions des traces après leur passage dans l'aimant.

f. Un mur de verre au plomb, complété par un détecteur à gerbes électromagnétiques chargé d'identifier les électrons. Cet ensemble est complété par deux détecteurs à radiation de transition, composés chacun d'un radiateur de lithium et d'une chambre proportionnelle à xénon.

g. Un compteur à Gaz Cerenkov permettant de séparer électrons et pions en dessous de 6 GeV/c.

h. Les chambres Gamma

Ce sont deux grandes chambres proportionnelles ($200 \times 100 \text{ cm}^2$) réalisées à Orsay et précédées chacune d'une plaque de plomb de 8 mm d'épaisseur. Placées juste avant le mur de verre au plomb, elles signalent et déterminent l'impact des photons.

i. Le compteur à neutrons

Construit à Orsay, ce compteur est formé de dix chambres proportionnelles séparées par des plaques de fer de 1 cm d'épaisseur.

1.B. Déclenchement de l'expérience

Le déclenchement hypéron est obtenu à partir de trois conditions principales :

La première est l'identification d'une particule sortant du canal par le DISC comme étant un hypéron. (Σ^- ou Ξ^- ou Ω^-).

Les deuxième et troisième conditions proviennent de deux conséquences de la désintégration elle-même, à savoir que l'état final comprend trois particules chargées dont un proton. Ces deux conditions sont matérialisées par un compteur à multiplicité sur lequel on applique un seuil correspondant à trois particules et un compteur à protons placé à 10 mètres derrière l'aimant, dans une région accessible aux seuls protons.

La grande majorité d-s données ont été prises à 100 GeV/c en polarité négative avec des flux de 10^6 particules par cycle machine au niveau du DISC, ce qui est le flux maximal tolérable par l'appareillage. La coïncidence de tous les déclenchements fournit un taux typique de 70 déclenchements par cycle machine.

1.C. Organisation du système de lecture (fig.IV.2)

a. Description

L'ensemble de l'information mesurée est stockée dans 80 registres, 3200 mémoires, 2200 voies TDC (convertisseurs temps-numérique) et 350 voies ADC (convertisseurs analogiques-numériques). Suivant le mode de déclenchement, cette information doit être lue 10 à 300 fois par cycle d'accélérateur (1,5 seconde de temps actif).

Cette fréquence implique l'utilisation des préprocesseurs de lecture ou "compacteurs" sélectionnant uniquement les données significatives pour chaque événement. Cette conception, à l'époque de la réalisation, n'était pas encore suffisamment répandue pour avoir été standardisée, ce qui fait que les modules "compacteurs" ont été développés après les interfaces avec les capteurs. De ce fait ils étaient différents et non accessibles par une DMA unique. Tous ces compacteurs ont été réalisés en standard pseudo-CAMAC et permettent de lire depuis un seul module vu par CAMAC plusieurs ensembles de châssis, en nombre bien supérieur à celui d'une branche. Deux des trois modules ont été réalisés par Orsay et sont affectés :

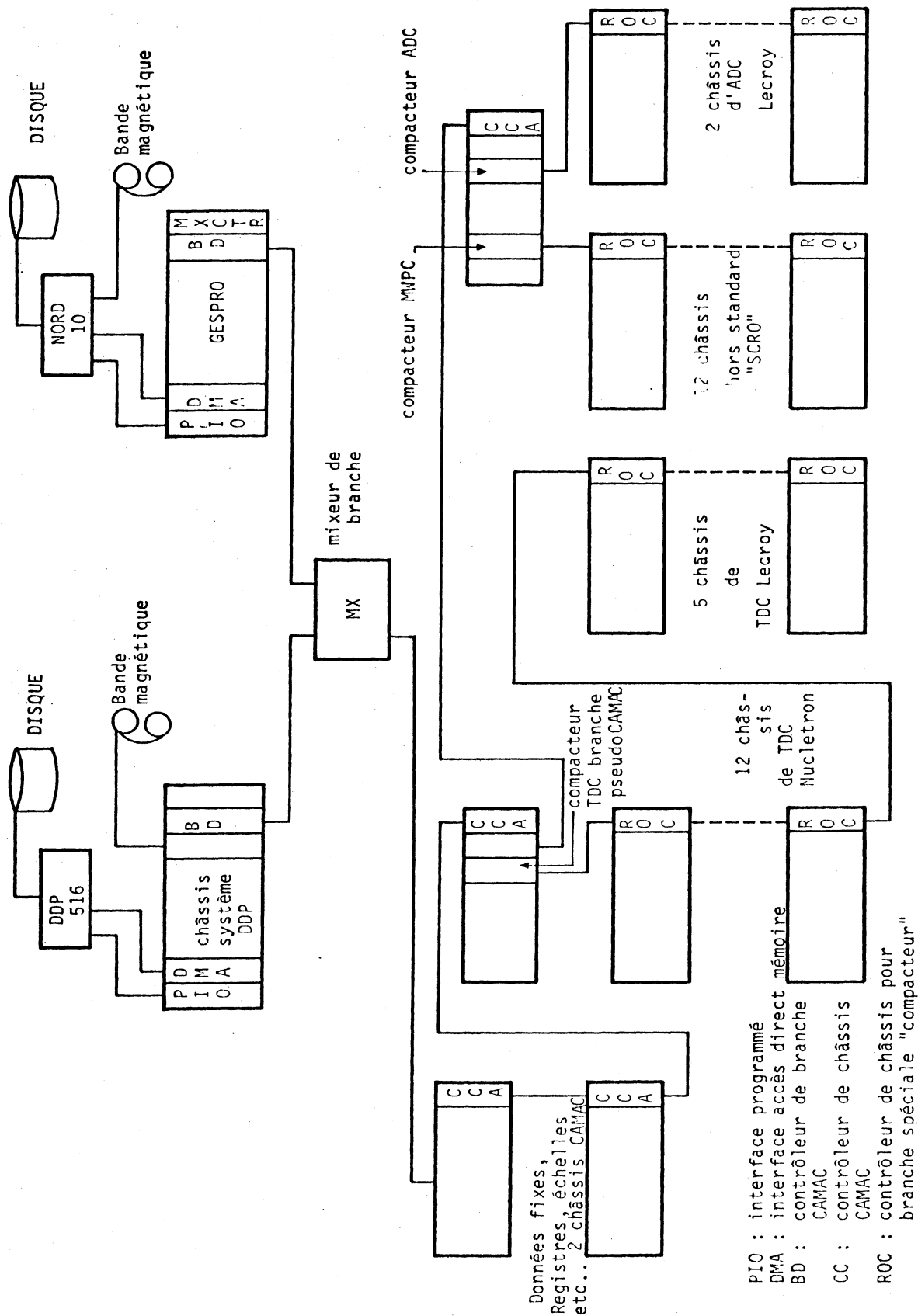


Fig.IV.2 : IIA2 : Organisation du système de lecture (seule la branche d'acquisition est représentée)

- le premier à la lecture des ADC avec suppression automatique des ADC non activés
- le deuxième à la lecture des MWPC avec suppression des fils non touchés et réarrangement des fils adjacents (cluster).
- le troisième a été fabriqué au CERN [5] et est chargé de la lecture de tous les modules TDC, tant ceux du télescope avant (TDC Lecroy 96 voies) que ceux du télescope arrière (TDC Nucletron 16 voies). Comme les MWPC et les ADC seuls les TDC touchés sont transférés vers le système d'acquisition.

b. Conséquences :

Les registres sont des modules CAMAC ordinaires, le nombre de données les concernant est constant et leur lecture en mode "auto-scan" ne pose pas de problème à la DMA CAMAC.

Les capteurs lus par compacteurs fournissent un nombre de mots variable sous forme d'adresses pour les données logiques ou d'adresses et valeurs digitalisées pour les données analogiques. Le nombre de mots lus est ajouté par le compacteur à la fin des données et des marqueurs dédiés repèrent les différents modules. Le processeur chargé du traitement sera donc contraint à lire l'ensemble des données pour pouvoir situer les différentes informations.

D'autre part les voies TDC, tout comme les fils des chambres à dérive coûtent fort cher, ce qui implique une optimisation du nombre de fils par plan de chambre par rapport au nombre de voies TDC disponibles par module.

Dans ces conditions le nombre de fils par plan n'est pas un multiple du nombre de voies TDC par module et tout programme de reconstruction de traces doit comporter une séquence de changement de base remplaçant l'adresse de la voie dans le module par l'adresse du fil dans le plan considéré.

Le nombre de fils par plans était de 116 et chaque module TDC contenait 96 voies pour le télescope avant. Nous verrons en détail sous E que ce changement de base joint à la lecture, au contrôle et au reformattage des données TDC a formé une des tâches les plus lourdes de GESPRO.

1.D. Philosophie du système d'acquisition

La conception du système d'acquisition découle de l'état de la technologie et des impératifs physiques de l'expérience.

La physiquenécessite la recherche d'événements rares en faisceaux intenses :

- En étude des omégas, pour 40 jours de prise de données à 10^6 particules par cycle machine $1,6.10^6$ événements ont été écrits sur bande magnétique et seulement 1500 Ω^- ont été retenus après l'analyse finale
- En prise de données leptoniques, pour 70 jours de faisceaux, $1,5.10^7$ événements ont été mis sur 600 bandes magnétiques, ce qui correspond à 4.10^{11} particules ayant traversé l'appareillage et a donné lieu à environ 150 000 événements leptoniques.

Dans ces conditions toutes les formes de filtrages sont recherchées, non seulement pour réduire le volume des données acquises, mais également pour réduire le temps mort et simplifier la tâche d'histogrammation puisque les histogrammes bruts (avant filtrage) ne correspondent pas à grand chose dans la mesure où ils ne comportent que très peu d'événements significatifs. Au moment de la conception (1975) les mini-ordinateurs, les mémoires vives et les systèmes de microprocesseurs en tranches coûtaient fort cher et avaient des capacités limitées.

Il n'était donc pas question de multiplier les processeurs. D'autre part le seul standard utilisable à l'époque était le standard CAMAC ou pseudo-CAMAC pour les compacteurs.

La seule possibilité d'introduction raisonnable d'un processeur de filtrage se situait au niveau du noeud du flux d'information constitué par le châssis système E.S.C. (Elliot Système Crate).

En effet, les conventions de ce système autorisent les modes de fonctionnement multi-branches, multi-sources et indirect (transfert de périphérique à périphérique). De plus l'interface DMA-NORD correspondant, dessiné au CERN, est utilisable par n'importe quel processeur ayant accès à l'Elliot System Crate (E.S.C.).

Le choix du miniordinateur utilisé s'est porté sur un NORD-10 pour les raisons suivantes :

- * Meilleurs rapports qualité-prix
- * Seul constructeur pouvant offrir pour le prix d'un mini-ordinateur (en 1975)
 - Mémoire virtuelle et pagination
 - Mémoire multiport (accès simultané)
- * compatibilité avec le CERN

Les charges du système ont été réparties comme décrites au chapitre I, à savoir le contrôle en ligne et la mise sur bande magnétique au NORD 10, l'acquisition, les gestions DMA, les gestions des mémoires et le filtrage à GESPRO.

La mémoire d'accumulation nécessaire pour tamponner le déversement était constituée d'un tambour magnétique, ce qui offrait pour l'époque le meilleur rapport vitesse-prix pour 256 K-mots

La mémoire du NORD 10 a été dynamiquement divisée en quatre partitions :

La première, fixe, comporte les programmes NORD

La deuxième contient les données en cours d'acquisition : la longueur de ce bloc est fixe, l'adresse initiale est allouée à GESPRO par NORD.

La troisième contient les données en cours de traitement NORD (histogrammatique, contrôle, communication à l'opérateur etc...)

La dernière contient les événements en cours de transfert vers le tambour.

Le tambour est vidé sur bande magnétique après le déversement du faisceau.

La gestion mémoire NORD, en ce qui concerne le bloc alloué à l'acquisition, est laissée à la charge de GESPRO. Cette conception, donnant au micro de grandes responsabilités, facilite beaucoup l'écriture des programmes d'acquisition et de filtrage puisque l'acquisition, les gestions DMA et la gestion mémoire sont réalisées par le même processeur (GESPRO) en adressage absolu.

En conséquence un système spécialisé a dû être écrit pour cette application. La synchronisation des tâches est réalisée par interruptions (externes, internes et de processeur à processeur).

Un canal mémoire (cf. § E) de GESPRO sert de boîte aux lettres pour permettre un jeu de messages entre NORD et GESPRO.

La figure IV.3. montre le système informatique complet, la figure IV.2. représente l'ensemble de la chaîne d'acquisition.

1.E. Configuration du microprocesseur GESPRO dans sa version WA2.

La figure IV.4 montre cette configuration où GESPRO est dédié CAMAC et inséré dans le châssis système. Il est organisé autour du noyau décrit précédemment avec 4K de 24 bits de mémoire centrale et 1K de 48 bits de mémoire de microprogramme.

Un ensemble de modules spécialisés à WA2 ont été ajoutés :

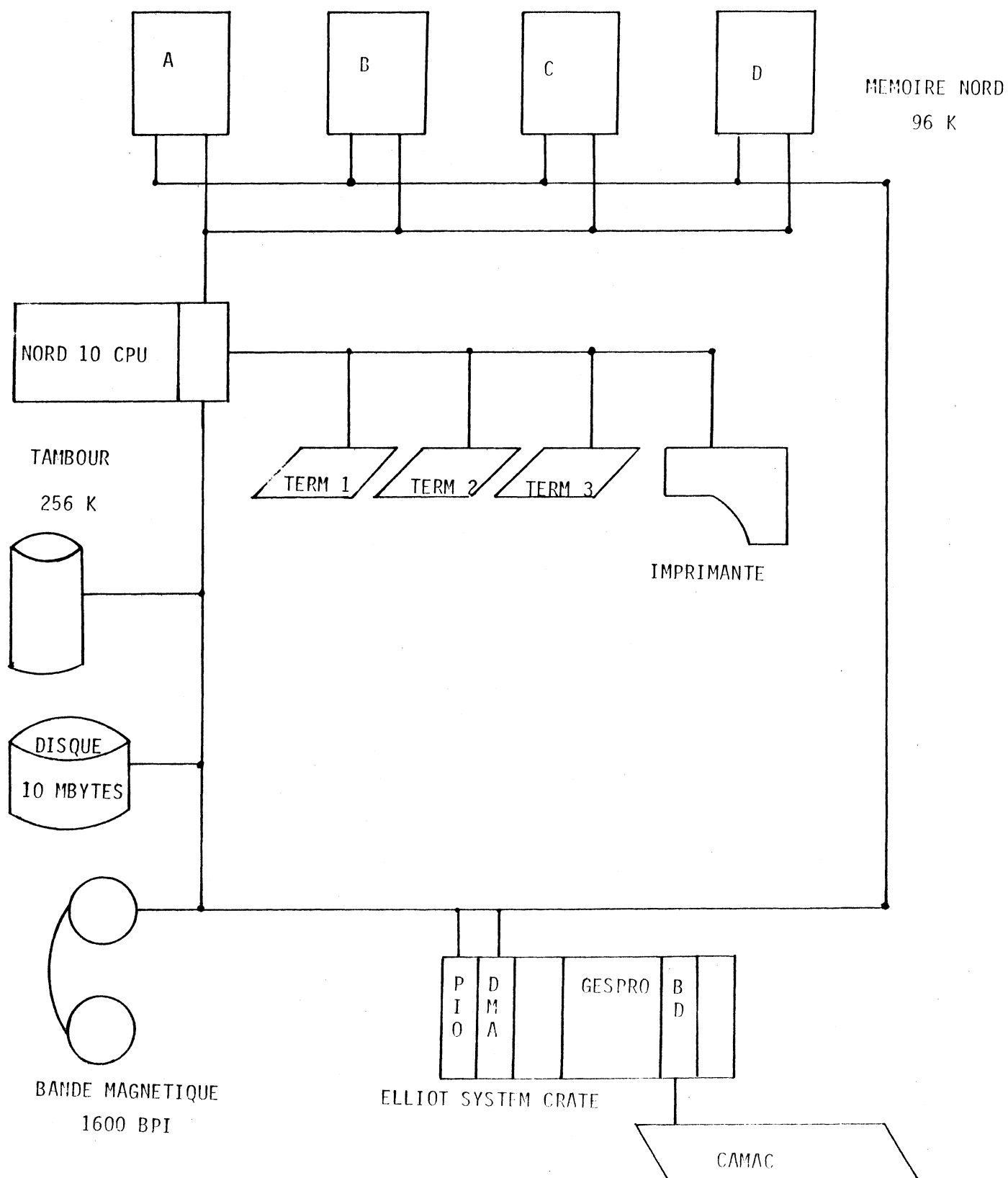


Fig.IV.3 : Système d'acquisition de l'expérience Hypéron 300

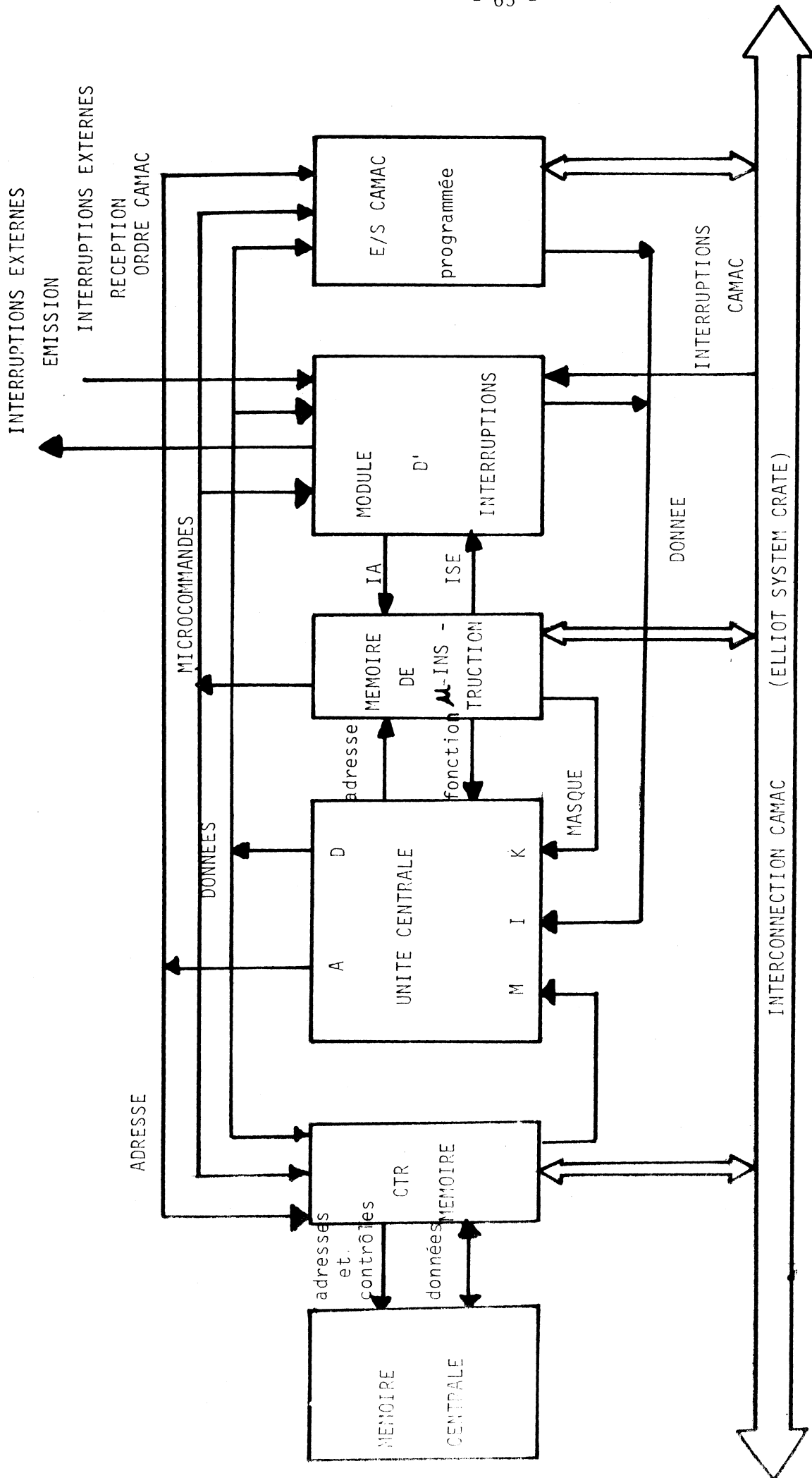


Fig.IV.4 : Configuration GESPRO Version WA2

- un tiroir DMA [19]
- un module d'entrée-sortie programmé [11]
- un module à 24 niveaux d'interruptions [11]
- un module de décalage
- un contrôleur mémoire à 8 canaux

Le tiroir DMA a été acheté dans le commerce en même temps que l'Elliott Système Crate, les autres unités ont été spécialement développées pour cette application :

a. Le module d'entrée-sortie programmée

Il permet d'exécuter les ordres CAMAC de lecture, d'écriture ou de fonction par programme et par microprogramme. Intégré à l'Elliott Système Crate (E.S.C.) il utilise ses conventions pour la logique de demande d'accès à l'interconnection à priorité série et pour l'envoi de l'ordre CAMAC sur les lignes W réallouées. En écriture, deux mots de 24 bits sont utilisés par GESPRO, l'un pour l'adresse, le mode et la fonction (IDC, F,B, C,N,A), l'autre pour la donnée. En lecture deux registres sont utilisés, l'un pour la donnée et l'autre pour les réponses Q,X. Ces données sont lues sur le bus I afin de ne pas interférer avec la mémoire. L'ensemble est synchronisé par câblage puisque le cycle GESPRO (150 ns) est beaucoup plus rapide que le cycle CAMAC (1 à 2 μ s). Pour ce faire un bit opérateur "utilisateur" (cf. chapitre III) décodé demande l'exécution d'un cycle. Le temps de réponse de 400 ns minimum a permis d'anticiper par câblage cette demande d'un cycle horloge. Les registres ordre, fonction et donnée d'écriture sont chargés à la fin de ce cycle, qui commande le signal d'occupation de l'interface.

L'horloge est resynchronisée sur S_1 , qui marque la fin d'occupation de l'interface et permet de lire les réponses Q et X. Dans le cas d'un cycle de lecture, les données sont lues lors du cycle suivant.

On remarque l'extrême rapidité des opérations puisque le démarrage anticipé de la demande masque une partie du précycle de l'E.S.C. et que le microprogramme se termine au maximum deux cycles après le début de S_1 soit avant la fin du cycle CAMAC lui-même.

En mode programmé, une instruction complète à référence mémoire et à quatre adresses de saut suivant les valeurs de réponses Q et X ne dure que 2,3 μ s.

Enfin le module d'entrée-sortie programmée utilise l'E.S.C. en mode I.D.C. Ce mode permet d'adresser deux périphériques dans le même cycle CAMAC, l'un en lecture et l'autre en écriture. D'où un transfert de périphérique à périphérique exécuté simultanément avec une lecture classique. Pour ce faire l'adressage du module "I.D.C." se fait sur 6 bits supplémentaires (les 6 bits à zéro imposent implicitement un cycle CAMAC ordinaire). Lorsqu'il reconnaît son code IDC pendant le précycle, le module concerné décode toutes les fonctions de lecture en écriture et inversement (ce qui correspond à l'inversion de la ligne F_{16}).

Ce mode de lecture, utilisé avec la DMA, nous a permis entre autre de transférer les données simultanément dans le processeur final (NORD) et dans le processeur de filtrage (GESPRO).

b. Le module d'interruption

Ce module fondamental permet de synchroniser les tâches sur les événements extérieurs et de les chaîner les uns aux autres. Il est notamment utilisé pour traiter les principaux signaux en provenance de l'expérience (début de burst, fin de burst, trigger) ainsi que pour les dialogues NORD-GESPRO (début de run, fin de RUN, pause, tambour plein etc...).

Cette unité à 24 niveaux dont 8 réservés au matériel et 16 réservés au logiciel permet une gestion automatique des niveaux de priorité (changement et sauvegarde des contextes transparents au logiciel).

Bien que n'ayant pas été utilisés lors de l'expérience deux des niveaux affectés aux matériels avaient été dédiés l'un aux appels CAMAC à raison de 64 sous-niveaux et 8 appels par branche et l'autre à des interruptions de communication par CAMAC (8 sous-niveaux).

Le module comprend donc en fait trois sous-modules dont seul le module de base, affecté à 6 niveaux matériels et 16 logiciels sera décrit ici.

Il comprend un registre d'état (mise en service - hors service), le registre de mémorisation des interruptions, un registre de masque pour tous les niveaux, la logique d'encodage et de synchronisation sur l'unité centrale et deux registres (niveau courant et niveau futur), permettant les activations des changements de contexte.

Les registres niveau courant et niveau futur sont directement visibles au logiciel sous la forme des adresses des contextes correspondants (fig.5)

Ce matériel optimisé permet d'exécuter un changement de contexte en 4 μ s.

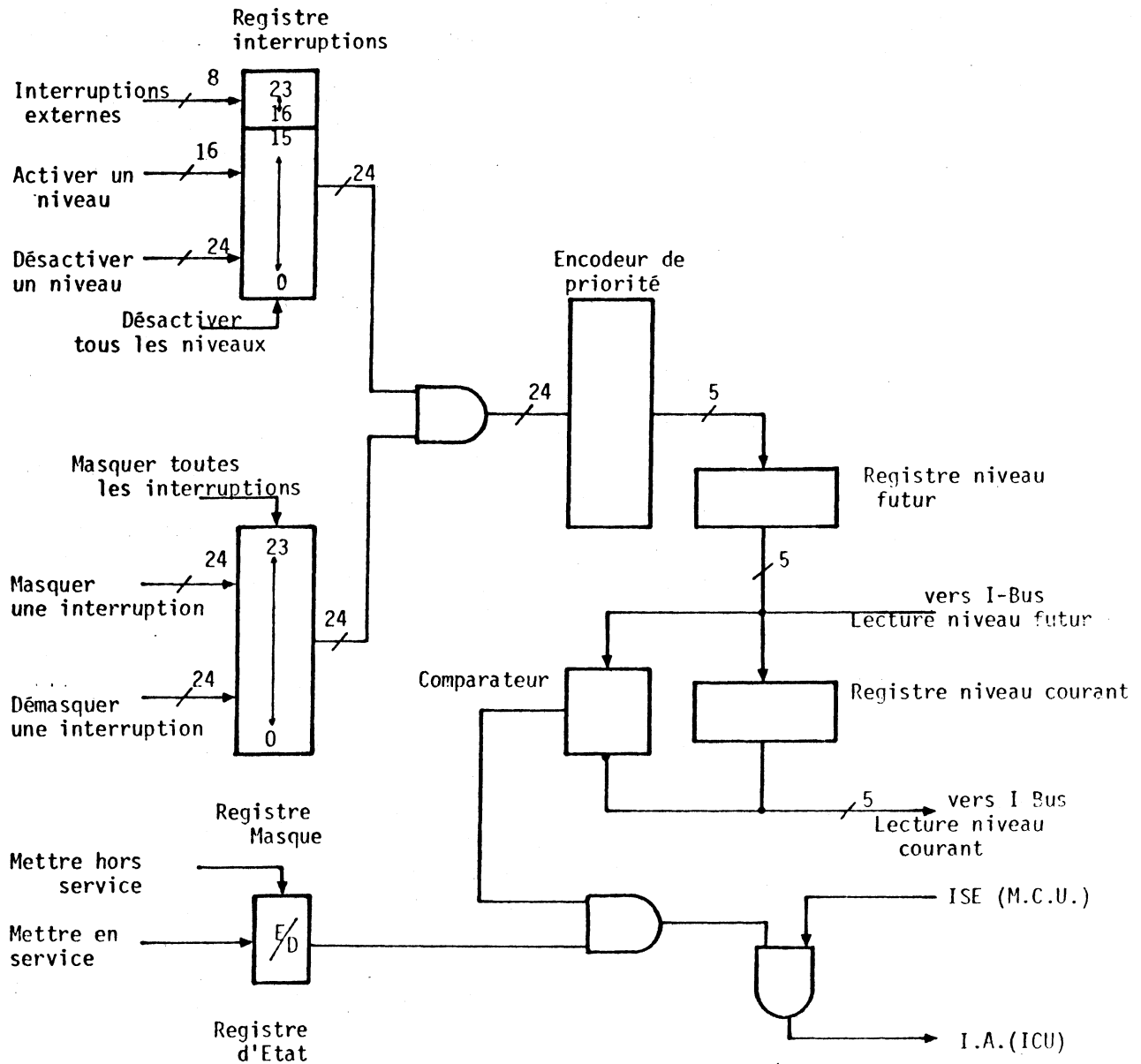


Fig.IV.5 :

Module de base des interruptions

c. Le shifteur câblé

Il s'agit en fait d'une rotation à droite du bus M vue par le bus I, et activée par un code opérateur. La lecture d'un mot mémoire est donc possible normalement par le bus M, mais aussi par le bus I, décalée de 8 bits à droite sans temps d'accès supplémentaire. Cet opérateur simple à exécution simultanée avec l'unité centrale est utilisée aussi bien pour les données que pour certaines instructions.

En effet les TDC fournissent le temps drift sur les 8 bits les moins significatifs et le numéro du fil dans le module sur les 8 bits les plus significatifs.

La lecture de ce numéro peut, grâce à cet opérateur être exécutée en une seule opération.

D'autre part, certaines instructions de transfert ou de lecture de blocs, écrites pour cette expérience, contiennent dans le même mot machine le nombre de mots du transfert et une adresse sur 16 bits. Ces deux informations sont chacune accessibles en une seule opération à travers l'opérateur de décalage, qui n'est constitué que d'une commande de multiplexage et d'un câblage croisé de M sur I.

d. Le contrôleur mémoire à 8 canaux

Cette unité permet l'utilisation de la mémoire centrale par 8 sources autonomes en mode CAMAC ou IDC dans l'E.S.C. Cette utilisation s'effectue en vol de cycle à l'unité centrale, avec en cas de conflit la priorité à CAMAC (accès le plus lent). Les différences de temps d'exécution (1,3 à 3 μ s pour CAMAC contre 150 ns pour GESPRO) imposent deux connections différentes.

L'une utilise l'interconnection du châssis système de l'E.S.C. l'autre un connecteur spécial sur la face avant (type I.U.B.) pour GESPRO.

Chaque canal d'accès est pourvu :

- d'un registre d'adresse mémoire (16 bits)
- d'un registre d'état de GESPRO par rapport à la source et de la source par rapport à GESPRO
- d'un registre de lecture (24 bits)
- d'un accès au registre d'écriture commun aux huit canaux. Les huit registres d'adresse sont auto-incrémentés avec pré-recherche au chargement pour permettre une utilisation en DMA.

Les huit registres de données sont indépendants pour permettre les entrelacements.

Les ordres CAMAC du controleur mémoire.

F 17	Ai	i = 8 à 15 :	charger l'adresse de base du canal i en mode lecture (Q = I , X = I)
F 21	Ai	i = 8 à 15:	charger l'adresse de base du canal i en mode écriture (Q = I , X = I)
F o	Ai	i = 8 à 15	lire une donnée par le canal i X = I Q=1 si le canal n'est pas en erreur (même lecture si Q = 1 avec F 16. IDC i en mode IDC)
F 16	Ai	i= 8 à 15	écriture une donnée par le canal i X = 1 Q=I si le canal n'est pas en erreur. (même écriture si Q = I avec Fo. IDC i en mode IDC).
F 1	Ai	i = 8 à 15	lire l'adresse courante du canal i
F 5	A8		réserver le premier canal libre lire son numéro sur les lignes R1 à R 3 Q = 1 (Q = 0 si aucun canal n'est libre)
F5	A9		même ordre que F5 A8 mais le numéro du premier canal libre est fourni sur les lignes W19 à W22 pour une utilisation future en mode IDC
F12	Ai	i = 8 à 15	libérer le canal i (Q = X = 1)

Fig.IV.6.

Un allocateur de ressource câblé permet, par le biais des lignes de dialogue, de rendre complètement transparent ces 8 accès.

La mémoire, vue du côté utilisateur de GESPRO, se comporte comme une mémoire monoaccès.

Connectée au contrôleur par le connecteur I.U.B., elle est modulaire et extensible avec un maximum de 32K mots (4K seulement ont été utilisés sur cette expérience).

Enfin un certain nombre de fonctions de service ont été implémentées afin de faciliter l'utilisation du module et de réduire le temps d'exécution.

Les canaux ont été hiérarchisés et une fonction CAMAC permet de lire le numéro du CANAL de plus haut rang non utilisé et de le réserver (réponse $Q = 0$ si les 8 canaux sont occupés).

De plus ce numéro peut être lu sur les lignes R_0 à R_3 (utilisation en mode CAMAC) ou R_{19} à R_{22} (utilisation en mode IDC).

La figure IV.6. résume les ordres CAMAC utilisés.

Sur les huit canaux indépendants disponibles, l'un est réservé aux communications entre NORD et GESPRO, un autre à la DMA GESPRO-NORD (événements traités), un troisième à la DMA d'acquisition (capteurs-mémoire GESPRO) et un quatrième lorsque l'on utilise cette DMA en mode IDC.

f. Mise en oeuvre de GESPRO

Quatre tâches principales ont été confiées à GESPRO :

L'acquisition, la détection des erreurs matérielles, le formatage des événements et le filtrage des données.

a. L'acquisition

La lecture des données est effectuée sous la responsabilité de GESPRO seul. L'accès CAMAC-NORD étant réservé aux tests, aux initialisations et aux dialogues fonctionnels NORD-GESPRO.

Puisque la notion de traitement de données et de filtrage d'événements a été introduite au niveau GESPRO, l'acquisition ne peut être dissociée de la gestion mémoire. Cette dernière est facilitée par l'utilisation de la DMA banalisée de l'E.S.C. et par le système NORD spécialisé écrit pour cette application et allouant dynamiquement un bloc mémoire à l'acquisition.

Deux types d'acquisitions sont utilisés, suivant que les informations lues sont à modifier par GESPRO ou simplement à transférer après vérification.

- L'information n'est pas modifiée :

La DMA est lancée par GESPRO en mode IDC. Les données sont donc transférées simultanément dans NORD et dans GESPRO. Si la vérification est positive, aucune intervention n'est nécessaire et l'acquisition continue. Si l'événement doit être rejeté, il suffit de remettre l'adresse DMA à sa valeur initiale pour que l'événement suivant se place en surimpression.

- L'information doit être modifiée :

Les données sont alors transférées dans la mémoire de GESPRO uniquement, soit par DMA (ce qui est possible avec la DMA de l'E.S.C. en mode IDC), soit par microprogramme. Le traitement est alors interne à GESPRO et le résultat est transféré par une DMA de processeur à processeur via CAMAC.

Ces deux types d'acquisitions peuvent être exécutés simultanément et laissent l'unité centrale NORD entièrement libre et non ralentie par vol de cycle puisque les accès CPU et DMA utilisent deux blocs séparés en mémoire "multiport".

b. La détection des erreurs matérielles

Il s'agit de détecter de façon simple et rapide les défauts de fonctionnement de la chaîne d'acquisition et d'en faire l'histogramme dans une table accessible à NORD. Ceci permet de contrôler l'efficacité du système de lecture au niveau de chaque événement et non par simple échantillonnage. Les réponses Q et X CAMAC, la vérification des mots d'états des principaux modules permettent d'obtenir un premier diagnostic, la figure 7 montre la table d'histogrammation GESPRO vue par NORD et retransmise périodiquement sur bande magnétique pour contrôle à l'analyse.

Ces tests sont suivis de vérifications plus évoluées de consistance des données :

- . Les registres forment un bloc de données de longueur constante, ce qui est facilement vérifiable par lecture du registre d'adresse de la DMA. Un marqueur formé par un registre de valeur fixe a été ajouté en fin de données afin de vérifier l'exactitude de la lecture.
- . Les quatre autres ensembles de détecteurs (ADC, MWPC, TDC Strasbourg et TDC Heidelberg) sont testés pour rejeter les événements vides ou en dépassement ou ceux pour lesquels le compacteur n'a pas fonctionné

Table des erreurs trouvees par GESPRO

CAMAC.	PIO 0	FAT.ERROR 5	DMA TIME OUT 0	DMA PX/RE 0	DMA CT/MT 0	NB JUM.BURST 0
FDATA.	LENGT 0	LAST WORD 0				
ADC.	NO DATA 0	OV. FLOW 0	TIME OUT 122			
MWPC.	NO DATA 228	OV. FLOW 3417	TIME OUT 0			
TDC.	NO DATA H 0	NO DATA S	OV.FLOW H 1276	OV.FLOW S 1766		
TDC.	OVERFLOW 0	TIME OUT 193				
TDC.	TO MANY 0 0	NOT ENOUGH 0 0	TO MANY-1 0	NOT ENOUGH-1 0	DECR.ADD.H. 20	DECR.ADD.S 0
REJEC.	- 1 FOL.BY 0 0	ROBD 0				
REJEC.	TOTAL 15144	F DATA 0	ADC 122	MWPC 11375	TDC 3647	
REJEC.	DMA TDC 0	DMA HEAD 0	DMA A/W 0			
TRIG.	NB TOTAL TRIG 28661	GOOD.EVEN. 13517	NB REJEC 15144	NB REJEC COND 0	MULT TDC 9876	HIT MWPC 7730

Resultats du run Omega janv. 1978

Declenchements	10292	100%
Evenements enregistres	2905	28%
Evenements rejetes	7387	72%

Detail des rejets

Rejets	7387	100%
Erreurs declenchement	1095	14%
Mauvaises donnees	2212	30%
Criteres physiques	4084	56%

correctement (lignes 3,4, 5 et 6 du tableau - Fig.IV.7)

. Enfin des vérifications plus poussées ont été mises en place, afin de vérifier la séquence de lecture elle-même : en effet chaque module TDC est lu par le compacteur par ordre d'adresse croissante. Les adresses des fils touchés doivent donc, entre deux marqueurs de changement de module, être toujours croissantes (tableau ligne 7)

Les séquences des marqueurs eux-mêmes (changement de module et fin de donnée d'un module) sont également vérifiées ainsi que le nombre de mots total (lignes 7 et 8).

c. Formattage de l'événement :

Comme nous l'avons déjà signalé, les compacteurs effectuent une lecture séquentielle au cours de laquelle ils peuvent ajouter des informations (marqueurs par exemple). Par contre, n'étant pas pourvu de mémoire d'événements, ils n'indiquent le nombre total d'informations qu'après lecture du dernier mot. Ce format convient très mal à l'analyse ultérieure (lecture mot à mot de tout l'événement pour situer les informations).

Cette tâche a donc été confiée à GESPRO qui placé au noeud des informations et pourvu d'une mémoire d'événements peut en modifier aisément la structure.

On a pu ainsi structurer l'événement sous forme d'une entête indiquant en premier lieu le nombre total des mots ainsi que toutes les indications susceptibles de faciliter l'analyse : Pointeurs vers chaque détecteur, nombre de mots de chaque détecteur, numéro de l'événement, numéro de version de programme GESPRO, nombre de mots ajoutés par le traitement GESPRO.

Cette façon de faire permet une évolution dans le temps avec modification de la structure et du nombre des informations sans modification du programme de lecture hors faisceau. Ceci nous a permis d'implémenter les différents algorithmes de vérifications, de filtrage et de traitement tout au long de la prise de données, au fur et à mesure qu'ils étaient formulés par les physiciens.

d. Les rejets (fig.IV.7 et IV.8)

A l'époque, il a été facile de faire admettre qu'il était utile de détecter les erreurs matérielles et même de jeter ces événements.

Par contre il a été plus difficile de faire admettre les filtrages sur des critères plus physiques ainsi que des traitements modifiant les valeurs des données.

Néanmoins, nous avons pu faire admettre de ne plus transférer sur bande magnétique les données TDC brutes (non traitées) et d'installer un certain nombre de microprogrammes de filtrage.

La figure IV.9 montre le traitement réalisé sur les données TDC. Chaque mot lu par le compacteur contient le temps de dérive et le numéro du fil dans le module TDC. Il est transformé en un autre mot contenant le même temps de dérive mais contenant en partie haute le numéro du fil dans le plan de chambre correspondant. Le décompte du nombre de fils touchés par plan permet de simplifier le programme d'analyse et d'appliquer un algorithme de rejet sur le nombre de fils touchés par plan.

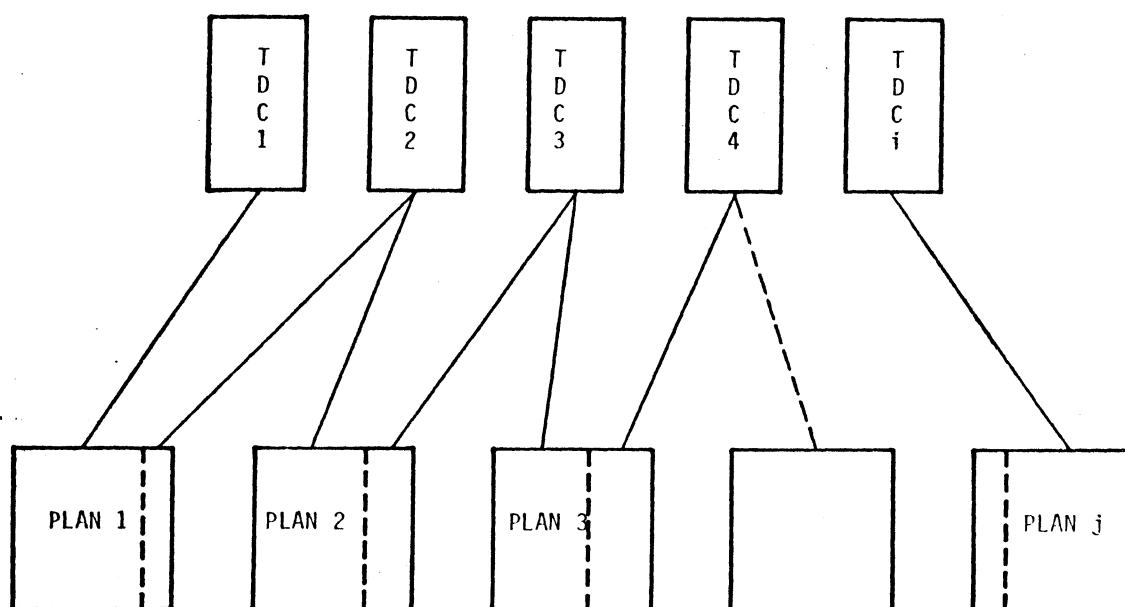
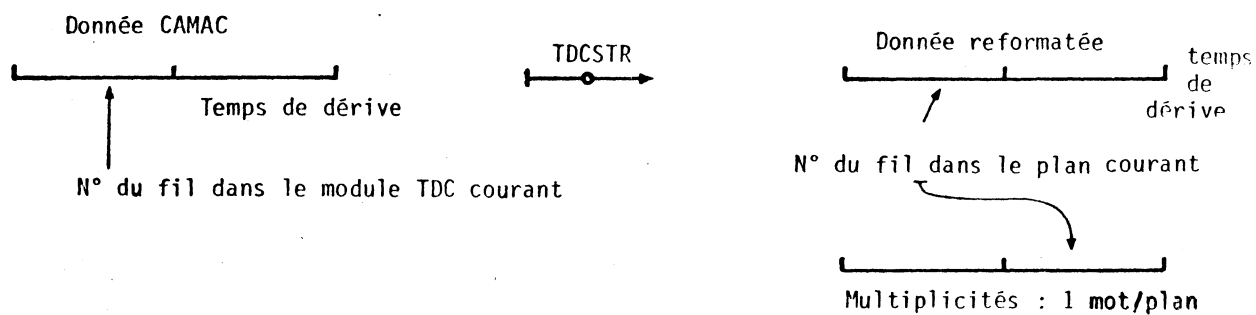
- Une coupure en multiplicité sur les chambres à dérive enrichit les événements à 3 traces devant l'aimant
- Une coupure en multiplicité maximum sur chaque détecteur (télescope AV, télescope AR, ADC, MWPC) et sur la longueur totale de l'événement permettent d'éliminer les événements inutilisables par suite d'un trop grand nombre d'informations.

Toutes ces coupures ont été mises au point en respectant toute une série de prudentes vérifications avant rejet définitif.

- 1) Analyse hors faisceau qui permet de déterminer les algorithmes
- 2) Procédure de test où tous les événements sont transmis sur bande, mais où un mot "drapeau" indique si l'événement aurait été conservé ou non et pourquoi. Cette procédure facilite le déverminage du matériel ainsi que l'analyse fine des rejets.
- 3) Rejet conditionnel :
C'est la procédure définitive qui permet de rejeter les événements en prise de données et de revenir en cas de panne ou de vérification à la procédure décrite sous 2).

Toutes les indications nécessaires sont également retransmises dans la table des statistiques GESPRO, périodiquement mise sur bande.

Enfin, lorsque l'ensemble des algorithmes devient définitif, on peut procéder à une optimisation en temps d'exécution.



1 TDC → 96 fils

1 PLANE → 116 fils

Figure IV.9

Traitement des convertisseurs temps-numérique (T.D.C.)

La figure IV.10 montre le résultat d'une telle optimisation. Elle compare les temps d'exécution obtenus par rapport à un système de lecture classique (gestion mémoire et DMA par un mini-ordinateur).

Les registres, les ADC et les MWPC sont lus en DMA, en mode IDC et les TDC sont lus par microprogrammes. Le traitement de ces derniers est effectué donnée par donnée en cours de lecture.

On remarque la simultanéité des tâches, traduite par :

- la lecture des registres pendant les initialisations programmées
- la lecture des ADC pendant la lecture et le traitement des TDC du télescope AR
- la lecture des MWPC pendant la lecture et le traitement des TDC du télescope AV
- le transfert des données TDC traitées de GESPRO vers NORD pendant les traitements ADC et MWPC.

On remarquera enfin que lorsqu'un algorithme donne lieu à rejet la fin de lecture est anticipée et on assiste à une diminution du temps mort.

L'ossature du programme est écrite en assembleur, les traitements des données ainsi que les initialisations des tableaux et les lancements des DMA sont microprogrammés.

Les résultats obtenus par microprogrammation d'une dizaine d'algorithmes sont considérables puisque pour 3 millisecondes de temps moyen d'exécution, il y a environ 8000 exécutions de micro-instructions et 800 exécutions d'instruction, soit un taux de microprogrammation d'environ 90 %.

I.5. Résultats

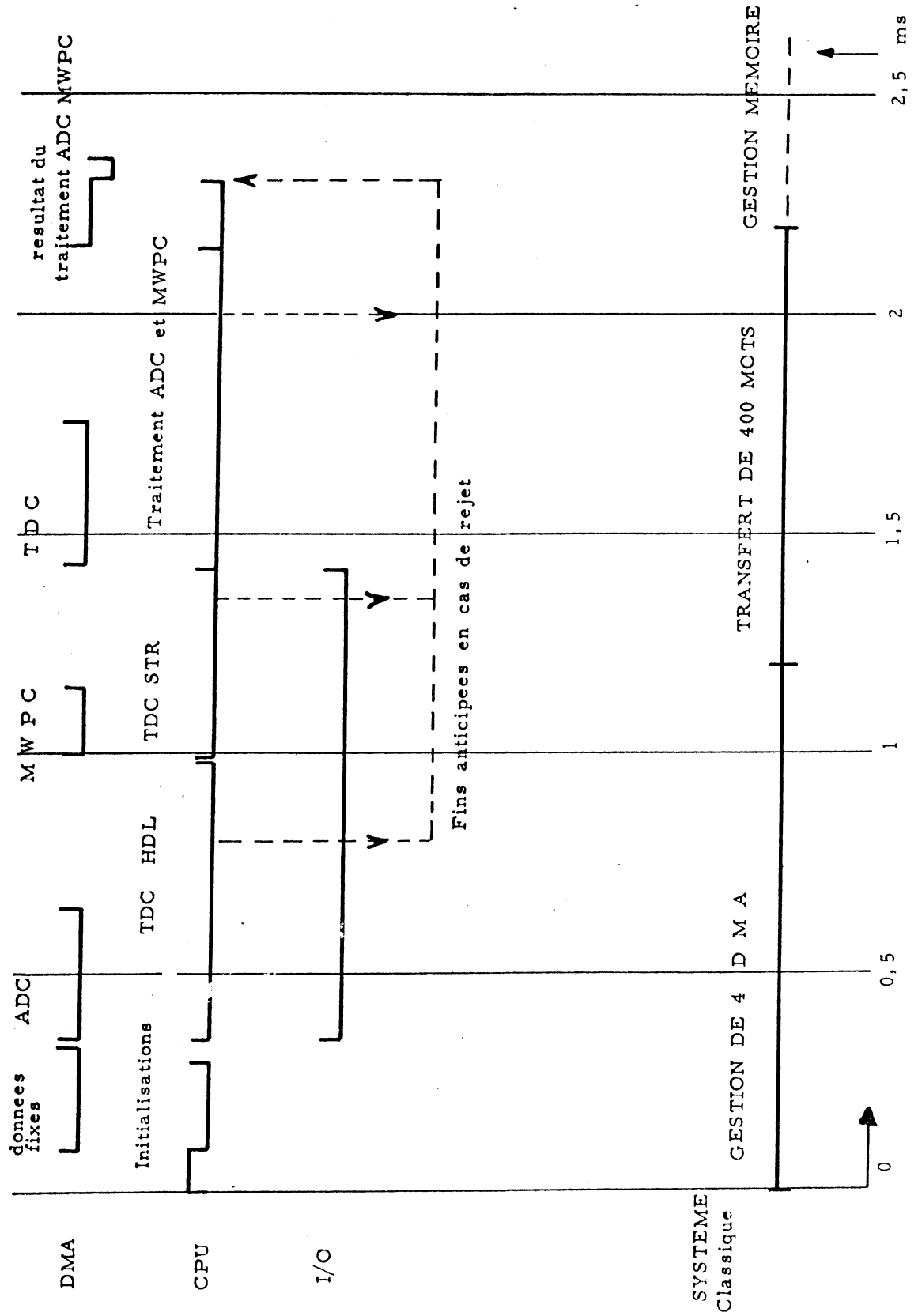
a. Taux de microprogrammation et facteur de qualité

pour un taux de microprogrammation de 90 %, le facteur de qualité défini comme le rapport du temps d'exécution en mode programmé au temps d'exécution réel est d'environ 10.

b. Le gain en temps mort

La figure 10 montre le chronogramme du programme d'acquisition par rapport à une lecture classique par mini-ordinateur en DMA. Le temps d'exécution est à peu près le même si aucun événement n'est rejeté (2,5 ms de temps d'exécution). Par contre le temps mort diminue lorsque le nombre de rejets augmente. En analysant le "RUN OMEGA" de janvier 1978 dont les résultats sont donnés fig.8 on constate que le temps mort a été réduit à 1,94 m. Soit un gain en temps mort de 22 %.

FIG. IV.10 : Chronogramme de l'acquisition WA2



c. gain en efficacité :

Le nombre d'événements validé par GESPRO et écrit sur bande par rapport au nombre d'événements sélectionné par le compteur Cerenkov DISC est d'environ 1/3. On peut classer (fig.8) ces rejets en trois catégories :

- la somme de toutes les détections d'erreurs (erreurs de DMA, erreurs de compacteur, de transfert de données, d'absence de données dans un capteur) participe pour 14 % à ces rejets
- la lecture de mauvaises données : (événements fortuits) empilement de déclenchement, événements trop grands, fournit 30 % de rejet
- l'application d'algorithme correspondant à des critères physiques simples (fenêtre d'amplitude, multiplicité...) apporte 56 %.

d. gain en histogrammation :

La figure IV.11 montre les différents temps d'exécution NORD en histogrammation ainsi que le pourcentage d'événements histogrammés par rapport au nombre d'événements mis sur bande.

On remarque que l'on gagne un facteur 8 sur ce rapport, ce qui augmente considérablement la qualité des histogrammes et permet un meilleur contrôle de l'appareillage. D'autre part les événements histogrammés ont été filtrés ce qui se traduit pour une plus grande lisibilité des histogrammes.

e. gain en temps d'analyse :

Pour cette expérience, le temps d'unité centrale gros calculateur consommé pour fabriquer les "bandes résumées" D.S.T. était à peu près proportionnel au nombre d'événements. Ce temps a été réduit d'un facteur 3 ce qui correspond à 100 heures unité centrale au lieu de 300, soit une économie de 200 heures CDC 6600 ; le gain correspondant au reformatage n'a pas été chiffré, étant donné la difficulté d'écriture d'un autre programme analyse.

f. gain en statistique des erreurs :

Pour chaque événement une table détaillée est remise à jour. Toutes les erreurs détectées y figurent ainsi que les résultats des traitements. Cette table est communiquée à NORD et mise sur bande magnétique à une fréquence programmable. Histogrammée par NORD elle permet un diagnostic rapide et sûr des incidents de fonctionnements des détecteurs et des chaînes d'acquisition.

1 6 Les faiblesses du système Hypéron :

- a) La mémorisation des informations des capteurs était faite de façon assez hétéroclite. Ce matériel, tout en étant réalisé dans la norme CAMAC, était lu par une multitude de procédures différentes qui ont considérablement alourdi le système d'acquisition.
Les rejets logiciels réalisés par GESPRO ne pouvaient être faits avant lecture de l'information du détecteur sur lequel ce rejet devait être réalisé. C'est pour cela que la séquence des rejets a été organisée pour traiter en priorité les capteurs dont l'information conduisait à la meilleure rejection. Ceci conduit à un gain limité en temps actif.
- b) Le taux de microprogrammation aurait pu être beaucoup plus élevé et le programme réduit à une dizaine d'instructions. Ce défaut provient des difficultés de microprogrammation de la machine ainsi que du manque de temps entre la définition des algorithmes et la production de données (nous ne disposions pas, pour cette expérience, de compilateur de micro-code).
- c) Deux choix ont limité considérablement les performances. Ces deux choix proviennent de l'époque de la conception (1974). En effet, à cette époque il était difficile de prévoir les évolutions technologiques pourtant toutes proches, concernant en particulier les mémoires vives et les vitesses des circuits. Ainsi le choix s'est porté sur un tambour en guise de mémoire d'accumulation ce qui pour une capacité de 256 K mots serait actuellement avantageusement remplacé par de la mémoire vive. Enfin la vitesse de transfert détecteurs-GESPRO est celle de CAMAC dans une branche de l'E.S.C. soit 2 us/mot environ. Cette vitesse limite les performances intrinsèques du système d'acquisition. Il faut remarquer cependant que le nombre de mots acquis à chaque burst ne justifiait pas une plus grande vitesse de transfert. D'autre part une partie de ce temps est récupéré en traitement grâce à la structure du processeur de filtrage.

Ces faiblesses peuvent être corrigées par l'emploi de plusieurs unités de filtrage simultanées et par un accroissement de la vitesse de lecture avant filtrage ce qui permet l'adaptation des flux.

Enfin un accès plus rapide entre le matériel de déclenchement et GESPRO peut permettre d'affiner le déclenchement par programme avant même la fin de la lecture rapide des données.

WA2	Temps d'histogrammation (en s) relatif aux chambres à dérive avant et après l'aimant			
longueur de de l'évène- ment	avec GESPRO		sans GESPRO	
	Télescope	Télescope	Télescope avant	Télescope arrière
267	0,34	0,38	0,5	0,92
481	1,25	1,62	1,90	4,21
310	2,97	2,96	4,51	7,83
pourcentage d'évène- ments his- togrammées	8 %		1 %	

Fig.IV.11. Gain en histogrammation

II. EXPERIENCE NA10 (1979.....) (Réf.5)

Collaboration CERN-Ecole Polytechnique (Palaiseau) - Université de Naples -
Université de Strasbourg - Ecole Polytechnique (Zürich)

II.A. Description de l'appareillage

Le but de cette expérience est l'étude de la production de paires de muons de hautes masses (au dessus de $4 \text{ GeV}/c^2$), à l'aide d'un faisceau intense de pions (jusqu'à $10^{10} \pi^-$ par pulse).

L'appareillage utilisé a été construit pour obtenir une grande acceptance et une haute résolution en masse. Il est représenté fig.IV.12. Il comprend :

- 1) une cible constituée d'un barreau de tungstène
- 2) un absorbeur de 4,8 m constitué de tungstène, de graphite et de fer destiné à éliminer les particules qui ne sont pas des muons.
- 3) quatre hodoscopes hexagonaux (R_1 à R_4) formés de six sextants chacun, orthogonaux à l'axe du faisceau et constitués de lattes de scintillateurs parallèles, reliés à des photomultiplicateurs.
- 4) un électro-aimant toroïdal constitué de six sextants égaux disposés autour de l'axe du faisceau.
- 5) huit chambres proportionnelles PC1 à PC8 constituées chacune de trois plans de fils parallèles répartis selon trois directions, chacune formant un angle de 60° avec la précédente.

Les quatre hodoscopes sont utilisés pour le déclenchement de l'acquisition.

II.B. Le déclenchement

Le déclenchement sélectionne les dimuons de haute masse en provenance de la cible. Ces événements sont noyés dans une grande quantité de dimuons de basse masse sans intérêt. Ce même problème a été posé à d'autres expériences et diverses solutions ont été envisagées. La complexité de l'électronique associée augmente avec l'intensité du faisceau incident. Une étude comparative de ces déclenchements est donnée en référence [20,21,22]

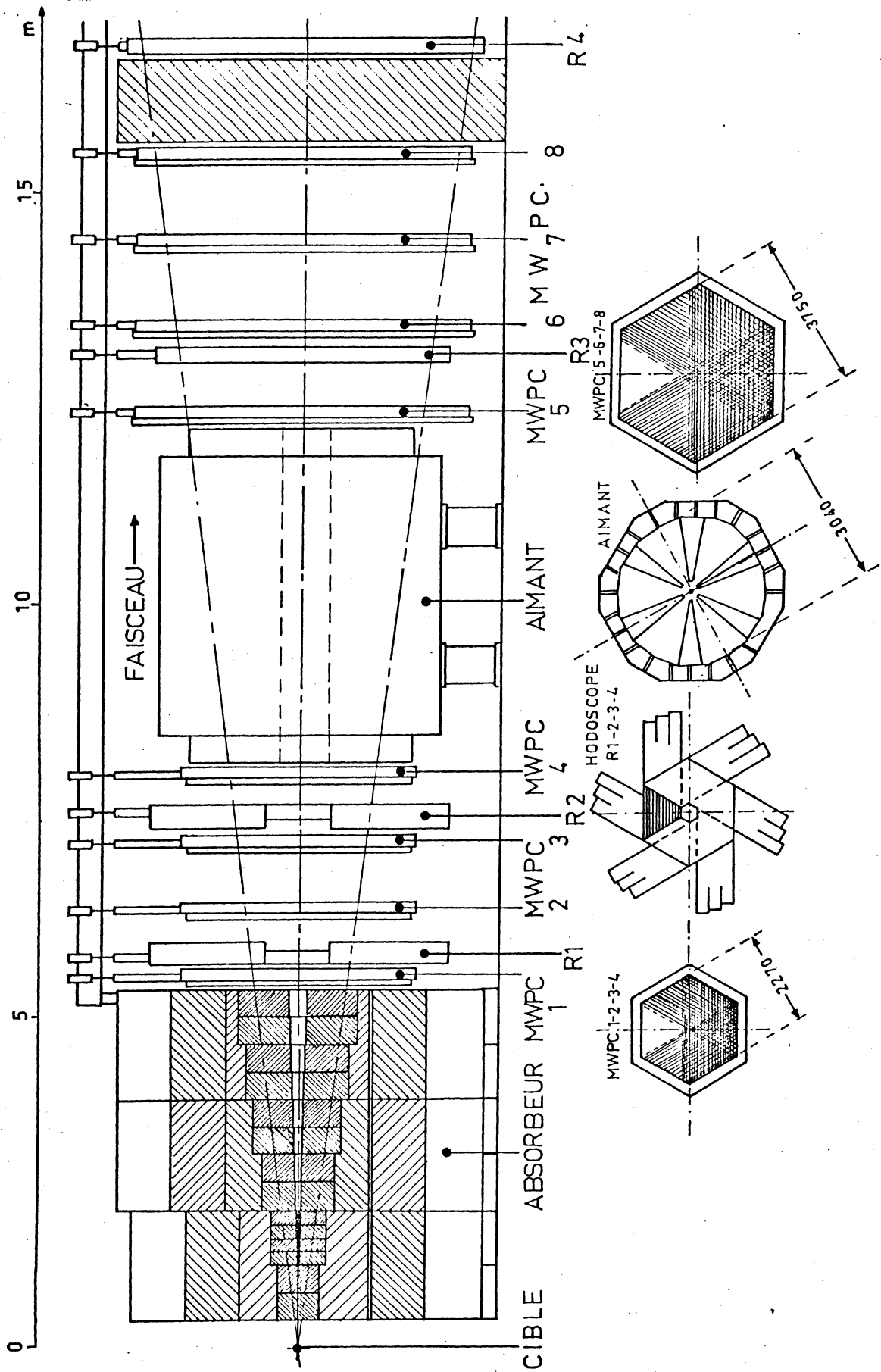


Fig. IV.12 : Appareillage NA10

Le déclenchement de l'expérience NA10 se fait suivant des critères analogues qui se distinguent essentiellement par une meilleure résolution sur le signal et une meilleure tenue au flux incident (fig.IV.13)

La tenue au flux incident est basée sur le choix des capteurs utilisés pour le déclenchement : compteurs à scintillations bien dimensionnés. La résolution sur le signal dimuon est obtenue :

- par optimisation de l'épaisseur et de la composition du filtre à muons
- en privilégiant les particules produites dans la cible au détriment de celles produites dans le dump ou de celles qui se propagent parallèlement au faisceau (dimensions et position des compteurs de R_1, R_2)
- en mesurant avec une bonne précision l'impulsion transverse des muons (dimensions et positions des compteurs de R_3, R_4)

La logique associée aux compteurs de R_1, R_2, R_3, R_4 détermine, dans 6 secteurs de 60° disposés autour de l'axe du faisceau :

- l'angle de production des muons ($R_1 * R_2 = V$)
- son angle de déviation dans l'aimant, c'est à dire leur impulsion transverse (logique VR3R4).

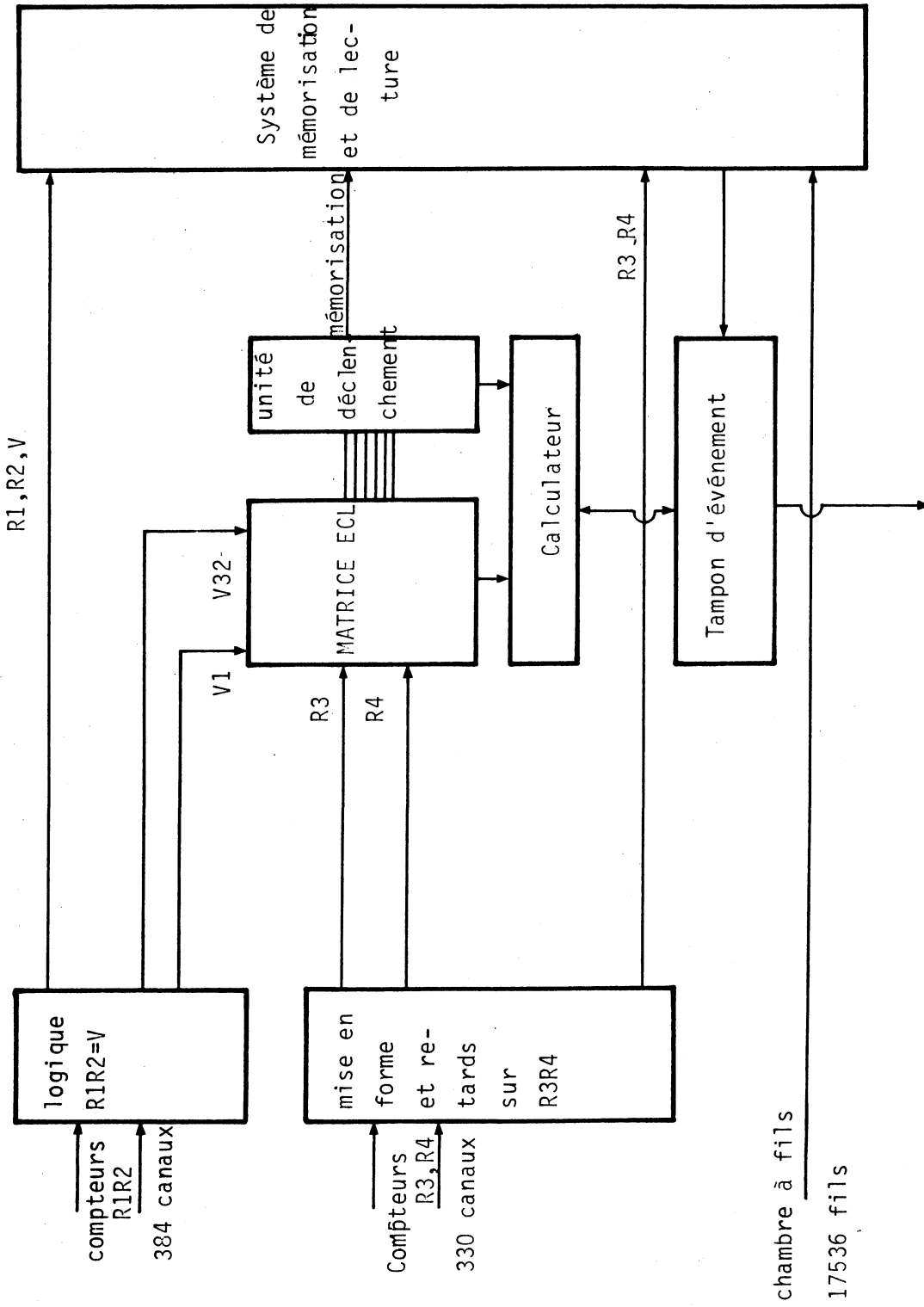
Sachant que la masse d'un dimuon est proportionnelle à la somme des impulsions transverses de chacun des muons, cette logique autorise un déclenchement sur la masse des dimuons produisant des muons dans des secteurs différents.

1. La logique associée au déclenchement

L'étage $V = R_1 R_2$ est réalisé par des discriminateurs rapides, afin de réduire au minimum les coïncidences fortuites ($V = 11$ ns). L'étage $VR_3 R_4$ est constitué de 192 éléments d'une matrice de coïncidences mesurant pour chaque V 32 valeurs de P_T correspondants aux 32 compteurs possibles sur R_4 validés par 23 compteurs de R_3 (fig.IV.14.1 et IV.14.2.).

Les P_T de chaque sextant sont alors disponibles sous deux formes différentes :

- Les P_T regroupés qui sont utilisés par une logique intersextant pour réaliser un déclenchement rapide et simple sur la masse du dimuon produit grâce à sa valeur approchée $P_{T1} + P_{T2}$. Les bornes de ces intervalles définissent quatre régions A-B-C-D.



vers l'acquisition

Fig.IV.13 : Mise en temps des données de tous les capteurs, mémorisation et emplacement d'un calculateur.

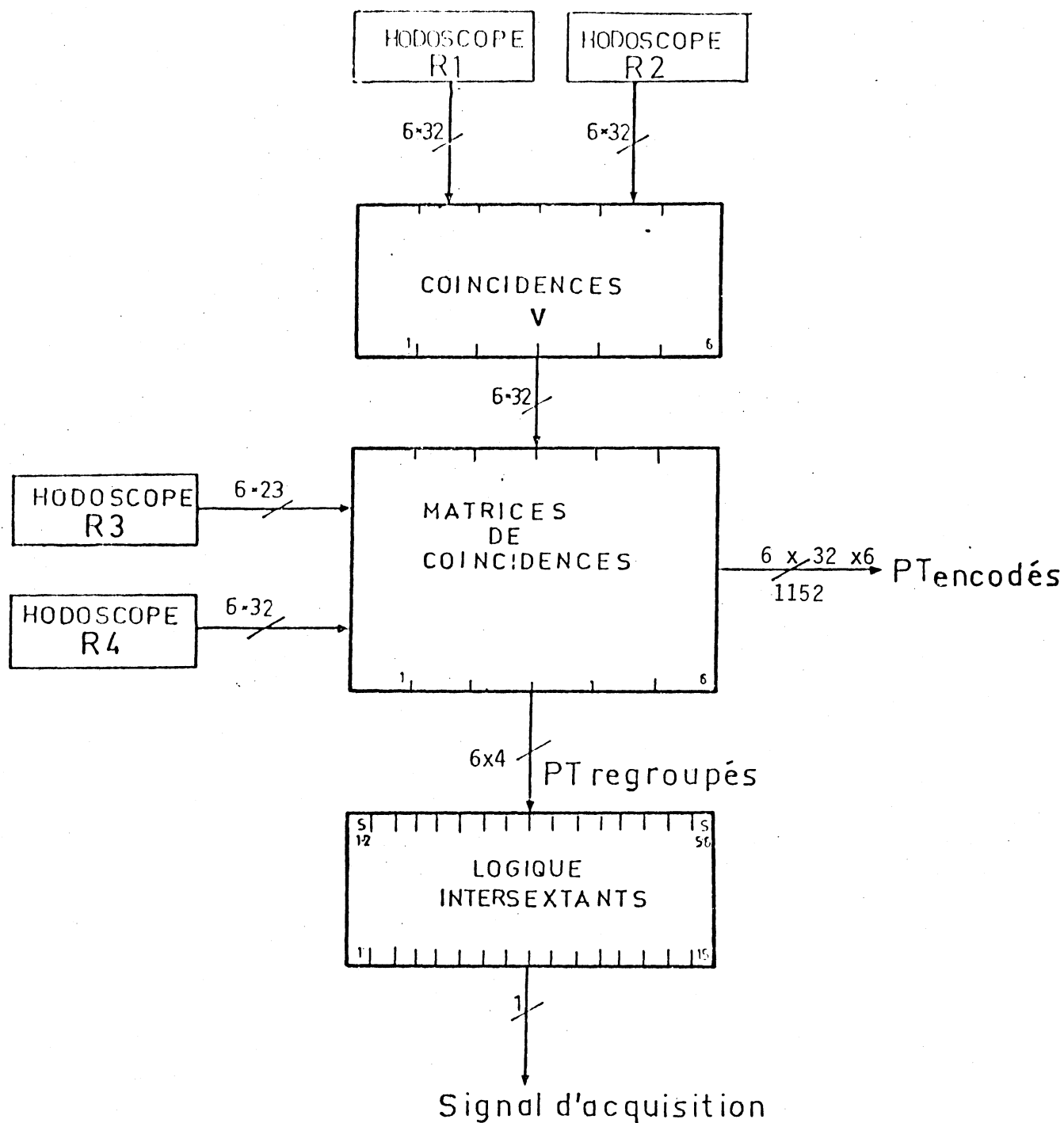


Fig.IV.14 : Déclenchement NA10

A		$P_T > 4 \text{ GeV/c}^2$
B	$2 \leq$	$P_T \leq 4 \text{ GeV/c}^2$
C	$1.4 \leq$	$P_T > 2 \text{ GeV/c}^2$
D	$0.6 \leq$	$P_T > 1.4 \text{ GeV/c}^2$

elles ont été choisies pour optimiser la résolution en masse au dessus du ψ
($P_{T1} + P_{T2} \geq 4 \text{ GeV/c}^2$)

- Les P_T encodés, sous forme des numéros de R_4 correspondant à chaque V, sont utilisés pour réaliser un déclenchement de second niveau utilisant la valeur exacte de la masse calculée par GESPRO. En cas de conflit entre plusieurs valeurs de P_T dans le même sextant seule la valeur correspondant à la plus grande masse est transmise. La valeur est codée sur 5 bits, un bit supplémentaire de présence permet un comptage simple des multiplicités. Ceci porte à 192 mots de 6 bits le volume des informations reçues par GESPRO indépendamment du système de lecture.

II.C. Organisation du système de lecture

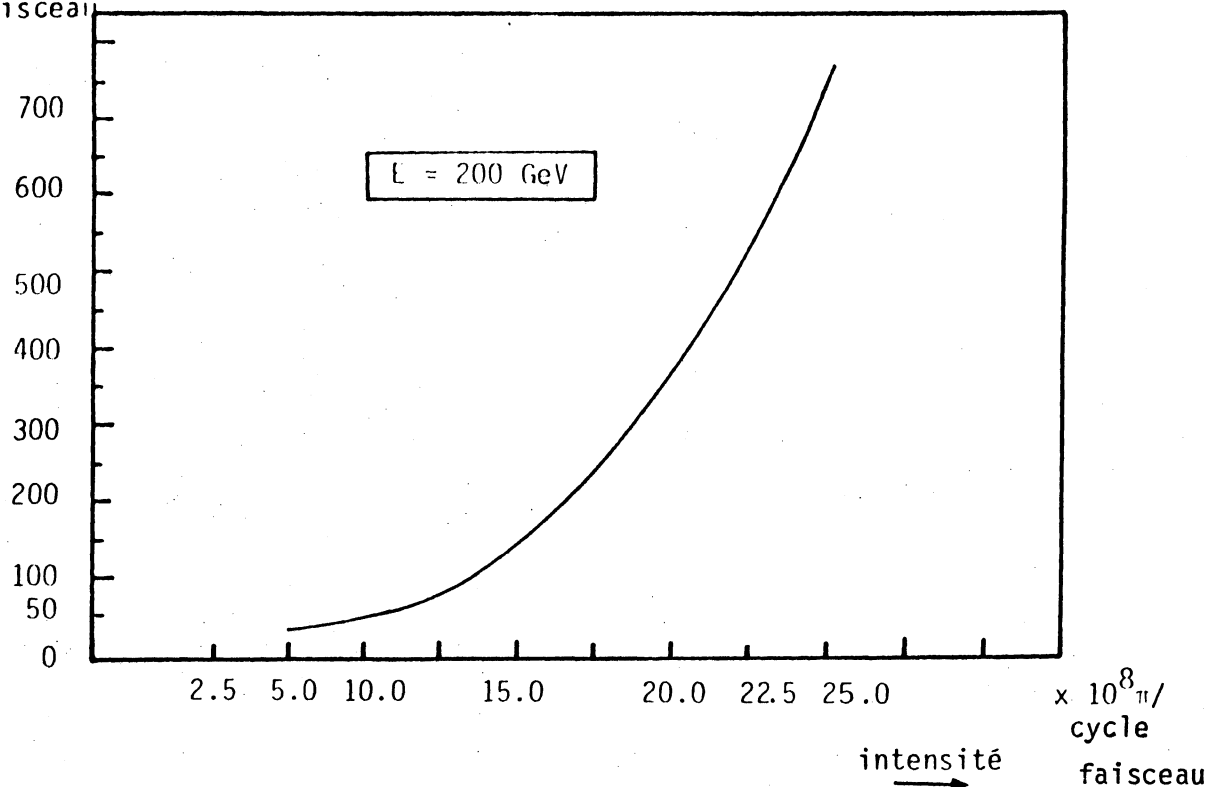
L'ensemble des capteurs représente environ 21000 voies. Ces informations sont compactées pour permettre au système d'acquisition de ne lire que les compteurs touchés. La figure 15 montre le nombre de déclenchements par cycle et le nombre de mots par événement après compactage. Dans ces conditions si l'on veut permettre des acquisitions à haute intensité ($\approx 2.10^9 \pi/\text{cycle}$) on doit faire face à un flux de données de 1000 événements de 1000 mots par seconde. Ceci est incompatible avec la vitesse d'acquisition de l'ordinateur maître (2 à 3 μs par mot) et avec la taille de la mémoire tampon disponible (180 K mots).

Enfin le temps mort obtenu avec un système classique du type compacteur CAMAC serait prohibitif puisqu'un transfert de 1000 mots représente environ 3 millisecondes, ce qui est à peu près dix fois trop lent par rapport au taux de déclenchements obtenu par la logique rapide.

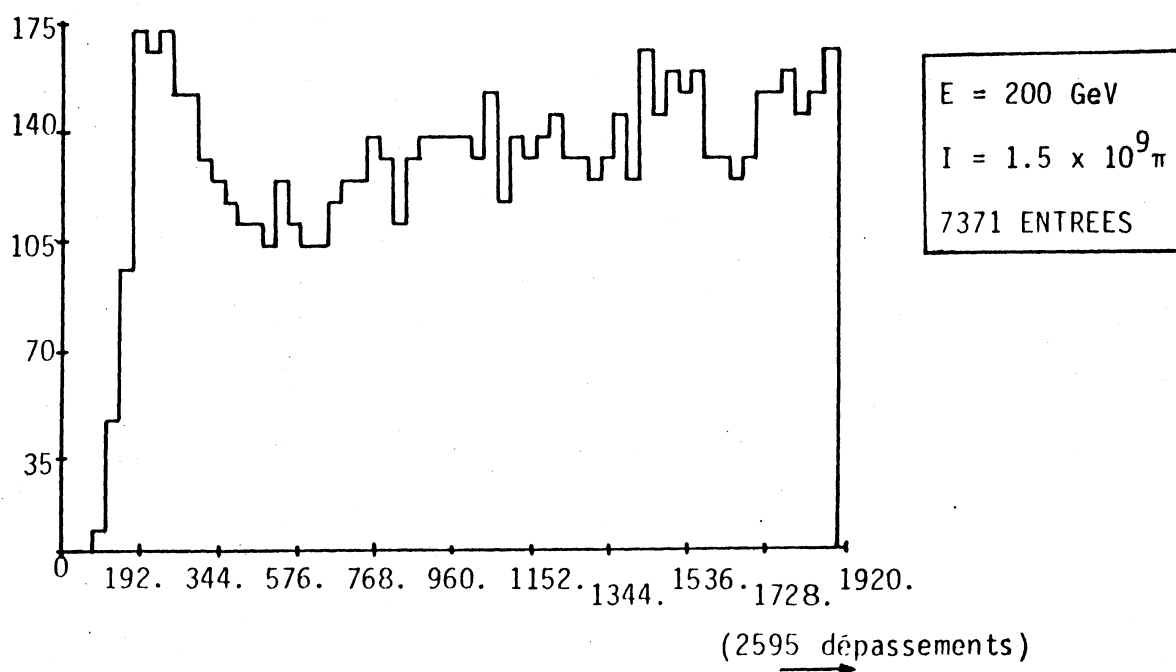
Pour ces raisons le système d'acquisition NA10 implique, par rapport au système d'acquisition WA2, un changement de Technologie et un changement de philosophie.

Changement de technologie : Le système de compactage-lecture a été réalisé en technologie ECL, et est plus de dix fois plus rapide que les systèmes précédents (Réf.7). Grâce à ce système de lecture (le R.M.H. ou Receiver Module Hybride), on obtient une vitesse de transfert de 150 ns par mot soit plus de 6 Méga mots

Nombre de déclenchements /
cycle faisceau



Variation du nombre de déclenchements en fonction de l'intensité



Histogramme du nombre total de mots par événement

Fig.IV.15 : Variation du nombre de déclenchements en fonction de l'intensité et histogramme du nombre total de mots par événement sans filtrage.

par seconde (fig.16).

Changement de philosophie : Les ordinateurs de contrôle en ligne constitués par un couple NORD 100-NORD500 n'acceptent qu'un mot toutes les $1,5 \mu s$ soit un flux dix fois plus faible. Une connection RMH-NORD, directe ou via CAMAC est donc impensable. L'originalité du système d'acquisition de cette expérience est la réalisation d'un ensemble autonome, modulaire et transparent permettant de réduire le flux. Ce système, fruit de la collaboration BRISTOL-GENEVE-STRASBOURG permet d'approcher la structure fonctionnelle idéale décrite au chapitre II, à savoir celle d'une mémoire rapide visible de tous les processeurs.

Dans ce contexte, le processeur d'acquisition (R.M.H.) est câblé, le processeur de gestion mémoire est câblé et les processeurs de filtrage sont constitués de processeurs GESPRO, dont le nombre, la configuration et les opérateurs câblés sont fonction des rejets réalisés et des capacités de transmission du système.

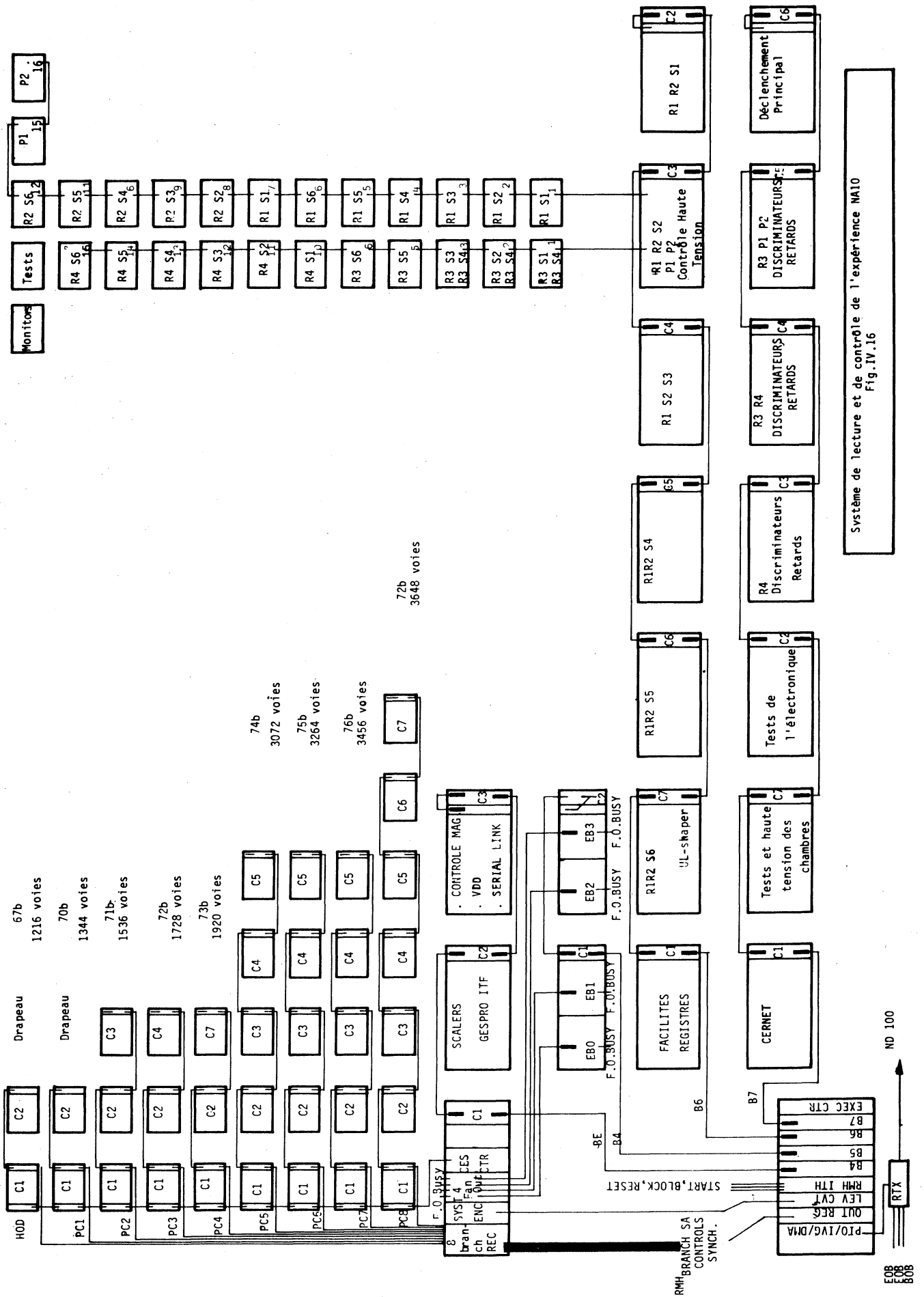
Dans le cas de l'expérience NA10, l'idéal est obtenu lorsque le nombre d'événements transmis à NORD est dix fois plus faible que le nombre de déclenchements.

Par ailleurs, un résumé de données fourni par les hodoscopes (p_t encodés) a été interfacé directement sur les calculateurs. Ceci permet d'agir à deux niveaux :

- 1) Niveau déclenchement logiciel : Traitement de l'événement avant et pendant son transfert dans la mémoire rapide (sur résumé de données) (fig.IV.17)
- 2) Niveau filtrage en ligne (sur données acquises)

Il est évident que le premier niveau est efficace si le temps de traitement est inférieur ou égal au temps de transfert RMH - Mémoire rapide alors que le deuxième niveau est rentable tant que son temps de traitement ne dépasse pas le temps de transfert mémoire rapide-NORD. Ceci se traduit chaque fois par un gain en temps actif et donc en sensibilité.

Comme nous l'avons vu au chapitre deux, la gestion d'une telle mémoire devient très complexe lorsque le nombre de processeurs qui l'utilisent augmente. Cette gestion a été simplifiée ici en sacrifiant le taux d'utilisation à la vitesse d'exécution : l'ensemble est divisé en mémoires d'événement et le contrôleur change de mémoire à chaque déclenchement. Chaque mémoire est à huit accès et permet donc les accès RMH, contrôleur, CAMAC et microprocesseur en vol de cycle.



Système de lecture et de contrôle de l'expérience NA10
Fig. IV.16



opérateur de recherche
du plus grand P_T

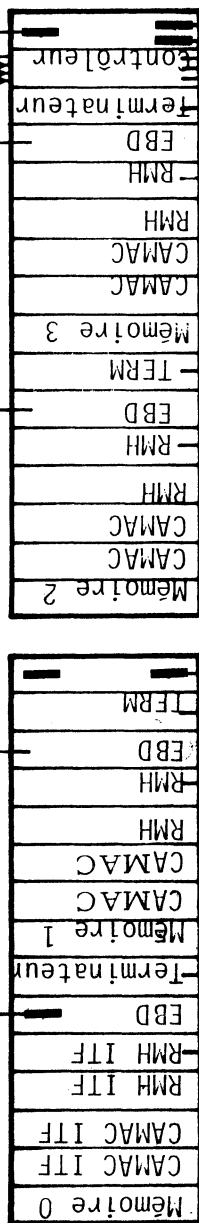
déclenchement - déversement
initialisation - temps mort

Accès
CAMAC

CAMAC
ITF

GESPRO
to CAMAC

CAMAC
BRANCHE 4



RMH F.03

RMH F.02

RMH F.01

RMH F.00

Chaque mémoire d'événement (EVB) peut recevoir un ou plusieurs interfaces avec un ou plusieurs processeurs de filtrage. L'ensemble est synchronisé par un contrôleur entièrement câblé qui assure les allocations des mémoires en acquisition, en analyse et, pour les événements non rejetés, en transfert vers NORD.

Le transfert EVB-NORD est réalisé en CAMAC, en mode multichâssis et multi-stations de telle sorte qu'une seule DMA lancée au début du burst soit suffisante pour assurer la totalité de l'acquisition. Grâce à cette philosophie l'ordinateur de contrôle ne voit que les deux interruptions début et fin de burst, tandis que chaque processeur de filtrage ne voit que les déclenchements qui lui sont destinés.

II.D.Simulations des différentes coupures et conception des calculateurs

1. Les simulations

Un programme du type Monte Carlo a été utilisé pour estimer l'effet des diverses coupures sur l'impulsion transverse des muons et sur la masse des dimuons. La géométrie utilisée est celle du dispositif expérimental pour des pions incidents d'impulsion $p = 225 \text{ GeV/c}$ (Réf.7).

Ces simulations ont permis de dégager les limites du déclenchement de premier niveau, en montrant qu'à grande impulsion transverse du dimuon, une coupure à 4 GeV/c^2 au déclenchement pouvait faire perdre des masses jusqu'à 6 GeV/c^2 . Il a donc été décidé d'utiliser une coupure de second niveau, agissant après une coupure prudente sur la formule approchée utilisée au premier niveau :

$M = p_{T1} - p_{T2}$. La formule exacte est :

$$M^2 = 2P_1P_2(1 - \cos\theta_1 \cos\theta_2 + \sin\theta_1 \sin\theta_2 \cos(\varphi_1 - \varphi_2)) \text{ où}$$

P_i est l'impulsion du muon et θ_i, φ_i ses angles polaire et azimutal. Nous avons ensuite introduit dans un programme de simulation toutes les valeurs possibles de P_i, θ_i et φ_i avec leur précision respective et nous avons examiné l'effet du nombre de bits utilisés dans un calcul réalisé en entier et en flottant. Nous avons examiné d'autre part les différents temps d'exécution en mode programmé et microprogrammé, étant entendu que la précision des compteurs autorise une tabulation complète des lignes trigonométriques.

2. Résultats des simulations

a) Résultats généraux

- La coupure en masse est améliorée d'un facteur 3 à 5 par rapport aux résultats de la formule approchée. .
- Lorsqu'un sextant fournit plusieurs mesures de p_t on ne retient que le plus grand p_t .
- dans le cas de deux traces de même moment transverse dans le même sextant on doit conserver la trace d'angle polaire θ le plus faible
- les événements de très hautes multiplicités seront à rejeter car ils détruisent totalement la sélectivité du déclenchement

b) Résultats spécifiques à la conception des calculateurs

- en tant que déclenchement de second niveau le temps du calcul de masse ne doit pas excéder le temps du transfert RMH-EVB (100 à 200 μ s suivant le nombre de mots).
- le nombre de bits ne peut être inférieur à 24 sous peine de perdre tout le bénéfice de l'application de la formule exacte par manque de précision du résultat.
- le calcul sera plus simple à exécuter si l'on dispose d'une arithmétique flottante (mélange d'addition et de multiplication)
- la recherche, pour chaque sextant, de la trace de plus grand moment transverse coûte trop cher en temps d'exécution si cette dernière est faite par programme ou par microprogramme.

En effet il faut lire 32 mots par sextant et exécuter une comparaison et un comptage demultiplicité, ce qui nécessite un minimum de 6 instructions, soit pour 192 données et 150 ns de temps de cycle unité centrale, plus de 180 μ s.

3. Conception des calculateurs

- La recherche du plus grand P_T par sextant sera exécutée par un opérateur câblé. Ce dernier sera conçu en logique séquentielle rapide. Le temps d'exécution obtenu est de 5 μ s par sextant. Pour ne pas pénaliser l'ensemble 6 opérateurs identiques seront activés simultanément (cf. § 2.F.).
- L'estimation du temps d'exécution d'un calcul de masse montre l'intérêt d'un opérateur arithmétique ultra-rapide, capable d'exécuter multiplications et conversions sur 24 bits. La plus grande vitesse a été obtenue

rendant tous les opérateurs combinatoires, ce qui permet d'obtenir un temps d'exécution de 200 ns par opération (cf. § 2.F.).

II.E. Description détaillée des mémoires d'événements "EVB"

1. Description générale

Cet ensemble, réalisé par la collaboration Bristol-CERN-Strasbourg permet le déclenchement sur la masse calculée par GESPRO ainsi que le filtrage en ligne.

Ce système a été conçu à partir de mémoires d'événement de 2K-mots et 8 accès chacune, de temps de réponse environ dix fois plus rapide que les temps de réponse des accès externes des mini-ordinateurs classiques. Un maximum de 32 mémoires gérées par un contrôleur câblé, est prévu pour être connecté à un système de lecture (le RMH dans notre cas), et à un système de transfert (DMA CAMAC EVB-NORD pour cette expérience). Ceci laisse donc cinq accès libres pour d'autres processeurs (acquisition filtrage ou transfert).

Physiquement chaque EVB se compose d'un sandwich de tiroirs au standard mécanique CAMAC reliés entre eux par un bus interne ECL.

Grâce à une gestion complète du contrôleur une seule DMA CAMAC par burst permet le transfert des seuls événements non rejetés vers les ordinateurs de contrôle. Ceci est obtenu en mettant le châssis système CAMAC en multiadressage :

$$B_{(4)} \cdot C_{(8)} \cdot N_{(24)} \cdot A_{(0)} \cdot F_{(2)}.$$

B est l'adresse de la branche où se trouve le contrôleur

C_8 : adressage de tous les châssis de la branche

N_{24} : adressage de toutes les stations du châssis

Chaque EVB peut prendre quatre états :

- libre (attente d'un déclenchement)
- acquisition (lecture des capteurs via le RMH)
- analyse (examen des données par le (les) processeur de filtrage)
- transfert des événements non rejetés vers les ordinateurs de contrôle

Le bus interne ECL comprend quatre parties, il est détaillé dans le tableau de la figure 18. Pour chaque EVB :

INIT : Initialisation (événement par événement)
RES : Remise à zéro (général)
DATA : 16 lignes de données (lecture-écriture)
Request : 8 lignes de demande d'accès
GRANT : 8 lignes d'acquiescement
BUSY-SYNC-ACKNOWLEDGE : 3 lignes de synchronisation
R/W : 2 lignes de lecture /écriture (possibilité d'action sur l'octet).
SDI : "start data input" début de la phrase d'acquisition-signal émis par le contrôleur
RDI : Reset data input - ordre de fin d'acquisition - signal émis par le contrôleur en cas d'interruption d'acquisition
DIC : data input completed signal de fin normal d'acquisition émis par le (les)processeurs d'acquisition
DIE : data input error : détection d'une erreur d'acquisition
STA : start analysis signal émis par le contrôleur indiquant le début de la phrase d'analyse des données par les processeurs de filtrage.
STO : stop analysis : signal de fin anormale d'analyse, émis par le contrôleur
GOOD : signal émis par le (la) processeur de filtrage indiquant que l'événement n'a pas été rejeté
BAD : signal indiquant que l'événement est à rejeter
ECT : "enable CAMAC Transfert" indique que le transfert EVB - ordinateur de contrôle est commencé
CTD : "CAMAC transfert done" fin normale du transfert.

Figure IV.18

Tableau des signaux de l'EVB

- la première partie est consacrée aux données (de lecture et d'écriture) sur 16 lignes
- la deuxième comprend 17 lignes d'adresse (16 sont prévues pour une possible extension à 64K mots par EVB et 1 réservée à l'adressage des registres de contrôle).
- la troisième est consacrée à la synchronisation processeurs-mémoire
- la dernière à 12 lignes de contrôle et d'état ; elle comprend deux lignes d'initialisation (INIT et RES) et 10 lignes définissant les états du système (tableau Fig.13)

Cette philosophie permet l'extension de ce système à plusieurs processeurs de filtrage et (ou) d'acquisition. A cet effet la ligne GOOD, en "ET CABLE" n'est vraie que si tous les processeurs de filtrage l'ont satisfaite, de même que la ligne BAD "OU CABLEE" est vraie dès que l'un des processeurs de filtrage l'a définie.

Le même type de connection est utilisé en acquisition : la ligne DIC (fin d'acquisition) n'est vraie que lorsque tous les processeurs d'acquisition l'ont activée.

2. L'interface "GESPRO" Mémoire d'événement

Cet interface, relié à GESPRO par quatre câbles en technologie ECL différentielle donne accès à un registre de contrôle et d'état ainsi qu'aux données EVB en lecture et écriture. Vue côté GESPRO l'EVB se comporte exactement comme de la mémoire centrale ce qui, en mode programmé, permet l'utilisation de toutes les instructions à référence mémoire. Ceci donne également à NORD, via CAMAC, l'accès aux données et registre de l'EVB.

Les signaux de contrôle ou d'état, utiles au processeur de filtrage ont été regroupés dans un registre. Ce registre d'état mémorise l'activité du bus interne, il est lu comme une case mémoire mais remis à zéro bit à bit.

Le registre de contrôle, également accessible comme une case mémoire, est activé bit à bit.

3. Séquences et dialogue

Le fait que GESPRO soit utilisé au niveau déclenchement et au niveau filtrage est illustré fig.19 par le diagramme des états du système où l'on remarque les deux possibilités de rejet (rejet avant fin d'acquisition sur résumé de données et rejet après acquisition sur analyse des données complètes).

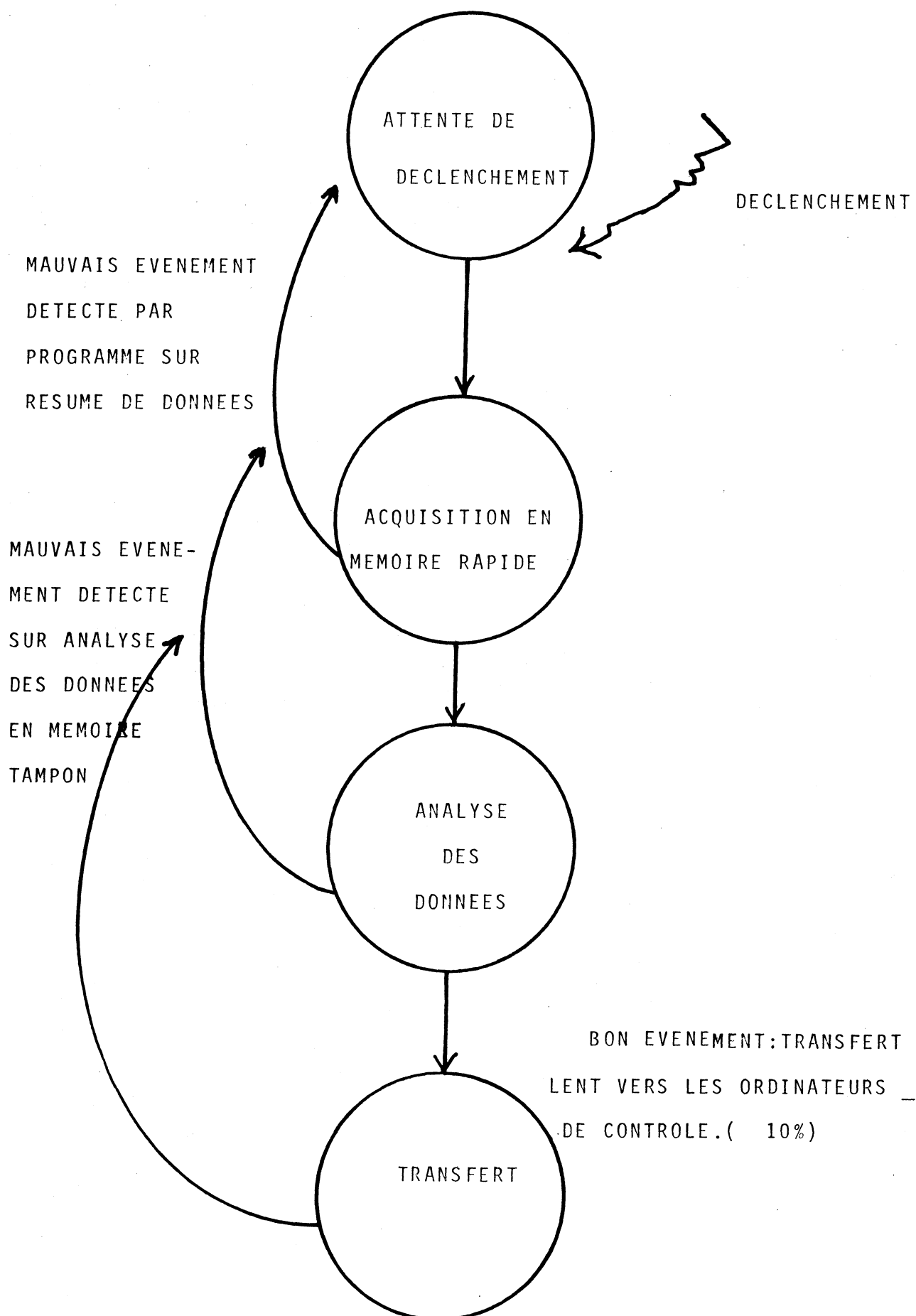


Fig.IV.19 : Diagramme des états du système

Le chronogramme des opérations sur le bus interne est représenté figure 20. Enfin deux détections d'erreur par dépassement de temps ont été câblés dans le contrôleur. La première permet d'interrompre l'acquisition (par INIT) lorsque le signal de fin d'acquisition n'est pas parvenu au contrôleur après un temps programmable. La seconde permet d'interrompre l'analyse lorsque ni l'un ni l'autre des signaux GOOD et BAD n'est parvenu au contrôleur après le temps normal prévu pour cette analyse.

II.F. Description de la configuration de GESPRO en version NA10

1. Généralités

GESPRO type NA10 comprend le module de base décrit au chapitre III, un module d'interruption simplifié (une grande partie des tâches de synchronisation ayant été câblée dans le contrôleur d'EVB), son interface EVB vu comme deux K-mots de mémoire centrale et deux mots spéciaux à action bit à bit.

Cet ensemble est complété par le module de recherche de plus grands moments transverses et par l'unité arithmétique rapide qui seront décrits plus loin.

Le temps de lecture RMH estimé était de 100 μ s et le temps d'un calcul de masse de 15 μ s. En tenant compte des temps de recherche dans les tables et dans les périphériques, le temps complet d'un calcul avait été estimé à 25 μ s. Dans le cas le plus défavorable représenté par les six sextants touchés et où aucune combinaison ne convient, 15 calculs sont à faire, ce qui nous amène à environ 400 μ s soit à 4 couples EVB-GESPRO si l'on veut éviter d'introduire un temps mort. Nous verrons par la suite que ce cas défavorable est rarissime et que l'un des quatre couples n'est pratiquement jamais sollicité en prise de données.

Enfin, si l'on se rapporte au chronogramme de la figure 15, on remarque que dans le cas d'erreur d'acquisition, deux signaux SDI peuvent être émis en très peu de temps. La limite, fixée par le contrôleur est de 15 μ s. SDI et STA sont deux signaux importants de synchronisation entre l'EVB et GESPRO. Dans le cas normal, le déclenchement de second niveau est terminé avant STA, ce qui rend ce signal peu critique. Il n'en est pas de même pour SDI qui doit être pris en compte très rapidement si l'on veut éviter les empilements.

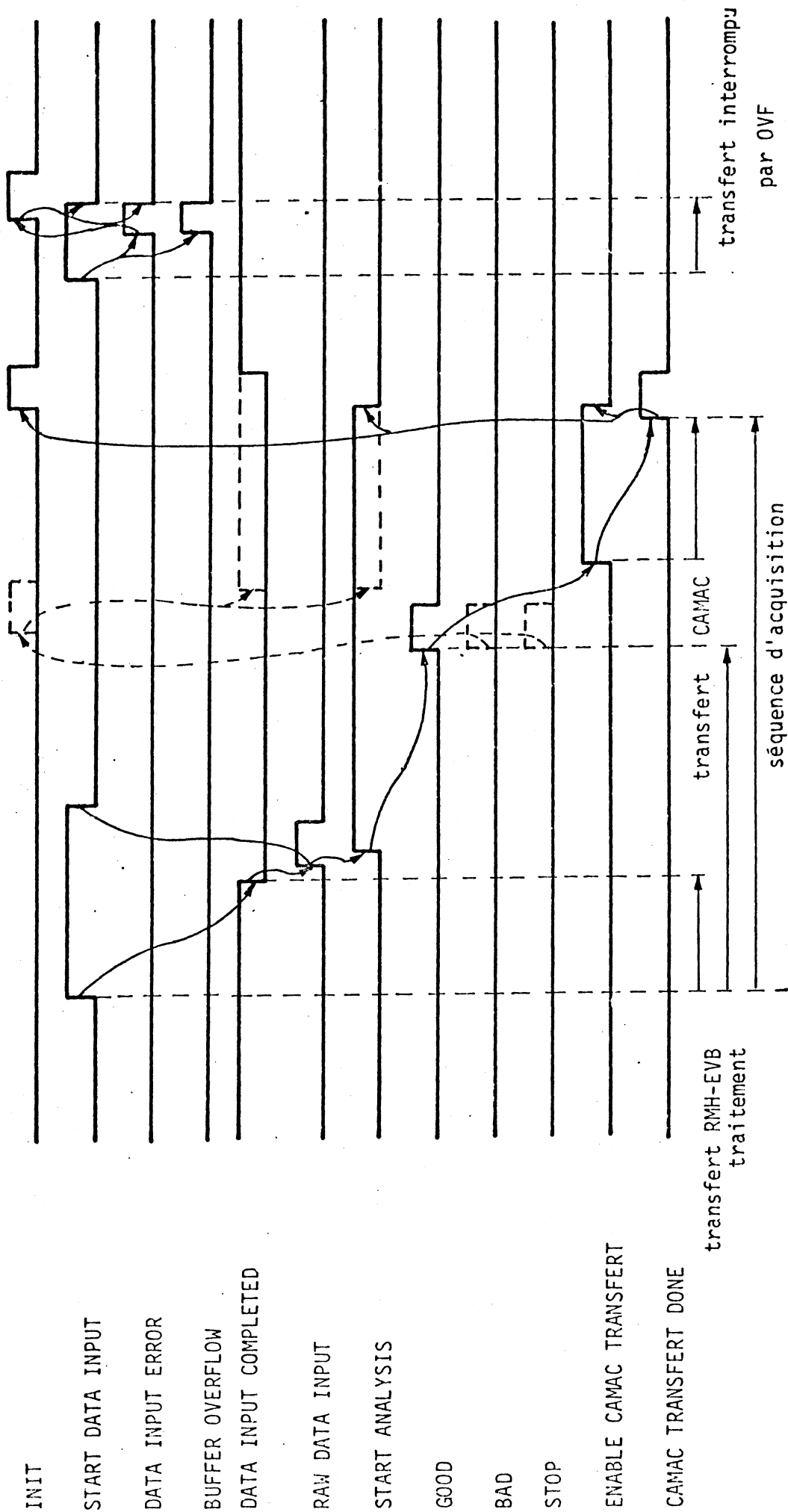


Fig.IV.20

Chronogramme des opérations sur le bus interne 100

A cet effet une interruption spéciale, ultra-rapide et de priorité absolue, masquable cependant, a été câblée sur le niveau zéro : Elle permet un débranchement en un cycle horloge (150 ns environ), sans changement de contexte, ce qui, au prix de quelques efforts de programmation, permet d'obtenir une efficacité absolue en réponse au contrôleur, et conserve le minimum de temps mort.

2. Les opérateurs câblés spécifiques à l'expérience

a) La recherche du plus grand P_T :

Elle est faite sur six unités qui ont pour rôle de permettre aux calculateurs de déterminer rapidement, grâce à un ensemble de tables, tous les paramètres qui leur sont nécessaires pour effectuer le calcul de masse.

Rappelons la formule exacte de la masse :

$$M^2 = 2 P_1 P_2 (1 - \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2 \cos(\varphi_1 - \varphi_2))$$

Pour chaque sextant l'opérateur câblé recherche la trace donnant le plus grand P_T , mémorise cette valeur ainsi que celle de l'angle θ associé, indique le nombre de traces contenues dans le sextant et retourne le numéro du sextant qui sert de mesure grossière de l'angle azimutal φ .

a.1) Fonctionnement

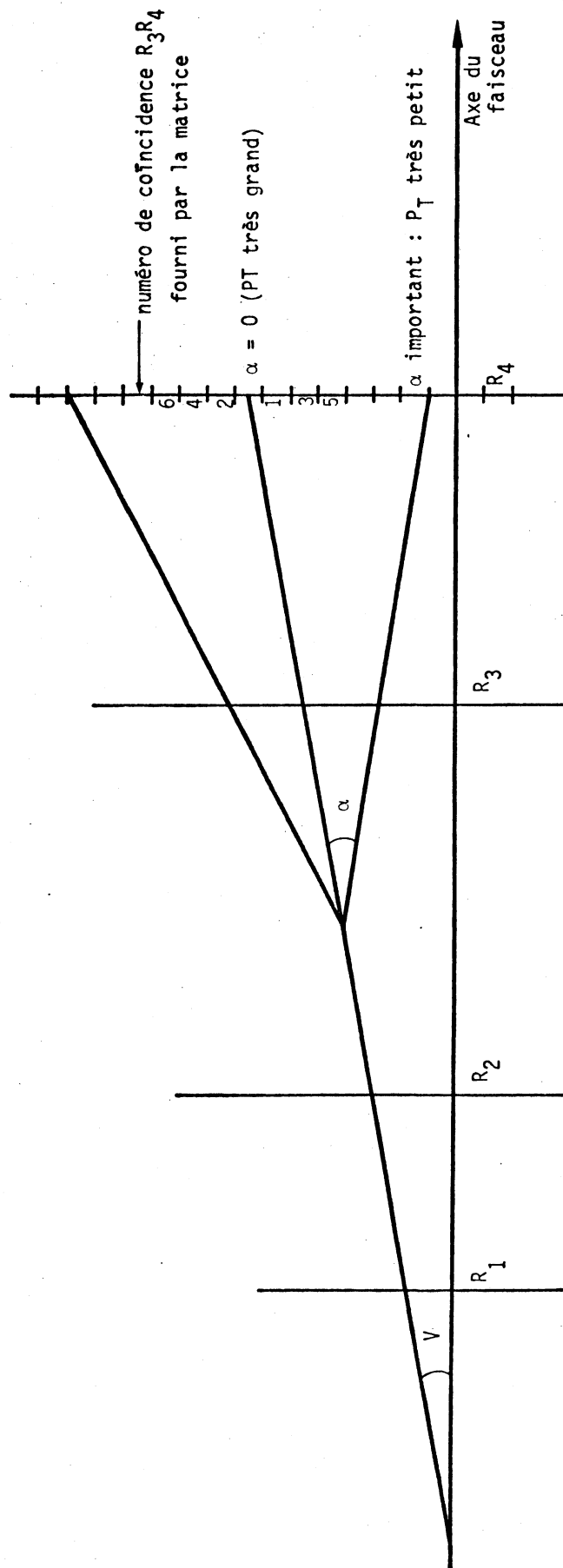
La figure 21 montre le procédé utilisé :

La matrice de coïncidence fournit sous forme binaire d'un nombre compris entre 0 et 31 le numéro de la coïncidence $R_3 R_4$ ou, s'il y en a plusieurs, le numéro de celle qui correspond à la plus faible déviation α donc au plus grand P_T .

Ce numéro n'est pas en lui-même une image du P_T qui est fonction de la déviation donc de $V(P_T = \frac{K}{\alpha})$.

Pour chercher le plus grand P_T parmi les $32 + 32$ combinaisons $VR_3 R_4$ fournies par la matrice pour chaque sextant, une table de transcodage à 1024 mots de 5 bits est nécessaire.

Cette table est adressée par les dix bits formés par les 5 bits de V et les 5 bits définissant les coïncidences $R_3 R_4$. (Les 5 bits de V sont implicites : c'est le numéro du connecteur correspondant à un module de matrice).



V : angle incident déterminé par la coïncidence $R_1 R_2$

α : angle de déflexion de l'aimant

Figure IV.21 :

Principe de détermination du P_T

grâce au numéro de coïncidence $R_3 R_4$ et à la direction incidente V

Les signaux incidents sont émis sous forme de 192 câbles de 6 paires différentielles ECL, à raison de 32 par sextant. Une paire indique la présence ou l'absence d'au moins une trace dans l'élément de matrice et les cinq autres donnent la valeur codée du numéro de coïncidence R_3R_4 .

Après mise en temps, l'ensemble des informations est enregistrée à l'instant défini par le déclenchement principal dans 192 registres de six bits (32 par sextant). Les six unités démarrent alors simultanément un cycle de 32 phases. A chaque phase un séquenceur adresse un nouveau registre, recherche dans une mémoire morte la valeur du P_T correspondant et la compare à la plus grande valeur obtenue précédemment. Si la valeur courante est supérieure, la remise à jour touche également le numéro du registre, image de l'angle θ . Simultanément un compteur est incrémenté si le bit de présence est vrai, ce qui, à la fin du cycle fournit la multiplicité. Le cycle de base de chaque séquenceur est de 150 ns ce qui permet d'obtenir les trois paramètres cherchés en 4,8 μ s.

Les six opérateurs sont rigoureusement identiques et opèrent en TTLS après conversion ECL-TTL.

a.2) Réalisation

Les données sont fournies en ECL et converties en TTL par des convertisseurs intégrés. Elles sont ensuite mémorisées dans des bascules S 374 choisies pour leur temps de réponse (< 10 ns) et leur sortie 3 états favorisant le balayage de chaque registre. Le séquenceur peut ainsi être réduit à un registre à décalage et à un compteur 5 bits. Chaque cycle exécute ensuite la même opération, purement combinatoire : adressage de la mémoire morte, comparaison du P_T courant à la valeur maximale précédente avec décision sur la fin du cycle. Les six unités ont été réalisées en circuit imprimé et les connecteurs des câbles en provenance de la matrice ont été fixés sur le panneau avant, au pas de 2,54 mm ce qui prend toute la longueur de la carte (50 cm environ).

Bien que relativement simple de conception, cet interface, de par sa taille, sa vitesse de fonctionnement, son nombre de connections et sa difficulté de réalisation (très grand circuit imprimé) a nécessité la réalisation d'un banc de test complexe comprenant un simulateur de matrice interfacé CAMAC et un interface CAMAC de lecture de telle sorte qu'un jeu de programmes écrit sur NORD permette un test complet avant mise en place sur le site de l'expérience.

b) L'opérateur arithmétique rapide

Cet opérateur a été conçu pour réaliser des opérations arithmétiques entières et flottante sur 24 bits en un temps d'exécution aussi court que possible. Les entrées sont représentées sur 24 bits en complément à deux et les flottants sur 24 bits également (16 bits de mantisse, signe, exposant en excès à 64 sur 7 bits). Tous les nombres sont donc représentés en un seul mot machine GESPRO pour minimiser les temps d'exécution. Les formats et les capacités sont représentés sur la figure 22. Cinq opérations de bases peuvent être effectuées en 200 ns chacune : deux multiplications (entière et flottante), les deux conversions entiers $\leftrightarrow$ flottants et la réduction au même exposant pour faciliter les additions. Toutes les opérations ont été dessinées en logique combinatoire.

b.1) Fonctionnement

- Les multiplications

Elles sont obtenues par utilisation d'un arbre de Wallace de 4 fois 12 bits utilisant les multiplieurs combinatoires à 12 bits et 80 ns TRW MPY 12 HJ. Cet arbre est accompagné de la logique nécessaire à son utilisation en entier et en flottant :

en entier signé (complément à deux) il faut ajouter la résolution de la propagation du signe des résultats intermédiaires, alors qu'en flottant, il faut traiter le problème des exposants et celui de la normalisation du résultat. Le diagramme de cet opérateur est représenté figure 23.

- Les conversions (figure 24 et 25)

La conversion entier-flottant tient compte du signe - une complémentation à deux est nécessaire en cas de nombre négatif en entrée. La mantisse est calculée par un opérateur de décalage à gauche combinatoire dont le nombre de décalages est déterminé par un encodeur de priorité. L'exposant est calculé par un additionneur.

La conversion inverse (flottant-entier) tient compte de deux signes : un exposant négatif engendre un résultat nul, un nombre négatif impose la complémentation à deux du résultat. Ce dernier est obtenu par extension à droite de la mantisse, suivie d'un décalage combinatoire à droite calculé grâce à l'exposant.

Un regroupement judicieux des deux schémas permet une utilisation commune des opérateurs principaux de ces deux conversions.

Format entier : 24 bits en complément à 2 :

capacité en entier

dè 8 338 607 : 011 111 111 111 111 111 111 111

à

- 8 388 608 : 100 000 000 000 000 000 000 000

Format flottant : 16 bits de mantisse normalisée

1 bit de signe de l'ensemble

7 bits d'exposant signé en excès à 64

capacité en flottant :

+ 0,9999695. 2⁶³ ⁰
|1111111111111111| 1| 1111111 |

à

+ 0,0000305. 2⁶⁴ ⁰
|1000000000000000| 1| 0000000 |

Fig.IV.22 : Format et capacités arithmétiques

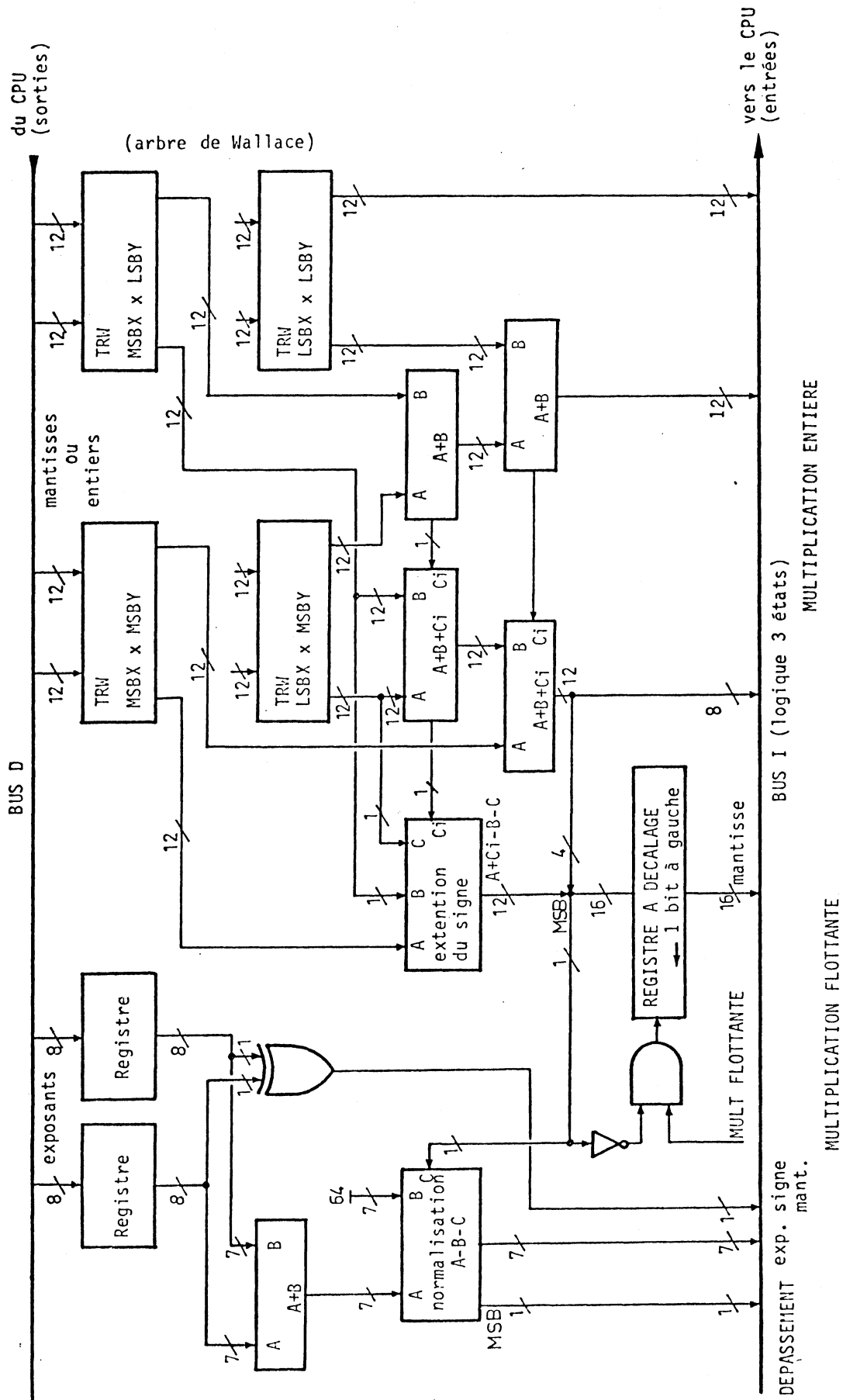
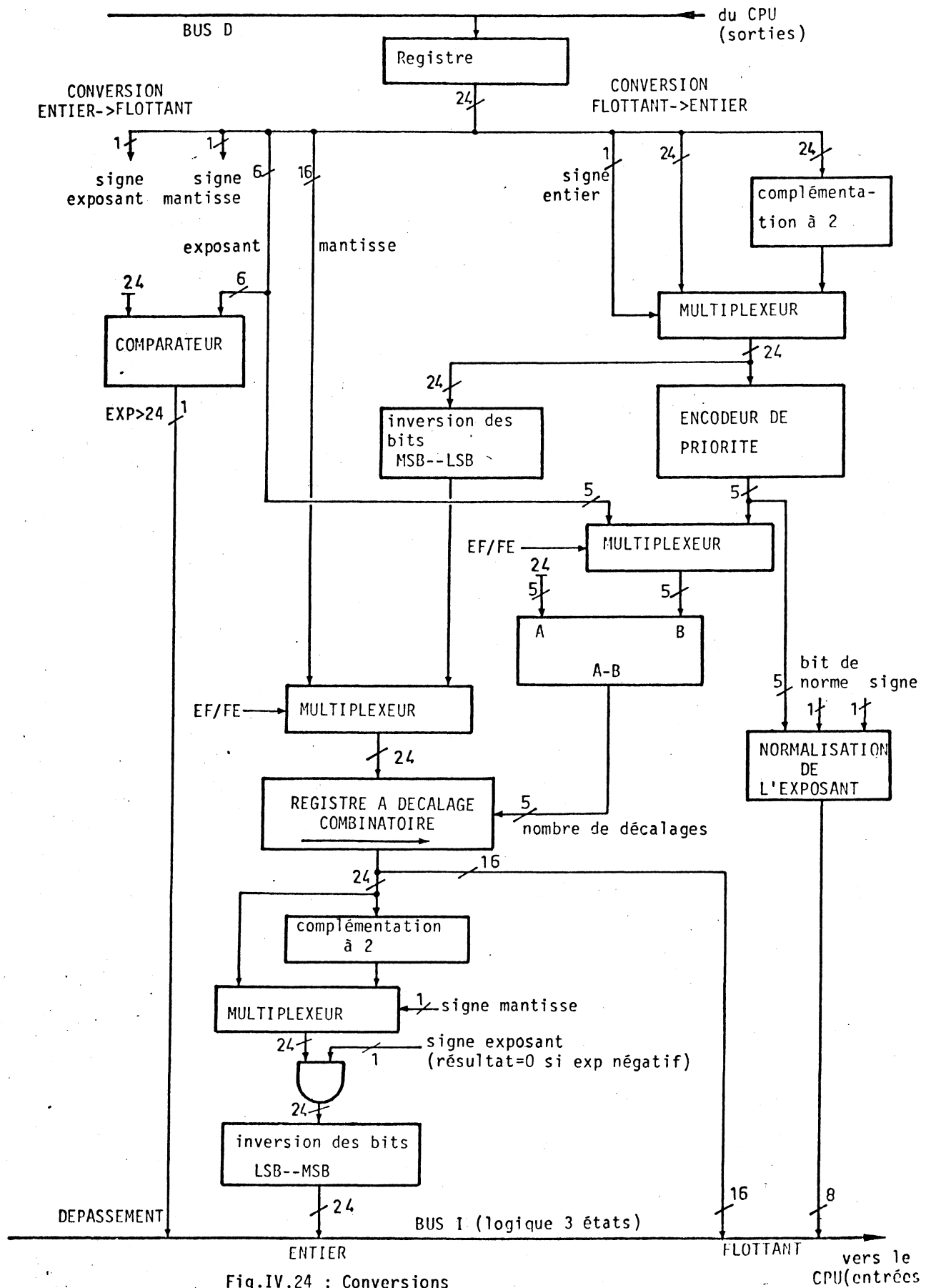


Fig.IV.23 : Multiplieur entier et flottant



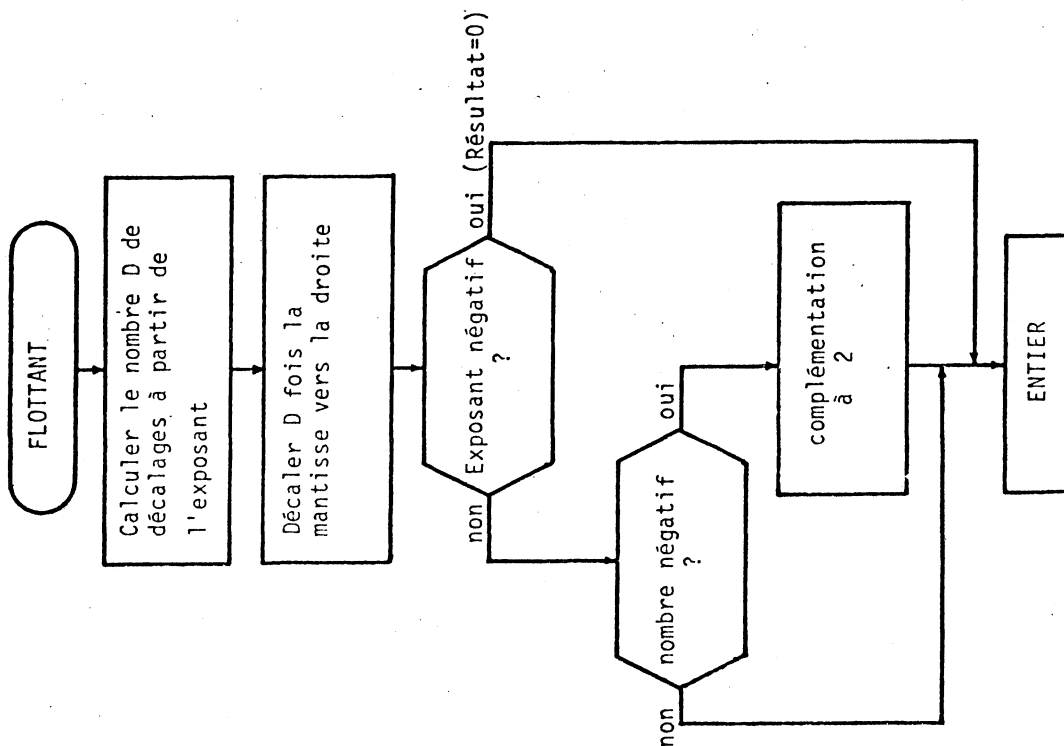
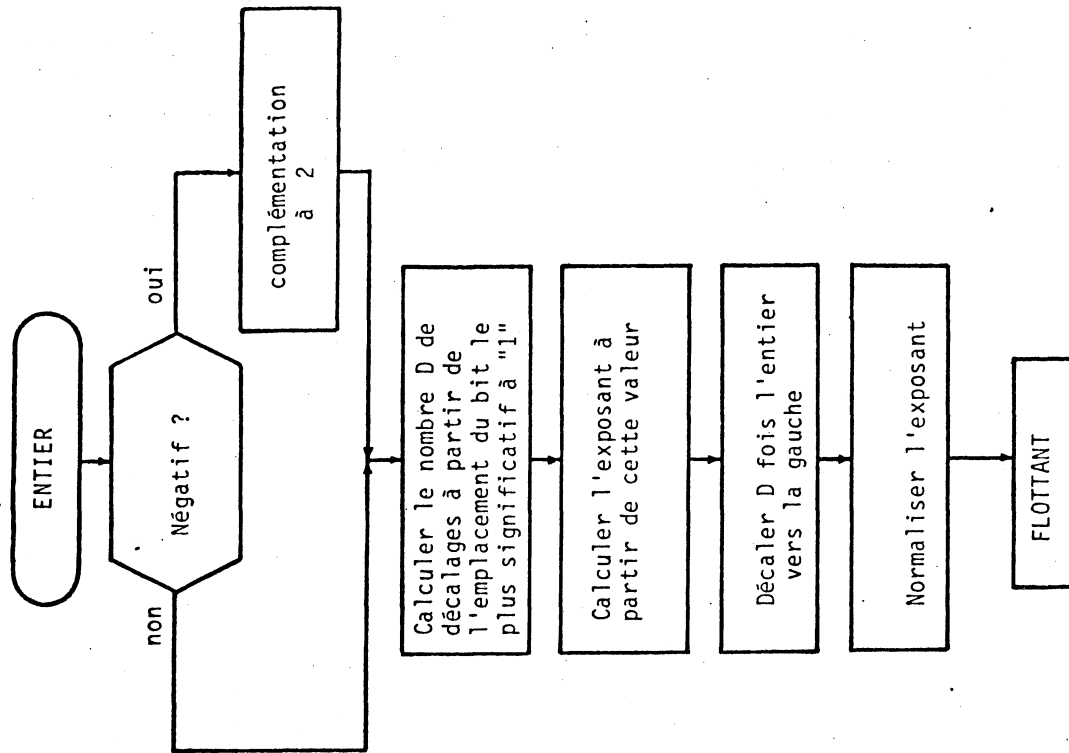


Fig.IV.25 : Conversions

- La réduction au même exposant, indispensable pour exécuter les additions flottantes est représentée figure IV.26.

Le principe utilisé consiste à dénormaliser le flottant le plus petit jusqu'à lui donner le même exposant que celui du plus grand. Pour ce faire un comparateur indique le plus grand exposant et positionne des multiplexeurs pour effectuer la soustraction des deux exposants dans le sens correct. Le résultat de cette soustraction donne le nombre de décalages à droite à effectuer sur le plus petit flottant pour lui donner l'exposant du plus grand.

Lorsque cette réduction est faite l'addition proprement dite est microprogrammée.

b.2) Réalisation et interface

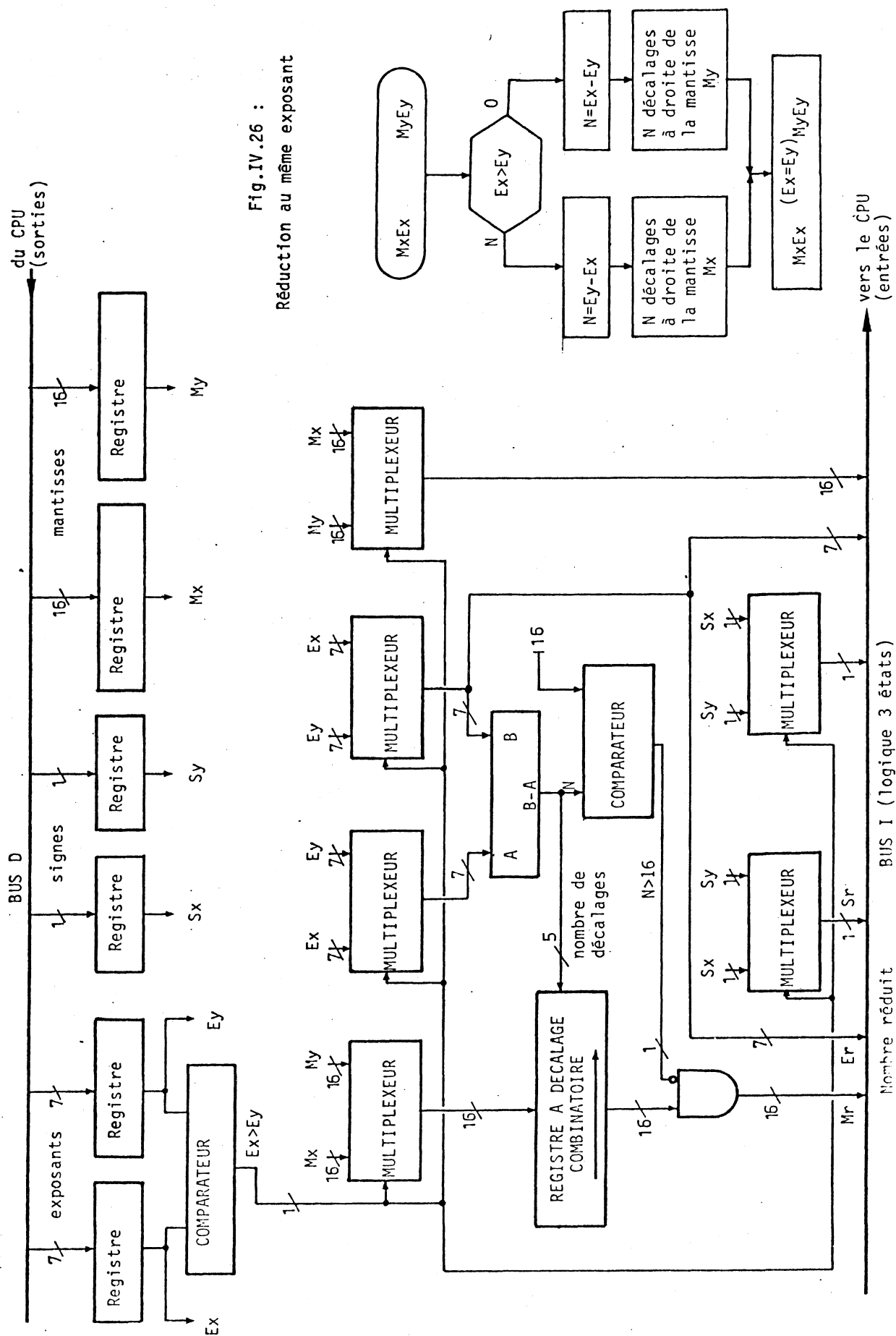
L'ensemble a été réalisé en TTLS et utilise environ 230 circuits. 6 unités ont été réalisées en wrapping automatique à l'aide de fichier descriptif.

La liaison opérateur-CPU utilise deux busses de 24 bits pour les données et un bus de contrôle.

Ce dernier se compose de 3 lignes d'adresses indiquant la fonction à exécuter (en lecture ou en écriture), d'une ligne de lecture-écriture, et d'une ligne d'occupation permettant à une unité centrale rapide de se synchroniser sur l'opérateur. Un indicateur de débordement est disponible sur une ligne séparée. Le tableau ci-dessous récapitule ces diverses fonctions. La connexion avec GESPRO a été réalisée de telle sorte qu'elle n'interfère pas avec la liaison mémoire. Dans ce but les lignes de fonctions ont été reliées non pas au bus d'adresse A mais directement à une partie du champ opérateur de la mémoire de microprogramme. Les lignes de données en lecture sont reliées au bus "I" (réservé aux périphériques). Le bus de données d'écriture est commun à la mémoire et à cette unité ce qui permet d'écrire dans l'une ou l'autre ou dans les deux à la fois.

Grâce à cette liaison directement accessible par microprogramme le temps d'exécution de la conversion d'un entier en flottant en mémoire centrale (lecture-conversion-écriture), peut être réduit au même temps d'exécution qu'une lecture-écriture mémoire soit une microseconde environ.

De même que l'unité de recherche du plus grand moment transverse, les tests de cette unité arithmétique ont nécessité la construction d'un interface spécial et l'écriture d'un ensemble de programmes de tests.



II.G. Mise en oeuvre des ensembles EVB-GESPROS

Grâce à la conception du matériel décrit ci-dessus et particulièrement bien adapté aux tâches d'acquisition et de filtrage, la programmation des calculateurs GESPROS est extrêmement simple dans son principe.

En particulier, les deux tâches les plus délicates sur l'expérience WA2, qui étaient la gestion mémoire et les gestions DMA n'existent plus ici puisqu'un lancement DMA suffit pour tout un bloc mémoire NORD et puisque la gestion des mémoires a été câblée, grâce à la conception des tampons d'événements. En conséquence il n'y a plus à traiter plusieurs niveaux d'interruptions et il n'y a plus non plus de dialogue de processeur à processeur.

Le principe, comme le montre la figure IV.19, consiste à attendre le déclenchement câblé (SDI) pour lancer le déclenchement logiciel, puis à attendre STA, signal de fin d'acquisition et de début d'analyse pour lancer s'il y a lieu les programmes ou microprogrammes de filtrages.

Cet état prend fin, soit à la fin du filtrage par l'un des signaux GOOD ou BAD émis par un des GESPROS, soit à l'interruption du filtrage par le contrôleur (STOP) en cas de dépassement de temps (Time Out). Après GOOD ou STOP l'événement est transféré vers NORD, à la fin de ce transfert ou après BAD l'EVB correspondant est remis en attente d'acquisition.

Tous les processeurs GESPROS sont initialisés par NORD Via CAMAC en début de prise de donnée ; Ils contiennent tous les mêmes programmes et les mêmes microprogrammes.

Leur mise en oeuvre comprend les tâches d'acquisition, de déclenchement de second niveau, de filtrage, de formattage et de contrôle.

a) L'acquisition :

Elle est entièrement automatisée par la chaîne DMA-contrôleur d'EVB - EVB-RMH. Un seul ordre de lancement DMA suffit donc pour remplir un bloc mémoire NORD sans autre action logicielle.

Lorsque le déclenchement de premier niveau est activé, il émet un signal qui mémorise les informations simultanément dans le système RMH et dans les cartes de recherche de plus grand PT.

Le contrôleur retransmet ce signal à l'EVB libre de plus haut rang et les deux tâches de lecture et de déclenchement logiciel démarrent simultanément. Cet état prend fin lors de la fin normale de la lecture par le signal correspondant. Une fin anormale en erreur (Dépassement en temps) permettrait de relancer l'EVB en cas de défaillance RMH.

b) Le déclenchement logiciel

Il a pour but essentiel, outre la réduction de données qu'il représente, de réduire le temps mort. Ce dernier est au minimum égal au temps de lecture RMH soit environ 100 μ s mais au maximum égal au temps de transfert NORD par CAMAC soit environ 1,5 ms.

Comme le taux de comptage incident est élevé, le déclenchement de premier niveau doit être très rapide (cf. § B) et malgré la performance de l'électronique employée seul un événement sur dix environ est intéressant pour l'analyse physique.

L'idéal serait donc de tuer les 9 autres pendant le temps de lecture RMH ce qui réduirait le temps mort de l'expérience au temps mort du RMH. Ce résultat a été obtenu en deux étapes : la première au niveau déclenchement et la seconde au niveau filtrage.

Au niveau déclenchement, le résumé des données fournies par les cartes de PT décrites sous le § F permet de calculer la masse du dimuon produit selon sa formule exacte (cf. § B).

Ce calcul démarre dès la fin de l'exécution de l'algorithme de tri par les cartes de PT soit 5 μ s après le déclenchement de premier niveau. Ces 5 μ s sont mises à profit pour examiner et remettre à jour le mot d'état de l'interface EVB qui contient alors la totalité des informations concernant l'événement précédent et en particulier les erreurs éventuelles.

La préparation des calculs peut alors commencer : il y a 1, 3, 6, 10 ou 15 calculs complets à exécuter suivant que 2, 3, 4, 5 ou les six sextants sont touchés. La lecture des 6 cartes permet de déterminer tous les paramètres nécessaires aux calculs selon la formule :

$$\frac{M^2}{2} = PT_1, PT_2 (1 - \cos\theta_1 \cos\theta_2 - \sin\theta_1 \sin\theta_2, \cos(\varphi_1 - \varphi_2))$$

Cette lecture fournit également les combinaisons des sextants touchés. Pour chaque sextant trois mots sont mis à jour : $\cos\theta$, $\sin\theta$, et PT recherché dans les deux tables TABV et TABPI (cf. figure 27) . Tous ces triplets sont rangés dans la table TABVAL, accessible grâce à la table de chaînage TABPT, adressée indirectement par la combinaison de sextants correspondants et qui contient également, pour chaque combinaison, le terme $\cos(\varphi_1 - \varphi_2)$. (Ce dernier terme est l'approximation obtenue à partir du milieu des sextants correspondants).

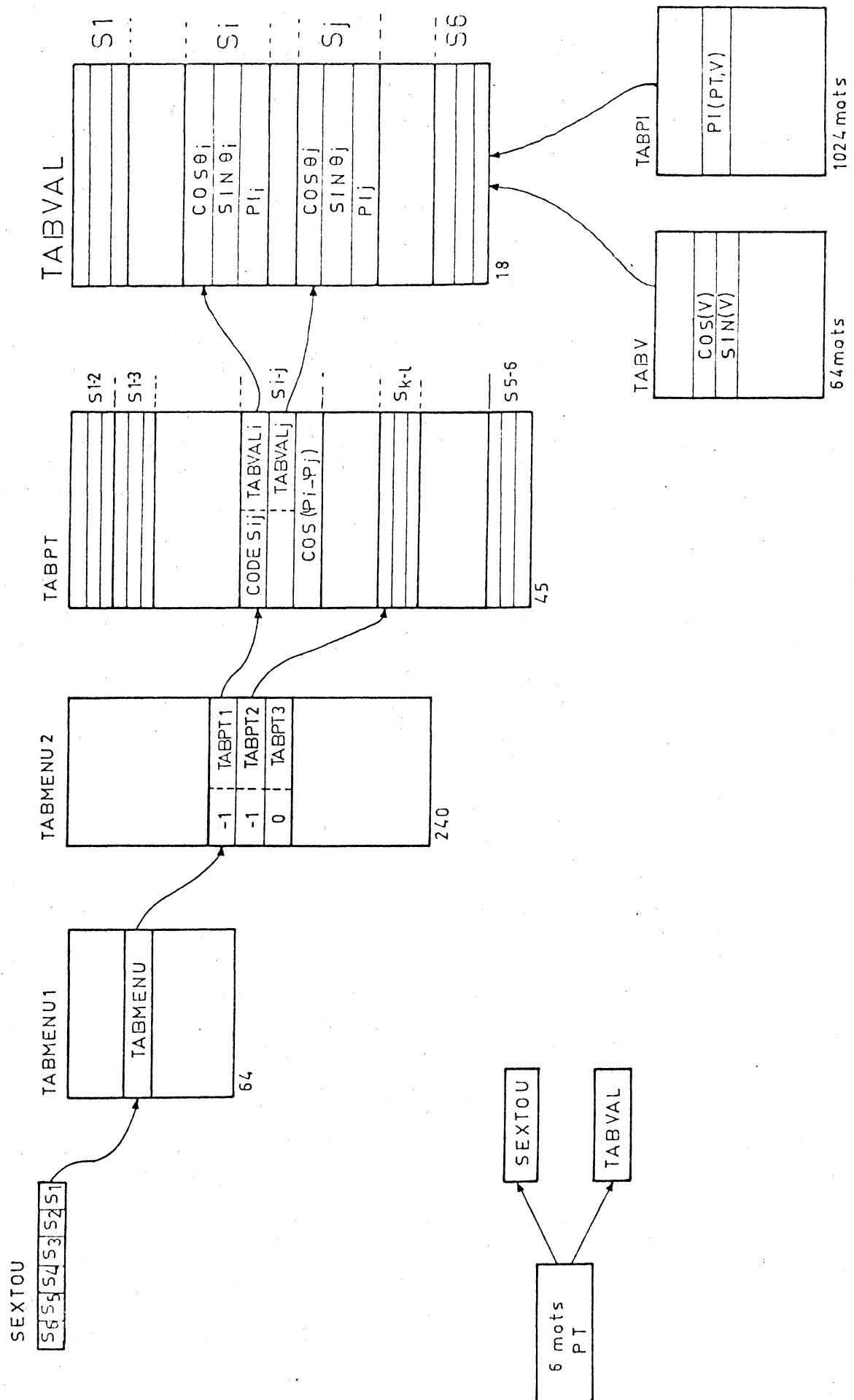


Fig.IV.27 : Tables de chaînage du calcul de masse

L'ensemble d'un calcul prend environ 13 μ s.

Ces opérations sont toutes contenues dans une seule instruction, qui outre la lecture des cartes, la mise à jour des tables et les calculs de masse effectuent la mise à jour de la plus grande masse trouvée et de la combinaison de sextants correspondante.

Dans le cas le plus défavorable, l'ensemble des opérations dure 360 μ s mais dans les cas les plus probables (2 et 3 sextants touchés), la durée d'exécution de l'instruction (50 et 96 μ s) reste inférieure au temps de lecture RMH.

Les événements sont alors classés en trois catégories :

- les événements de haute masse, minoritaires, qui intéressent la physique
- les événements de masse intermédiaire (PSI) qui représentent un grand intérêt pour le contrôle de l'appareillage (cf. § d)
- les événements de basse masse sans intérêt.

La figure 28 montre l'histogramme des temps d'exécution obtenu entre le déclenchement de 1er niveau et la fin du déclenchement de second niveau (lecture RMH et déclenchement logiciel). Une première décision peut alors être prise qui dépend du mode de fonctionnement. Les événements n'intéressant pas la physique peuvent être éliminés et l'EVB correspondant remplacé en attente d'un nouveau candidat. Ils peuvent être simplement étiquetés ce qui permet un meilleur contrôle tout en conservant le gain en analyse hors faisceau grâce à l'utilisation de l'étiquette (cf. § d).

Dans ce dernier cas tous les candidats sont admis au stade suivant : le filtrage sur données acquises.

c) Le filtrage (ou analyse des données acquises)

Cette analyse effectue la dernière réduction de données avant mise sur bande magnétique. Le nombre d'événements à analyser est donc à ce stade relativement faible (10^2 environ par cycle pour notre expérience).

Le temps disponible est lui aussi plus important : il est du même ordre de grandeur que le temps de transfert EVB-NORD soit de l'ordre de 2 ms, et au maximum égal au temps de mise sur bande magnétique si l'on obtient un fort taux de rejet (seul les événements rejetés en moins de 2 ms donnent lieu à réduction de temps mort).

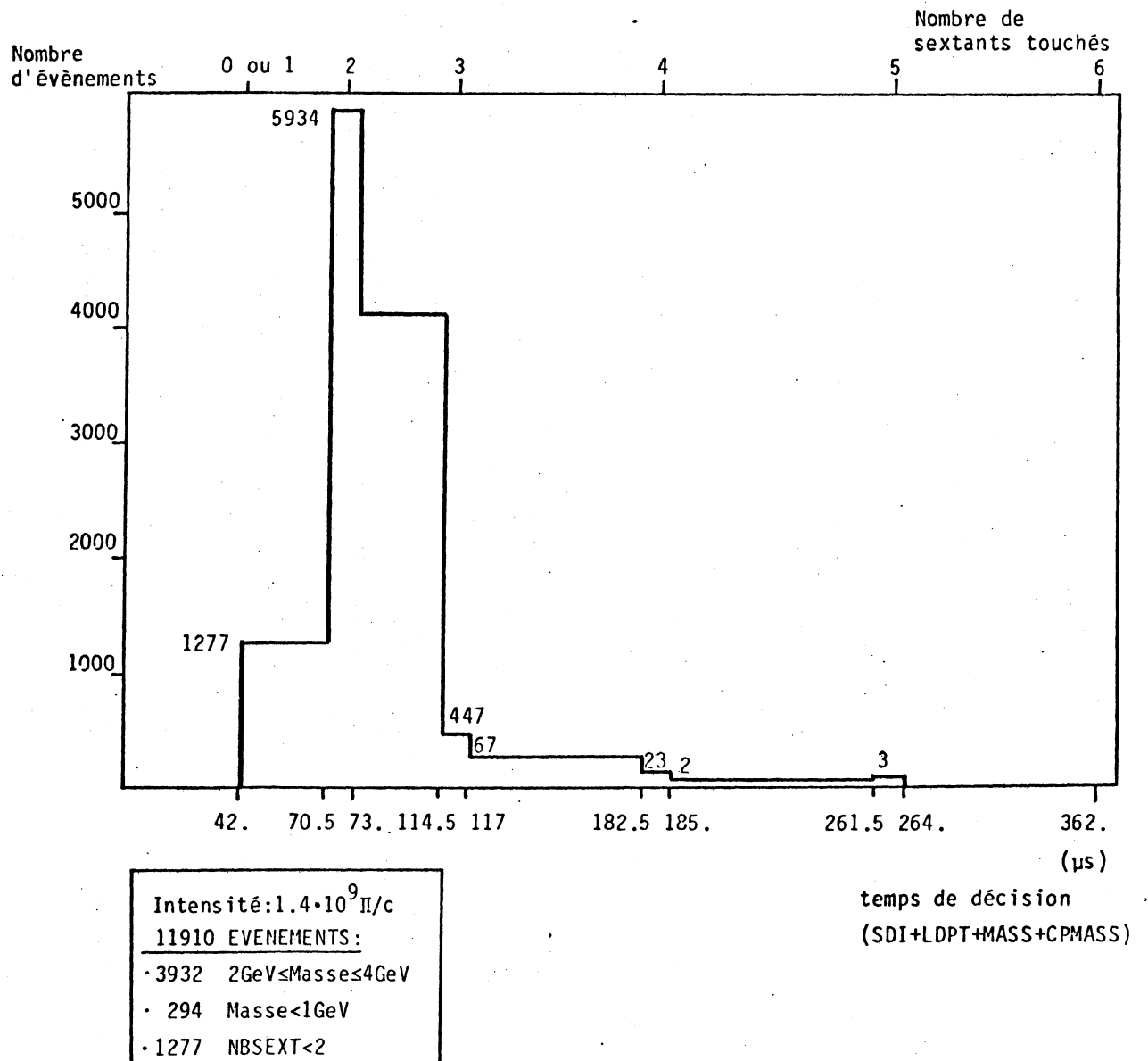


Fig.IV.28 : Histogramme des temps d'exécution du déclenchement logiciel

Toutes les informations mesurées sont disponibles au moment du filtrage, mais l'efficacité du filtrage sur les données hodoscopes s'est révélée suffisante, de sorte qu'aucun traitement sur les chambres n'a été retenu. Nous verrons au paragraphe consacré aux performances qu'en cas de nécessité, ce type de traitement pourrait être rajouté car une grande partie du temps GESPRO reste disponible.

Le filtrage proprement dit se décompose en deux parties, la première comporte tous les tests de validité de l'événement, la seconde concerne des coupures agissant en fonction de la nature de l'événement.

La première partie consiste à vérifier les marqueurs de la chaîne d'acquisition, et la consistance des données (adresses hodoscopes comprises dans la fenêtre correspondant à leur emplacement physique).

Enfin la validité de la taille de l'événement est vérifiée : cette valeur doit être supérieure à 36 mots (entête minimum) et inférieure à la limite programmée dans l'EVB.

La deuxième partie exécute une coupure (IRAWMAX) sur la taille des événements (les événements longs à haute multiplicité sont irreconstructibles). Une autre coupure (NTCO) est appliquée au nombre de compteurs touchés. Lorsque la première coupure est placée à 800 données et la seconde à 40 compteurs, les facteurs de rejets sont respectivement de 2,4 et de 1,2 pour des temps d'exécution de 90 μ s et 220 μ s après STA.

L'alignement des compteurs azimutaux P_1P_2 est ensuite vérifié pour la combinaison de sextants ayant donné la plus haute masse (facteur de rejet 1, 3, temps : 310 μ s après STA).

Enfin un test d'alignement général V , R_3 et R_4 peut être demandé : il est plus sélectif que la logique VR_3R_4 qui a été simplifiée pour préserver la modularité.

Notre programme d'analyse complet est exécuté en moins de 400 μ s comme le montre le tableau ci-dessous, pour un facteur de rejet supérieur à 5. Plus de détails au sujet de cette programmation peuvent être trouvés dans les réf. [16] et [23].

Type de rejet	Réjection	Temps d'exécution à partir de STA
MEMORY OVERFLOW	1.65	0 (câblé)
IRAWMAX (800)	2.37	90 μ s
NTCO (40)	1.2	270 μ s
P1P2	1.3	310 μ s
VR3R4	1.1	360 μ s

Fig.29

Indépendamment de toute coupure, un test permet d'étiqueter les événements ayant au moins trois sextants touchés. Ce test facultatif permet d'éviter le calcul de masse pour ces événements et de les transmettre pour analyse ultérieure.

d) Les contrôles en ligne

La sécurité de fonctionnement d'un tel système est primordiale, surtout lorsque l'on procède à l'élimination d'événements sur des critères complexes non vérifiables par le simple examen en ligne d'un événement. L'une des caractéristiques de notre système de filtrage est justement la sécurité obtenue grâce à l'utilisation simultanée de deux procédés. Le premier consiste à faire en sorte que tous les processeurs se contrôlent mutuellement. Ainsi les calculateurs GESPROS contrôlent les marqueurs RMH et la validité du déclenchement. De plus ils contrôlent également la croissance des adresses compteurs et leur validité.

D'autre part ils histogramment toutes les caractéristiques des événements qui les concernent : chaque événement est étiqueté par un mot d'état qui signale toute anomalie. Un mode de fonctionnement particulier permet de traiter l'événement sans le jeter afin de permettre la vérification hors faisceau de l'exactitude de la coupure concernée. La valeur de toutes ces étiquettes est histogrammée et la table ainsi constituée est lue par NORD et visible à l'opérateur. Tous les signaux de contrôle (SDI, STA, GOOD et BAD) et toutes les branches de sortie des différents programmes, aussi bien en mode test (non rejet) qu'en mode rejet sont comptabilisées.

Les histogrammes NORD peuvent ainsi être comparés aux histogrammes calculés en ligne.

Un autre procédé de contrôle concerne non seulement la chaîne d'acquisition mais également la validité des coupures, afin de s'assurer qu'aucun biais n'est introduit par les réductions de données. A cet effet un échantillonnage d'événements tirés au hasard dans une proportion programmable ne subit aucune coupure mais est étiqueté avec les résultats de tous les tests. La validité des coupures peut ainsi être vérifiée de façon certaine à tout moment de la prise de données grâce à la double histogrammation des événements biaisés et non biaisés. Enfin le comptage des événements de masse intermédiaire (PSI) est une constante caractéristique de l'expérience. La vérification de cette constante s'est révélée le moyen le plus sûr de vérifier l'ensemble de l'appareillage : toute variation du pourcentage de PSI dans les particules incidentes

indique un mauvais fonctionnement de l'acquisition.

Toute cette procédure réalisée par les microprocesseurs GESPRO libère le couple NORD 100-500 d'une forte quantité de travail et lui permet de se consacrer efficacement à l'histogrammation, au contrôle de l'appareillage, à la communication opérateur-machine et à la reconstruction de traces.

Un résumé complet d'une prise de données de juin 1982 est fournie en annexe, on peut y remarquer l'efficacité du contrôle et les résumés détaillés sur les rejets réalisés (cf. § H).

On remarquera également que la totalité des événements sont histogrammés.

II.H. Performance de l'ensemble EVB-GESPRO - Résultats expérimentaux et déclenchements spécialisés.

a) Performance

Le résumé de la prise de données de juin 1982 fourni en annexe montre les performances du système. Quatre couples EVB-GESPRO ont examiné plus de 10^5 événements dont seuls 20 000 ont été enregistrés. Le premier processeur a travaillé sur 94568 événements alors que le dernier n'est intervenu que 43 fois - autrement dit, pour cette prise de données, deux couples auraient été suffisants. Des essais réalisés à très haute intensité, à $3,5 \cdot 10^9$ particules par cycle, n'ont pas saturé le système.

La figure 28 donne l'histogramme des temps de calcul du déclenchement logiciel. Le temps moyen obtenu est de 72 μ s par événement, à intensité moyenne pour l'expérience ($1,5 \cdot 10^9$ particules par cycle).

A intensité moyenne le nombre de mots par événement avant coupure GESPRO est de 500. Après coupure il est de 200. Nous avons vu d'autre part que le taux de rejet était de 5 environ soit pour 250 événements par cycle, 50 événements mis sur bande. Le volume total des données a donc été réduit de $1,25 \cdot 10^5$ mots à 10^4 , soit un taux de réduction de 12,5.

Enfin, au point de vue performance technique pure, lorsque tous les rejets sont activés à haute intensité, le taux de rejet est supérieur à dix (calcul de masse compris), ce qui, compte tenu du temps nécessaire à la prise de décision (fig.27 et fig.28), n'introduit aucun temps mort supplémentaire au temps mort RMH, et permet l'analyse de plus de 1000 événements par cycle machine.

ANALYSE DE LA PRISE DE DONNÉE N° 1973 de Juin 1982

Condition de la prise de données :

- intensité $\approx 2.10^9 \pi$ par cycle.
- quatre couples EVB-GESPRO actifs.
- nombre d'événements examinés par le système : 104947
dont 94568 par le premier couple EVB-GESPRO et 49 par le dernier
- mode de fonctionnement :
- Le calcul de masse n'est pas utilisé comme déclenchement mais en mode étiquetage (donc après les rejets).
- Les coupures en multiplicités sont toutes les deux actives
- La validité VR_3R_4 est en mode étiquetage seulement

Résultats correspondants au couple N° 0 :

- 94568 événements examinés - 18161 écrits sur bande magnétique soit un facteur de rejet de 5,2 pour les deux coupures activées.
- L'histogramme en masse des événements non biaisés comparé à celui de la masse des événements biaisés montre l'amélioration apportée au contrôle sans perte d'efficacité (cf. Annexe)
- Si tous les rejets étaient activés on obtiendrait pour 94568 déclenchements :
 - une réduction à 32386 par coupure câblée sur le nombre de mots
 - une réduction à 18161 par les filtrages activés sur cette prise de données
 - une réduction à 3635 par les autres filtrages, en mode étiquetage sur cette prise de données.

Cette dernière réduction se décompose en :

1 erreur matérielle RMH

- 2156 événements non biaisés à retrancher de cette statistique
- 1849 rejetés par le test VR_3R_4
- 1486 dont le nombre de sextants touchés n'est pas au moins égal à deux
- 44 étiquetés "Trimuons"
- 8990 rejetés par le calcul de masse donnant 555 masses faibles et 8435 masses intermédiaires. A noter que ce calcul de masse, utilisé ici en mode étiquetage donc après rejet a été effectué sur 12625 événements et a donc un pouvoir de coupure de 3,4.

Enfin la somme de toutes les coupures fournit une réjection de 90933 événements pour 94568 entrées soit un taux de réjection de 26, ce qui réduit le nombre d'événements moyen par déversement de 389 à 15.

La précision du calcul de masse effectué par GESPRO est évidemment limitée par la précision des capteurs (dimension des scintillateurs).

Les histogrammes de la figure 30 montrent la dispersion en masse et l'erreur introduite par l'utilisation des scintillateurs par comparaison avec la masse calculée hors faisceau grâce aux chambres à fils (85,3 MeV en de valeur moyenne soit 2 % à la coupure, 476 MeV de largeur à mi-hauteur soit 12 % à la coupure).

b) Résultats expérimentaux

L'examen des histogrammes des événements non biaisés permet de montrer que les coupures n'entraînent aucune inefficacité.

Dans ces conditions le temps actif est meilleur que 95 % et le seul temps mort provient du transfert RMH-EVB. S'il était gênant, ce dernier pourrait encore être réduit d'un facteur deux par réinitialisation du RMH en cours de transfert à la fin du résultat du calcul de masse (72 μ s).

La fiabilité de cet ensemble est vérifiée à chaque prise de données par examen des histogrammes et des relations de fermeture consistant à vérifier que le nombre de déclenchements correspond bien au nombre des événements rejetés à chaque coupure augmenté du nombre d'événements enregistrés.

- Le gain en temps mort obtenu est égal au rapport de la vitesse CAMAC à la vitesse RMH, soit un gain supérieur à dix.
- Le gain en efficacité est encore supérieur puisque les événements non rejetés sont plus courts que les événements bruts.
- Le gain en contrôle et en histogrammation n'est pas chiffrable exactement puisqu'un système sans filtrage serait incapable de tenir les mêmes performances. Il faut noter cependant que la qualité des histogrammes est nettement supérieure et que par voie de conséquence le contrôle en ligne est plus efficace (fig.31 : comparaison des événements biaisés - non biaisés)

c) Déclenchements spécialisés

La simplicité d'utilisation des micro-processeurs GESPROS et leur temps de réponse ont permis de les utiliser pour calibrer les détecteurs. Sur l'expérience NA10, nous avons mis au point des déclenchements spécialisés pour

nombre d'évènement

- 120 -

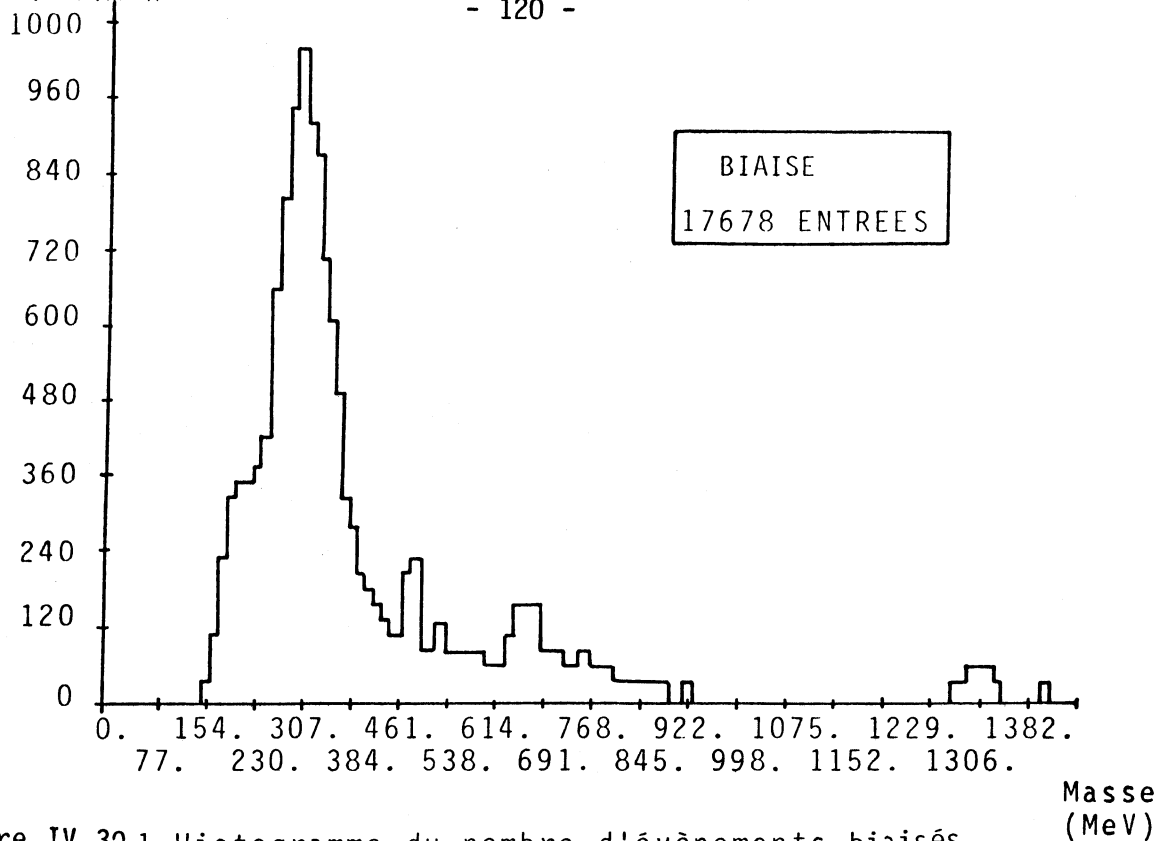


Figure IV.30.1. Histogramme du nombre d'évènements biaisés en fonction de la masse du dimuon

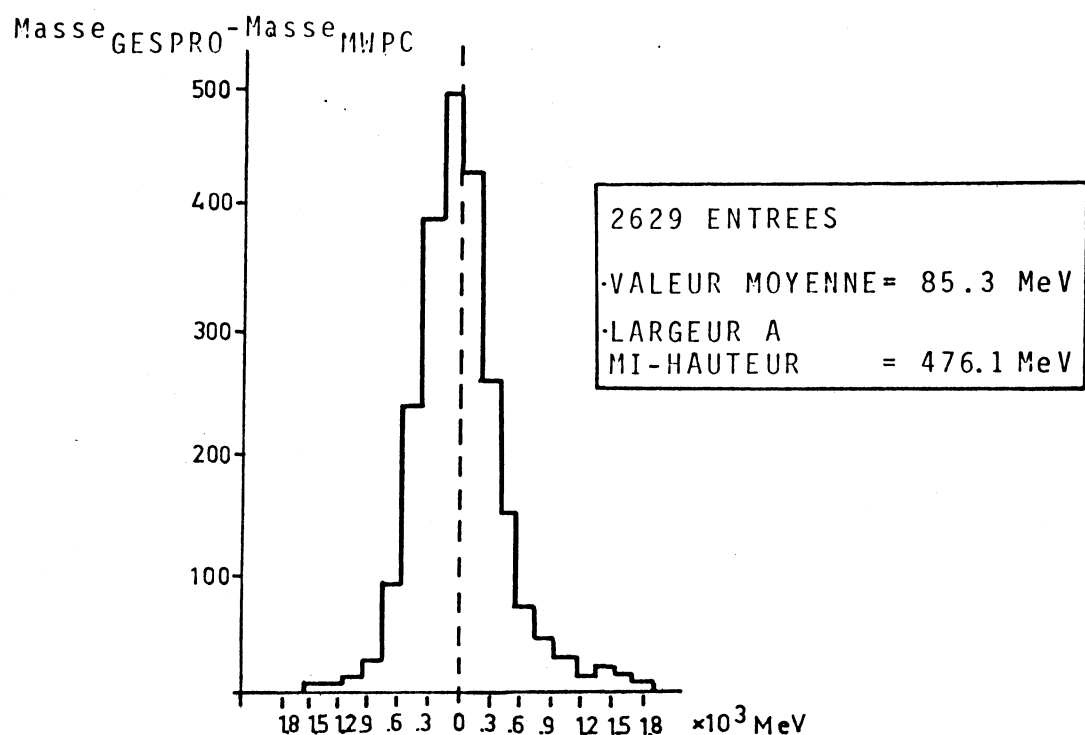


Figure IV.30.2. Résolution en masse: différence entre la masse calculée en ligne par GESPRO et la masse calculée hors ligne à partir des données chambres.

des applications particulières.

Dans ce cas on ne s'intéresse plus à la physique mais on recherche des critères de sélection sur une partie de l'appareillage non concernée par la mesure envisagée en vue d'enrichir les événements du type permettant cette application particulière.

Ainsi, pour déterminer l'alignement des chambres proportionnelles, on coupe le champ dans l'aimant et on programme un déclenchement une particule (4 compteurs $R_1 R_2 R_3 R_4$). Ceci fournit des événements très propres utilisables pour l'alignement de l'appareillage.

Pour déterminer l'efficacité du déclenchement standard, on a programmé un déclenchement de tests exigeant :

- un muon mesuré de façon standard dans l'un des sextants
- deux muons au niveau des compteurs P1P2
- deux traces pointant sur la cible au niveau de GESPRO. Pour ce faire on applique un critère de multiplicité et une corrélation entre les fils touchés.

L'action de GESPRO enrichit ce déclenchement d'un facteur 5 et permet l'acquisition avec un temps actif de 95 %. Sans cet enrichissement, les statistiques équivalentes auraient été obtenues en cinq fois plus de temps.

Cette dernière application de GESPRO est capitale dans la mesure où elle permet de réaliser rapidement les déclenchements spéciaux indispensables à la calibration de l'appareillage : efficacité des compteurs des hodoscopes R_i , efficacité des logiques $R_1 R_2$, $V R_3 R_4$ et de la logique intersextant.

C O N C L U S I O N

L'évolution des systèmes d'acquisitions entre 1975 et 1982 consacre l'application des microprocesseurs aux systèmes de contrôles et de tests, et, dans une moindre mesure aux systèmes de filtrage en ligne.

Au cours des deux expériences WA2 et NA10, le microprocesseur GESPRO a subi une évolution considérable : En effet, il a été initialement introduit pour gérer l'acquisition d'une grande variété de capteurs (GESTION PROGRAMMEE des entrées-sorties) et décharger les miniordinateurs de contrôle en traitant les erreurs matérielles dans leurs totalités et non par échantillonnage.

Au fur et à mesure de l'évolution de l'expérience WA2, des critères de sélection ont été implémentés par microprogrammation. Ils ont permis d'améliorer le rapport signal sur bruit tout en diminuant le temps mort et en apportant un gain important en temps de calcul. Cette évolution s'est accentuée sur l'expérience de production de dimuons.

L'introduction du déclenchement logiciel a permis un gain d'un ordre de grandeur sur la sélectivité et sur le temps mort, un contrôle plus strict du fonctionnement des capteurs et une réduction du temps d'analyse.

Au travers de ces applications on voit que GESPRO, de processeur d'acquisition est devenu processeur de traitement, processeur de déclenchement et de filtrage puis instrument perfectionné de mesure.

Ces étapes témoignent de ses capacités d'adaptation : de simple élément de transmission, il est devenu capteur actif, capable de prendre des décisions suivant une grande variété de paramètres :

- nombre de compteurs touchés
- nombre de fils touchés
- amplitude de signaux
- temps d'arrivée des signaux détectés dans les capteurs
- corrélation entre compteurs pour calculer des directions
- calculs arithmétiques complexes
- corrélation entre fils s'il est nécessaire d'obtenir une meilleure précision, etc...

Les futures expériences de hautes énergies laissent prévoir une évolution dans le sens de la multiplication des machines spécialisées, tout en laissant prévoir également une augmentation en taille et en nombre des machines banalisées.

Cette multiplication des ordinateurs spécialisés imposera un effort au niveau de la conception de leur logiciel afin de les rendre utilisables comme une machine ordinaire.

En effet si la deuxième version de GESPRO est particulièrement simple par rapport à la première, il n'en reste pas moins vrai que cette machine demeure inutilisable sans les conseils d'un spécialiste. Si le nombre des machines de ce type augmente (plusieurs dizaines de programmations différentes par expérience), il devient indispensable de les mettre à la portée des physiciens tout en conservant leurs performances.

Ces efforts seront donc à soutenir essentiellement au niveau logiciel, soit par traduction de langages connus, soit par la mise au point d'un langage propre adapté à ce type de tâche.

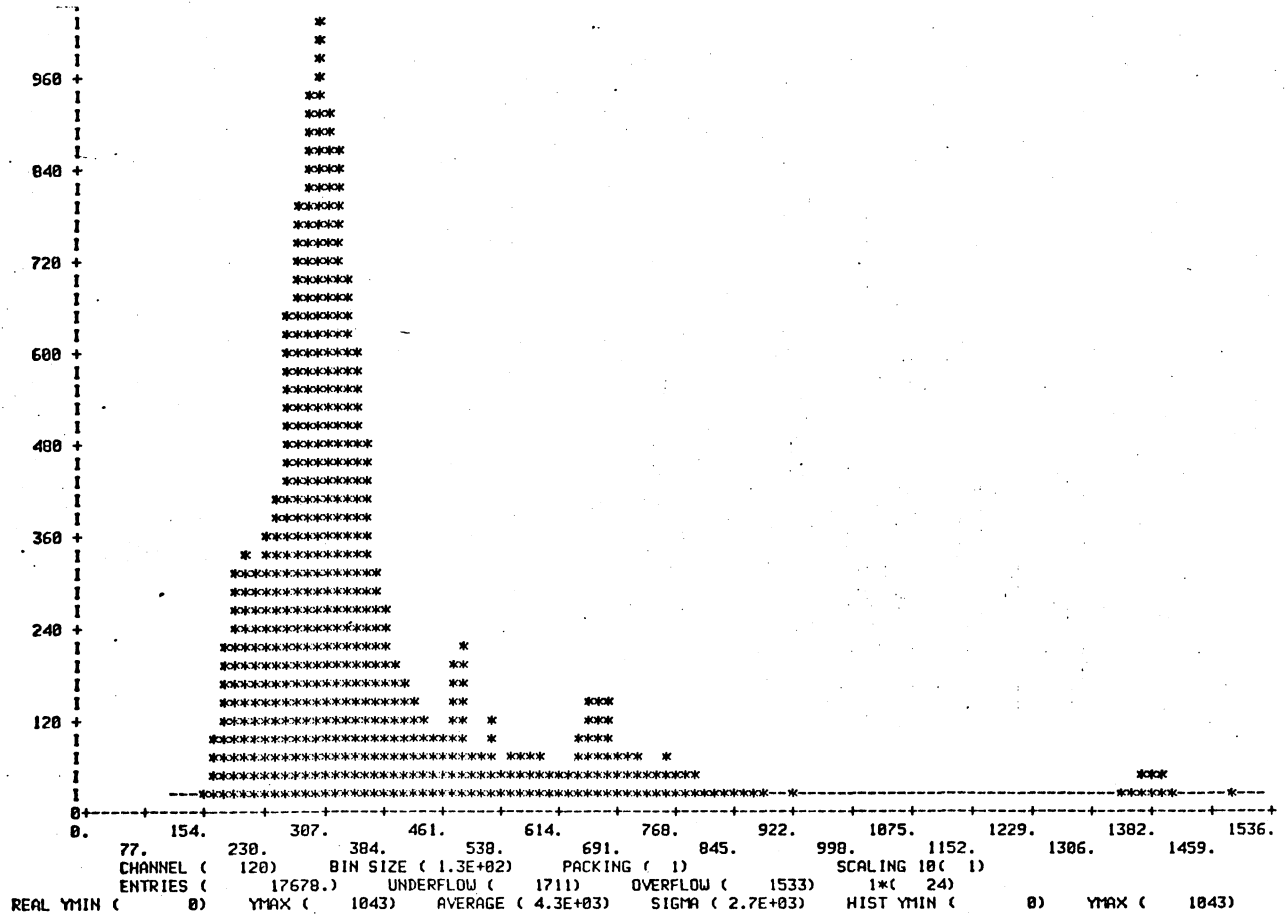
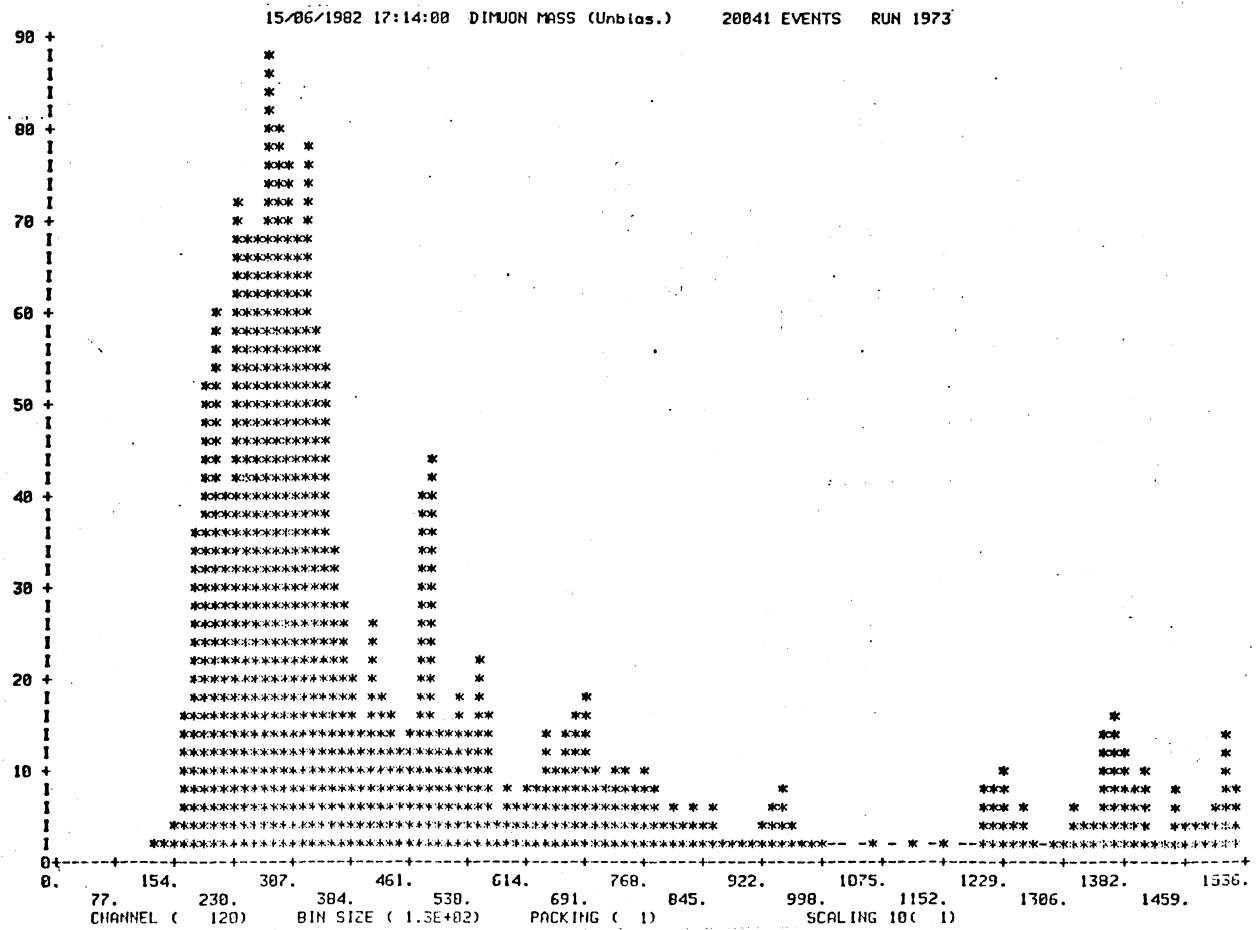


Fig.IV.31 : Histogrammes de la masse calculée en ligne pour des événements
biaisés et non biaisés.



ANNEXE AU CHAPITRE II

Etude comparative des différents micro-processeurs rapides et évolution de 1976 à 1981

La première utilisation se situe en 1975 pour l'étude du système d'acquisition destiné à l'expérience "HYPERON". Le seul processeur rapide du marché est l'Intel 3000.

De 1976 à 1980 pendant le développement du système Hypéron puis pendant les projets de structure destinés aux expériences NA10-VA2 trois processeurs en tranche sont disponibles : l'Intel 3000, l'AMD 2900 et le Motorola 10 800. Leur étude comparative citée ici, se situe en 1978-1979.

I. TYPE :

I.3000 Tranche de 2 bits, 100 ns de temps de cycle, technologie TTLS. Structure multibus.

AMD 2900 Tranche de 4 bits, 95 ns de temps de cycle, technologie TTLS, Structure unibus.

Motorola 10 800 : Tranche de 4 bits - 55 ns de temps de cycle, technologie ECL - demande quelques composants supplémentaires, Structure unibus.

II. NOMBRE DE REGISTRES

AMD 2900 : C'est ce circuit (2901) qui possède le plus de possibilité à ce point de vue : 16 registres et une structure à 3 adresses. Cet élément peut à ce sujet être considéré comme complet.

I 3000 : 12 registres seulement mais le choix des deux busses d'entrée et des deux busses de sortie mémorisés avec action d'un cinquième buss de masquage étend les possibilités.

La structure à deux adresses seulement est une faiblesse

Motorola 10800 : Cet élément ne possède qu'un accumulateur et un registre de sortie, ce qui impose le câblage de circuits extérieurs. C'est la plus grande faiblesse de cette famille.

III. STRUCTURE DE L'UNITE ARITHMETIQUE ET LOGIQUE

AMD 2900 : Un seul buss en entrée et un seul buss en sortie. Cette structure très pauvre pénalise lourdement cette famille qui ne se prête pas aux entrées-sorties rapides. Le ralentissement occasionné de ce fait (facteur deux environs) ne peut être masqué sauf câblage d'une unité de commande supplémentaire, pour la mémoire de donnée par exemple. De même les données mémoire, les données externes (acquisition) et internes (interruption, diagnostics, masquage ...) se partagent le buss unique d'entrée.

En conséquence l'adjonction d'un multiplexeur rapide commandable par un champ supplémentaire est la seule solution ne pénalisant par trop les performances. Il s'ensuit une augmentation du nombre de bits de la mémoire de commande avec ralentissement du temps de cycle. Sur ce point le plus faible de la série l'AMD doit être considéré comme incomplet.

I 3000 : La structure à cinq busses du I 3002 est son point le plus fort. Deux busses de sortie munies de registre tampon mémorisent adresse et données simultanément ou séquentiellement. Leurs sorties à 3 états permettant la mise en parallèle d'autres unités identiques ou non. Deux busses d'entrée séparés permettent d'éviter le multiplexage du buss mémoire et d'optimiser ainsi sa vitesse d'exécution. Les périphériques étant connectés à un autre buss, on peut masquer les temps d'accès par programmation judicieuse des appels mémoire et périphérie. De plus les données recueillies peuvent subir un traitement supplémentaire simultané de masquage grâce au cinquième buss ; on peut ainsi réaliser simultanément une opération logique et une opération arithématique à destination quelquefois séparées.

Cet élément peut être considéré comme exceptionnellement performant en entrée-sortie et économique en longueur du champ correspondant de micro-commande.

MOTOROLA 10800 : L'élément central est organisé autour de 3 busses dont deux sont bidirectionnels.

Moins évolué que l'I 3000 mais plus que l'AMD le 10800 offre sur ce point un compromis valable.

IV . LES FONCTIONS DE L'UNITE ARITHMETIQUE ET LOGIQUE

AMD 2900 : Le jeu d'instruction du 2901 est bien fait clair et complet.

3 sous-champ bien définis permettent une microprogrammation aisée sur 9 bits, proche d'une parfaite symétrie.

Les indicateurs sont complets (zéro, retenue, signe) ce qui rend agréable la programmation d'opération arithmétique malgré le manque d'opérateur simultané de masquage

Le plus efficace sur ce point, l'AMD offre de plus le meilleur compromis nombre de bits/possibilités

I 3002 : Un codage très complexe des fonctions a permis de réduire à 7 le nombre de bits de fonctions. Cependant ce codage extrême nuit à la compréhension des fonctions.

Avantage fondamental provenant de la structure à 5 busses ; les entrée-sortie sont puissantes. L'exemple suivant est significatif. (MAR est le registre de sortie adresse, R_n un registre pouvant être le registre de sortie donnée, K le buss d'entrée de masque et CI la retenue)

$$R_n \vee K \rightarrow \text{MAR}$$

$$R_n + K + \text{CI} \rightarrow R_n$$

Si R_n est l'accumulateur on remarque :

- utilisation simultanée de 3 busses dans la même micro instruction
- exécution de deux types d'opérations simultanées (l'une logique et l'autre arithmétique avec retenue)
- la même micro instruction existe en référence mémoire et utilise donc 2 busses simultanément

Les fonctions offertes sont réalisables sur le bus M, sur le bus I ou sur les registres.

Un seul indicateur multiplexé en entrée et un seul en sortie est astucieusement dédoublé dans l'unité de contrôle de la mémoire de commande.

L'utilisation de ces deux mémorisation s'avère puissante mais complexe. La grande faiblesse est l'absence d'indicateur purement arithmétique (ni retenue ni signe).

En résumé cet élément est économique en bits (champs de 7 bits) il est très puissant pour les opérations logique - faible en arithmétique avec manque de rotation arithmétique - manque d'indicateurs de retenue et de signe . De par sa structure multibus se prête manifestement très bien aux entrées-sorties et aux systèmes nécessitant des opérations logique simultanées.

MOTOROLA 10800 : C'est l'élément dont les fonctions ont été les moins codées. Il s'en suit une grosse consommation de bits : 17 bits pour ce seul champ P.

Les opérations arithmétiques et logiques sont complètes, les indicateurs également avec possibilité d'arithmétique BCD.

V. UNITE DE CONTROLE DE LA MEMOIRE DE MICROPROGRAMME

AMD 2909 : C'est un élément cascade à 4 digits sans unité de contrôle proprement dite. L'adressage est incrémental, le moins efficace dans ce cas. Il peut être assimilé à un compteur et un multiplexeur à 4 entrées permettant le choix entre :

- l'incréméntation de l'adresse
- une pile pour sous-programme
- deux sources extérieures.

Le lien avec l'instruction, seul capable d'indiquer efficacement la suite des opérations, n'existe pas et ne peut qu'être rajouté sous forme d'élément discret et d'une ROM de translation. L'interruptibilité n'est pas traitée.

Cette unité est le point le plus faible d'AMD, le contrôle des modes d'accès n'est pas lié à l'instruction, l'adressage n'est pas efficace, le nombre de bits de commande est élevé pour un résultat médiocre - l'interruptibilité n'est pas traitée.

I 3001 : Cet élément unique à très grand codage et forte puissance ne couvre qu'un espace de travail restreint (512 mots) cependant le mode d'extension par décomposition en plan est indiqué par le constructeur, et la puissance des modes d'adressage permet la réutilisation systématique des cases déjà écrites par débranchement arborescent.

Basé exclusivement sur la méthode d'adressage la plus codée, relative et conditionné au code opération de l'instruction courante le 3001 se souvient d'une partie du code opération ce qui lui permet de retrouver son chemin même après une référence mémoire.

Il traite de plus la double mémorisation de l'indicateur avec débranchement conditionnel sur la valeur courante ou l'une des valeurs mémorisée.

Enfin la structure de microprogrammation arborescente à origine Fetch-phase lui permet de traiter efficacement le problème de l'interruptibilité par émission d'un signal spécial, anticipé, au début de chaque microprogramme. Toute cette puissance n'est accompagnée d'aucune possibilité d'utilisation vraiment simple (incrémentale) de l'adressage.

Les outils logiciel doivent donc être très sophistiqués et le constructeur n'en propose aucun.

Cette carence est sans doute la cause du faible succès rencontré par cet élément : en l'absence de système de développement, les utilisateurs lui ont préféré des concurrents moins efficaces mais d'utilisation beaucoup plus simple.

MOTOROLA 10801 : Cette unité offre un bon compromis entre codage maximum et utilisation simple.

Les sauts conditionnés au code opérations sont possibles. L'utilisation de sous-programme est prévue. A noter qu'un registre spécial permet la répétition d'une microinstruction unique.

Les interruptions peuvent être traitées par le biais des indicateurs (forme de débranchement plutôt que d'interruption proprement dite)

CONCLUSION

Chaque famille a ses avantages et inconvénients mais aucune ne peut être considérée comme complète. Les inconvénients les plus flagrants sont :

- absence de registre (10800)
- faiblesse arithmétique (I 3000)
- structure unibus (10800 - 2900)
- Difficulté d'utilisation et absence de système de développement de INTEL.

Les avantages principaux :

- Très bon jeu de microinstruction (2900)
- Très bonne structure en entrée-sortie (I3000)
- Bon ou très bon système d'adressage (10800 - I3000)
- Très grand codage donc très petit mot de mémoire de commande (I3000)

F-GROUP	R-GROUP	MICRO-FUNCTION
0	I II III	$R_n + (AC \wedge K) + CI \rightarrow R_n, AC$ $M + (AC \wedge K) + CI \rightarrow AT$ $AT_L \wedge (I_L \wedge K_L) \rightarrow RO \quad LIVE(I_H \wedge K_H) \wedge AT_H \rightarrow AT_H$ $AT_L \wedge (I_L \wedge K_L) \vee AT_H \wedge (I_H \wedge K_H) \rightarrow AT_H$
1	I II III	$K \vee R_n \rightarrow MAR \quad R_n + CI + K \rightarrow R_n$ $K \vee M \rightarrow MAR \quad M + CI + K \rightarrow AT$ $(AT \vee K) + (AT \wedge K) + CI \rightarrow AT$
2	I II III	$(AC \wedge K) - 1 + CI \rightarrow R_n$ $(AC \wedge K) - 1 + CI \rightarrow AT$ $(I \wedge K) - 1 + CI \rightarrow AT$
3	I II III	$R_n + (AC \wedge K) + CI \rightarrow R_n$ $M + (AC \wedge K) + CI \rightarrow AT$ $AT + (I \wedge K) + CI \rightarrow AT$
4	I II III	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO \quad R_n \wedge (AC \wedge K) \rightarrow R_n$ $CI \vee (M \wedge AC \wedge K) \rightarrow CO \quad M \wedge (AC \wedge K) \rightarrow AT$ $CI \vee (AT \wedge I \wedge K) \rightarrow CO \quad AT \wedge (I \wedge K) \rightarrow AT$
5	I II III	$CI \vee (R_n \wedge K) \rightarrow CO \quad K \wedge R_n \rightarrow R_n$ $CI \vee (M \wedge K) \rightarrow CO \quad K \wedge M \rightarrow AT$ $CI \vee (AT \wedge K) \rightarrow CO \quad K \wedge AT \rightarrow AT$
6	I II III	$CI \vee (AC \wedge K) \rightarrow CO \quad R_n \vee (AC \wedge K) \rightarrow R_n$ $CI \vee (AC \wedge K) \rightarrow CO \quad M \vee (AC \wedge K) \rightarrow AT$ $CI \vee (I \wedge K) \rightarrow CO \quad AT \vee (I \wedge K) \rightarrow AT$
7	I II III	$CI \vee (R_n \wedge AC \wedge K) \rightarrow CO \quad R_n \oplus (AC \wedge K) \rightarrow R_n$ $CI \vee (M \wedge AC \wedge K) \rightarrow CO \quad M \oplus (AC \wedge K) \rightarrow AT$ $CI \vee (AT \wedge I \wedge K) \rightarrow CO \quad AT \oplus (I \wedge K) \rightarrow AT$

SYMBOL	MEANING
I, K, M	Data on the I, K, M busses
R _n	Contents of register n (R-Group 1)
AC	Contents of the accumulator
AT	Contents of AC or T, as specified
CI	Data on the carry input
CO	Data on the carry output
L, H	As subscripts, designate low and high order bit
+	2's complement addition
-	2's complement subtraction
$\wedge$	Logical AND
$\vee$	Logical OR
$\oplus$	Exclusive NOR
$\rightarrow$	Deposit into

MNEMONIC	DESCRIPTION	FUNCTION							NEXT ROW					NEXT COLUMN				
		AC							MA						MA			
		6	5	4	3	2	1	0	8	7	6	5	4		3	2	1	0
JCC	Jump in Current Column	0	0	d ₄	d ₃	d ₂	d ₁	d ₀	d ₄	d ₃	d ₂	d ₁	d ₀		m ₃	m ₂	m ₁	m ₀
JZR	Jump in Zero Row	0	1	0	d ₃	d ₂	d ₁	d ₀	0	0	0	0	0		d ₃	d ₂	d ₁	d ₀
JCR	Jump in Current Row	0	1	1	d ₃	d ₂	d ₁	d ₀	m ₈	m ₇	m ₆	m ₅	m ₄		d ₃	d ₂	d ₁	d ₀
JCE	Jump in Column/Enable	1	1	1	0	d ₂	d ₁	d ₀	m ₈	m ₇	d ₂	d ₁	d ₀		m ₃	m ₂	m ₁	m ₀
JFL	Jump/test F-latch	1	0	0	d ₃	d ₂	d ₁	d ₀	m ₈	d ₃	d ₂	d ₁	d ₀		m ₃	0	1	f
JCF	Jump/test C-flag	1	0	1	0	d ₂	d ₁	d ₀	m ₈	m ₇	d ₂	d ₁	d ₀		m ₃	0	1	c
JZF	Jump/test Z-flag	1	0	1	1	d ₂	d ₁	d ₀	m ₈	m ₇	d ₂	d ₁	d ₀		m ₃	0	1	z
JPR	Jump/test PR-latches	1	1	0	0	d ₂	d ₁	d ₀	m ₈	m ₇	d ₂	d ₁	d ₀		p ₃	p ₂	p ₁	p ₀
JLL	Jump/test Left PR-bits	1	1	0	1	d ₂	d ₁	d ₀	m ₈	m ₇	d ₂	d ₁	d ₀		0	1	p ₃	p ₂
JRL	Jump/test Right PR-bits	1	1	1	1	1	d ₁	d ₀	m ₈	m ₇	1	d ₁	d ₀		1	1	p ₁	p ₀
JPX	Jump/test PX-bus	1	1	1	1	1	d ₁	d ₀	m ₈	m ₇	m ₆	d ₁	d ₀		x ₇	x ₆	x ₅	x ₄

d_n Data on adress control line n

m_n Data in microprogram adress register bit n

p_n Data in PR-latch bit n

x_n Data on PX-bus line n

f,c,z Contents of F-latch,C-flag,Z-flag

Code opération

x ₇	x ₆	x ₅	x ₄	p ₃	p ₂	p ₁	p ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

ST, R _n	[@] ADR	R _n → [(ADR)]	104 + R _r
STI, R _j , R _i	ADR	R _j → (R _i + ADR)	302 + R _j
*STIM, R _n	ADR, DATA	DATA → (ADR)	253
*STM, R _n	ADR, DATA	DATA → (ADR) ↓ (ADR + (i-1))	264 + R _n

Met data ds un tableau commençant à
ADR et ayant une longueur i contenu
dans (R_n)

LOAD :

Met le contenu d'une adresse ou une data dans un registre.

LD, R_n [@]	ADR	$[(ADR)] \rightarrow R_n$	$320 + R_n$
LDI, $R_j R_i$	ADR	$(ADR + R_i) \rightarrow R_j$	$300 + R_j$
LDD, A [@]	ADR	$[(ADR)] \rightarrow A$	230
		$[(ADR)+1] \rightarrow B$	
LDD, IE [@]	ADR	$[(ADR)] \rightarrow D$	231
		$(ADR + 1) \rightarrow E$	
*LIM, R_n	DATA	DATA $\rightarrow R_n$	$200 + R_n$
		24 bits	
*IML, R_n	DATAHL	DATAHL $\rightarrow R_n$	"
		8,16 bits	
SETA, R_n	DATAL	DATAL $\rightarrow R_n$	$240 + R_n$
		16 bits	

(La data étant sur 16 bits cette instruction ne prend
qu'un mot. Elle est donc plus rapide que LIM)

MOVE : transfert d'adresse à adresse

*MOVE	SADR, DADR, NBMOT	$(ADR1 + i) \rightarrow (ADR2 + i)$	
		$i = 0, NBMOT$	44

OPERATIONS ARITHMETIQUES

. addition

ADD, R_n	ADR	$R_n + (ADR) \rightarrow R_n$	$334 + R_n$
ADDI, R_j, R_i	ADR	$R_j + (ADR + R_i) \rightarrow R_j$	$314 + R_j$
*ADIM, R_j	DATA	$R_j + DATA \rightarrow R_j$	$212 + R_j$
*IMADD, R_j	DATAHL	$R_j + DATAHL \rightarrow R_j$	$212 + R_j$

RADD, R _j	si	j = A	A + B → A	360 + R _j
	si	j = B	A + B → B	
COPY, SR _n , DR _n	DATAL	SR _n + DATAL → DR _n , SR _n		340 + SR _n * 4 + DR _n
				<u>16 bits</u>

. soustraction

SUB, R _n	ADR	R _n - (ADR) → R _n	324 + R _n
SUBI, R _j , R _i	ADR	R _j - (ADR + (R _i)) → R _j	316 + R _j
*SUBIM, R _j	DATA	R _j - DATA → R _j	207 + 8 * R _j
*IMSUB, R _j	DATAHL	R _j - DATAHL → R _j	"
RSUB, R _j		si j = A A - B → A	362 + R _j
		si j = B B - A → B	

. autres opérations

TZR, R _n	Ø → R _n	123 + 4 * R _n
CP, R _n	$\overline{R_n} \rightarrow R_n$	364 + R _n

Opérations logiques

ET

AND, R _j [@] ADR	R _j ∧ [(ADR)] → R _j	220 + R _j
ANDI, R _j , R _i ADR	R _j ∧ (ADR + R _i) → R _j	306 + R _j
*ANDIM, R _j DATA	R _j ∧ DATA → R _j	204 + 8 * R _j
*IMAND, R _j DATAHL	R _j ∧ DATAHL → R _j	"
RAND, R _n	A ∧ R _n → A	120 + 4 * R _n

OU

OR, R _j [@] ADR	R _j ∨ [(ADR)] → R _j	222 + R _j
ORI, R _j , R _i ADR	R _j ∨ (ADR + R _i) → R _j	310 + R _j
*ORIM, R _j DATA	R _j ∨ DATA → R _j	205 + 8 * R _j
*IMOR, R _j DATAHL	R _j ∨ DATAHL → R _j	"
ROR, R _n	A ∨ R _n → A	121 + 4 * R _n

OU EXCLUSIF

XOR, R _j [@] ADR	R _j ⊕ [(ADR)]	→ R _j	224 + R _j
XORI, R _j , R _i ADR	R _j ⊕ (ADR + R _i)	→ R _j	312 + R _j
XORIM, R _j DATA	R _j ⊕ DATA	→ R _j	206 + 8 * R _j
* IMXOR, R _j DATAHL	R _j ⊕ DATAHL	→ R _j	"
RXOR, R _n	A ⊕ R _n	→ A	122 + 4 * R _n

INCREMENTATION + et -

-__1 : DCM [@] ADR	[(ADR)] - 1 → [(ADR)]	60
	RAC + 1 si [(ADR)] = 0	
	SKip if not zero	
DCMI, R _i ADR	(ADR + R _i) - 1 → (ADR + R _i)	305
	RAC + 1 si (ADR + R _i) = 0	
	SKip if not zero	
DRJ, R _n ADR	R _n - 1 → R _n	22 + 4 * R _n
	saut à ADR si R _n = 0	
+__1 : INM [@] ADR	[(ADR)] + 1 → [(ADR)]	62
	RAC + 1 si [(ADR)] = 0	
	SKip if not zero	
INMI, R _i ADR	(ADR + R _i) + 1 → (ADR + R _i)	304
	RAC + 1 si (ADR + R _i) = 0	
	SKip if not zero	
IRJ, R _n ADR	R _n + 1 → R _n	21 + 4 * R _n
	saut à ADR si R _n ≠ 0	

*LBYT, R_n [@] ADR, MASK 330 + R_n

Rotation de 8 à droite avec masquage du résultat

TRAITEMENT DU BIT

*CPZRO [@]ADR, CONF	$[(\overline{ADR})] \wedge \overline{CONF}$ saut à RAC + 2 si $("0")_{ADR} \neq ("0")_{CONF}$ sinon saut à RAC + 3	77
*CPONE [@]ADR, CONF	$[ADR] \wedge CONF$ saut à RAC + 2 si $("1")_{ADR} \neq ("1")_{CONF}$ sinon saut à RAC + 3	73
*SETONE [@]ADR, CONF	$(ADR) \vee CONF \rightarrow (ADR)$	74
*SETZRO [@]ADR, CONF	$(ADR) \wedge CONF \rightarrow (ADR)$	70

TRAITEMENT DU MOT

. <u>shift à gauche</u> : décalage à gauche - on remplit à droite avec 0		
SHLA, NBRE	LSB → LSB + 1 0 → LSB	373
SHLB, NBRE	idem	370
.. <u>shift à droite</u> : décalage à droite - on remplit à gauche avec 0		
SHRA, NBRE	MSB → MSB - 1 0 → MSB	371
SHRB, NBRE	"	372
. <u>rotation à droite</u>		
RORA, NBRE	$A_0 \rightarrow A_{24}$	375
RORB, NBRE	$B_0 \rightarrow B_{24}$	374
. <u>rotation à gauche</u>		
ROLA, NBRE	$A_{24} \rightarrow A_0$	377
ROLB, NBRE	$B_{24} \rightarrow B_0$	376

INSTRUCTIONS DE CONTROLE

JMP [@] ADR	saut à [(ADR)]	42
JMPI, R_n [@] ADR	saut à [(ADR) + R_n]	$2 + 4 * R_n$
JZRO, R_n [@] ADR	saut à [(ADR) si $R_n = 0$	$20 + 4 * R_n$
JNZR, R_n [@] ADR	saut à [(ADR) si $R_n \neq 0$	$23 + 4 * R_n$
JTC [@] ADR	saut à [(ADR) si carry = 0	50
JMPM [@] ADR	saut à [(ADR) + 1] avec	63
ou	marquage : RAC + 1 $\rightarrow$ [(ADR)]	
CALL		

INSTRUCTIONS "SYSTEME"

ION	valider les interruptions	140
IOFF	invalider les interruptions	141
MSKAL	masquer tous les niveaux	142
CLAL	désactiver tous les niveaux	143
MSKLV NIV	masquer le niveau NIV	144
DSKLV NIV	démasquer le niveau NIV	145
CLLV NIV	désactiver le niveau NIV	146
ACTLV NIV	activer le niveau NIV	147
RCLV	lire le niveau courant	166, 4000
GIVUP	quitter le niveau courant en le désactivant	167, 4000

MULTIPLIEUR

* FLOAD, R_n ADR	$R_n FL + (ADR)_{FL} \rightarrow R_n FL$ saut à RAC + 1 si overflow (RAC + 1) = JMP ERROR	$260 + R_n$
* FLOADM, R_n ADR	idem avec (RAC + 1) = JMPM ERROR	"
* FLMUL, R_j ADR	$R_j FL * (ADR)_{FL} \rightarrow R_j FL$ saut à RAC + 1 si overflow (RAC + 1) = JMP ERROR	$270 + R_n$

```

STATUS OF DATA ACQUISITION
17:13:54 15/06/1982
243 Beg. of burst 243 End of burst RUN 1973 ACTIVE
Buffer status 0 0 0 0 Partial bursts 0
Buffer entries 229 226 0 0 0 0 0 0
Current pointer 37934 44441 Pointer 37934 Max. pointer 37934
243 Calls to DAQ 469 DMA transfers 0 DMA errors
0 LAM errors 0 WAIT errors 14 Empty buffers
0 Time-outs LAM 0 Time-outs EOB
243 NS00 too big 0 NS00 too small 0 NS00 too slow
20000 Req. events 20041 Proc. events
Truncated events : 0
Events with > 2048 words : 0
Word count overflow : 0
Bursts without BOE flags : 0
Events with RMH errors : 3
RMH address less than 67b : 0 words 0 events
Bad branch flags sequence : 0 words 0 events
Bad hodoscope data : 0 words 0 events
Bad chamber data : 7 words 3 events
RMH br. 20041 20040 20040 20041 20040 20040 20041
Event Buffer entries 18144 1796 95 6
Event Buffer WC errors 191 14 1 0
Event Buffer empty events 0 0 0 0

```

```

BEAM STATUS
17:13:56 15/06/1982
243 Beg. of burst 243 Active bursts RUN 1973 ACTIVE
(229 Bursts with beam)

BEAM MONITORS : 229 ENTRIES
Burst 243 mean sigma
EBP Argonium 2.791E+05 2.806E+05 1.106E+04
EBP / T90HJ (true) 9.115E+01 8.499E+01 1.683E+00
EBP / T90HS (true) 1.072E+02 1.057E+02 2.221E+00
EBP / T90V (true) 1.095E+02 1.136E+02 2.396E+00
EBP / (HJ+HS) (true) 4.926E+01 4.712E+01 6.802E-01
THJ (tg) / THJ (bg) 9.017E-01 8.991E-01 8.652E-03
THJ (bg) / THJ (fr) 1.000E+00 1.005E+00 6.594E-04

LINK WITH EA COMPUTER : 211 ENTRIES
Burst 243 mean sigma
Intensity on T8 3.700E+12 3.734E+12
EBP / T8 protons 7.543E-08 7.459E-08 1.209E-09
Protons on T4 66 E+11
T8 / T4 5.606E-01 5.630E-01 6.157E-03
P0 control word 000000 * GREEN *
39 Bursts with bad symmetry out of 211 bursts

```

Run nb. 1973
SPS period 3 subperiod 1
Beam particle: pi - at 200 GeV/c
Nb. of argonium foils: 1
Target: Z=74 length=120 mm

Trigger type : Magic Box
Magic box trigger : dimuon

Opposite sextants				Non-adjacent sextants				Adjacent sextants			
D	C	B	A	D	C	B	A	D	C	B	A
+	+	+	+	+	+	+	+	+	+	+	+
A	I	H	I	A	I	H	I	A	I	H	I
+	+	+	+	+	+	+	+	+	+	+	+
B	I	H	I	B	I	H	I	B	I	H	I
+	+	+	+	+	+	+	+	+	+	+	+
C	I	I	I	C	I	I	I	C	I	I	I
+	+	+	+	+	+	+	+	+	+	+	+
D	I	I	I	D	I	I	I	D	I	I	I
+	+	+	+	+	+	+	+	+	+	+	+

H = high masses I = int. masses L = low masses

Prescaling of J/psi: 1
Sampling rate of unbiased triggers: 15
4 event buffers enabled: 0, 1, 2, 3
Tests done by GESPRO:
Test 1: Trimuon mode: flag only
Test 2: Mass calculation mode: flag only
Test 3: Mass in J/psi region (prescaling mode: flag only
Prescaling region: 2000 - 4000 Max factor: 1
Test 4: Cut on nb. of raw data words mode: rejection
Max. nb. of raw data words: 800
Test 5: Cut on nb. of hodosc. words mode: rejection
Max. nb. of hodoscope words: 40
Test 7: V-R3-R4 not valid mode: flag only

Writing on tape NA1416 on unit 0

Total number of events: 20041

RUN 1973 17:14 15/06/1982
243 BURSTS 20041 EVENTS
GESPRO STATUS

Gespro nb.	0	1	2	3	SUM
version nb.	4	4	4	4	
status	run	run	run	run	
SDI	94568	9693	637	49	104947
STA	32386	3279	206	13	35884
GOOD total	18161	1797	96	7	20061
BAD	14225	1482	110	6	15823
T.O.ANA (STOP)	0	0	0	0	0
buff input err	0	0	0	0	0
T.O.buff ACQ	0	0	0	0	0
buff overflow	62182	6414	430	35	69061
GOOD unbiased	2156	198	8	0	2362
RMH err GESPRO	1	0	0	0	1
Norm. J/psi	146	14	1	0	161
tests (fl+rej)	000137	000137	000137	000137	
rejections	000030	000030	000030	000030	

RUN 1973 17:14 15/06/1982
243 BURSTS 20041 EVENTS
STATUS OF TRACKING

19997 Tracked events

5662	Events with	1 full tracks
3140	Events with	2 full tracks
104	Events with	3 full tracks
3	Events with	4 full tracks
0	Events with	5 full tracks
0	Events with	6 full tracks
0	Events with	7 full tracks
0	Events with	8 full tracks
0	Events with	9 full tracks
0	Events with	10 full tracks
0	Events with	11 full tracks
0	Events with	12 full tracks
0	Events with	13 full tracks
0	Events with	14 full tracks
0	Events with	15 full tracks
0	Events with	16 full tracks

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 9 OF SCALER DUMP

CH.	LAST BURST	ECL SCALER 5		LAST BURST	ALL BURSTS
		ALL BURSTS	CH.		
1	0	0	17	0	0
2	2149	4913376	18	0	0
3	0	0	19	0	0
4	845064	194793178	20	0	0
5	0	0	21	0	0
6	0	0	22	0	0
7	0	0	23	0	0
8	0	0	24	0	0
9	0	0	25	0	0
10	0	0	26	0	0
11	0	0	27	0	0
12	0	0	28	0	0
13	0	0	29	0	0
14	0	0	30	0	0
15	0	0	31	0	0
16	0	0	32	0	0

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 14 OF SCALER DUMP : GESPRO 0 REJECTION LOG

REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	SEXTANT COMBINATIONS	
			TIJ	LAST BURST
1	0	0	12	1
2	0	0	34	1
3	0	0	56	1
4	48	9999	13	3
5	15	4226	15	4
6	0	0	24	0
7	0	0	26	0
8	0	0	35	2
9	0	0	46	2
10	0	0	16	2
11	0	0	36	1
12	0	0	23	1
13	0	0	25	6
14	0	0	45	34
15	0	0	14	4
16	0	0	00	1

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 15 OF SCALER DUMP : GESPRO 1 REJECTION LOG

REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	SEXTANT COMBINATIONS		
			TIJ	LAST BURST	ALL BURSTS
1	0	0	12	0	13
2	0	0	34	0	39
3	0	0	56	0	30
4	7	1069	13	0	25
5	2	413	15	0	62
6	0	0	24	0	34
7	0	0	26	0	8
8	0	0	35	0	44
9	0	0	46	0	21
10	0	0	16	0	25
11	0	0	36	1	61
12	0	0	23	0	49
13	0	0	25	1	127
14	0	0	45	6	811
15	0	0	14	1	121
16	0	0	00	0	21

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 16 OF SCALER DUMP : GESPRO 2 REJECTION LOG

REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	SEXTANT COMBINATIONS		
			TIJ	LAST BURST	ALL BURSTS
1	0	0	12	0	0
2	0	0	34	0	1
3	0	0	56	0	2
4	0	75	13	0	2
5	0	35	15	0	1
6	0	0	24	0	2
7	0	0	26	0	2
8	0	0	35	0	1
9	0	0	46	0	0
10	0	0	16	0	3
11	0	0	36	0	2
12	0	0	23	0	3
13	0	0	25	0	9
14	0	0	45	0	67
15	0	0	14	0	15
16	0	0	00	0	0

17:14 15/06/1982 20041 EVENTS
 RUN 1973 243 BURSTS
 PAGE 17 OF SCALER DUMP : GESPRO 3 REJECTION LOG

REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	SEXTANT COMBINATIONS		
			TIJ	LAST BURST	ALL BURSTS
1	0	0	12	0	0
2	0	0	34	0	0
3	0	0	56	0	0
4	0	5	13	0	0
5	0	1	15	0	0
6	0	0	24	0	0
7	0	0	26	0	0
8	0	0	35	0	0
9	0	0	46	0	0
10	0	0	16	0	0
11	0	0	36	0	1
12	0	0	23	0	0
13	0	0	25	0	1
14	0	0	45	0	4
15	0	0	14	0	0
16	0	0	00	0	0

17:14 15/06/1982 20041 EVENTS
 RUN 1973 243 BURSTS
 PAGE 18 OF SCALER DUMP : GESPRO 0 FLAGGING LOG

REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	SEXTANT COMBINATIONS		
			TIJ	LAST BURST	ALL BURSTS
1	0	44	12	1	79
2	32	8435	34	1	118
3	2	555	56	1	123
4	0	0	13	0	66
5	0	0	15	2	118
6	0	0	24	0	111
7	9	1849	26	0	67
8	0	0	35	0	115
9	0	0	46	0	103
10	0	0	16	2	115
11	0	0	36	2	136
12	0	0	23	1	158
13	0	0	25	0	275
14	0	1	45	1	594
15	6	1486	14	0	225
16	9	2155	00	0	1

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 19 OF SCALER DUMP : GESPRO 1 FLAGGING LOG

REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	SEXTANT COMBINATIONS		
			TIJ	LAST BURST	ALL BURSTS
1	0	1	12	0	8
2	2	809	34	0	7
3	1	57	56	1	15
4	0	0	13	0	5
5	0	0	15	0	18
6	0	0	24	1	12
7	2	188	26	0	6
8	0	0	35	0	13
9	0	0	46	0	7
10	0	0	16	0	8
11	0	0	36	0	8
12	0	0	23	0	21
13	0	0	25	0	24
14	0	0	45	1	76
15	2	163	14	0	17
16	1	198	00	0	0

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 20 OF SCALER DUMP : GESPRO 2 FLAGGING LOG

REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	SEXTANT COMBINATIONS		
			TIJ	LAST BURST	ALL BURSTS
1	0	0	12	0	0
2	0	40	34	0	0
3	0	4	56	0	2
4	0	0	13	0	0
5	0	0	15	0	0
6	0	0	24	0	1
7	0	9	26	0	1
8	0	0	35	0	0
9	0	0	46	0	0
10	0	0	16	0	0
11	0	0	35	0	2
12	0	0	23	0	0
13	0	0	25	0	1
14	0	0	45	0	3
15	0	16	14	0	3
16	0	8	00	0	0

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 21 OF SCALER DUMP : GESPRO 3 FLAGGING LOG

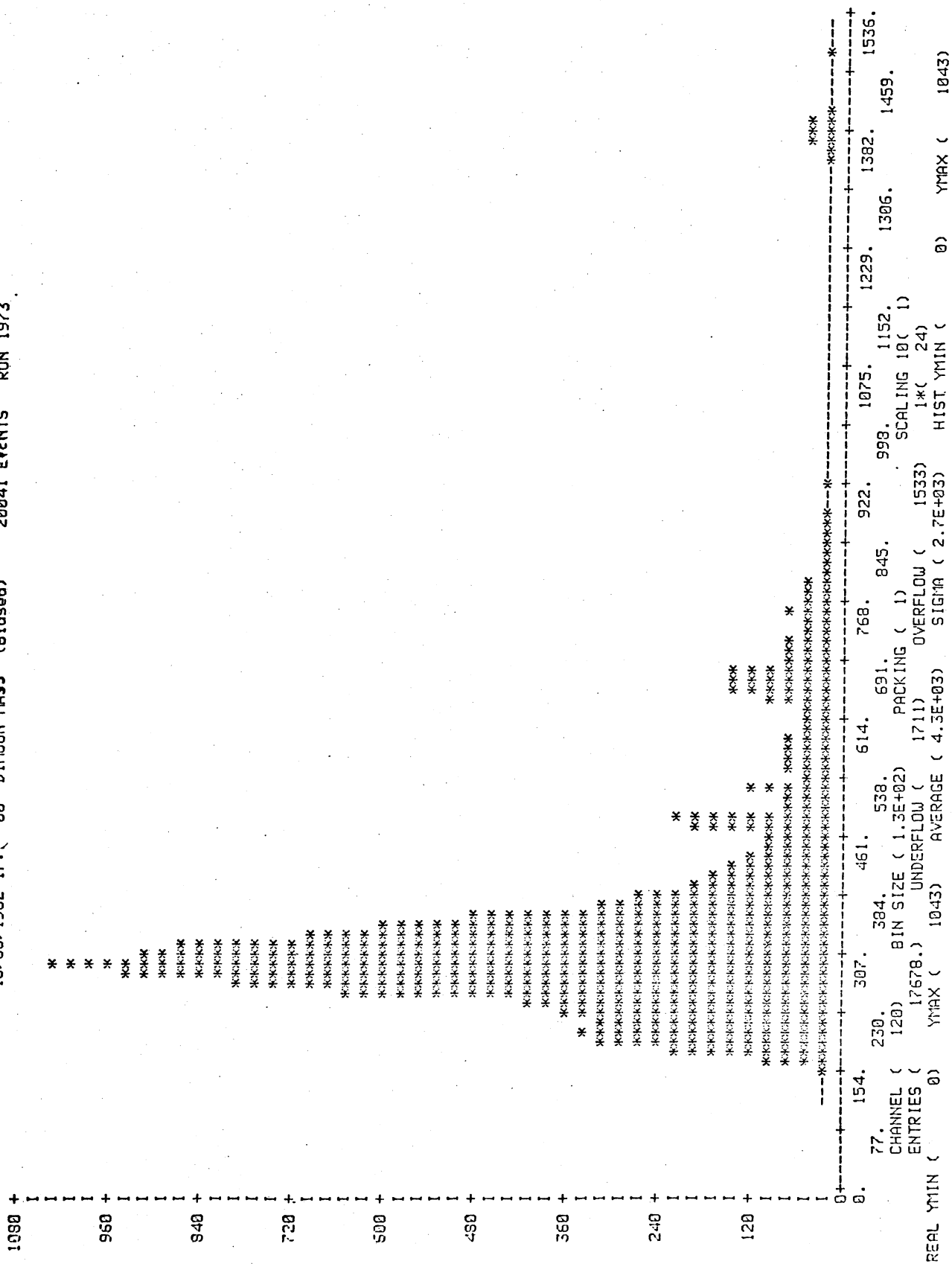
REJ	REJECTION LAST BURST	CRITERIA ALL BURSTS	TIJ	SEXTANT COMBINATIONS LAST BURST	ALL BURSTS
1	0	0	12	0	0
2	0	4	34	0	0
3	0	0	56	0	0
4	0	0	13	0	0
5	0	0	15	0	1
6	0	0	24	0	0
7	0	1	26	0	0
8	0	0	35	0	0
9	0	0	46	0	0
10	0	0	16	0	0
11	0	0	36	0	0
12	0	0	23	0	0
13	0	0	25	0	0
14	0	0	45	0	0
15	0	1	14	0	0
16	0	0	00	0	0

RUN 1973 17:14 15/06/1982 20041 EVENTS
243 BURSTS

PAGE 22 OF SCALER DUMP : GESPRO STATUS

GESPRO NB.	VERSION NB.	STATUS	1	2	3	SUM
SDI	94568	RUN	4	4	4	104947
STA	32386	637	49	13	7	35884
GOOD TOTAL	18161	206	96	6	0	20061
BAD	14225	110	0	0	0	15923
T.O.ANA (STOP)	0	0	0	0	0	0
BUFF INPUT ERR	0	0	0	0	0	0
T.O.BUFF ACQ	0	0	0	0	0	0
BUFF OVERFLOW	62182	430	35	0	0	69061
GOOD UNBIASED	2156	8	0	0	0	2362
RMH ERR GESPRO	1	0	0	0	0	1
NORMAL. J/PSI	145	14	1	0	0	161
TESTS (FL+REJ)	000137	000137	000137	000137	000137	000137
REJECTIONS	000030	000030	000030	000030	000030	000030

15/06/1982 17:1 00 DIMUON MASS (Biased) 20041 EVENTS RUN 1973



REFERENCES

1. Experiments with high-energy charged hyperons at the SPS CERN/SPSC/173-11, 15 June 1973
2. On-line filtering, in proc. 1978 CERN School of Computing, Jadwisin, Poland. 1978-13, p.65 - C.VERKERK
3. Use of fast processors for on-line triggering and event filtering in proc. Topical conference on the application of microprocessors to high energy physics experiments, J.D.BERST, J.LECOQ, J.M.MEYER, M.PERRIN
4. Microprocessor in european high energy physics experiments ECFA Working Group on data processing standards. Report of the microprocessing sub-group. May 1982 - ECFA/82/58
5. A Guide to ROMULUS/REMUS data acquisition Systems, CERN-EP-Electronics NOTE 79-02 (1979).
6. Compacteurs Orsay pour ADC et MWPC. Notices Techniques ORSAY 1976.
7. A fast and flexible data acquisition system for multiwire proportional chambers and other detector. J.B.LINDSAY and al. Nucl. Instr. Meth. 156, 329 (1978)
Provinal specifications :
Type 9294 CERN-EP 9294-1A4/1981-08.05
Type 4237 CERN-EP 4237-3A4/1976-02.10
Type 4236 CERN-EP 4236-3A4/1975-07.22
8. M.MUR-MORPION : a fast hardware processor for straight line finding in MWPC. in Proc. Wire chamber Conf. Vienna Austria - 1980 - Nucl. Instr. Meth. 176, 277 (1980).
9. J.LECOQ, M.PERRIN, J.M.MEYER, C.BOULIN and G.METZGER, Multiprocessor acquisition system including a special purpose CAMAC dedicated microprocessor, presented at CAMAC SYMPOSIUM, HAMBOURG 1978 - CRN-HE-78-16 Centre de Recherches Nucléaires, Strasbourg.
10. Etude d'un système d'aide à la microprogrammation - Mme MARMONNIER - Rapport Interne - Université de Haute Alsace - Mulhouse
11. Contribution à l'étude et à la réalisation d'un processeur micro-programmable, ultra-rapide, destiné à la gestion des tâches CAMAC. Thèse de 3^e cycle C.BOULIN, Université Louis Pasteur Strasbourg I-1976.
12. Trigger NA10 in Trigger Electronics in CERN Experiments. CERN-EP. Electronics Note 80-04 (1980). A.DEGRE

13. How can fast programmable devices enhance the quality of particle experiments.
G.LUTJENS - in proc. Topical Conference on the application of microprocessors
to high-energy physics experiments - CERN - Geneva - 4-6 May 1981.
14. Use of intelligent devices in high-energy physics experiments, C.VERKERK,
in proc. 1980 CERN School of computing, Vraona-Attiki, Grèce 14-27 Sept. 1980
15. En particulier : mémoires mortes ou vives : comment les tester ? C.BOUCHET,
Electronique et applications industrielles - 1980 280
et Séquence de test pour mémoires à semi-conducteurs - J.MONTOIS -
Electronique et applications industrielles - 1975-206.
16. Filtrage en ligne des événements sur une expérience de production de dimuons de
haute masse.
Thèse de 3è cycle : J.P.Froberger 1980 - Université Louis Pasteur Strasbourg I
17. Conception et réalisation d'un système de développement exécutable sur
mini-ordinateur NORD10 pour un microprocesseur spécialisé. Thèse de 3è cycle -
G.HUOT/1978 - Université Louis Pasteur Strasbourg I
18. "Désintégrations semi-leptoniques d'hypérons" - Daniel FROIDEVAUX - Thèse
d'Etat - 6 Juillet 1981 - Paris Sud Centre d'Orsay - LAL-81/12
19. P10 and DMA controller-NORD10 interface with the GEC Executive suite -
Rapport CERN EP (1976) - J.P.VANUXEM
20. Design of a fast mass dependant trigger. G.E.HOGAN - NIM 165, 7 (1979)
21. A Trigger system using cathode read out chamber and fast computing of
Kinematical quantities - NIM 174, 379 (1980) J.BOUCROT and al.
22. The M7-a high speed digital processor for second level trigger selections.
IEEE Trans. Nucl.Sci. NS25-698/1978, T.F.Droege - I.Gaines. Turner.
Cf. Ref.20, 21 et : Thèse de 3è cycle R.DUBE, Université de Paris Sud, Orsay,
LAL 79/6 (Mars 1979)
23. Apport des processeurs microprogrammables au système d'acquisition et au
contrôle en ligne de l'expérience NA10. Thèse de 3è cycle. Ph.KLEIN -
Université Louis Pasteur Strasbourg 1.

BIBLIOGRAPHIE

- Proposal for a fast acquisition system (F.A.S.) Note CRN/HE 74-10
J.D.BERST - Y.CHATELUS - J.M.MEYER
- Confining CAMAC Systems controllers (technical description) G.E.C. Elliott
1973 - D.DRURY.
- EURATOM Reports on CAMAC : 4100 E, 4600E, 9600 E
- Tous documents constructeurs MOTOROLA - AMD - INTEL en particulier :
3002 Control processing element - Document intel MCS-269-0275/27,5 K
3001 Microprogram control unit - Document Intel MCS 149.774.1K
3214 Priority interrupt control Unit - Document Intel MCS 396-0275/27,5 K
- Contribution à l'étude des systèmes informatiques en temps réel en physique
des particules élémentaires. Thèse de doctorat d'Etat, J.M.MEYER - Université
de Haute Alsace - 1977
- High Resolution study of the inclusive production of massive muon pairs by
intense beams. CERN/SPSC/77-10
- Practical data acquisition problems in large high energy Physics experiments
R.W.DOBINSON, CERN - in proc. 1980 CERN School of computing.
- Le spectromètre à dimuons de l'expérience NA10. Son système de déclenchement -
Thèse de 3^è cycle - A.COC 1980 - Université de Paris VII
- Proceeding of CERN Schools of computing 1974-1976-1978-1980
- Communications à : "Topical Conference on the application of microprocessors
to high energy physics experiments - CERN Genève - Mai 1981 - En particulier :
les utilisations de microprocesseurs en tranche
- MONICA - A programmable microprocessor for track recognition in an e^+e^-
experiment at PETRA. H.J.STUCKENBERG
- CAMAC BOOTER ("CAB" System) - a versatile microcomputer for high rate CAMAC
data acquisition, E.BARRELET and al.
- User aspects of MICE (J.A.BLANC and al.) et CAT and MICE : Computer-assisted
Ionography with a microprogrammable engine. A.JEAVONS and al.
- On line filtering at the CERN-ISR - H.FREHSE
- The microprogrammable processor ESOP in the AFS Trigger system, S.CAIRANTE and
al.
- On-line use of the 168/E emulator at the CERN ISR SFM Detector. Ch.BERTUZZI
and al.
- Experiences using the 168/E microprocessor within the European collaboration
(EMC). D.R.BOTERILL - A.W.EDWARDS

- The 168/E at CERN and the Mark 2, an improved processor design - D.LORD and al.
- BDG 500 Branch driver controller. A.DIJLESMAN, W.SCHOEPS
- The bit slice micro-processor "GESPRO" as a project in the UA2 Experiment. Ch.BECAM and al.
- GA103 : a microprogrammable processor for on-line filtering. A.CALZAS, G.DAMON, B.BOUQUET.

R E M E R C I E M E N T S

Ce travail a été effectué dans le groupe "Compteurs" de la Division des Hautes Energies du CRN à Strasbourg, sous la direction de Monsieur M.CROISSIAUX que je tiens à remercier pour l'accueil qu'il m'a réservé dans son laboratoire.

Je veux remercier Monsieur le Professeur G.METZGER, Directeur du Laboratoire d'Electronique et d'Instrumentation Nucléaire de Mulhouse d'avoir accepté la direction de la thèse et la présidence du jury.

J'adresse également mes remerciements à MM. AUBRUN et VERKERK d'avoir accepté de participer au jury.

Les résultats présentés ici n'ont été obtenus que grâce aux travaux de plus de sept ans de collaboration dont je remercie ici collectivement tous les membres.

Tous mes travaux n'ont été possibles que grâce à la présence et au travail incessant de M. René MORAND. Je le remercie chaleureusement pour tout ce qu'il a donné tout au long de ces deux expériences et pour ses conseils concernant la rédaction de cette thèse.

Je dois beaucoup à Monsieur Jean Marie MEYER pour son soutien constant et en particulier pour sa persévérance dans les phases difficiles de l'expérience WA2.

Monsieur M.PERRIN a partagé avec moi les difficultés de cette première expérience, qu'il soit ici chaleureusement remercié.

Monsieur J.J.BLAISING a accepté avec dévouement et efficacité d'assurer le dialogue Physicien-Automaticien qu'il soit ici remercié.

Monsieur Jean Daniel BERST assure depuis mon arrivée au laboratoire la bonne marche du service technique. Que ses résultats, dans des situations souvent difficiles soient ici une fois encore amicalement salués.

Je remercie également tous les étudiants stagiaires qui ont participé au développement du matériel, ainsi que MM. BOULIN, FROBERGER et KLEIN qui ont participé au développement du logiciel.

Je tiens enfin à remercier le secrétariat du laboratoire et tous ceux qui ont contribué à la réalisation de frappe des dessins et des tirages de cette thèse. Je remercie surtout Mme GOETZ pour la rapidité et la qualité de sa frappe.

RESUME

Cette thèse a pour but de montrer l'intérêt de l'utilisation de micro-processeurs **microprogrammables** en acquisition et en filtrage d'événements de physique des **particules** hautes énergies.

La première partie est consacrée à l'analyse du problème et à la mise en évidence des principales difficultés.

La deuxième partie décrit la **solution** retenue. Cette dernière consiste à insérer dans **la** structure informatique initiale une grande efficacité de transfert et de traitement des données au moyen d'une architecture **multiprocesseur** à intelligence répartie, adaptable à chaque expérience.

La troisième partie décrit en détails le processeur **GESPRO**, les outils logiciels conçus pour cette réalisation, ainsi que les performances obtenues.

La dernière partie est consacrée à l'analyse des résultats obtenus depuis 1975 sur deux expériences (**WA2** et **NA10**) au SPS du **CERN** à Genève.

ABSTRACT

This **thesis** describes the **interest** of using microprogrammable **micro-**processors for **on-line** triggering and event **filtering** in high energy physics experiments.

The first part is used to analyse this problem and to show the principal difficulties.

The second part describes **our** solution, which consist of adding in the initial informatic structure a great transfert and treatment capability by a **multiprocessor** architecture, **suitable** to each experiment.

The third part is dedicated to the description of **our** fast processor **GESPRO** and of its software tools specially written for this realization. Performances obtained are **then** given.

The last part is used to **detail** the results obtained **since** 1975 on two experiments (**WA2** and **NA10**) on the SPS of **CERN** at Geneva.